User Empowerment in Adaptive Video Streaming over Best-Effort Networks

by

Leonardo Peroni

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Telematic Engineering

Universidad Carlos III de Madrid

Tutor/Advisor: Sergey Gorinsky

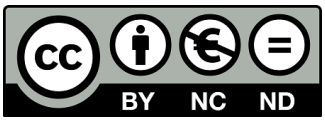
January 2025

User Empowerment in Adaptive Video Streaming over Best-Effort Networks

Prepared by:
Leonardo Peroni, University Carlos III of Madrid
contact: 100455778@alumnos.uc3m.es

Under the advice of:
Sergey Gorinsky, IMDEA Networks Institute

This work has been supported by:



Acknowledgments

First and foremost, I would like to thank my advisor, Sergey Gorinsky, for the opportunity to pursue this PhD in his research group and for his guidance throughout this journey, particularly for granting me the freedom to choose my research direction. He has introduced me not only to the technical aspects of research but also to a range of equally important and nuanced considerations, including the more curious, anecdotal, political, and human sides of our work.

I also extend my gratitude to Dongsu Han for his guidance during my time at KAIST, with a special thanks to all the members of the Intelligent Network Architecture Lab for providing not only an inspiring research environment but also a warm welcome.

Since listing everyone would be too lengthy, I want to express my appreciation to all the members of the IMDEA team who have accompanied me on this journey.

Finally, I would like to thank my parents. Although I may not have fully utilized their emotional support, I know they have always been there for me. On the other hand, I have certainly benefited from their consistent financial support, which has never wavered.

Published and Submitted Content

This thesis is based on the following published papers:

[1] Leonardo Peroni, Sergey Gorinsky. An End-to-End Pipeline Perspective on Video Streaming in Best-Effort Networks: A Survey and Tutorial. Published in *ArXiv:2403.05192, September 2024*. https://arxiv.org/abs/2403.05192. (Currently under submission to a journal)

- This work is fully included and its content is reported in Chapter 2.
- The author fully participated in the writing of the paper and his role in this work focused on performing the literature review and investigation into videostreaming ecosystem and infrastructures within best-effort networks.
- The material from this source included in this thesis is not singled out with typographic means and references.

[2] Leonardo Peroni, Sergey Gorinsky, Farzad Tashtarian, Christian Timmerer. Empowerment of Atypical Viewers via Low-Effort Personalized Modeling of Video Streaming Quality. Published in *Proceedings of the ACM on Networking 1, CoNEXT3, 1–27, November 2023*. https://doi.org/10.1145/3629139.

- This work is fully included and its content is reported in Chapter 3.
- The author's role in this work is focused on the design, implementation and experimentation with regarding of the concepts proposed and the writing of the paper.
- The material from this source included in this thesis is not singled out with typographic means and references.

[3] Leonardo Peroni, Sergey Gorinsky, Farzad Tashtarian. In-Band Quality Notification from Users to ISPs. Published in *IEEE 13th International Conference on Cloud Networking (CloudNet)*, Rio de Janeiro, Brazil, November, 2024. https://doi.org/10.1109/CloudNet62863.2024.10815908.

- This work is fully included and its content is reported in Chapter 4.
- The author's role in this work is focused on the design, implementation and experimentation with regarding of the concepts proposed and the writing of the paper.

- The material from this source included in this thesis is not singled out with typographic means and references.

[4] **Leonardo Peroni**, Sergey Gorinsky. Quality of Experience in Video Streaming: Status Quo, Pitfalls, and Guidelines. Published in *16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bengaluru, India, January 2024. https://doi.org/10.1109/COMSNETS59351.2024.10427330.

- This work is fully included and its content is reported in Chapter 5.
- The author's role in this work is focused on the design, implementation and experimentation with regarding of the concepts proposed and the writing of the paper.
- The material from this source included in this thesis is not singled out with typographic means and references.

Abstract

Video streaming is not only the largest source of Internet traffic but also one of the most economically significant industries, particularly in its adaptive form, where content is delivered in multiple quality levels via the hypertext transfer protocol (HTTP). Users purchase streaming services from streaming platforms (SPs) and network access from Internet service providers (ISPs), both of which strive to enhance user satisfaction, known as quality of experience (QoE), through contrasting methods. QoE is crucial for the ecosystem's economy and technological advancements, but its subjective nature complicates measurement and application. Consequently, easier-to-use yet less accurate QoE proxies, termed QoE models, become prevalent.

Service providers, particularly SPs, increasingly incorporate active user involvement to boost QoE. This trend, known as user empowerment, offers users active opportunities to enhance their streaming experience. It targets increasing QoE for those who are willing to invest effort, while not impacting those who prefer a passive role. Because user participation requires balancing effort and reward, empowerment strategies must manage this trade-off effectively while ensuring simplicity and privacy. From the providers' viewpoint, these strategies serve as extensions of core services, necessitating cost-effectiveness and easy integration. This approach enhances QoE and gives engaged users more control, fostering trust, loyalty, and a competitive advantage for the platform.

Currently, user empowerment techniques in video streaming are still nascent. This thesis aims to bridge the gap in user empowerment within adaptive video streaming, focusing on enhancing QoE through active engagement with SPs and ISPs. The work presents four key contributions: 1) A holistic exploration of the video streaming landscape, emphasizing adaptive streaming of long-form videos over current Internet architecture, while reviewing and classifying state-of-the-art methods and identifying promising development directions. 2) A method for creating personalized QoE models by engaging users in brief assessment sessions, resulting in significant QoE improvements through active learning. 3) A novel mechanism for in-band QoE communication from users to ISPs, utilizing the SP's client interface for QoE estimation and transmission, supported by a prototype demonstrating its feasibility. 4) An evaluation of shortcomings in QoE practices and models, providing guidelines for improvement.

Table of Contents

List of Tables

List of Figures

List of Acronyms

1D	One-Dimensional
2D	Two-Dimensional
A2C	Advantage Actor Critic
A3C	Asynchronous Advantage Actor Critic
ABMA+	Adaptation and Buffer Management Algorithm
ABR	Adaptive BitRate
AC	Actor Critic
ACAA	Affective Content-Aware Adaptation
ACKTR	Actor Critic using Kronecker-factored Trust Region
ACR	Absolute Category Rating
AE	AutoEncoder
AIMD	Additive-Increase Multiplicative-Decrease
ALTO	Application-Layer Traffic Optimization
ANT	Accurate Network Throughput
AP	Access Point
AR	Augmented Reality
ARBITER+	Adaptive Rate-Based InTElligent http stReaming
ARTEMIS	Adaptive bitRaTE ladder optiMIzation for live video Streaming
AV1	Aomedia Video 1
AVC	Advanced Video Coding
AVoD	Advertising-based Video on Demand
B-frame	Bipredictive frame
BANQUET	BAlaNcing QUality of Experience and Traffic
BBA	Buffer-Based Algorithm
BO	Bayesian Optimization
BOLA	Buffer Occupancy based Lyapunov Algorithm
CDN	Content Delivery Network
CHN	Content Harvest Network
CMAF	Common Media Application Format
CMCD	Common Media Client Data

CMSD	Common Media Server Data
CNN	Convolution Neural Network
CP	Content Provider
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CTU	Coding Tree Unit
CU	Coding Unit
DASH	Dynamic Adaptive Streaming over Http
DCT	Discrete Cosine Transform
DD	Deep Downscaler
DDPG	Deep Deterministic Policy Gradient
DDS	Dnn-Driven Streaming
DNN	Deep Neural Network
DO	Dynamic Optimizer
DP	Dynamic Programming
DPI	Deep Packet Inspection
DPPO	Distributed Proximal Policy Optimization
DQL	Deep Q-Learning
DRL	Deep Reinforcement Learning
DST	Discrete Sine Transform
DT	Decision Trees
EA-MCTF	Encoder-Aware Motion Compensated Temporal Filter
EC2	Elastic Compute Cloud
ELASTIC	fEedback Linearization Adaptive STreamIng Controller
ERUDITE	dEep neuRal network for optimal tUning of aDaptive vIdeo sTreaming controllErs
EVC	Essential Video Coding
EVSO	Environment-aware Video Streaming Optimization
FCC	Federal Communications Commission
FESTIVE	Fair, Efficient, and Stable adapTIVE algorithm
FL	Federated Learning
FastTTPS	Fast video Transcoding Time Prediction and Scheduling
fps	frames per second
GAN	Generative Adversarial Network
GOP	Group Of Pictures
GP	Gaussian Processes
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
GS	Greedy Sampling

HAS	Http Adaptive Streaming
HDR	High Dynamic Range
HDS	Http Dynamic Streaming
HDTV	High-Definition TeleVision
HEQUS	HEvc-based QUadtree Splitting
HEVC	High Efficiency Video Coding
HLS	Http Live Streaming
HMD	Head-Mounted Display
HR	High-Resolution
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTP3	HyperText Transfer Protocol version 3
HTTPS	HyperText Transfer Protocol Secure
HYBJ	Jump-enabled HYBrid coding
I-frame	Intra-frame
IAA	Interest-Aware Approach
IF	Influence Factor
IGS	Improved Greedy Sampling
IL	Imitation Learning
ILP	Integer Linear Programming
INFLOW	Intelligent Network FLOW
IQN	In-band Quality Notification
ISP	Internet Service Provider
ITU	International Telecommunication Union
IXP	Internet eXchange Point
iLQR	iterative Linear Quadratic Regulator
iMPC	ilqr based Model Predictive Control
iQoE	individualized Quality of Experience
JND	Just-Noticeable Difference
k**-NN**	k-Nearest Neighbors
LCEVC	Low Complexity Enhancement Video Coding
LEAP	Learning-based Edge with cAching and Prefetching
LNC	Layered Neural Codec
LO	Lyapunov Optimization
LOLYPOP	LOw-LatencY PredictiOn-based adaPtation
LR	Low-Resolution
LSTM	Long Short-Term Memory
LwTE	Light-weight Transcoding at the Edge
M/D/1/K	Markovian Deterministic Single-server finite-Capacity

MAE Mean Absolute Error
MILP Mixed-Integer Linear Programming
MINLP Mixed-Integer NonLinear Programming
MIP Mixed-Integer Programming
ML Machine Learning
MLMP Meta-Learning framework for Multi-user Preferences
MOS Mean Opinion Score
MPC Model Predictive Control
MPEG Moving Picture Experts Group
MR Mixed Reality
NAT Network Address Translation
NB Naive Bayes
NDN Named Data Networking
NP Nondeterministic Polynomial time
NVENC NVidia ENCoder
OSCAR Optimized Stall-Cautious Adaptive bitRate
OTT Over-The-Top
P-frame Predictive frame
P2P Peer-To-Peer
P4P Proactive network Provider Participation for P2P
PANDA Probe AND Adapt
PI Proportional-Integral
PIA PId-control based Abr streaming
PID Proportional-Integral-Derivative
PLCC Pearson Linear Correlation Coefficient
PPO Proximal Policy Optimization
PPV Pay-Per-View
PREPARE Playback RatE and Priority Adaptive bitRatE selection
PSNR Peak Signal-to-Noise Ratio
PoC Proof of Concept
QBC Query By Committee
QL Q-Learning
QP Quantization Parameter
QT QuadTree
QUAD QUality-Aware Data-efficient streaming
QUETRA QUEuing Theory-based Rate Adaptation
QUIC Quick Udp Internet Connections
QoE Quality of Experience
QoS Quality of Service

RDS	Relational Database Service
RF	Random Forest
RIGS	Randomized Improved Greedy Sampling
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
ROI	Region Of Interest
RS	Random Sampling
RTMP	Real-Time Messaging Protocol
S3	Simple Storage Service
SAM	Sequential Auction Mechanism
SARA	Segment-Aware Rate Adaptation
SDN	Software-Defined Networking
SL	Supervised Learning
SMASH	Supervised Machine learning Approach to adaptive video Streaming over Http
SP	Streaming Platform
SQUAD	Spectrum-based QUality ADaptation
SR	Super Resolution
SRAVS	Super-Resolution based Adaptive Video Streaming
SRCC	Spearman's Rank Correlation Coefficient
SRT	Secure Reliable Transport
SS	Smooth Streaming
SSIM	Structural Similarity Index Measure
ST	Space-Time
STALLION	STAndard Low-Latency vIdeo cONtrol
SVC	Scalable Video Coding
SVoD	Subscription-based Video on Demand
SVR	Support Vector Regression
TCP	Transmission Control Protocol
TF-IDF	Term Frequency-Inverse Document Frequency
TR	ThroughputRule
TVoD	Transactional Video on Demand
UC	Uncertainty Clustering
UDP	User Datagram Protocol
UL	Unsupervised Learning
URL	Uniform Resource Locator
VCE	Video Coding Engine
VDN	Video Delivery Network
VISCA	VIdeo Super-resolution and CAching

VMAF	Video Multimethod Assessment Fusion
VR	Virtual Reality
VVC	Versatile Video Coding
Vabis	Video adaptation bitrate system
Video ATLAS	Video Assessment of TemporaL Artifacts and Stalls
VoD	Video on Demand
WebRTC	Web Real-Time Communication
XGB	eXtreme Gradient Boosting
XSVR	eXtended SVR

1 Introduction

1.1. Video Streaming

Video streaming dominates Internet traffic for over a decade, and its growth shows no signs of slowing. [5] estimates that the volume of video traffic quadruples, from 2017 to 2022, increasing its share of total Internet traffic from 75% to 82%, with live streaming surging 15-fold. [6] forecasts a similar trend, predicting that by 2024, the proportion of the population using video streaming services reaches 18.3%, rising to 20.7% by 2027, while [7] shows that 83% of households in the U.S. currently subscribe to at least one streaming service. The growth of the video streaming industry is also evident in its economics: it currently holds a value of $544 billion and is expected to reach $1.9 trillion by 2030. Revenue is set to hit $43.97 billion in 2024 and rise to $54.22 billion by 2027, reflecting an annual growth rate of 7.53% [8].

Several factors drive this expansion, including technological advances, diverse content, and ease of access. High-speed Internet significantly enhances the streaming experience, with global fixed broadband speeds rising to 110.4 Mbps and mobile speeds tripling to 43.9 Mbps from 2018 to 2023 [9]. Streaming platforms (SPs) attract subscribers with exclusive content and broad device compatibility, with networked devices per capita increasing to 3.6 by 2023 [9]. The COVID-19 pandemic also boosts viewership, increasing streaming time by 44% from 2019 to 2020 and by 13% from 2020 to 2021 [10, 11].

In this prominent industry, a diverse ecosystem of stakeholders exists, each with distinct roles. SPs, like Netflix, Amazon Prime Video, and Youtube, offer streaming services through a maintained infrastructure and following various revenue models. The platforms obtain videos from content providers (CPs). Users act as the final consumers of the videostreaming experience with varying levels of satisfaction. Content delivery networks (CDNs) facilitate low-latency video distribution by storing content on cache servers located near users, equipment manufacturers produce devices necessary to access the services, while Internet service providers (ISPs) provide the network connectivity that enables user access [12].

ISPs' relationship with SPs is particularly contentious, as both act as providers to the end user who pays for their different services. This creates a shared incentive to improve the quality of video streaming services to maintain a good reputation and gain a competitive edge. However, despite having the technical means to collaborate effectively [13], their approaches clash [14]. CPs prefer to utilize proprietary tools, such as specific media players and application enhancements, while insisting that ISPs adhere to the principles of network neutrality, which prohibits preferential treatment of data based on its type [15], except during temporary congestion. This stance aims to avoid paying ISPs for quality enhancements and to maintain their advantageous economic position. Moreover, this framework protects smaller CPs that may lack the financial resources to pay for content prioritization, fostering fair competition. Consequently, SPs are increasingly raising concerns about fairness to strengthen their position and are encrypting their traffic, which hinders ISPs' ability to manage that traffic effectively. ISPs manage service enhancements by violating net neutrality, prioritizing certain data flows or reshaping traffic. They resist net neutrality largely due to dwindling traditional revenue from multimedia services, which face competition from SPs [16]. In short, SPs want ISPs to function as neutral pipes, ensuring all data is treated equally, while ISPs seek greater control over traffic management, creating tension between the two.

Independently of the disputes among service providers, users play a crucial role in the economics of this ecosystem regardless of the specific business model adopted. Subscription-based video on demand (SVoD) platforms, like Netflix, rely on recurring fees from users. Advertising-based video on demand (AVoD) models, such as YouTube, offer free content supported by ads, with success tied directly to viewer engagement and ad interaction. Transactional video on demand (TVoD) services prioritize user choice by allowing them to pay per content. Hybrid models combine elements of both subscriptions and ads, while freemium models, like YouTube Premium, provide free access with ads alongside an option to upgrade to a premium experience. Pay-per-view (PPV) models charge users for individual content, emphasizing the user's willingness to pay for exclusive access [17–19]. It is clear that expanding and retaining the user base ultimately determines the success or failure of a service. To achieve this, all stakeholders, whether directly or indirectly, strive to enhance the user satisfaction associated with their services.

The roles and relationships of actors are constantly evolving making it difficult to capture an accurate, static description of the industry as a whole. For instance, it is increasingly common for CPs and SPs to overlap, with SPs often owning and operating their own CDNs [20]. Such complexity also reflects in the technological infrastructure that powers videostreaming services, where SPs operate at the application level, perceiving ISPs indirectly through the available network bandwidth for streaming. Conversely, ISPs operate at the network level and have limited visibility into the streaming process at the application level.

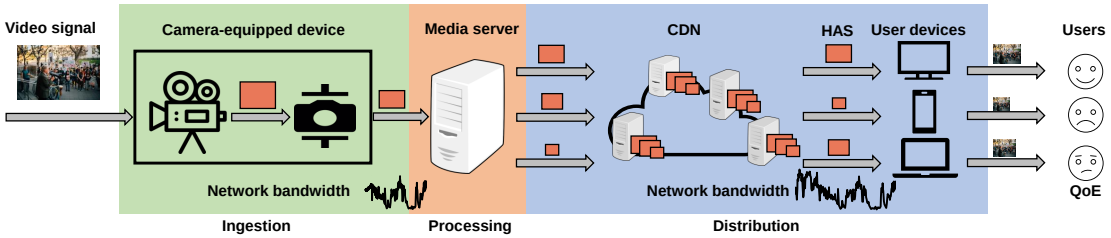


Figure 1.1: Application-level perspective on video streaming.

1.1.1. Application-Level Perspective

From the application-level perspective, there are various streaming infrastructures that arise from different video modalities, each with its specific objectives, and from the coexistence of various underlying network technologies. For example, a specific infrastructure strives to maximizing bandwidth efficiency for 360-degree videos [21] while another strives to achieve ultra-low latency for video conferencing [22]. Similarly, software-defined networking (SDN) [23] and named data networking (NDN) [24] exemplify different alternative networking architectures, which alter the videostreaming processes built on top of them.

The most common videostreaming pipeline, which serves as the foundational infrastructure for this dissertation, involves streaming two-dimensional (2D) long-form video over the current best-effort Internet. Figure 1.1 illustrates the infrastructure which includes three main stages: ingestion, processing, and distribution. The ingestion stage focuses on efficiently uploading encoded raw footage from a camera-equipped device to a media server. The processing stage, which primarily occurs within the media server, handles the storage and transformation of the ingested video. Finally, the distribution stage is responsible for delivering the video from the media server to a user's device for decoding and playback, typically relying on HTTP adaptive streaming (HAS) protocols which treat media content as standard web content and deliver it in small chunks over the HTTP protocol [25]. Specifically, the media server transcodes the uploaded video into multiple representations, each with different combinations of bitrate and resolution, resulting in varying file sizes. The transcoded video is then divided into chunks. A CDN scalably disseminates the video to heterogeneous user devices for decoding and playback. On the user's device, the video player employs an adaptive bitrate (ABR) algorithm to sequentially and iteratively select the appropriate chunk representations based on network performance fluctuations, ensuring smooth video playback and delivering various levels of satisfaction with the service, also known as quality of experience (QoE).

1.1.2. Network-Level Perspective

The underlying network infrastructure supports all stages of the video streaming pipeline by providing the necessary connections to other providers and users, generally overlooking the applications built on top. ISPs control and manage data flows, which are defined as data channels identified by a five-tuple description that includes source and destination IP addresses, port numbers, and protocols. Even though there is a growing adoption of QoE as a performance metric, there remains a strong focus on quality of service (QoS) [26], which aims to quantify the overall performance of a service through objective measures like packet loss rate, available bandwidth, jitter and latency. ISPs play a crucial role in managing QoS through traffic optimization, congestion reduction, and ensuring reliable connections, with factors such as network topology and routing protocols significantly impacting the efficiency of video delivery.

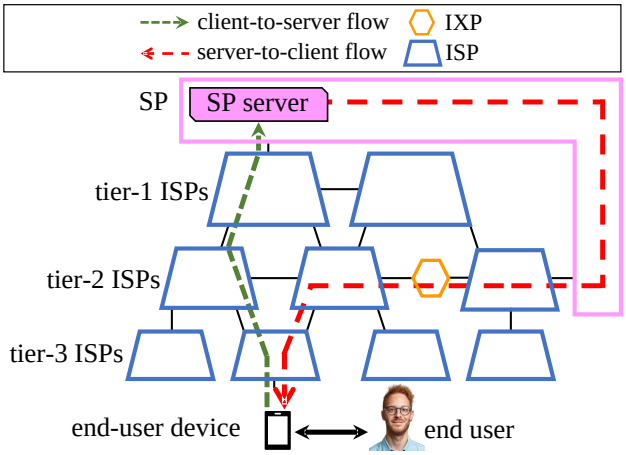


Figure 1.2: Network-level perspective on the distribution stage of video streaming.

ISPs operate in a tiered structure. Figure 1.2 represents an example of communication in the distribution stage between end user and SPs from the network-level perspective where the central tier-2 ISP acts as a transit customer [27] of a tier-1 ISP [28], buys partial transit [29] from another tier-1 ISP, peers with one tier-2 ISP via a private interconnection [30], purchases a remote-peering service [31] to reach an Internet exchange point (IXP) [32, 33], publicly peers at this IXP with another tier-2 ISP, and sells transit to two tier-3 ISPs.

1.2. QoE

QoE reflects a user's overall satisfaction with an application based on their personal perception, making it a subjective measure. QoE is influenced by various factors, such as network connectivity, device type, and application content [34, 35]. Originating from QoS [26] in packet-switched networking, QoE differs in two key ways. Firstly, QoE focuses on the user's subjective experience rather than objective system performance. Secondly, while QoS encompasses multiple performance metrics, QoE is a comprehensive concept

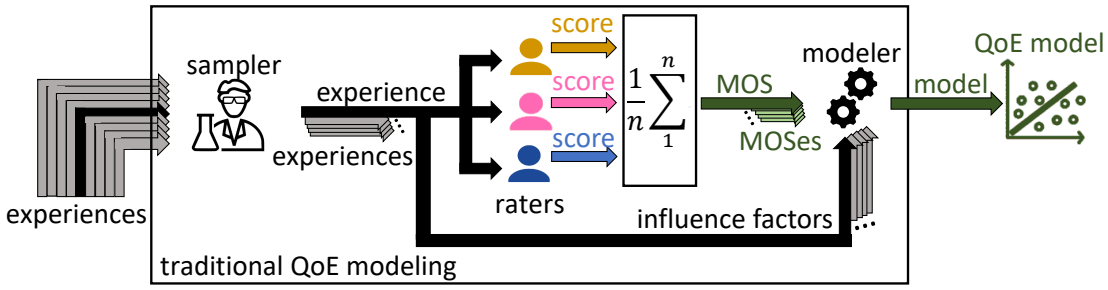


Figure 1.3: Traditional QoE modeling.

that captures the user's overall satisfaction with the application. The QoE subjectivity implies that direct assessment of QoE involves subjective tests where human raters provide scores for experiences presented to them using scoring scale. There exist various of them including those standardized by the international telecommunication union (ITU) [36] such as absolute category rating (ACR) that uses integers from 1 to 5 to represent bad, poor, fair, good, and excellent levels [36] or the 100-point scale where ranges 1-20, 21-40, 41-60, 61-80, and 81-100 represent bad to excellent QoE, respectively [37–39]. While these discrete scales are common, alternative methods include continuous scales, assessments of QoE degradation, and pairwise comparisons of experiences [40, 41].

Although subjective assessments offer high accuracy, they are costly and time-consuming, limiting their practical use and leading to the development and widespread adoption of QoE models. These models automatically estimate QoE from objective influence factors (IFs), such as stall duration and bitrate changes across consecutive chunks [42, 43], serving as proxies for actual user experience, and representing a crucial tool for designing, operating, and evaluating modern video streaming systems. Figure 1.3 represents the typical QoE models construction, where raters—users involved in subjective evaluations—participate in a series of assessments. Each assessment centers on one experience, which consists of a sequence of video chunks characterized by IFs. The sampler, commonly a human expert conducting the assessments [44], selects an experience for each assessment from an experience set. Each rater provides an individual score for every presented experience [45]. A mean opinion score (MOS) averages the individual scores by all raters and represents the QoE perception by the average rater. Based on the MOSes and IFs of the rated experiences, the modeler constructs a QoE model by approximating the functional relation between the MOS and IFs. Existing QoE models differ in their function forms [46, 47] and approximation methods [48, 49].

1.3. User Empowerment

The economic success of a video streaming service depends on attracting new users and retaining existing ones, as these factors directly impact revenue for all stakeholders [19,

50]. As a result, QoE-driven technological advancements become the industry's primary focus. This includes QoE-optimized ABR algorithms for SPs [46, 51] and QoE-aware traffic management strategies for ISPs [52]. In addition, SPs are increasingly adopting strategies that account for active user involvement of various kinds, but mainly through direct feedback. For instance, Skype and WhatsApp prompt users to rate their experience on a scale, provide comments, and evaluate specific features, after a call, to help identify particular problems [53]. Zoom actively collects feedback not only after calls but also through regular surveys, asking users about their overall satisfaction and desired features, such as virtual backgrounds or breakout rooms [54]. Additionally, real-time feedback is being increasingly solicited to improve services and tailor them to individual users. For example, platforms like Netflix and Amazon Prime Video request users to rate content after viewing [55], refining their recommendation systems to better personalize future content suggestions [56–58]. There are even proposals for ABR algorithms that factor in manual user preferences [59]. Similarly, various ISPs offer apps and online tools that allow customers to provide feedback and troubleshoot issues, like Verizon's My Fios or AT&T's Smart Home. CPs are also beginning to embrace this trend, as seen in the interactive features of "Black Mirror: Bandersnatch", which enables the viewer to make choices influencing the story's direction and outcome [60].

User empowerment refers to a set of strategies and techniques aimed at providing users with more opportunities to actively enhance their streaming experience, in line with the evolving trend of user involvement. While any provider in the video streaming ecosystem can adopt these strategies, SPs and, to a lesser extent, ISPs have direct ability to enhance user experience by designing their applications and network services to facilitate user interaction. User empowerment seeks to improve QoE for active users who are willing to invest effort in shaping their experience, such as providing feedback, installing programs, or increasing interactions, which require additional effort in the form of extra time, focus, or delays in the service. This approach involves a trade-off between the benefits gained and the effort required and not all users are inclined to take an active role in increase their QoE, with many preferring to remain passive consumers rather than engage in the process. This reluctance stems from various reasons. For example, video streaming services originates as passive experience for users, who are accustomed to consuming content in this manner. Many users may feel that their satisfaction is already maximized with the current service, leading to little desire for customization. Some users may lack trust in the effectiveness of QoE improvements due to past negative experiences. Others might perceive themselves as having limited technical skills, which causes them to believe their efforts could be futile or even counterproductive. Concerns about privacy also play a role. User empowerment primarily focuses on a distinct group of users who believe their engagement can improve their satisfaction, while preserving the current experience for those who prefer to remain passive.

In this context, user empowerment includes a range of strategies that aim to enhance existing services rather than completely overhaul them, this means that these approaches focus on adding features to the service while preserving its core functionality. Importantly, user empowerment relies on a thorough understanding of QoE and its advancements, especially through the application of QoE models, which are essential for improving video streaming experiences.

Applying user empowerment strategies has the clear potential to attract new users who seek a more active role and has the capability to significantly enhance QoE for users who are harder to satisfy and more inclined to abandon and switch providers. Moreover, by giving users a sense of control and ownership over their interactions, user empowerment builds trust and fosters a positive emotional connection. This helps create a virtuous cycle of engagement and satisfaction, ultimately increasing revenue. It also motivates passive users to become more active. As these newly engaged users are drawn to the unique features of the service, they are likely to develop stronger loyalty, further reinforcing the cycle of user involvement and satisfaction.

All these beneficial effects come at the cost of addressing challenges that fall into two categories: accessibility for users and efficient implementation for service providers. To achieve widespread adoption, it is crucial to strike the right balance between the level of enhancement provided and the amount of user effort or involvement required. If the process becomes too demanding, even the most enthusiastic users may be discouraged from participating. The solution should offer clear and simple interactions that are easy to understand and execute for all users, as otherwise, it could lead to frustration and feelings of inadequacy. Additionally, it must ensure an adequate level of privacy, either through the enforcement of appropriate security policies or by designing interactions that are not privacy-sensitive. From the service provider's perspective, these solutions may require meaningful but non-disruptive changes to the existing infrastructure, necessitating a careful analysis of their cost-effectiveness and ease of integration.

1.4. Challenges and Contributions

This thesis aims to address the existing gap in user empowerment within video streaming services by enhancing QoE through active engagement of users with both SPs and ISPs. User empowerment in video streaming is still in its early stages, offering significant room for improvement, in contrast to fields like content personalization and e-commerce, where it is already well-established and continues to grow, largely driven by the extensive use of user feedback.

The thesis achieves its goal through four key contributions, organized into as many chapters, which branch out across five closely interconnected entities: the streaming pipeline, SP, ISP, QoE, and user empowerment, as illustrated by Figure 1.4. Chapter 2

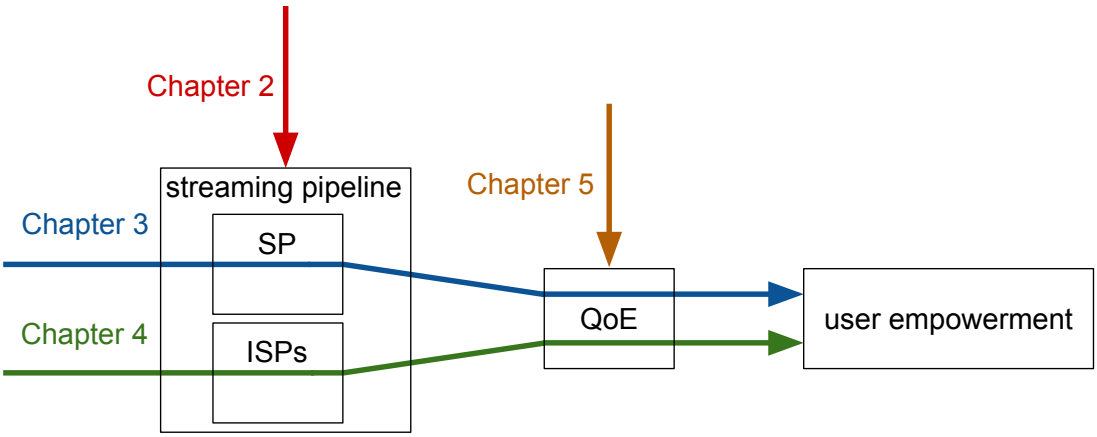


Figure 1.4: Relationship between chapters and key topics of the thesis.

lays the groundwork for a thorough understanding of the subsequent chapters by exhaustively exploring the video streaming landscape through a holistic view of the most prominent streaming pipeline in the current Internet, supported by an in-depth survey of state-of-the-art technologies. Chapter 3 proposes an innovative method to empower users through collaboration with SPs to generate personalized QoE models. Similarly, Chapter 4 introduces a novel mechanism that empowers users to signal QoE impairments to server-to-client ISPs responsible for QoE-impairing congestion. Together, these chapters form the core of the thesis, proposing two innovative approaches to enhance user empowerment on a voluntary basis. Chapter 5 uncovers poor practices and improper usage of QoE in adaptive video streaming that hinder the effectiveness of user empowerment strategies and proposes guidelines for improvement.

1.4.1. End-to-End Pipeline Perspective on Video Streaming

Chapter 2 offers a comprehensive overview of the videostreaming landscape by analyzing the dominant videostreaming pipeline. This helps establish the context for the following chapters of the thesis and highlights areas where user empowerment can be effectively addressed.

Challenges. The vastness of the video streaming landscape, combined with its fast pace of evolution, makes it challenging to collect and organize up-to-date information in a comprehensive and coherent manner. This difficulty is further exacerbated by recent profound changes, such as the widespread adoption of machine learning—particularly deep learning—and the proliferation of diverse pipeline infrastructures tailored to specific streaming modalities or designed around specific network architectures. Additionally, the task is complicated by the fact that recent surveys on the topic often focus narrowly on specific aspects of the pipeline in a siloed manner, overlooking other components and obscuring potential cross-sectional approaches. This fragmented view makes it difficult

to identify the most promising areas for the implementation of strategies aimed at user empowerment.

Contributions. This chapter explores the video streaming technological landscape by analyzing the pipeline for long-form 2D videos over the best-effort Internet, supported by CDNs and client-side ABR algorithms, from a holistic perspective. It offers an extensive review of recent research, organized under a novel classification scheme based on problem-solving methodologies, along with tutorial material that serves as technical background. The analysis identifies three main stages of the pipeline: ingestion, processing, and distribution, highlighting the interactions between these stages. Particular emphasis is placed on the distribution stage, specifically on ABR algorithms and QoE, as these are crucial areas for empowering users. Additionally, the chapter discusses real-world applications, current trends, and promising future directions, some of which are further explored in other chapters of the thesis.

1.4.2. User Empowerment via Low-Effort Personalized QoE Modeling

Chapter 3 explores user empowerment through collaboration between users and SPs by designing a novel method to generate personalized QoE models, which improve the users' QoE.

Challenges. The collaboration between users and SPs can be achieved through various methods, each offering different levels of user empowerment. Achieving higher accuracy typically requires more effort from users, either in terms of time or through more complex interactions, which can result in frustration and potential abandonment of the service. Simultaneously, any solution must be easily integrable into existing SP infrastructures with minimal implementation effort to ensure its viability in real-world scenarios. Striking the optimal balance between accuracy, user effort, and integrability is challenging, as these goals conflict.

Contributions. The chapter introduces *individualized quality of experience (iQoE)*, a novel method for creating personalized QoE models by engaging users in a brief series of subjective assessment sessions. The method is designed to enhance the platform's personalization portfolio by enabling voluntary user participation in the personalization process. iQoE collects direct feedback through clearly defined goals and operates iteratively, utilizing an active learning technique coupled with a model based on support vector regression (SVR). The method is both sample-efficient and highly accurate, with the greatest benefits seen among atypical viewers—those whose QoE perceptions significantly deviate from the median (10% of the population). An evaluation with 120 subjects recruited via Microworkers shows that iQoE improves average accuracy by at least 42% for all users and by at least 85% for atypical viewers, with session durations of approximately 22 minutes, compared to 10 baseline models. These results confirm that iQoE successfully improves QoE through user empowerment in collaboration with SPs.

1.4.3. In-Band Quality Notification from End Users to ISPs

Chapter 4 explores user empowerment through collaboration between users and ISPs by designing a novel QoE impairment notification mechanism. This mechanism enhances ISPs' per-flow QoE-aware traffic management, which in turn leads to higher users' QoE.

Challenges. Delivering QoE feedback from users to ISPs faces several challenges. First, the hierarchical structure of the Internet makes it difficult for users to identify the correct ISP to contact. Additionally, network address translation (NAT) alters the flow's 5-tuple description, obscuring user identifiers and making it impossible for ISPs to identify users, thus necessitating in-band solutions (transmission within the same channel as the primary data). Proposed protocols for direct communication between users and ISPs are complicated due to their out-of-band nature and the disruptive changes they would require to the current Internet infrastructure [61, 62]. Furthermore, when communication occurs through SP channels, encryption by SPs conceals the content, restricting ISPs' ability to decrypt it due to high complexity and limited extent of exposed information.

Contributions. The chapter introduces *in-band quality notification (IQN)*, a practical mechanism for in-band QoE signaling from end users to relevant ISPs. We design IQN to operate via voluntary installation of additional software on user devices. IQN leverages the SP's client interface to estimate and communicate QoE information to all ISPs along the delivery path, akin to an SOS message, without prescribing how ISPs should use this information. As a proof of concept (PoC), the chapter presents YouStall, a prototype that estimates and signals YouTube Live's stalls to an ISP emulated by an Amazon elastic compute cloud (EC2) instance [63]. Evaluations show that IQN-assisted ISP-side inference estimates the duration of significant stalls (those lasting at least 400 ms) with an average mean absolute error (MAE) and root mean square error (RMSE) of 231 and 288 ms, respectively, while the average stall duration exceeds 1.4 s. These results confirm that iQN successfully improves QoE through user empowerment in collaboration with ISPs.

1.4.4. QoE in Video Streaming: Status Quo, Pitfalls, and Guidelines

Chapter 5 focuses on QoE modeling as a key concept underlying user empowerment, unveiling suboptimal practices and proposing corrective actions.

Challenges. Identifying QoE's critical areas is challenging because it requires a thorough review of the literature to uncover common patterns of misuse, which often can only be recognized through firsthand experience and analysis of dataset, which are particularly rare for individual scores. While pinpointing these issues is valuable, for such insights to be truly impactful, a well-structured set of guidelines and countermeasures must be defined, supported by solid evidence and compelling arguments to ensure they are practical and applicable.

Contributions. The chapter examines critical areas of QoE in ABR video streaming by reviewing the current landscape, identifying key problems and common pitfalls, and offering practical advice and recommendations for improvement, drawing on two large datasets of individual QoE perceptions. The insights are organized into categories of test conducting, model building, and model usage, with the aim of assisting newcomers, raise awareness, and serve as a wake-up call for the broader community to address these issues.

2 End-to-End Pipeline Perspective on Video Streaming

Video streaming continues to dominate Internet traffic, involving both end users and service providers in a diverse economic and technical ecosystem. SPs such as Netflix, YouTube, Amazon Prime Video, and Twitch manage the infrastructure where CPs upload videos, while users access streaming services designed to meet their QoE under various revenue models. CDNs enhance video distribution performance through caches and geographically dispersed servers, while ISPs provide network connectivity defined by QoS metrics. Entities often take on multiple roles, with their relationships constantly evolving. Similarly, there are different videostreaming infrastructures, defined by the underlying network infrastructure, like peer-to-peer (P2P), SDN or NDN, and specific streaming modes, like 360-degree or video conferencing.

This chapter provides an extensive overview of recent research on CDN-assisted HAS of 2D videos over best-effort networks with client-side ABR algorithms, which constitutes the major paradigm for video streaming on the current Internet. The focus is on long-form 2D videos, though many reviewed designs also apply to short-form and 360-degree videos. The chapter covers extensive material, reviewing over 200 papers and offering essential tutorial content to enhance accessibility, and serving as a foundation for understanding the entire thesis. Furthermore, the overview delves into the goals and technical interactions among videostreaming actors, which are essential for distinguishing between areas that show promise for the application of user empowerment strategies and those that are less favorable. It enhances the understanding of the rationale behind currently successful strategies, aiding in the uncovering of their complexities, and provides technical insights for designing novel approaches.

This chapter presents an innovative end-to-end perspective on the video pipeline, in contrast to other surveys that focus on isolated stages and neglect cross-stage relationships and a holistic view. We divide the pipeline into three stages: ingestion, processing, and distribution, as illustrated in Figure 1.1. In the ingestion stage, a camera-equipped device captures raw footage, encodes it to reduce file size, and uploads the video to a media server. The processing stage includes video storage, segmentation, and transcoding to

create multiple versions based on an encoding ladder. In the distribution stage, a CDN efficiently delivers the video to various user devices for decoding and playback.

The chapter introduces a novel classification scheme for the reviewed works (Section 2.2), with the pipeline at the top of the hierarchy, followed by the stages of ingestion (Section 2.3), processing (Section 2.4), and distribution (Section 2.5). The distribution stage includes an additional layer that addresses aspects of ABR, CDN, and QoE. Lower classification levels categorize each work based on its methodology for addressing specific problems, with works organized chronologically within the same methodology. We enhance the descriptions of these works by including tables that highlight additional characteristics. The chapter then discusses real-world applications (Section 2.6), trends, and future directions (Section 2.7), concluding with final remarks (Section 2.8).

2.1. Background

2.1.1. End-to-End Streaming Pipeline

The end-to-end streaming pipeline starts at the ingestion stage with the capture of raw video by a camera-equipped device, with a codec applying spatial and temporal compression to the raw footage for reducing the video size, and subsequent upload of the encoded video over the Internet to a media server. This stage attracts significant research efforts, driven by the growing interest in video analytics and live streaming. Ingestion-stage designs aim to improve analytics accuracy, encoding complexity, video quality, bandwidth utilization, and upload latency, which is especially important for interactive applications.

The processing stage, which primarily involves internal operations within the media server, handles the storage and transformation of ingested video. Transformation tasks, such as video segmentation and transcoding, enable the pipeline to manage heterogeneity in network connectivity and device capabilities. Video segmentation divides the video into smaller chunks, while transcoding converts these chunks into multiple representations with different resolutions, bitrates, and frame rates (measured in frames per second, or fps). Numerous research efforts target the integration of transcoding with tasks at the ingestion and distribution stages to optimize pipeline performance.

The distribution stage deals with video delivery from the media server to a user device that decodes and plays back the content. In addition to ISPs, which provide network connectivity, this stage also involves CDNs to disseminate the video from their edge servers to user devices with low latency and high QoE. To tackle the heterogeneity of user devices and variable network conditions, a media player on the user device runs an ABR algorithm. This algorithm dynamically selects, from a server-supplied manifest

file describing available video representations, the most appropriate representation for the next requested chunk. The distribution stage, involving many stakeholders and directly interfacing with end users, presents diverse problem formulations. Consequently, its ABR, QoE, and CDN aspects attract the largest research efforts along the end-to-end pipeline. Design and evaluation, particularly at the ingestion and distribution stages, often rely on a QoE model that expresses the user's subjective QoE as a function of objectively measurable IFs, such as stall duration and video quality. QoE modeling is an interdisciplinary topic. The construction of QoE models relies on subjective testing, informed by user experience design. Network engineering plays a crucial role in QoE by managing upload and distribution infrastructure to deliver content with low latency and high bitrate, for which QoE models account either directly or via other IFs. Data science increases the predictive power of QoE models through learning-based techniques.

2.1.2. 2D Streaming Modes

Video on demand (VoD) is the most dominant of the two streaming modes considered in this thesis. This mode closely aligns with real-world applications of major SPs, such as Netflix, and specifically involves serving pre-stored video from a media server. The reliance on the media server effectively decouples the ingestion and distribution stages: while distribution operates in real time, ingestion occurs beforehand under less stringent latency constraints. As a result, VoD employs different communication designs at the ingestion and distribution stages.

Live streaming refers to an increasingly popular variant that requires real-time operation of the entire pipeline from video capture to playback. "Real-time" is a relative notion where acceptable latency depends on the particular application. This thesis considers HAS-based live streaming for applications such as live broadcasting. HAS improves its support for live streaming through a variety of techniques, such as reducing chunk duration, delivering a chunk in multiple fragments, and prefetching expected chunks by the CDN edge server.

2.1.3. Streaming Protocols

The current streaming ecosystem involves a large number of protocols with their popularity varying across the pipeline stages. Figure 2.1 depicts the usage of streaming protocols by 391 global broadcasters in sports, radio, gaming, and other industries [64]. Apple's HTTP live streaming (HLS) constitutes the most popular protocol due to its dominance at the distribution stage. The main competitors of HLS at this stage are dynamic adaptive streaming over HTTP (DASH) [65], which is an open standard maintained by the moving picture experts group (MPEG),

Microsoft's smooth streaming (SS), and Adobe's HTTP dynamic streaming (HDS). However, both SS and HDS experience declining usage compared to HLS and DASH. The common media application format (CMAF) [66] refers to an emerging container format designed to improve compatibility between HLS and DASH and to support low-latency HAS.

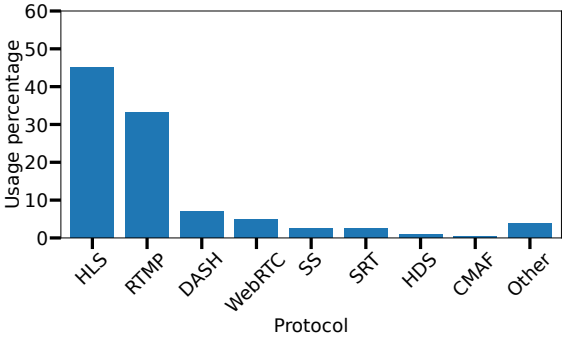


Figure 2.1: Usage of streaming protocols by broadcasters [64].

The real-time messaging protocol (RTMP) maintains its prominence as the leading upload protocol at the ingestion stage. Web real-time communication (WebRTC) and secure reliable transport (SRT) are newer protocols challenging the dominant role of RTMP at this stage. Whereas RTMP uses the transmission control protocol (TCP) as its transport protocol, both WebRTC and SRT rely instead on the user datagram protocol (UDP) to support low-latency upload.

2.1.4. Previous Surveys

A large number of earlier surveys tackle the important topic of video streaming. Due to the complexity of the end-to-end streaming pipeline, these surveys often focus on individual stages or specific elements within a stage. For instance, [25,67,68] concentrate on ABR algorithms at the distribution stage. [69–71] address QoE in video streaming and emphasize QoE modeling, while [72] deals with QoE management in novel network architectures. [73] surveys video streaming over multiple wireless paths. Whereas [74] discusses CDN support for video streaming and other traffic classes, [75] covers cloud-based video streaming. In contrast to the previous surveys, our work offers a holistic overview of video streaming across the entire end-to-end pipeline. In addition to CDN support, QoE, and ABR algorithms at the distribution stage, this chapter also reports on advances in video streaming at the ingestion and processing stages. Besides, we offer an up-to-date perspective by highlighting more recent research findings in the field.

2.1.5. Related Topics Beyond the Chapter Scope

While this chapter offers a new end-to-end pipeline perspective on HAS of long-form 2D videos over the best-effort networks, the rich area of video streaming contains related topics outside the chapter scope. In particular, we do not report on P2P solutions exemplified by WebRTC [22,76] where a camera-equipped device transmits video directly to a user device without any assistance from a media server. By deviating from the HAS pipeline and relying on UDP instead of TCP, such P2P solutions seek to provide

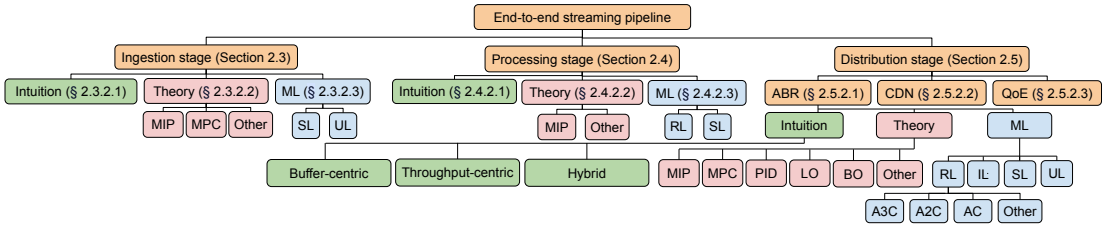


Figure 2.2: Classification scheme of the survey.

ultra-low end-to-end latency for effective support of interactive applications, such as video conferencing [77].

Compared to long-form videos, streaming of short-form videos differs significantly in its requirements and solutions. With a common duration of 15 to 60 s, depending on the SP, a short-form video requires much less storage and bandwidth, making it feasible to implement techniques such as prefetching the entire video [78], relying on progressive download instead of HAS [79], using equal-size rather than equal-duration chunks [80], simplifying the ABR algorithm, or even transmitting the entire video at a single bitrate [81]. In short, short-form streaming is a vast topic deserving a separate survey.

360-degree video streaming also faces distinct challenges. To deliver immersive experiences in virtual reality (VR), augmented reality (AR), and mixed reality (MR), 360-degree videos require specialized equipment for capture and playback. This includes camera arrays, omnidirectional cameras, curved screens, and head-mounted displays (HMDs). The creation and presentation of seamless panoramic videos involve advanced stitching and projection methods [82]. To manage the higher storage, processing, and bandwidth requirements, 360-degree video streaming employs tile-based [83] and viewport-based [84] techniques, which outside the scope of the chapter.

Future Internet architectures, such as SDN and NDN, offer radically new opportunities for video streaming and other applications [85], but the chapter reviews video streaming designs within the current Internet architecture.

2.2. Classification Scheme

Figure 2.2 presents the classification scheme introduced by the chapter. The end-to-end streaming pipeline represents the top level of the hierarchy. The next level distinguishes between the ingestion, processing, and distribution stages of the pipeline, with respective designs discussed in Sections 2.3, 2.4, and 2.5. To reflect the complexity and diversity of the reviewed works, the classification scheme includes branches of varying breadth and depth. For example, we classify the distribution-stage designs into ABR, CDN, and QoE categories. The lower levels of the scheme categorize each work according to its problem-solving methodology, also with varying breadth and depth of the

classification branches. In each final category, we present its designs in chronological order and additionally describe them with respect to various characteristics. While Section 2.2.1 elaborates on the methodology-based classification, Section 2.2.2 discusses the additional characteristics.

2.2.1. Methodology-Based Classification

The methodology-based subdivision in our classification scheme differentiates between methods according to their reliance on intuition, theory, or machine learning (ML), as discussed below.

2.2.1.1. Intuition-Based Methods

In an intuition-based method, a human expert leverages domain knowledge and trial-and-error experimentation to develop a simple heuristic solution. It is common for an informal intuition-based method to undergo subsequent formal analysis, supplying insights into the underlying principles. An intuition-based heuristic might prove broadly applicable beyond the initial problem. A notable example is the additive-increase multiplicative-decrease (AIMD) algorithm [86], originally designed for network congestion control and now employed widely in video streaming and other fields. For intuition-based ABR algorithms, we include a deeper level of classification that considers buffer-centric, throughput-centric, and hybrid categories, where ABR decisions rely on playback-buffer occupancy, network-bandwidth estimate, or both, respectively.

2.2.1.2. Theory-Based Methods

A theory-based method abstracts specific details to formulate a problem within a general formal theory and systematically applies principles of rational logic to derive a solution, often with guarantees of correctness and performance. In comparison to intuition-based methods, the derived solution might be less intuitive or even counterintuitive. Mixed-integer programming (MIP) constitutes a prominent theory-based method for formulating and solving optimization problems [87]. Control-theoretic techniques, such as model predictive control (MPC), proportional-integral-derivative (PID) controllers [88], and Lyapunov optimization (LO) [89], commonly underpin solutions in video streaming. Bayesian optimization (BO) [90] represents a popular statistical optimization method. Our classification utilizes these MIP, MPC, PID, LO, and BO categories commensurately with the diversity of reviewed works: MIP and MPC at the ingestion stage, only MIP at the processing stage, and all five categories for ABR algorithms at the distribution stage. The sixth category, called other, contains theory-based techniques applied less frequently in video streaming, such as dynamic programming (DP) [91].

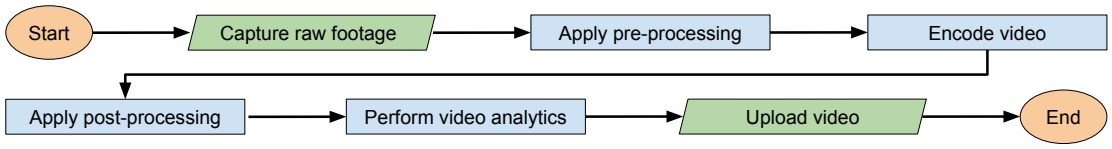


Figure 2.3: The ingestion stage of the end-to-end VoD streaming pipeline.

2.2.1.3. ML-Based Methods

ML trains models on sample data to generalize and produce accurate predictions on new data, rather than following explicit instructions. The focus of ML is on learning generalizable patterns and minimizing error on unseen samples, distinguishing it from theory-based methods that optimize solely for the given data. While promising better performance, ML raises concerns about higher overhead and poorer explainability. We classify ML techniques into four categories based on their model training methodology as reinforcement learning (RL), imitation learning (IL), supervised learning (SL), and unsupervised learning (UL) [92]. At the distribution stage, our classification scheme further divides RL into asynchronous advantage actor critic (A3C), advantage actor critic (A2C), actor critic (AC) [93], and other methods. At the processing and ingestion stages, the reviewed works employ RL, SL, or UL.

2.2.2. Additional Design Characteristics

Besides applying the classification scheme from Figure 2.2, we also use a set of characteristics to describe each reviewed design. This set varies depending on the design category. For every design, we specify its core technique, a free-form characteristic of the design's main distinguishing trait, and codec compatibility. For example, when discussing ML-based designs, the core technique characterizes the used model, such as a decision tree (DT), random forest (RF), naive Bayes (NB), multilayer perceptron (MLP), convolution neural network (CNN), generative adversarial network (GAN), autoencoder (AE), generative pre-trained transformer (GPT), or another deep neural network (DNN) [94, 95].

2.3. Ingestion Stage

2.3.1. Background

We proceed by providing additional background on ingestion, with Figure 2.3 illustrating this stage for VoD with a flowchart. The stage starts with the camera-equipped device capturing raw footage. Then, the device applies pre-processing, such as color correction and balancing, and encodes the video with a codec. After post-processing,

such as artifact filtering, the ingestion stage provides optional support for video analytics, e.g., object detection and recognition. The stage concludes with uploading the encoded video to the media server. Hosting the media server in cloud infrastructure is increasingly common in both VoD and live streaming [96].

2.3.1.1. Video Encoding

Compression, which might occur during both ingestion and processing, is either lossy or lossless. Lossy compression reduces storage and bandwidth needs by discarding some information while maintaining high content quality. Spatial compression removes redundancy within a frame, e.g., by using discrete cosine transform (DCT) and quantization, and encodes the result to reduce the bit count. Temporal compression, which is more computationally demanding, reduces redundancy across multiple frames through motion estimation and compensation. The codec or post-processing applies filtering to correct block boundaries, mosquito noise, ringing, and other artifacts introduced by lossy compression [97].

A compressed video involves different frame types. Intra-frames (I-frames), resulting from spatial compression, serve as reference points, prevent error accumulation, and facilitate video search. Predictive frames (P-frames) use motion compensation based on previous frames, while bipredictive frames (B-frames) leverage both preceding and following frames. A group of pictures (GOP) refers to a sequence starting with an I-frame, followed by P-frames and B-frames. A single container format file stores the encoded video along with audio, synchronization, subtitle, and other metadata.

Video encoding is computationally intensive, with innovations aimed primarily at faster processing. Alongside algorithmic advances, hardware-accelerated encoding becomes more prevalent and offloads tasks to specialized components. Examples include Nvidia encoder (NVENC), which shifts video encoding from the central processing unit to the graphics processing unit (GPU), and video coding engine (VCE), a GPU-integrated unit dedicated to video compression.

Encoding parameters: Latency, throughput, video quality, and other compression metrics represent conflicting optimization goals. A codec manages these trade-offs using various parameters. The frame size in pixels defines a video resolution, with higher resolutions improving image sharpness while requiring more storage and bandwidth. Ideally, the video resolution should match the display resolution. A frame rate denotes the frequency of frames and needs to be high enough to ensure smooth motion perception: while frame rates of 24 to 60 fps are common, some scenarios such as gaming might require up to 120 fps [98]. A GOP structure describes GOPs with two parameters N and M, where N expresses the GOP size in frames, and M captures the distance between two consecutive anchor frames (I-frames and P-frames). Larger GOPs with more B-frames

reduce the video size but increase processing complexity and latency. A bitrate refers to the number of transferred or processed bits per second.

Codecs: As one of the most widely adopted codecs, advanced video coding (AVC) or H.264 refers to a compression standard based on macroblocks and motion compensation [99]. Its features include an integer DCT, variable block-size segmentation, inter-frame prediction over multiple frames, and in-loop deblocking filtering. H.264 is the most popular codec due to its widespread support by commercial devices [100].

High efficiency video coding (HEVC) or H.265, a successor to H.264, achieves up to 50% better compression efficiency while maintaining the same video quality. It replaces 16×16 macroblocks with coding tree units (CTUs) up to 64×64 in size and uses both integer DCT and discrete sine transform (DST). HEVC simplifies deblocking filtering, making it easier to parallelize [101]. Despite superior performance, adoption is slow due to royalty issues and limited browser support.

VP9, an open royalty-free codec developed by Google and utilized on YouTube, employs 64×64 superblocks with quadtree (QT) partitioning and intra-frame prediction with six oblique directions for linear extrapolation of pixels. While less efficient in compression than H.265 [102], VP9 reduces encoding latency and enjoys broad browser support.

AOMedia video 1 (AV1), a royalty-free successor to VP9, diversifies coding options for better video input handling and uses rectangular DCTs, asymmetric DSTs, and superblocks up to 128×128. It also employs in-loop and loop-restoration filters. While AV1 incurs higher computational complexity to improve compression efficiency over H.265 [103], subjective tests of video quality show minimal differences [104].

Scalable video coding (SVC) extends H.264 by enabling layered encoding into multiple streams, where enhancement layers build upon the base layer. These layers improve the frame rate, resolution, bitrate, or combinations thereof [105]. Although less efficient in compression than H.264, SVC better manages highly variable bandwidth [106].

Versatile video coding (VVC) or H.266 [107], adopted in 2020, is a successor to H.265 that supports lossless and subjectively lossless compression. It aims to support a wide range of video applications through layered coding and flexible bitstream handling. VVC offers significant improvements in compression efficiency over H.265 and requires more computational resources [108]. The royalty situation for VVC is still uncertain.

Essential video coding (EVC) [109], introduced in 2020 as well, features innovations such as a binary-ternary tree structure, split unit coding order, and adaptive loop filter. It improves compression efficiency by about 30% over H.265, albeit with five times the computational complexity [110]. EVC is available in both royalty-based and royalty-free profiles.

Low complexity enhancement video coding (LCEVC) [111] constitutes a novel approach to video enhancement. LCEVC adds an enhancement layer to a base layer

encoded with a different codec, with an objective to reduce both encoding and decoding complexity.

2.3.1.2. Perceptual Compression

Unlike codecs that reduce statistical redundancy, perceptual video compression leverages properties of the human visual system to reduce the video size without compromising the perceived quality. It identifies regions of interest (ROIs) as spatial, temporal, or spatio-temporal areas critical for perception and encodes ROIs losslessly, while applying stronger compression to less critical parts. This process involves two phases: detecting ROIs with techniques ranging from user input to non-visual information [112], and ROI-aware encoding, which might take place during pre-processing or actual encoding [113].

2.3.1.3. Super Resolution (SR)

SR [114] refers to a computer-vision task that reconstructs high-resolution (HR) images from low-resolution (LR) versions. In video streaming, SR reduces network-bandwidth consumption by transmitting LR frames and reconstructing HR video at the recipient. While traditional SR relies on spatial-frequency substitution and geometric techniques, modern ML-based approaches employ GANs, CNNs, and other DNNs [115]. Despite improving video quality and bandwidth efficiency, ML-based SR techniques face challenges such as poor generalization, high parameter dimensionality, and balancing inference accuracy with speed.

2.3.2. Recent Results

Recent works at the ingestion stage commonly tackle tasks such as video encoding, analytics, and upload. These studies evaluate the effectiveness of their solutions within the stage via metrics of bandwidth utilization, encoding complexity, video quality, analytics accuracy, upload latency, and computational overhead. Additionally, some studies assess user experience by means of QoE models. To achieve their goals, the reviewed designs explore various approaches, including assistance from SR, transport-layer signals, and edge servers.

This section, along with Table 2.1, organizes our discussion of the recent works according to the methodology-based classification scheme presented in Section 2.2.1: intuition, theory (MIP, MPC, and other), and ML (SL and UL). In addition to the core technique and codec characteristics explained in Section 2.2.2, Table 2.1 describes each ingestion-stage design based on five stage-specific binary characteristics: (1) SR usage, (2) utilization of a well-defined QoE model in design or evaluation, (3) reliance on

Table 2.1: Designs at the ingestion stage of the end-to-end streaming pipeline (*u* abbreviates *unspecified*).

Name [reference]	Method		Year	Core technique	Codec	SR	QoE model	Transport-layer signals	Edge infrastructure	Bandwidth-efficiency evaluation
[116]	Intuition		2015	dynamic encoding ladder	H.264	✘	✘	✔	✘	✘
[117]			2016	switch between upload protocols	*u*	✘	✘	✘	✘	✘
[118]			2017	AIMD-based encoding-rate control	H.264	✘	✘	✘	✘	✘
NeuroScaler [119]			2022	zero-inference selection of anchors	VP9	✔	✘	✘	✘	✘
Vantage [120]	Theory	MIP	2019	regression heuristic	VP8, a VP9 predecessor	✘	✔	✔	✘	✘
LiveSRVC [121]		MPC	2021	SR	H.264	✔	✔	✘	✘	✔
[122]		Other	2017	DP, greedy heuristics	SVC	✘	✔	✘	✘	✘
CHN [123]			2019	knapsack-like problem, greedy rounding heuristic	*u*	✘	✘	✔	✔	✘
[124]			2019	relaxation-based heuristic	*u*	✘	✔	✘	✔	✘
DDS [125]			2020	adaptive feedback control	H.264	✘	✘	✘	✘	✔
LiveNAS [126]			2020	concave optimization problem, gradient ascent	*u*	✔	✘	✔	✘	✔
[127]	ML	SL	2017	CNNs	H.265-based	✘	✘	✘	✘	✘
[128]			2020	CNNs	ROI-based	✘	✘	✘	✘	✘
CrowdSR [129]			2021	unspecified DNNs	*u*	✔	✘	✘	✘	✘
DIVA [130]			2021	AlexNet variants (CNNs)	H.264	✘	✘	✘	✘	✔
MobileCodec [131]			2022	CNNs	MobileCodec	✘	✘	✘	✘	✘
DeepFovea [132]		UL	2019	Wasserstein GAN	DeepFovea	✘	✘	✘	✘	✘
Reducto [133]			2020	*k*-means clustering	H.264	✘	✘	✘	✘	✔
[134]			2023	AE	data-scalable	✘	✘	✘	✘	✘

transport-layer signals, (4) leverage of edge infrastructure, and (5) bandwidth-efficiency evaluation in the reviewed work.

2.3.2.1. Intuition-Based Methods

Guided by measurements of TCP uplink throughput in a radio access network, [116] intuitively reduces the number of bitrate levels in the encoding ladder and thereby conserves bandwidth. This technique combines real-time and historical throughput data, using the former for ongoing sessions and the latter at the start of sessions or

during handovers. [117] proposes dynamic selection of the upload protocol by a mobile broadcasting application. The application considers latency, join-time, goodput, and overhead metrics, picks one of them, evaluates this metric in real time, and periodically decides whether to switch to another upload protocol. While this method performs as well as the best protocol for each individual metric, the switching between protocols incurs undesirable delay. [118] monitors the average inter-arrival time of video frames and dynamically adjusts the encoding rate on a camera-equipped mobile device via the AIMD algorithm. By increasing the average encoding rate and decreasing the packet loss, the algorithm improves real-time upstreaming under changing network conditions. *NeuroScaler* [119] enhances the scalability of SR-based live streaming by lowering both overhead and encoding time of SR. The design includes a novel scheduler and enhancer of the anchor frames used by SR. The anchor scheduler leverages codec-level information to select the anchor frames in real time without any neural inference. The anchor enhancer complements a video codec with a simple image codec and employs the latter for compression of the anchor frames only.

2.3.2.2. Theory-Based Methods

Vantage [120] refers to a MIP-based approach that targets social live streams and improves QoE for time-shifted viewers through frame retransmissions. When bandwidth allows, it retransmits earlier frames at a higher bitrate, enhancing the experience for viewers watching with time shifts. Vantage employs MIP for retransmission scheduling. *LiveSRVC* [121] is an MPC-based solution for live-stream ingestion, aiming to decrease bandwidth usage and latency via SR. It compresses I-frames at the camera side and trains an SR model online to reconstruct them on the server. Guided by estimated uplink bandwidth, SR processing time, and accuracy, LiveSRVC uses MPC to select the I-frame compression ratio and chunk bitrates.

Other theory-based ingestion works include [122], which, similar to Vantage, strives to maximize video quality in live streaming for multiple clients with heterogeneous upload latencies. The design involves a series of algorithms that leverage a greedy low-complexity DP-based approach. Conversely, the *content harvest network (CHN)* [123] achieves both low latency and efficient bandwidth utilization during ingestion by employing edge devices as relays to direct traffic from broadcasters to servers. To determine the path for each broadcaster, CHN employs two strategies on different time scales. Whereas finding a globally optimal path is a nondeterministic polynomial time (NP) and NP-hard problem, a centralized server periodically solves it via a polynomial-time greedy rounding algorithm. [124] selects both the upload server and encoding bitrate to jointly maximize the video rate and minimize end-to-end latency. It develops algorithms for both one-hop-overlay and full-overlay architectures. The one-hop-overlay algorithm is an optimal polynomial-

time solution. The paper proves NP-completeness of the full-overlay problem and solves it with an efficient heuristic solution based on convex relaxation.

DNN-driven streaming (DDS) [125] refers to a theory-based solution where the camera-equipped device optimizes bandwidth usage across two streams to enhance inference accuracy while minimizing bandwidth consumption in analytics applications. The first stream transmits low-quality video to the server, which identifies ROIs for DNN inference. The second stream provides high-quality video for the detected ROIs, improving inference accuracy while managing bandwidth efficiently. DDS applies a Kalman filter to estimate base bandwidth and adjusts bandwidth usage by tuning the resolution and quantization parameter (QP). With a similar focus on camera uploads, *LiveNAS* [126] employs SR for high-quality live streaming. Along with the live video, the camera-equipped device also uploads high-quality frame patches. The server utilizes these patches to train a DNN for SR in real time. LiveNAS allocates upload bandwidth between the live video and patches by means of gradient ascent to maximize both video quality and DNN accuracy, while minimizing overhead for ingest clients.

2.3.2.3. ML-Based Methods

[127, 128] present SL-based codecs for perceptual compression, targeting improvements in coding efficiency. Compared to standard codecs, these designs increase video quality and decrease storage requirements while decreasing the encoding speed. [127] extends the H.265 codec by incorporating a hybrid compression algorithm that employs a CNN for spatial saliency and then extracts temporal saliency from motion information in the compressed domain. [128] introduces an ROI codec that combines CNNs with an entropy codec to achieve better encoding efficiency than previous ROI codecs, though its decoding performance is less effective. *CrowdSR* [129] enhances live streaming from low-end devices via SR-based video uploading. It periodically trains an SR model with high-quality video patches from similar content broadcasters. CrowdSR outperforms existing counterparts in regard to the peak signal-to-noise ratio (PSNR) [135] and structural similarity index measure (SSIM) [136]. In contrast, *DIVA* [130] improves video analytics efficiency by leveraging both camera-equipped device and server. It processes only key video frames on the camera-equipped device to avoid unnecessary uploads. Utilizing the sparse analytical data, the server trains CNNs, specifically variants of AlexNet [137], and sends them back to the camera-equipped device to identify I-frames for upload. This iterative approach enhances analytics performance and operates 100 times faster than real-time video. Recent work on neural codecs includes *MobileCodec* [131], which adopts a DNN architecture and SL to support efficient coding for mobile devices. It features an inter-frame module with fully convolutional operations, asymmetrical design for faster real-time decoding, and activation quantization with simulated straight-through gradient estimation.

Applying UL to perceptual compression, *DeepFovea* [132] proposes foveated coding that strengthens compression for areas outside the fovea. The codec employs a GAN to reconstruct realistic peripheral video from a minimal set of frame pixels. It operates quickly enough for HMDs and delivers superior perceptual quality in subjective evaluations. In contrast, *Reducto* [133] aims to reduce bandwidth consumption in UL-based video analytics. It tracks basic features, such as pixel and edge differences, and identifies relevant features for specific queries. Reducto relies on k-means clustering to establish a dynamic threshold for frame filtering at the camera-equipped device. By filtering out less important frames, it reduces upload traffic while maintaining analytics accuracy.

Among the latest advancements in neural codecs, [134] introduces a data-scalable codec that employs AEs trained with UL and custom loss function. This codec enhances compression quality with each new packet received and achieves the highest quality when there is no packet loss.

2.3.3. Main Takeaways

The review of recent research on ingestion-stage designs reveals a strong focus on live streaming. This emphasis stems from the growing importance and significant technical challenges of the live mode. The stricter end-to-end latency constraints of live streaming affect the ingestion stage and promote a trend toward integrated end-to-end streaming solutions. Another trend is the increasing computational role of camera-equipped devices able to offload processing from media servers. This offloading delivers faster video analytics and decreases upload bandwidth consumption. ML-based methods are increasingly prominent at the ingestion stage, either as core algorithms or supporting components. In particular, ML-based SR methods receive considerable attention and success, with a prevalence of adapting existing models and effective training strategies rather than developing new ML techniques.

2.4. Processing

2.4.1. Background

The processing stage lies between ingestion and distribution in the end-to-end streaming pipeline. It operates on dedicated or cloud servers and performs various tasks to support the adjacent stages. The essential task at the processing stage of the HAS pipeline is transcoding [138], which converts encoded video into multiple representations. Since transcoding produces compressed videos, it shares similarities with the video compression performed during the ingestion stage, making the background information in Section 2.3.1.1 relevant. However, there are key differences. While the primary goal of

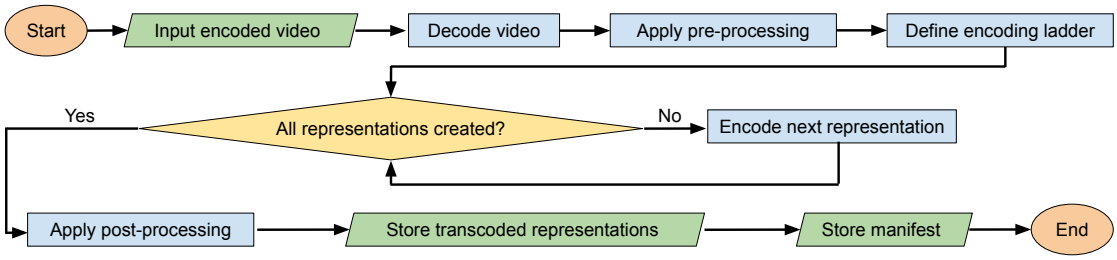


Figure 2.4: Transcoding at the processing stage of the end-to-end streaming pipeline.

encoding on the camera-equipped device is to efficiently utilize the ingestion bandwidth, transcoding leverages the superior computational and storage resources of the media server to create compressed videos suitable for distribution to a wide range of user devices.

Figure 2.4 presents a flowchart of transcoding. After receiving encoded video as input, the task decodes the video and applies pre-processing, such as noise filtering. Then, the task defines an encoding ladder by specifying the target bitrate, resolution, and frame rate of each representation. Transrating and transsizing refer to transcoding where the generated representations differ only in their bitrate or resolution, respectively. For each rung of the encoding ladder, the process re-encodes the decoded video to create a corresponding new representation. After post-processing, such as subtitle embedding, the transcoding task stores the created representations on the media server and records the encoding ladder in a manifest file.

Alongside transcoding, which is intrinsic to the HAS pipeline and directly impacts end-to-end streaming performance, the processing stage performs a variety of auxiliary tasks. Video splitting divides the video into smaller chunks for HTTP compatibility, typically ranging from 2 to 10 s in duration, with this variation significantly affecting the quality of video streaming [139]. Video editing alters the video content, e.g., by adding advertisements or removing censored material. Traditionally carried out at the processing stage, video analytics employs techniques from computer vision for object detection and image segmentation, classification, and recognition. Video storage on the media server is particularly important for VoD, where videos need to remain available over extended periods.

2.4.2. Recent Results

Our review of recent research at the processing stage focuses on its main task of transcoding. These studies typically aim to reduce processing time, energy consumption, storage needs, and bandwidth usage.

Again, we present the reviewed works according to the methodology-based classification scheme outlined in Section 2.2.1: intuition, theory (MIP and other), and ML (RL and SL). Besides the core technique and codec, which are relevant across all

Table 2.2: Transcoding designs at the processing stage (*u* abbreviates *unspecified*).

Name [reference]	Method		Year	Core technique	Codec	Type	Performance	Infrastructure
[140]	Intuition		2017	statistics-driven early termination	H.264 → H.265	*u*	processing	*u*
[141]			2018	joint crypto-transcoding	H.264, H.265	*u*	processing	*u*
EVSO [142]			2018	frame-rate adjustment	H.264	offline	energy	*u*
LwTE [143]	Theory	MIP	2021	MILP, binary search	H.265	hybrid	storage, processing	edge
ARTEMIS [144]			2023	MILP	*u*	online	processing, bandwidth	CDN
[145]		Other	2015	Markov model	H.264	hybrid	processing	CDN
[146]			2018	context-aware ladder optimization	H.264	offline	bandwidth	*u*
[147]			2020	knapsack-like optimization problem	H.264	hybrid	energy	*u*
MAMUT [148]	ML	RL	2018	multi-agent QL	H.265	online	processing, energy	*u*
DeepLadder [149]			2021	AC, dual-clip PPO, DNN with 1D CNNs	H.264	online	bandwidth, storage	*u*
[150]		SL	2018	DTs	H.265	online	processing, energy	*u*
[151]			2018	RFs	H.265	*u*	processing	*u*
FastTTPS [152]			2020	MLP	H.264	*u*	processing	*u*
HEQUS [153]			2021	NB classifiers	H.265 → VVC	*u*	processing	*u*

stages, Table 2.2 also characterizes the processing-stage designs based on: (1) optimization type as online, offline, or hybrid, (2) processing, energy, storage, energy, or bandwidth as performance improvement objectives, and (3) explicit consideration of edge or CDN infrastructure.

2.4.2.1. Intuition-Based Methods

To support fast low-complexity transcoding from H.264 to H.265, [140] employs intuitive statistics-driven heuristics for different types of coding units (CUs). These heuristics allow for early termination of CU partitioning and prediction unit mode selection. [141] deals with transcoding video streams encrypted in the H.264 or H.265 formats. Because decrypting and re-encrypting these streams introduces significant latency, this work develops a joint crypto-transcoding scheme that enables transcoding of encrypted video streams without decrypting them or exposing the decryption key at intermediate devices. To reduce energy consumption on mobile devices, *environment-aware video streaming optimization (EVSO)* [142] considers the device's battery status and generates encoding ladders that adjust the frame rate of different video chunks based on a new metric of perceptual similarity.

2.4.2.2. Theory-Based Methods

To minimize storage and processing requirements, both *light-weight transcoding at the edge (LwTE)* [143] and *adaptive bitrate ladder optimization for live video streaming (ARTEMIS)* [144] rely on MIP. In LwTE, the edge server performs partial transcoding based on the optimal CU partitioning structure received from the origin server. By applying binary search to a mixed-integer linear programming (MILP) formulation, LwTE heuristically distinguishes between popular and unpopular video chunks. For unpopular chunks, it stores only the highest bitrate level and generates lower bitrate levels on the fly through metadata-accelerated transcoding. In contrast, ARTEMIS dynamically defines the encoding ladder for live streaming sessions by considering content complexity, network conditions, and detailed client feedback in a standard format [154]. ARTEMIS advertises many representations via a mega-manifest file and employs MILP to select a smaller subset of these representations for the encoding ladder.

Other theory-based works include [145], where a CDN performs online just-in-time transcoding of a video chunk to the needed bitrate only when a user requests it. This design relies on a Markov model to predict the bitrate requested for the next chunk, enabling the CDN to start delivering the transcoded chunk immediately upon receiving the request. [146] explores context-aware encoding and formulates encoding-ladder definition as an optimization problem that models the client's bandwidth estimates and viewport sizes as stationary random processes. To support energy-efficient transcoding, [147] selects between three options: offline transcoding, online transcoding, and serving the chunk at a lower than requested bitrate. The selection seeks to maximize video quality within a limit imposed on the total transcoding time, formulates a knapsack-like problem, and solves the problem via a greedy heuristic.

2.4.2.3. ML-Based Methods

MAMUT [148] and *DeepLadder* [149] are RL-based designs for efficient real-time transcoding. MAMUT employs multi-agent Q-learning (QL) in an environment with multiple users, where three agents collaboratively adjust the number of encoding threads, QP, and processor frequency. This optimization seeks to maximize a reward function that combines the frame rate, bitrate, PSNR, and power consumption. On the other hand, DeepLadder leverages content features, available bandwidth, and storage costs to transcode each chunk according to an encoding ladder defined via a dual-clipped version of proximal policy optimization (PPO).

[150, 151] apply SL to limit the encoder's parameter search and thereby reduce transcoding time. [150] employs DTs to constrain the maximum CTU depth, aiming to balance transcoding time, energy consumption, and video quality. [151] accelerates cascaded pixel-domain transcoding by employing two RF classifiers to set upper and

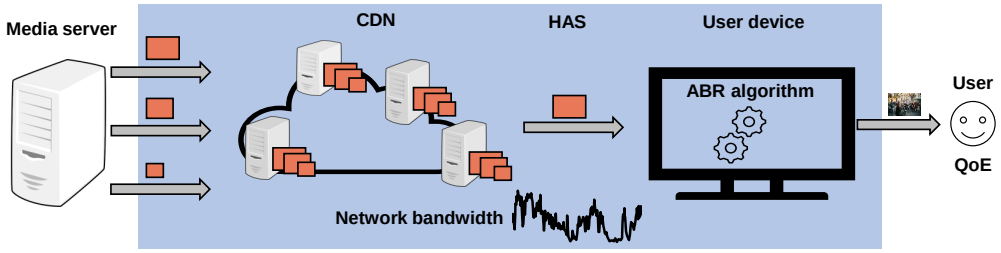


Figure 2.5: The distribution stage of the end-to-end VoD streaming pipeline.

lower limits on the CTU depth. *Fast video transcoding time prediction and scheduling (FastTTPS)* [152] considers features of source videos, trains an MLP to predict transcoding time, and leverages the predictions to schedule parallel executions of transcoding tasks. *HEVC-based quadtree splitting (HEQUS)* [153] reduces the encoder's parameter search for transcoding from H.265 to VVC. It trains NB classifiers to partition the first QT level into 128×128 blocks and uses the H.265 CU partitioning to guide QT splitting decisions for 64×64 blocks and lower levels.

2.4.3. Main Takeaways

Recent research efforts at the processing stage put a key focus on faster processing and lower power consumption. In particular, transcoding acceleration enables on-the-fly definition of encoding ladders, which not only decreases storage demands but also aligns with the growing trend toward live streaming. The explicit consideration of distribution-stage infrastructure, such as CDN or edge servers, reflects a closer integration across pipeline stages. ML-based methods are increasingly prominent in processing-stage designs and, in contrast to the ingestion stage, tend to employ simple models rather than deep networks. Additionally, most designs assume the use of H.264 or H.265 codecs rather than more advanced options.

2.5. Distribution

2.5.1. Background

The end-to-end streaming pipeline concludes with the distribution stage, which delivers the requested video to the user device and plays it on the screen. Figure 2.5 illustrates the distribution stage of HAS for VoD. At this stage, the user device requests one video chunk at a time from the CDN, which caches each chunk in multiple representations provided by the media server. The CDN supports scalable low-latency delivery by utilizing its extensive network of edge servers spread across different geographical regions. The ABR algorithm on the user device dynamically chooses the

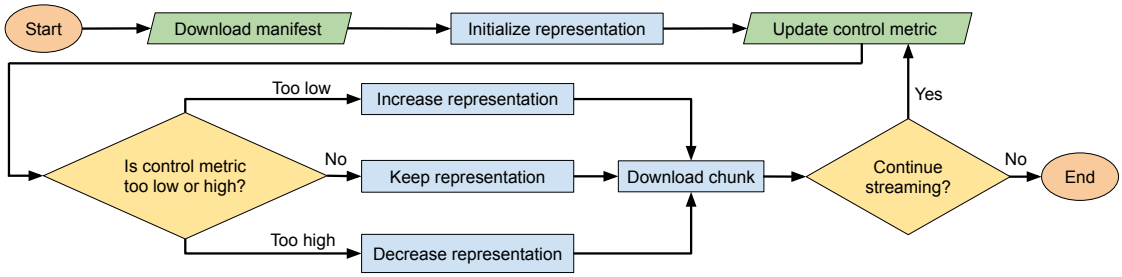


Figure 2.6: The ABR algorithm.

appropriate representation for the next requested chunk based on predictions of varying network bandwidth. This algorithm aims to balance uninterrupted playback with high video quality, ultimately ensuring high QoE for the user. Live streaming employs shorter chunks, downloads them from the camera-equipped device to the user device in real time, and imposes more stringent requirements on distribution, prompting different approaches to CDN support and QoE improvement. This chapter focuses on the key ABR, CDN, and QoE aspects of the distribution stage.

2.5.1.1. ABR Algorithms

The ABR algorithm dynamically selects the chunk representation and serves as a cornerstone of HAS, with HLS and DASH being the predominant HAS protocols. While HLS commonly employs a chunk duration of 6 s (10 s originally) and is compatible with the H.264 or H.265 codecs, DASH typically has a chunk duration between 2 and 10 s and is codec-agnostic. The chapter focuses on the prevailing HAS approach that uses client-side ABR algorithms.

Figure 2.6 depicts the ABR algorithm as a flowchart. At the start of the streaming session, the client downloads a manifest file from the media server. The manifest includes an encoding ladder that describes the available representations for each video chunk in terms of their bitrate, resolution, and frame rate. The ABR algorithm updates a control metric based on the monitored network conditions. For example, the control metric is typically playback-buffer occupancy and network-bandwidth estimate in, respectively, buffer-centric and throughput-centric ABR algorithms. If the control metric indicates that the current representation is too low, the ABR algorithm increases the representation for the next chunk to enhance video quality. If the control metric is too high, the algorithm decreases the representation to avoid video stalling and rebuffering, which occur when chunks arrive too late for smooth playback. Otherwise, the representation remains unchanged. In all three cases, the client downloads the next chunk in the selected representation. This cycle of updating the control metric, selecting the appropriate representation, and downloading the chunk continues until the end of the streaming session.

Representation selection is challenging due to a priori unknown network conditions, mismatches between the manifest-file descriptions and actual chunk bitrates, large gaps between the bitrates of adjacent representations, and conflicting performance objectives. Because optimal ABR control is an NP-hard problem [155], practical ABR algorithms employ various heuristics, e.g., predicting the available network bandwidth from the client's historical throughput measurements.

2.5.1.2. CDN Support

A CDN refers to a system of cache servers distributed across wide geographical areas to improve the performance of content delivery from SPs to end users [156]. The CDN stores videos and other content collected from SPs' origin servers in cache servers placed near users, reducing network traffic and enabling low-latency content delivery [157]. Though originally optional, CDNs are indispensable in the modern Internet ecosystem and handle estimated 56% and 72% of all Internet traffic in 2017 and 2022, respectively [5].

Economic relationships with SPs form the basis for classifying CDNs as public, private, or hybrid. A public CDN, e.g. Akamai [158], acts as a third party and charges the SPs for its content-delivery services. A private CDN belongs to the same organization as the SPs utilizing it, while a hybrid CDN serves both internal and external SPs. Due to CDNs' differences in scalability, pricing, and QoE across regions and time [159], SPs often deliver content over multiple CDNs.

Standards such as common media client data (CMCD) [154] and common media server data (CMSD) [160], introduced in 2020 and 2022 respectively, enable information exchange between a CDN and clients to support data analysis and QoE monitoring. Additionally, edge infrastructure extends the original CDN concept by involving ISPs in content caching and offers new options for video streaming [161].

2.5.1.3. QoE

In contrast to the earlier notion of QoS, which encompasses individual network-level metrics such as packet loss, latency, and throughput, QoE captures the user's subjective satisfaction with the overall performance of a streaming service [35]. QoE is crucial for SPs because user satisfaction strongly correlates with customer attraction and retention and, ultimately, provider revenues. However, user perception of service performance is complex and depends on numerous IFs, such as network bandwidth, latency, and video quality [162].

Assessing QoE is challenging due to its subjective interdisciplinary nature. Direct measurement typically involves subjective tests where users rate their streaming experience. These tests typically take place in controlled lab environments and follow well-established protocols informed by user experience design [71]. Online crowdsourcing

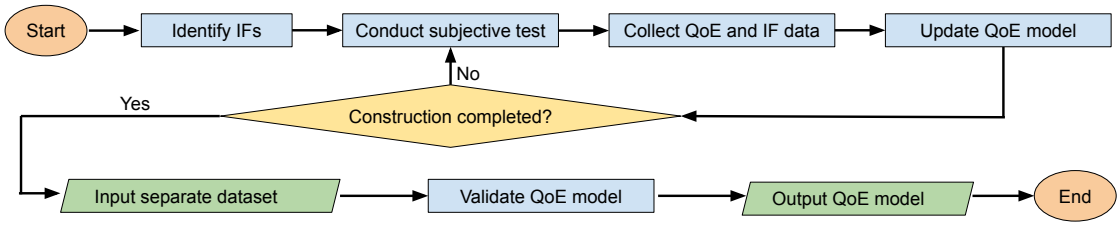


Figure 2.7: QoE modeling.

improves testing scalability and weakens control over experimental settings [163]. The predominant approach to QoE evaluation is indirect and relies on subjective tests to build a QoE model that expresses QoE as a function of objectively measurable IFs. Data science enhances the predictive power of QoE models via ML-based methods. QoE models commonly represent QoE in terms of the mean opinion score (MOS) [164], the average rating given by users in a subjective study.

Figure 2.7 illustrates QoE modeling that constructs a QoE model iteratively. The process identifies the IFs of the QoE model and enters a cycle of conducting a subjective test, recording the respective IF values, collecting a user-provided QoE score, and mapping the IFs to the score to update the QoE model. This iterative refinement allows the current model to inform the configuration of the subsequent test, thereby reducing the number of subjective tests needed to develop an accurate QoE model [2]. After the construction is complete, the process utilizes a separate dataset to validate the model and outputs the validated QoE model.

Once constructed, a QoE model supports automatic QoE computation based on the objectively measurable IFs, eliminating the need for human feedback and enabling QoE evaluation at scale. Existing QoE models vary widely in terms of the IFs considered and the methods used for construction [165]. Despite the significance of QoE and its models, their treatment often lacks standardization and rigor, creating opportunities for improvement [4].

2.5.2. Recent Results

2.5.2.1. ABR Algorithms

Recent research on ABR algorithms aims to improve QoE for end users, either directly or indirectly. Direct approaches explicitly incorporate a QoE model into the control metric of the ABR algorithm, e.g., employing a QoE model as the reward function in an RL-based ABR algorithm. Indirect approaches focus on individual IFs, such as the bitrate, PSNR, SSIM, and video multimethod assessment fusion (VMAF) [166] to capture video quality. In addition to video quality and its stability, prominent IFs in the design and evaluation of ABR algorithms include the frequency and duration of video stalls. Efficient utilization of

network bandwidth and its fair distribution among multiple sessions are common design goals. For live streaming, ABR algorithms also prioritize reducing latency.

We structure our coverage of ABR designs in accordance with the classification scheme given in Section 2.2.1: intuition-based (buffer-centric, throughput-centric, and hybrid), theory-based (MIP, MPC, PID, LO, BO, and other), and ML-based (RL, IL, SL, and UL), with the RL-based ABR algorithms categorized further by their reliance on A3C, A2C, AC, or other methods. Tables 2.3, 2.4, and 2.5 summarize the intuition-based, theory-based, and ML-based ABR algorithms, respectively. In addition to the universal core technique and codec characteristics, each table describes the reviewed works with respect to: (1) their application mode as VoD or live streaming, (2) SR usage, (3) employment of a QoE model in design or evaluation, (4) bandwidth-efficiency evaluation, and (5) bandwidth-fairness evaluation.

Intuition-based methods: Laying the groundwork for buffer-centric designs, a series of *buffer-based algorithms (BBAs)* [167] map the occupancy level of the playback buffer to a control metric. BBA-0 employs piecewise linear mapping of the buffer occupancy to a bitrate. BBA-1 performs the mapping to a chunk size. BBA-2 extends BBA-1 by estimating the available network bandwidth and increasing the bitrate more aggressively during a startup phase. *Segment-aware rate adaptation (SARA)* [168] enhances the manifest file with chunk sizes and switches between its four adaptation modes depending on the buffer occupancy. Aiming to eliminate video stalls, the *adaptation and buffer management algorithm (ABMA+)* [169] relies on buffer-occupancy mapping to characterize the rebuffering probability.

Among throughput-centric schemes, [170] caches video chunks on an access point (AP) to support effective ABR streaming over the wireless link from the AP to the client. While the AP selects chunks for prefetching into the cache, the client determines which chunks to request from either the AP or a remote server. *Low-latency prediction-based adaptation (LOLYPOP)* [171] targets live streaming and strives to improve QoE by optimizing the operating point, a metric that combines latency, stall frequency, and bitrate-change frequency. LOLYPOP predicts TCP throughput over periods ranging from 1 to 10 s and assesses the prediction error. Interestingly, the study finds that the simple method of using the last sample as the prediction is the most accurate. Developed for streaming over mobile networks, *adaptive rate-based intelligent HTTP streaming (ARBITER+)* [172] addresses dynamic network conditions and bitrate variability through techniques such as tunable smoothing and hybrid throughput sampling. *Playback rate and priority adaptive bitrate selection (PREPARE)* [173] is a throughput-centric ABR algorithm that accounts for client priority and playback speed. PREPARE improves average bitrate and stability by involving the server into prediction of the network bandwidth. Designed for live streaming, *standard low-latency video control (STALLION)* [174] uses a sliding window to measure the mean and standard deviation of both bandwidth and latency. The implementation of

Table 2.3: Intuition-based ABR algorithms at the distribution stage of the end-to-end streaming pipeline (*u* abbreviates *unspecified*).

Name [reference]	Method	Year	Core technique	Codec	Mode	SR	QoE model	Bandwidth efficiency evaluation	Bandwidth fairness evaluation
BBA [167]	buffer-centric	2014	linear piecewise mapping	*u*	VoD	✘	✘	✘	✘
SARA [168]		2015	switch between adaptation modes	*u*	VoD	✘	✘	✘	✘
ABMA+ [169]		2016	rebuffering-probability characterization	*u*	VoD	✘	✘	✔	✘
[170]	throughput-centric	2015	proxy caching	*u*	VoD	✘	✘	✘	✘
LOLYPOP [171]		2016	stall-probability prediction	H.264	live	✘	✘	✘	✘
ARBITER+ [172]		2018	hybrid throughput sampling	H.264, H.265	VoD	✘	✘	✘	✔
PREPARE [173]		2019	server-client cooperation	*u*	VoD	✘	✘	✔	✔
STALLION [174]		2020	sliding-window measurement	*u*	live	✘	✘	✘	✘
FESTIVE [175]	hybrid	2014	stateful delayed bitrate update	*u*	VoD	✘	✘	✔	✔
PANDA [176]		2014	AIMD-based estimation	*u*	VoD	✘	✘	✔	✔
SQUAD [177]		2016	spectrum minimization	*u*	VoD	✘	✘	✘	✔
Oboe [178]		2018	offline parameter optimization	*u*	VoD	✘	✘	✘	✘
BANQUET [179]		2021	brute-force search	H.264	VoD	✘	✔	✔	✘

STALLION in dash.js, a popular streaming client, outperforms the client's built-in ABR algorithm by significantly increasing the bitrate and decreasing the number of stalls.

Representing a hybrid approach, the *fair, efficient, and stable adaptive algorithm (FESTIVE)* [175] combines several mechanisms to ensure efficiency, fairness, and stability in ABR streaming to multiple clients. These mechanisms include randomized scheduling of chunk requests, harmonic-mean estimation of network bandwidth, and stateful bitrate selection with delayed updates. Pursuing similar goals, *probe and adapt (PANDA)* [176] incorporates estimation, smoothing, quantization, and scheduling techniques and, in particular, applies AIMD to estimate network bandwidth. Accounting for interactions between DASH and TCP, *spectrum-based quality adaptation (SQUAD)* [177] improves QoE by minimizing the spectrum, a metric that reflects bitrate variation. For a given ABR algorithm, *Oboe* computes an offline map of network conditions to an optimal configuration of algorithm parameters and automatically tunes these parameters online in response to current network conditions [178]. *Balancing quality of experience and traffic (BANQUET)* [179] aims to minimize the traffic volume while providing the QoE level specified by either the user or the streaming provider. To estimate the impact of bitrate choices on traffic and QoE, BANQUET employs brute-force search across all

Table 2.4: Theory-based ABR algorithms at the distribution stage of the pipeline (u abbreviates *unspecified*).

Name [reference]	Method	Year	Core technique	Codec	Mode	SR	QoE model	Bandwidth efficiency evaluation	Bandwidth fairness evaluation
OSCAR [180]	MIP	2016	MINLP	H.264	VoD	✘	✔	✔	✘
RobustMPC and FastMPC [46]	MPC	2015	harmonic-mean estimation	u	VoD	✘	✔	✘	✘
IAA [181]		2018	TF-IDF	u	VoD	✘	✔	✘	✘
LDM [182]		2020	frame dropping	H.264	live	✘	✔	✘	✘
Fugu [183]		2020	transmission-time prediction	H.264	VoD	✘	✔	✘	✘
iMPC [184]		2021	iLQR-based linearization	H.264	live	✘	✔	✘	✘
PIA [185]	PID	2017	PI control with linearization	u	VoD	✘	✔	✘	✘
QUAD [186]		2019	least-square optimization	H.264, H.265	VoD	✘	✔	✔	✘
BOLA [187]	LO	2020	utility maximization	u	VoD	✘	✔	✘	✘
Elephanta [188]		2020	renewal system	u	VoD	✘	✔	✘	✘
ERUDITE [189]	BO	2019	parameter configuration	u	VoD	✘	✔	✘	✘
[190]		2021	Gaussian processes	u	VoD	✘	✔	✘	✘
QUETRA [191]	Other	2017	M/D/1/K queuing	u	VoD	✘	✔	✘	✔
ACAA [192]		2019	DP	u	VoD	✘	✔	✘	✘

possible bitrate patterns for the next few chunks via predictions of buffer transitions and throughput.

Theory-based methods: *Optimized stall-cautious adaptive bitrate (OSCAR)* [180] represents a MIP-based approach. For a transient range of the buffer occupancy, it models the available network bandwidth using the Kumaraswamy distribution and formulates bitrate adaptation over a sliding look-ahead window as a mixed-integer nonlinear programming (MINLP) problem. OSCAR's optimization objective combines a switching penalty with bitrate utility.

MPC forms a prominent basis for recent ABR algorithms. Contributing several innovations, [46] introduces two MPC-based algorithms: *RobustMPC* and *FastMPC*. While RobustMPC performs better, FastMPC incurs significantly lower overhead. Additionally, this paper proposes a QoE model that underpins many subsequent ABR designs. As an MPC enhancement aimed at improving QoE, the *interest-aware approach (IAA)* [181] adjusts the bitrate by considering the user's interest in video scenes. IAA embeds content properties into the manifest file, allowing the client to analyze these properties and quantify the user's interest in the content via the term frequency-inverse document frequency (TF-IDF) method. *LDM* [182] utilizes MPC for live streaming and drops frames to ensure low latency. *Fugu* [183] is an MPC-based approach that predicts transmission time for each chunk via a DNN trained via SL in situ, i.e., in the actual deployment environment. To achieve low latency, *iLQR based model predictive control (iMPC)* [184]

combines MPC with the iterative Linear Quadratic Regulator (iLQR). iMPC employs MPC to predict the available network bandwidth and iteratively linearizes the control system around its operation point to determine the bitrate via iLQR.

Relying on PID as its main method, *PID-control based ABR streaming (PIA)* [185] removes the derivative (D) component from the standard PID controller and linearizes the closed-loop control system to maintain the buffer occupancy at a targeted level. PIA also equips this proportional-integral (PI) controller with mechanisms for faster initial ramp-up, reduction of bitrate fluctuation, and avoidance of bitrate saturation. Using the same PI controller as PIA, *quality-aware data-efficient streaming (QUAD)* [186] strives to maintain video quality at an intended level to prevent stalls, enhance playback smoothness, and reduce bandwidth consumption.

LO-based designs include the *buffer occupancy based Lyapunov algorithm (BOLA)* [187], which jointly optimizes playback smoothness and bitrate utility under a rate stability constraint. BOLA provides theoretical guarantees on the achieved utility and performs excellently in practice. *Elephanta* [188] addresses the diversity of QoE perception among different users. It offers an interface for users to adjust QoE perception parameters, models video streaming as a renewal system, and selects the bitrate by minimizing a user-specific function that combines penalties and drift.

By employing BO, the *deep neural network for optimal tuning of adaptive video streaming controllers (ERUDITE)* [189] configures the parameters of the *feedback linearization adaptive streaming controller (ELASTIC)* [193] to jointly optimize QoE and control robustness. At runtime, ERUDITE uses an offline-trained CNN to tune the controller parameters in accordance with real-time bandwidth measurements and video features. [190] develops a context-aware ABR algorithm to maintain QoE at the minimum level acceptable to the user. This algorithm leverages Gaussian processes to determine the target QoE level and then selects a bitrate via BANQUET.

Among other theory-based ABR algorithms, *queuing theory-based rate adaptation (QUETRA)* [191] uses the Markovian deterministic single-server finite-capacity (M/D/1/K) queuing model to assess the buffer occupancy. The algorithm takes into account the buffer capacity and network bandwidth, adjusting the bitrate to keep the buffer approximately half-full. QUETRA is notable for not requiring parameter tuning and performs well across various heterogeneous scenarios. To address the diversity of QoE perception among users, *affective content-aware adaptation (ACAA)* [192] considers the emotional relevance of content for different users. ACAA characterizes video chunks and users with confidence levels for six basic emotions, formulates a QoE maximization problem based on this emotional information, and solves the problem by means of DP.

ML-based methods: *Pensieve* [51] revolutionizes ABR streaming by applying deep RL (DRL) and, in particular, the A3C method. Pensieve formulates bitrate selection as a DRL problem and solves it using A3C where the function approximator combines one-

Table 2.5: ML-based ABR algorithms at the distribution stage of the pipeline (u abbreviates *unspecified*).

Name [reference]	Method		Year	Core technique	Codec	Mode	SR	QoE model	Bandwidth efficiency evaluation	Bandwidth fairness evaluation
Pensieve [51]	RL	A3C	2017	DNN with 1D CNNs	u	VoD	✘	✔	✘	✘
NAS [194]			2018	content-aware DNNs, SR	H.264	VoD	✔	✔	✔	✘
SRAVS [195]			2020	CNN, SR	u	VoD	✔	✔	✘	✘
Grad [196]			2020	DNN with 1D CNNs, HYBJ	SVC	VoD	✘	✔	✔	✘
[197]			2020	LSTM, manifest update	H.264	VoD	✘	✔	✘	✔
ANT [198]			2021	CNN, k-means clustering	u	VoD	✘	✔	✘	✘
FedABR [199]			2023	CNN, LSTM, FL	H.264	VoD	✘	✔	✘	✘
Ahaggar [200]		A2C	2023	DPPO, DNN with 1D CNNs	H.264	VoD	✘	✔	✔	✘
Fastconv [201]		AC	2019	CNNs	H.264	VoD	✘	✔	✘	✘
MLMP [202]			2020	PPO, LSTM	u	VoD	✘	✔	✘	✘
Vabis [203]			2020	ACKTR, DNNs	u	live	✘	✔	✘	✘
Stick [204]			2020	DDPG, DNN with 1D CNNs	H.264	VoD	✘	✔	✘	✘
Tiyuntsong [205]		Other	2019	self-play RL, GAN	u	VoD	✘	✘	✘	✘
Ruyi [59]			2022	DQL, DNN with CNNs	H.264	VoD	✘	✔	✘	✘
PiTree [206]	IL		2019	DTs	u	VoD	✘	✘	✘	✘
[207]			2020	DTs	u	VoD	✘	✘	✘	✘
Comyco [208]			2020	DNN	H.264	VoD	✘	✔	✘	✘
SMASH [209]	SL		2020	RFs	H.264	VoD	✘	✘	✘	✘
Karma [210]			2023	GPT	H.264	VoD	✘	✔	✘	✘
Swift [211]	UL		2022	AEs	LNCs	VoD	✘	✔	✔	✘

dimensional (1D) CNNs and fully connected layers. The DNN supports different encoding ladders. To accelerate state transitions, Pensieve trains its DNN with a chunk-level simulator, a technique that influences many subsequent DRL-based ABR approaches.

NAS [194] is another A3C-based algorithm that leverages content-aware DNNs and anytime prediction to improve QoE via SR. For each video, the server trains multiple DNNs of different sizes and performance levels. The client picks the largest DNN able to operate in real time. Furthermore, each DNN is scalable and consists of multiple layers. This enables the client to progressively download the entire DNN, immediately benefit from the downloaded DNN layers, and dynamically select a DNN configuration for SR of the current frames. NAS employs A3C to balance bitrate selection with progressive DNN download. *Super-resolution based adaptive video streaming (SRAVS)* [195] also combines A3C with SR. Using an SR CNN [212] for video reconstruction, SRAVS maintains separate downloading and playback buffers. The separation decouples bitrate selection from reconstruction decisions, allowing for independent optimization of both processes.

Grad [196] applies A3C to design ABR algorithms for SVC-encoded videos. It

mitigates SVC-related coding overhead and improves QoE through *jump-enabled hybrid coding (HYBJ)*, where a single layer delivers multiple levels of video-quality enhancement. [197] jointly maximizes QoE and fairness in video streaming to multiple clients over a shared bottleneck link. Its A3C actor incorporates a long short-term memory (LSTM) layer, and the server dynamically configures the manifest file based on transport-layer signals about the loss rate. With throughput measurements underlying many ABR algorithms, *accurate network throughput (ANT)* [198] seeks to precisely model the full spectrum of available network bandwidth. ANT performs k-means clustering of throughput traces over short periods, trains a CNN for cluster-specific bandwidth prediction over the next period, and utilizes the prediction to select the bitrate via A3C. Also based on A3C, *FedABR* [199] provides faster training and preserves data privacy via federated learning (FL). After receiving from multiple clients their locally trained ABR policies, the FedABR server produces a global aggregate ABR policy and disseminates it back to the clients for further refinement of their ABR algorithms based on local data.

Ahaggar [200] trains A2C, a synchronized variant of A3C, with distributed PPO (DPPO) for server-side bitrate adaptation across multiple clients. Ahaggar leverages CMCD and CMSD for communication with clients and accelerates learning in new network conditions through meta-RL. Additional ABR solutions in the general AC category include *Fastconv* [201], which supports the fast training of a simple AC network by prepending an adapter that converts highly fluctuating input features into a more stable signal. The *meta-learning framework for multi-user preferences (MLMP)* [202] utilizes multi-task DRL with PPO for policy updates, ensuring that bitrate adaptation for different users accounts for user-specific sensitivities to three QoE metrics. Designed for low-latency live streaming, the *video adaptation bitrate system (Vabis)* [203] relies on actor critic using Kronecker-factored trust region (ACKTR) in its server-side ABR algorithm and operates at the granularity of frames to synchronize state information during training and testing. Vabis also incorporates three playback modes on the client side and a specialized ABR regime for poor network conditions. *Stick* [204] combines the deep deterministic policy gradient (DDPG) with BBA to improve ABR performance and reduce computational costs. Stick uses DDPG to train an AC network that controls the buffer-occupancy boundaries within the BBA approach.

Tiyuntsong [205] and *Ruyi* [59] represent other RL-based ABR solutions. Tiyuntsong employs self-play RL, where two ABR algorithms compete against each other in the same streaming environment. The rewards for the RL agents come from wins and losses in this ongoing competition, rather than from traditional QoE metrics. Additionally, each RL agent in Tiyuntsong utilizes a GAN to extract hidden features from extensive historical data. Ruyi integrates user preferences into its QoE model and leverages the model to train a deep QL (DQL) algorithm. Ruyi allows users to provide their preferences in real time, enabling adaptation to these dynamic preferences without model retraining.

Relying on IL, *PiTree* [206] employs teacher-student learning in a simulated video player to convert DNN-based and other sophisticated ABR algorithms into accurate DT representations, thereby enabling the efficient online operation of these algorithms. Inspired by PiTree, [207] uses DTs to reconstruct proprietary ABR algorithms in a human-interpretable manner allowing domain experts to inspect, understand, and modify the DT representations of the algorithms. *Comyco* [208] incorporates a solver to generate expert ABR policies aimed at maximizing QoE and trains a DNN by cloning the behavior of these expert policies. It embraces lifelong learning through continuous updates of the DNN with newly collected traces.

Supervised machine learning approach to adaptive video streaming over HTTP (SMASH) [209] and *Karma* [210] are SL-based designs. SMASH trains an RF classifier on outputs of nine existing ABR algorithms across various streaming scenarios, while Karma employs causal sequence modeling on a multidimensional time series and trains a GPT via SL to enhance the generalizability of ABR decisions. Based on UL, *Swift* [211] addresses the challenges of coding overhead and latency in layered coding. It incorporates a chain of AEs to create residual-based layered codes on the server side, a single-shot decoder on the client side, and a Pensieve-like ABR algorithm compatible with *layered neural codecs (LNCs)*.

2.5.2.2. CDN Support

Like ABR algorithms, CDNs ultimately aim to improve QoE for end users. Recent research on CDN support focuses on achieving this goal by improving the integration of CDNs into the streaming pipeline. This includes coordinating with transcoding designs at the processing stage, collaborating with client-side ABR algorithms, deploying CDN servers, assigning users to appropriate servers, and enhancing caching performance. The proposed solutions are either specific to video streaming or also applicable to other types of traffic. Additionally, some studies investigate the utility of edge computing for video streaming.

Intuition-based works include the *sequential auction mechanism (SAM)* [213], which operates in a crowdsourced CDN where third-party edge devices supplement CDN servers and charge SPs for leased cache space. Another example is *intelligent network flow (INFLOW)* [214], an intuition-based design for dynamically selecting the most suitable CDN from multiple options. It uses measurements from video players to predict available network bandwidth and latency via LSTM. Guided by these predictions and business constraints, INFLOW intuitively selects the appropriate CDN for each player and updates the manifest file accordingly.

Theory serves as a major foundation for CDN designs. The *video delivery network (VDN)* [215] exemplifies video-specific CDN optimizations and incorporates a centralized control plane that constructs distribution trees for videos to enable scalable and highly

responsive CDN operation. VDN formulates the tree construction as an integer program and approximates the program through initial solutions and early termination. To improve upon traditional CDN caching heuristics, *AdaptSize* [216] utilizes a Markov model for content admission into the cache. To address both caching and transcoding in a radio access network, [217] formulates an integer linear programming (ILP) problem to minimize CDN costs and solves it with a greedy heuristic. For enhancing ABR performance, [218] monitors video streaming of two popular SPs across three major CDNs and develops a CDN-aware variant of RobustMPC. In contrast, *FastTrack* [219] aims to minimize the probability that stall duration in CDN-assisted video streaming exceeds a predefined threshold. FastTrack achieves this by formulating a non-convex optimization problem, dividing it into four subproblems, and solving them iteratively with an algorithm that replaces the non-convex objective function with convex approximations. To improve QoE by combining SR with edge computing, *video super-resolution and caching (VISCA)* [220] caches LR chunks at the edge, accounts for chunk quality and request frequency in the eviction policy, increases resolution via SR, and streams videos to players via an edge-based ABR algorithm.

Representing ML-based solutions, *learning-based edge with caching and prefetching (LEAP)* [221] employs a DNN to prefetch and cache chunks at the edge, predicting QoE in scenarios of cache hit vs. cache miss. Meanwhile, *RL-Cache* [222] utilizes a feedforward neural network for cache admission and trains this network via a new DRL method that relies on direct policy search.

2.5.2.3. QoE

Since QoE models are essential for both evaluating and designing video streaming systems, research in this area heavily focuses on the interdisciplinary topic of QoE modeling. The main objective is to increase the predictive power of QoE models by applying advanced data-science techniques and incorporating new IFs, such as content characteristics and user engagement. Recent studies also emphasize personalization of QoE models to provide better service for individual users.

Reliance on intuition is common in QoE modeling. *YouQ* [223] contributes a novel modeling technique that supports subjective tests on Facebook's social media platform. Many ABR proposals come with intuition-based improvements to QoE models. For example, while [46] introduces a QoE model that includes video quality as an IF, BOLA [187] redefines this factor as the logarithm of the ratio between the bitrate and the lowest bitrate in the encoding ladder. Comyco [208] further changes this IF to VMAF. In contrast, the QoE model proposed by *SENSEI* [224] incorporates dynamic sensitivity to video content.

Recent theory-based research explores the relationship between user engagement and QoE. Whereas the queuing-theoretic analysis in [225] shows a strong correlation

between these two notions, *VidHoc* [226] utilizes user engagement as a proxy for QoE in its modeling. Specifically, VidHoc dynamically limits available network bandwidth and leverages the collected data to construct a personalized QoE model via regret minimization.

Among ML-based studies, [227] predicts QoE from facial expressions and gaze direction, while [228] considers DTs, RFs, and k-Nearest Neighbors algorithm (k-NN) for QoE prediction based on user engagement and other factors. *P.1203* [229] refers to a standard QoE model that utilizes RFs to predict MOS on a five-point scale. *LSTM-QoE* [230] models QoE via an LSTM network. Meanwhile, *video assessment of temporal artifacts and stalls (Video ATLAS)* [231] expresses QoE by applying SVR to features related to perceptual quality, rebuffering, and memory effects. To personalize QoE models, [232] performs FL on sparse data and accounts for changes in IFs over time. Guided by user experience design and involving the user in a brief series of subjective assessments, *iQoE* [2] iteratively constructs an accurate personalized QoE model through active learning. Lastly, *Jade* [233] relies on DRL with PPO to train a QoE model based on the relative ranks, rather than the absolute values, of subjective scores.

2.5.3. Main Takeaways

Recent research on the ABR, CDN, and QoE aspects at the distribution stage primarily focuses on ABR algorithms, particularly for the VoD streaming mode. ABR algorithms increasingly rely on ML and, especially, DRL and actor-critic methods. Studies on ABR algorithms for live streaming are less extensive, partly because the HAS paradigm offers fewer opportunities for latency reduction, which is critical for live streaming. Additionally, the adoption of DRL-based ABR algorithms in live streaming is challenging due to their high computational demands. For similar reasons, the use of SR at the distribution stage remains relatively rare compared to the ingestion stage. Regarding codecs, the research tendency mirrors that at the processing stage, with a predominant reliance on H.264 or H.265 as opposed to cutting-edge proprietary alternatives. However, recent work with new layered codecs shows promising results.

The general trend toward integrated designs is evident at the distribution stage, particularly in research on CDN and QoE aspects. ABR designs that are CDN-aware or utilize well-defined QoE models become more common. Additionally, personalized QoE modeling represents an active research area. On the other hand, cooperation between the application and lower network layers struggles to gain traction. As a result, application-layer ABR algorithms primarily focus on bandwidth efficiency, while fairness in network sharing remains mostly the responsibility of the transport layer.

2.6. Real-World Applications

Commercial SPs play a major role in shaping the HAS practice for long-form 2D videos. These companies inform the technical community and general public about their technologies through corporate blogs, white papers, open-source tools, standardization efforts, and academic partnerships. Additionally, researchers provide independent insights by measuring and reverse-engineering proprietary technologies.

2.6.1. Netflix

Netflix regularly utilizes its technology blog to share in-depth insights into its practices and innovations. While supporting H.264, H.265, VP9, and AV1 codecs, Netflix prefers VP9 and AV1 to reduce licensing costs and enhance accessibility. As a major developer of AV1 [234], Netflix actively advocates for the codec and offers AV1 streaming on a range of devices, including television sets [235]. Netflix also supports AVCHi-Mobile and VP9-Mobile, which are profiles of AVC/H.264 and VP9 tailored for mobile devices [236]. Additionally, Netflix employs the *dynamic optimizer (DO)* [237], a codec-agnostic system that segments video into shots and constructs representations optimized for visual perception. Aiming to enhance QoE, Netflix applies its *deep downscaler (DD)* [238] in video preprocessing to scale down from HR to LR while preserving important visual details. DD leverages ML-based SR techniques and trains CNNs and GANs via SL. At the distribution stage, Netflix relies on Open Connect [20, 239], its proprietary CDN that integrates a backbone network with tens of thousands of local servers deployed across more than one hundred countries. Developed by Netflix, VMAF [166] is an open-source technique that employs SL to fuse PSNR, SSIM, and other metrics of video quality. For Netflix and many third parties, VMAF serves as a preferred metric for assessing video quality in applications related to ABR and QoE. Besides, Netflix tailors a variant of VMAF for high dynamic range (HDR) [240] video, which supports a wider range of color and luminance.

2.6.2. YouTube

Similar to Netflix, YouTube supports H.264, H.265, VP9, and AV1 [241]. However, YouTube focuses on VP9 as the default codec for most videos, employs H.264 for compatibility, and gradually adopts AV1, particularly for high-quality streaming [242, 243]. On average, YouTube encodes its VoD content into 20 representations with various combinations of bitrate and resolution. For live streaming, YouTube generates five or six representations [241] and operates in low and ultra-low latency modes, requesting chunks at intervals of 2 s and 1 s, respectively [244]. The distribution stage leverages the YouTube CDN [245], which Google maintains specifically for serving YouTube videos.

Independent measurements suggest that YouTube's proprietary ABR algorithm employs quick UDP Internet connections (QUIC) [246] or TCP flows to download multiple chunks concurrently [247], utilizes less than 60% of the available network bandwidth, sizes the playback buffer to a large duration of 80 s, and redownloads chunks in higher representations [248].

2.6.3. Amazon Prime Video

Prime Video, which is a streaming service offered by Amazon, typically supports H.264 and H.265 codecs and develops proprietary optimizations for video encoding. For example, it introduces the *encoder-aware motion compensated temporal filter (EA-MCTF)* [249] for video preprocessing in conjunction with H.265 to improve video quality while maintaining low encoding time overhead. At the distribution stage, Prime Video primarily relies on CloudFront, Amazon's own CDN, while also leveraging third-party CDNs to enhance performance, ensure reliability, and optimize delivery across different regions [250]. Additionally, Prime Video fosters technological innovation through academic partnerships. For instance, it explores spatio-temporal learning of video quality [251] and encoding parameter choices in HDR video [252]. Similar to Netflix, Amazon Prime Video develops *ChipQA* [253] as a no-reference metric of video quality based on space-time (ST) chips, which are localized segments of video.

2.6.4. Twitch

Twitch primarily employs the H.264 codec with NVENC hardware acceleration [254] and advances its support for H.265 [255], VP9 [256], and AV1 [255]. To overcome the limitations of open-source transcoding tools, Twitch develops its own transcoder, which enhances downsampling and metadata insertion [257]. Like other major SPs, Twitch relies on its own CDN for distribution [258], complemented by third-party CDN services. Independent measurements of Twitch's proprietary ABR algorithm indicate that it typically fills nearly 20 s of the buffer before starting playback, utilizes less than 60% of the available network bandwidth, and assesses video quality by accounting for human perception [248].

2.7. Trends and Future Directions

After reviewing recent research in Sections 2.3 through 2.5 and real-world applications in Section 2.6, we now distill current prominent trends and discuss future research directions in video streaming.

2.7.1. Trends

2.7.1.1. Continued Growth of Live Streaming

Live streaming continues to expand in traffic and attract increasing attention from researchers. At the ingestion stage, research focuses on enhancing video capture, analytics, compression, and upload to ensure low latency. The processing stage also sees some work related to live streaming, such as on-the-fly transcoding. At the distribution stage, the main research focus remains on VoD rather than live streaming.

2.7.1.2. Increasing Diversity of Devices

The camera-equipped devices, media servers, CDN servers, and user devices that make up the streaming pipeline are diverse in type and capability. This diversity continues to grow as new devices emerge alongside legacy equipment. Ongoing changes in device capabilities enable novel pipeline configurations. For instance, smart cameras now support deep learning and play a larger role in video analytics and encoding-ladder definition, tasks traditionally handled by servers. Additionally, ABR algorithms increasingly shift from the classic client-side paradigm toward greater server-side support. Furthermore, the server infrastructure diversifies its economic models by involving CDN, edge, and cloud operators at the distribution stage. Device heterogeneity is most pronounced at both ends of the pipeline, driven by interest in new streaming modes and improvements in QoE.

2.7.1.3. Integration Across the End-to-End Pipeline

Live streaming and advanced devices drive the trend toward unified solutions across the streaming pipeline, promising more efficient designs and improved end-to-end performance. For example, the distinction between initial compression in camera-equipped devices and transcoding in media servers becomes blurry, as recent designs dynamically split coding tasks between the camera-equipped device and media server to support low latency, conserve energy, decrease storage requirements, and reduce bandwidth consumption. Similarly, video analytics adopts joint designs operating at both ingestion and processing stages. SR methods are increasingly important for managing low network bandwidth during video ingestion and distribution. ABR algorithms and processing-stage tasks, such as transcoding, benefit from greater awareness of CDN, edge, and other distribution infrastructures. QoE models play a growing role in evaluating designs not only at the distribution stage, which directly interacts with end users, but also throughout the entire streaming pipeline.

2.7.1.4. Shift Toward ML Methodologies

The availability of devices with larger memory and processing capabilities also drives a greater reliance on ML methods in streaming designs. Recent results across all three stages of the streaming pipeline consistently show that ML gains popularity over intuition and theory as the basis for problem solving. With cheaper memory and processing power, the interest in resource-intensive data-driven techniques is unsurprising. However, the reviewed works reveal significant divergence in the ML models and training approaches employed at different stages. Ingestion-stage designs tend to rely on UL or SL with DNNs, such as CNNs. Processing-stage solutions predominantly train simpler models, such as DTs and RFs, via SL. At the distribution stage, DRL represents the most common approach, with actor-critic methods being particularly prominent. These differences highlight challenges for the integration trend, as designing a unified ML-based solution that works effectively across all stages might be difficult.

2.7.1.5. Design for Better Trade-Offs

Video streaming is a complex problem with conflicting objectives related to performance and resource consumption, making it infeasible to optimize all metrics simultaneously. Hence, practical solutions aim to offer attractive trade-offs. Technological advances impact the availability and relative costs of network bandwidth, memory, processing, energy, and other resources, thereby affecting which trade-offs are achievable. The shift toward ML methodologies, discussed in Section 2.7.1.4, exemplifies new desirable trade-offs. Additionally, the integration trend broadens the range of viable trade-offs by allowing more flexible placement of functionalities across the pipeline. This search for better trade-offs is evident in the wide adoption of SR techniques, which reduce network bandwidth consumption at the cost of increased processing requirements.

2.7.2. Future Directions

Building on the trends discussed in Section 2.7.1, we project future developments in the field and examine their potential and challenges.

2.7.2.1. ML-Based Streaming

Driven by increasingly affordable memory and processing power, the shift toward ML methodologies is likely to continue. Another key enabler is the wealth of unexplored opportunities, as many existing ML techniques have yet to be applied to streaming problems. For example, applying transformers to streaming deserves further investigation. Additionally, rapid advances in DNN architectures and training approaches continue to yield novel ML methods, potentially forming the basis for innovative streaming designs.

However, this abundance of research opportunities also presents challenges, particularly due to uncertainty about which directions hold the greatest promise. Specifically, as noted in Section 2.7.1.4, there is no clear understanding of which ML methods are most effective in supporting designs that span multiple stages of the streaming pipeline. The proliferation of ML designs across different stages also raises questions about interoperability and mutual influence. While ML-based streaming matures, we are likely to see the development of methods tailored specifically for video streaming, rather than continued reliance on generic ML techniques.

2.7.2.2. Pipeline-Wide Designs

The trends of stage integration and new trade-offs, as discussed in Sections 2.7.1.3 and 2.7.1.5, converge into a future direction geared toward pipeline-wide solutions. A recent surge in research at the ingestion stage suggests a more balanced approach to all three stages and their traditional roles. Cross-stage designs now benefit from the ability to shift or split tasks between stages, optimizing resource utilization and performance. For example, moving certain analytics functions from media servers to camera-equipped devices has the potential to save network bandwidth and reduce upload latency. While pipeline-wide designs hold tremendous promise and numerous unexplored opportunities, it is desirable for unified solutions to maintain flexibility, ideally through loose coupling. SR is likely to play a key role in these designs due to its ability to operate across all three stages of the end-to-end pipeline.

2.7.2.3. Transition to Advanced Codecs

While the surveyed research predominantly employs H.264 or H.265 due to their wide availability, a promising future direction is to build streaming systems around state-of-the-art codecs such as VVC, EVC, and LCEVC. Since cutting-edge codecs are often proprietary, research in this area is likely to involve reverse-engineering efforts, open-source initiatives, and collaborations with codec developers.

2.7.2.4. More ABR Research with Different Foci

ABR designs vary widely in complexity and performance. Recent research often focuses on complex high-performing DNN-based algorithms, while deployed systems typically use simpler solutions of lower effectiveness. This divergence indicates the need for ABR designs with a better balance between complexity and performance. A promising direction is to improve interpretability of DNN-based ABR solutions, leading to stronger confidence in their robustness. Alhough studied in other domains, work on understanding black-box ABR algorithms and converting them to simpler interpretable forms [206] is relatively scarce and needs further investigation. Another promising research direction is

automatic tuning of ABR algorithms. Early efforts, such as [178, 189], explore parameter tuning via simulations. Developing efficient automatic tuning techniques for advanced DNN-based ABR algorithms represents an appealing future research area.

2.7.2.5. Personalized Streaming

Despite significant variations in QoE perception among users [202], streaming services typically rely on one-size-fits-all QoE models that capture QoE as MOS, often failing to accurately reflect individual users' experiences. Personalization of QoE models shows immense promise for the enhancement of streaming services. However, inferring a user's QoE perception non-intrusively is challenging due to the complexity of human cognition, emotions, and actions. Interdisciplinary collaborations that integrate insights from network engineering, data science, and user experience design offer significant potential for progress in this area. When constructing a personalized QoE model requires explicit feedback on subjective QoE perception, this feedback should be expressible, actionable, and minimal to ensure accurate QoE modeling without overburdening the user. The application of transfer learning and the development of multiple MOS-based QoE models for different reference groups, with each user assigned the most representative model, are practical alternatives. However, these methods also need to address concerns about accuracy and overhead.

2.7.2.6. Application-Network Interaction

Video streaming within the HAS paradigm operates on top of TCP as the standard transport protocol. The independent allocation of network bandwidth by application-layer ABR logic and transport-layer congestion control algorithms creates problems for the efficiency, fairness, and stability of bandwidth utilization [259]. To address these issues, some surveyed ABR designs exploit existing transport-layer signals, while others tackle the problems by modifying the transport or network layers [260, 261]. The emergence of QUIC [246] as a promising transport protocol reinvigorates research interest in the interactions between streaming applications and underlying protocols. However, the area of application-network interaction remains underexplored, presenting opportunities for better understanding and developing integrated solutions.

2.7.2.7. Increased Focus on Newer Modes

While this chapter covers recent developments in VoD and live modes of 2D HAS, we anticipate a growing shift in interest from VoD to live streaming. CPs, SPs, and end users flock to live streaming because live content is now easy to create, profitable to distribute, and appealing to consume. From a research perspective, live streaming introduces new challenges, such as further reducing end-to-end latency within the HAS paradigm. Beyond

2D videos, 360-degree video streaming becomes increasingly important due to the wider availability of specialized equipment like omnidirectional cameras and HMDs. In addition to 360-degree video streaming, AR, VR, and MR applications, epitomized by the vision of the metaverse [262], are poised to continue attracting significant attention from both industry and research communities.

2.8. Conclusion

This chapter, supplemented by tutorial materials, provides a holistic survey of the end-to-end video streaming pipeline, encompassing the ingestion, processing, and distribution stages. It establishes essential context for a thorough understanding of the thesis's subsequent chapters. By examining the interactions among video-streaming stakeholders, this chapter highlights promising areas for developing user empowerment strategies, deepens the understanding of existing approaches, and offers insights for designing new solutions. The chapter focus on HAS of long-form 2D videos over CDN-assisted best-effort networks via client-side ABR algorithms reflects a dominant paradigm of modern Internet video streaming. Reviewing over 200 research papers, the chapter covers key topics such as video compression, upload, transcoding, bitrate adaptation, CDN support, and QoE modeling. A new classification scheme organizes the reviewed designs by their problem-solving methodology, whether based on intuition, theory, or ML. We distinguish between MIP, MPC, PID, LO, and BO as theoretical foundations and RL, IL, SL, and UL categories of ML, with further refinement of RL into A3C, A2C, AC, and other methods. In addition, we characterize each design by its core technique and traits such as codec compatibility and SR usage. This classification and trait characterization enhance the systematic understanding of video streaming research. To connect with real-world applications, we also report on practices and innovations by major SPs, such as Netflix and YouTube. The chapter distills prominent current trends, including the continued growth of live streaming, shift towards ML methodologies, integration across the end-to-end pipeline, and design for better trade-offs, fueled by increasing device diversity. Looking ahead, the chapter identifies promising future research directions: pipeline-wide optimization, integration of advanced codecs, further expansion of ABR research, support of personalized streaming, enhanced application-network interaction, and stronger emphasis on newer modes of streaming. These areas represent the forefront of innovation and potential in the field.

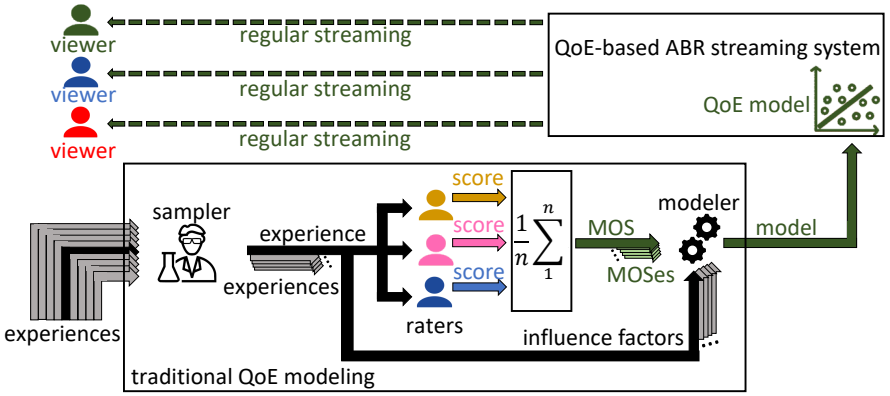
3 User Empowerment via Low-Effort Personalized QoE Modeling

QoE and QoE models are crucial for technological advancements in adaptive video streaming [120, 260, 263], particularly in the design of ABR algorithms [46, 51, 178, 183, 264, 265]. As described in Section 1.1, the process of constructing QoE models involves engaging raters to provide subjective scores for experiences selected by a sampler. These experiences consist of sequences of video chunks, with each chunk characterized by specific IFs. The scores are averaged into MOS, which are then used by a modeler, along with the IFs, to approximate the relationship between the IFs and MOS. Figure 3.1a extends Figure 1.3, illustrating how an ABR streaming system integrates the generated QoE model to optimize streaming performance for the viewers that consume regular streaming.

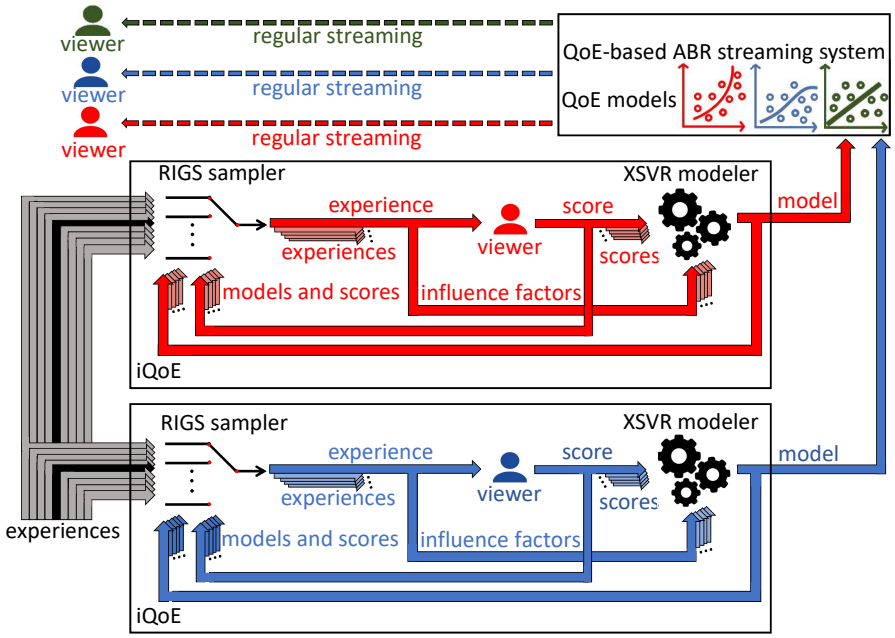
Although this approach reduces subjective testing overhead for most viewers, as raters are far fewer, it creates a one-size-fits-all model that may not adequately reflect the experiences of atypical viewers, whose QoE perceptions differ from both the average and the reference group. This issue is further exacerbated by standard subjective testing methods, which often exclude viewers with systematic rating differences or inversions [40]. Throughout the chapter we refer to atypical viewers as the 10% of the population whose QoE perception deviates the most from the median, where the specific percentile selected is not crucial since the results remain qualitatively unchanged. This statistical approach that focuses solely on statistical outliers is commonly used in various scientific fields like psychopathology [266].

This chapter addresses the issue of atypical viewers by proposing a novel solution called *individualized quality of experience (iQoE)* for creating personalized QoE models by leveraging explicit, expressible, and actionable feedback from viewers in exchange for an enhanced QoE. We focus on atypical viewers because they stand to benefit the most from QoE personalization; however, we envision iQoE incorporation into ABR streaming systems to make it accessible to all viewers.

iQoE involves the viewer, which also acts as the sole rater, into a short series of assessments and exercises active learning to iteratively select experiences for the



(a) One-size-fits-all QoE model for all viewers



(b) Personalized QoE models for atypical viewers

Figure 3.1: Reliance of QoE-based ABR streaming on: (a) traditional QoE modeling and (b) iQoE.

assessments. It leverage the interaction between a novel *randomized improved greedy sampling (RIGS)* strategy as a sampler and an *eXtended SVR (XSVR)* model, which utilizes an extended set of 10 IFs with a support vector regressor (SVR) [267]. Figure 3.1b shows iQoE working mechanism where the blue and red viewers choose to utilize iQoE to construct personalized QoE models used to provide personalized streaming to them. In contrast, the green viewer opts not to use iQoE, so the ABR streaming system applies a one-size-fits-all QoE model for standard streaming to this viewer.

iQoE represents a user empowerment strategy closely tied to QoE modeling, as it enables users to actively enhance their QoE by allowing them to shape the relationship

between IFs and their personal scores through feedbacks. This approach strengthens collaboration between SPs and users, with SPs responsible for implementing the mechanism and offering it as an additional tool within their portfolio of personalized services.

iQoE marks a significant shift from previous approaches to QoE personalization. Existing methods either infer QoE perception without explicit viewer feedback [268–270], often with unproven accuracy, or allow viewers to adjust parameters of a generic QoE model [188, 202, 271], though viewers may not know the optimal settings. iQoE's innovation lies in using explicit, expressible, and actionable feedback directly from the viewer.

The chapter provides a comprehensive evaluation of iQoE through online subjective studies involving 120 raters who submitted a total of 14,400 individual scores, recruited via the Microworkers platform [272]. The results demonstrate that 50 assessments, completed in approximately 22 minutes, are sufficient to build a personalized QoE model that achieves an average accuracy improvement of at least 42% for all viewers and at least 85% for atypical viewers. The collected dataset is made publicly available. Additionally, large-scale simulations using a novel synthetic profiling technique broaden the evaluation by examining iQoE design choices, parameter sensitivity, and generalizability. The findings confirm that iQoE delivers accurate QoE personalization with minimal effort required from the viewer.

3.1. Background on QoE Modeling

QoE modeling uses various methods, resulting in models with different scoring scales, influence factors, and applications. While subjective assessments are traditionally done in controlled lab environments, online crowdsourcing offers easier assessments with less control over conditions [273]. A common scale uses five levels from 1 to 5 [36]. To achieve finer granularity, this chapter uses a 1 to 100 scale, where ranges 1-20, 21-40, 41-60, 61-80, and 81-100 correspond to bad, poor, fair, good, and excellent QoE [39].

Prior studies consider a multitude of influence factors that include metrics of video quality and streaming systems. The former class consists of such metrics as the PSNR [135], SSIM [136], and VMAF [166]. Within the latter class, system factors from the application layer increasingly attract more attention than network-layer metrics. In particular, stall duration $\mathcal{T}_n$ and bitrate $\mathcal{R}_n$ of chunk n are archetypal influence factors in the ABR streaming systems that partition a video into a sequence of $\mathcal{N}$ chunks and encode each chunk for multiple bitrates [274]. Other examples of system factors are number l and average duration d of stalls during the playback [275].

There is no consensus either on the best way to map influence factors into QoE. Whereas some QoE models are closed-form expressions, construction of QoE models via

ML becomes common. The chapter considers 10 existing QoE models. For brevity, we label each of the models with a single letter, as specified below. A number of prominent ABR streaming systems rely on different instances of the following general closed-form QoE model:

$$Q_1 = \kappa \sum_{n=1}^{\mathcal{N}} q(\mathcal{R}_n) - \lambda \sum_{n=1}^{\mathcal{N}-1} |q(\mathcal{R}_{n+1}) - q(\mathcal{R}_n)| - \mu \sum_{n=1}^{\mathcal{N}} \mathcal{T}_n, \tag{3.1}$$

where κ, λ, and μ are tunable parameters, and $q(\cdot)$ denotes a function of the bitrate. We consider **models B** [46], **G** [187], **R** [276], **S** [183], and **V** [208] that instantiate $q(\mathcal{R})$ in Equation 3.1 as the identity function, $\log(\mathcal{R}/r)$ with r denoting the lowest bitrate, PSNR, SSIM, and VMAF, respectively. Similar to model V, **model N** [261] underlies the SDNDASH architecture. Widely known as the FTW model, **model F** [47] belongs to another type of closed-form QoE models and employs an exponential function with parameters α, β, γ, and δ:

$$Q_2 = \alpha e^{-(\beta d + \gamma) l} + \delta. \tag{3.2}$$

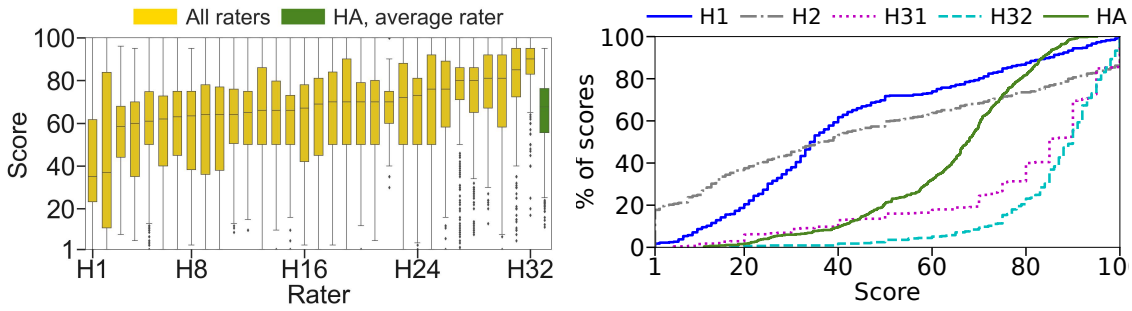
Among the QoE models constructed via ML, **model L** [277] represents state-of-the-art approaches based on deep learning and predicts QoE via a LSTM network. Relying on RF, **model P** [229] refers to the standard P.1203 model. Finally, **model A** [231] denotes the QoE model constructed by Video ATLAS via SVR on VMAF and other influence factors.

QoE models serve various purposes and return values from different ranges. For example, models B, G, and V primarily act as bases for internal improvement of ABR streaming systems [46, 51, 187, 208] and might produce negative values, complicating their value interpretability by humans. On the other hand, models L and P yield values between 1 and 5, as in the standard five-level scale for subjective scores. The heterogeneity of the value ranges undermines direct comparison of QoE models.

3.2. Motivation

3.2.1. Promise of Personalized QoE Modeling

While the introduction defines atypical viewers as the 10% of the population who are the furthest from the median QoE perception, we examine by how much the atypical viewers deviate in their QoE perception from the one-size-fits-all MOS-based QoE models. Specifically, we analyze the Waterloo-IV dataset that reports individual scores of experiences corresponding to all combinations of five videos of different genres (sports, nature, movies, video games, and slides), two encoders, five ABR algorithms, nine network traces, and three varieties of viewing devices [278]. Each experience in the dataset consists of seven chunks, with the playback of each chunk taking



(a) All raters vs. average HA

(b) Four atypical raters vs. HA

Figure 3.2: Inaccuracy of traditional QoE modeling.

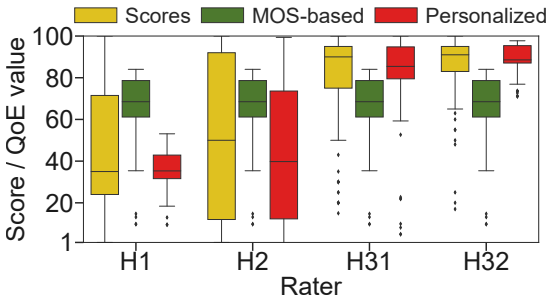


Figure 3.3: Promise of personalized modeling for atypical viewers.

4 s without stalls. Waterloo-IV employs the 1-100 scoring scale and a reference group of 92 raters. The group excludes five other raters who score experiences negligently or otherwise abnormally. Waterloo-IV is the largest recent dataset of its kind, as other datasets either provide MOSes without individual scores [279–283] or characterize chunks with smaller sets of influence factors [48, 284]. Our analysis focuses on the 32 raters who use high-definition television (HDTV) devices to watch 450 experiences where 10 influence factors characterize each chunk.

Figure 3.2a depicts in gold the individual scores by each rater, orders the 32 raters based on the median score, and respectively refers to them as raters H1 through H32. The scores across the reference group are quite distinct in terms of both median and variance, e.g., the gap between the first and third quartiles ranges from 12 to 73. To the right of rater H32, Figure 3.2a plots in green the MOSes, i.e., the QoE perception by the average rater labeled as HA. Raters H1, H2, H31, and H32 cover 10% of all 32 raters and comprise the four atypical raters in this population. Their respective median scores of 35 (i.e., poor), 37 (poor), 85 (excellent), and 90 (excellent) are quantitatively far from the median score of 68 (i.e., good) by average rater HA. Figure 3.2b zooms in on the scores of all experiences by the average and four atypical raters. The results reveal *substantial numerical differences in the QoE perception between the average and atypical raters.*

To evaluate how the choice of individual scores versus MOSes as the basis for QoE modeling affects the model accuracy, we consider QoE model A from Section 3.1 and construct five versions of it: a MOS-based version and, for each of the four atypical raters, a personalized model trained on the individual scores by this rater. We train and test the models on 70%, i.e., 315, and remaining 30%, i.e., 135, of all 450 experiences, respectively.

For each atypical rater H1, H2, H31, and H32, Figure 3.3 plots the individual scores by this rater, QoE values produced by the MOS-based model, and QoE values produced by the rater's personalized model. The personalized QoE modeling enormously enhances the model accuracy and, on average across the four atypical raters, reduces the numerical gap between the median score and median QoE value by more than 31 times. Hence, *personalized QoE modeling brings promise of significant quantitative improvements in the model accuracy for the atypical raters.*

The above analysis is for the atypical members of the vetted reference group in a subjective study. Atypical viewers who are not raters, such as the abnormal red viewer intentionally excluded from the reference group in Figure 3.1, might have even greater numerical gaps from the typical QoE perception and benefit more from personalized QoE modeling.

3.2.2. Design Goals

Because Section 3.2.1 indicates that atypical viewers are not only statistically different but also quantitatively far from the typical QoE perception captured by a MOS-based QoE model, we advocate personalized QoE modeling for atypical viewers and establish our design goals in contrast with three alternative means for accuracy improvement of the traditional QoE modeling.

While a vast majority of all viewers in the traditional approach do not exert any modelling effort, one possibility for retaining this attractive property is to form multiple reference groups, build a separate MOS-based QoE model for each group, and associate a viewer with the group that represents the QoE perception by this viewer most accurately. If the viewer and raters of the associated reference group are similar in their QoE perception, the numerical discrepancy between the viewer's QoE perception and MOS-based QoE model of the group is likely to be small. Although the general technique of multiple reference groups works reasonably well in other application domains such as recommendation systems [285, 286], our evaluation in Section 3.4 demonstrates that this approach does not sufficiently mitigate the inaccuracy of the MOS-based QoE modeling due to the great heterogeneity of QoE perception among humans. In addition, the task of associating each viewer with a representative reference group might be difficult to accomplish without interacting with the viewer. The above discussion leads us to our first goal:

Goal 1. *The construction of an accurate QoE model for an atypical viewer should rely on perception feedback from this viewer.*

The feedback requirement does not necessarily imply that the viewer has to explicitly score experiences as the raters do in the traditional QoE modeling. An intriguing prospect is indirect inference of the viewer's QoE perception, e.g., through automatic monitoring of

the viewer's gaze direction, facial expression, engagement, or viewing activities [227, 269, 287–289]. Unfortunately, such inference techniques might require special equipment or raise privacy issues [227]. Moreover, due to the complexity of overall human behavior, this alternative is yet to prove its suitability for accurate QoE modeling. Hence, we consider only explicit mechanisms for the viewer's feedback:

Goal 2. *The mechanism for perception feedback should be explicit.*

By itself, the feedback explicitness does not assure that the feedback is useful. A possible avenue for personalized QoE modeling is to ask a viewer for personal preferences, e.g., for values of the κ, λ, and μ parameters in Equation 3.1, and leverage the provided preferences to personalize a generic QoE model [188, 202, 271]. However, a viewer usually does not know how to articulate personal preferences well enough to make the resulting QoE model accurate. Thus, we pursue the following goal:

Goal 3. *The solicited feedback should be of a kind expressible by the viewer and actionable for accurate QoE modeling.*

When combined, Goals 1 through 3 limit the design options to methods that build an accurate QoE model for an atypical viewer by collecting explicit actionable feedback from this viewer. However, to encourage the viewer's participation, the model construction should require only light contributions from the viewer. This leads us to our fourth goal:

Goal 4. *The amount of effort contributed by a viewer into the construction of an accurate personalized QoE model for the viewer should be small.*

3.3. Design

3.3.1. iQoE Overview

This section designs iQoE to achieve the goals established in Section 3.2.2. iQoE meets Goals 1, 2, and 3 by adopting the assessment-based conceptual structure of the traditional QoE modeling and reducing the group of raters to the viewer for whom the method constructs the personalized QoE model, as illustrated in Figure 3.1b. Engaging the viewer as the sole rater satisfies Goal 1. The viewer explicitly scores experiences, which conforms to Goal 2, and in the same expressible actionable manner as in the traditional QoE modeling, thereby complying with Goal 3.

The fulfillment of Goal 4 is challenging and constitutes the main focus of this chapter. To keep the effort of the viewer low, iQoE limits the viewer's involvement to a short series of H assessments that cumulatively consume a small amount of the viewer's time. This section derives a simple and yet effective iterative design for iQoE where a new automatic RIGS sampler and XSVR modeler require little

memory and execute quickly on the client side without causing a perceptible wait for the viewer during the assessment series. iQoE exercises active learning so as to be sample-efficient and accurate. Interactions between RIGS and XSVR steer iQoE to produce an accurate QoE model despite the limited length of the assessment series.

Although the deliberate emphasis of this chapter is on the atypical viewers because they constitute the largest beneficiaries of QoE personalization, we design iQoE as an option available to any viewer. By not opting in, a typical or just uninterested viewer avoids any subjective-test overhead. Regular streaming to such viewers continues relying on the one-size-fits-all QoE model.

Turning the viewer into the sole rater during the construction of the personalized QoE model has positive side effects. Since the viewer becomes the only party utilizing the constructed QoE model, iQoE incentivizes the viewer to perform the assessments conscientiously. Also, the same number of assessments by the viewer typically results in a more accurate QoE model than in the traditional MOS-based modeling with many raters. Besides, because the viewer is likely to train the personalized QoE model in the settings of regular streaming, the QoE-model accuracy might increase due to the more specific context than in traditional lab-based subjective tests. That said, we defer to future work a comprehensive study of the impact by different contexts and contents.

Algorithm 1 *iQoE*(E)

1: $K \leftarrow E$; $J \leftarrow \varnothing$; $M \leftarrow \varnothing$; $Q \leftarrow \varnothing$ ▷ initialization
2: **for** $t = 1, \ldots, H$ **do**
3: $e \leftarrow \text{RIGS}(K, J, M, Q)$ ▷ sampling
4: $s \leftarrow$ the viewer's score of e ▷ assessment
5: $K \leftarrow K - \{e\}$; $J[t] \leftarrow e$; $M[t] \leftarrow s$
6: $Q \leftarrow \text{XSVR}(J, M)$ ▷ modeling
7: **end for**
8: **return** Q ▷ final QoE model
9: **procedure** RIGS(K, J, M, Q)
10: **if** $t \leq h$ **then**
11: $e \xleftarrow{\text{R}} K$ ▷ random sampling
12: **else**
13: $e \leftarrow \text{argmax}_K \min_J D_{jk}$ ▷ IGS sampling
14: **end if**
15: **return** e ▷ experience for the next assessment
16: **end procedure**
17: **procedure** XSVR(J, M)
18: **if** $t \geq h$ **then**
19: $Q \leftarrow$ the SVR model trained on J and M
20: **end if**
21: **return** Q ▷ current QoE model
22: **end procedure**

Algorithm 1 reports the pseudocode for the iQoE construction of model Q as a personalized QoE model for the viewer. Set E of experiences serves as an input to the algorithm and comprehensively covers the conditions possible during regular streaming. iQoE acquires set E in advance, e.g., by generating it through simulations or as a real dataset supplied by the ABR streaming system. Algorithm 1 tracks the non-rated and rated experiences in set K and array J, respectively, and stores the scores of the rated experiences in array M. Initially, set K contains experience set E while J, M, and Q are empty (Line 1 of Algorithm 1). iQoE performs H iterations (Lines 2–7) to produce the final QoE model (Line 8). Each iteration t uses the RIGS sampler to select experience e

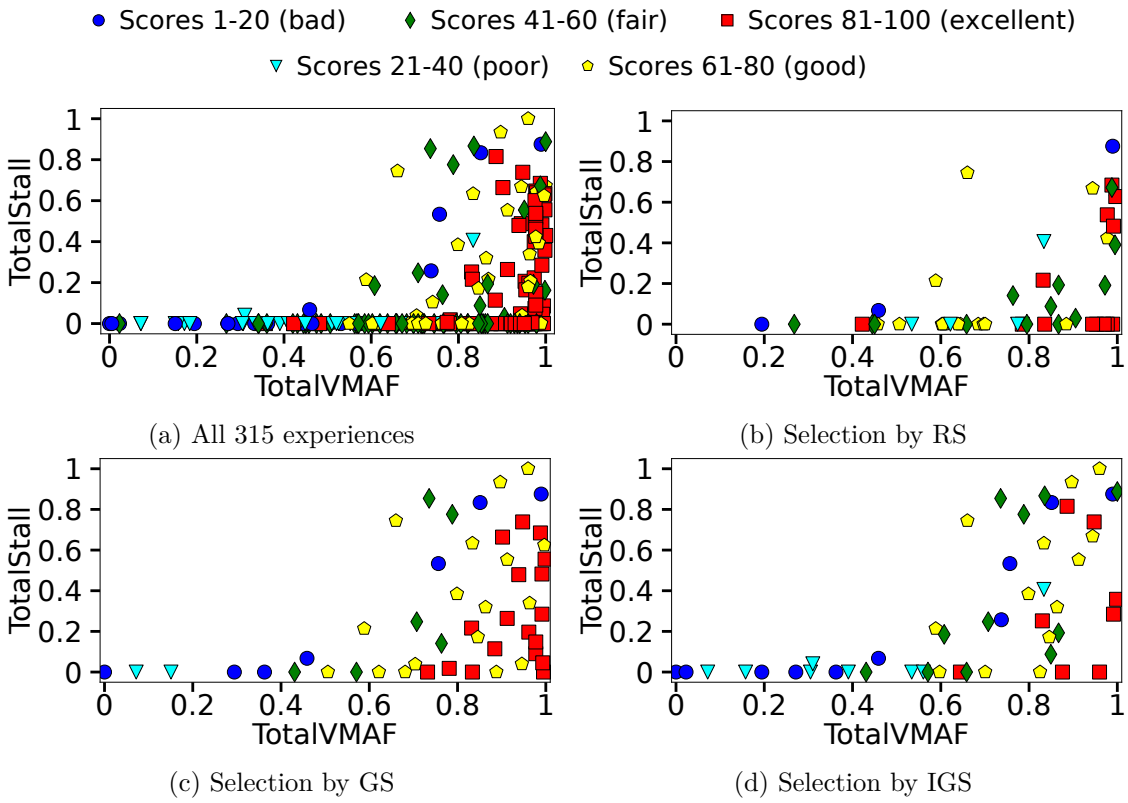
for the next assessment (Line 3), obtains score s of this experience by the viewer (Line 4), moves experience e from set K to array J and records score s in the corresponding element of array M (Line 5), and then updates QoE model Q by applying the XSVR modeler to arrays J and M (Line 6). Sections 3.3.2 and 3.3.3 elaborate on the designs of our RIGS sampler and XSVR modeler, respectively.

3.3.2. RIGS Sampler

We strive for an effective simple design of the automatic RIGS sampler. The primary objective is to pick a series of H experiences from set E so that the final QoE model becomes as accurate as possible. On the other hand, the design simplicity is important because of enabling the sampler to select an experience quickly without introducing a discernible wait for the viewer between successive assessments. Set E characterizes each of its experiences e with features from set C. Whereas these $|C|$ features form an input space, the viewer's score s_j of rated experience e_j in array J and value $Q(e_k)$ produced by QoE model Q for non-rated experience e_k in set K belong to an output space of subjective scores and QoE values. Intuitively, the experiences selected by an effective sampling strategy should provide a balanced coverage of set E in both input and output spaces so that the constructed QoE model successfully deals with both diversities in experiences and their perception by the viewer. The sampler's task to select H out of the $|E|$ experiences faces the following extra complication: while the feature values of all experiences are available in advance, the score of an experience becomes known only after the viewer assesses the experience.

Our derivation of the sampling strategy for RIGS illustrates relevant issues by utilizing again the real Waterloo-IV dataset described in Section 3.2.1. For clarity, we consider one rater, set E with 315 experiences (which are the same as the training experiences in Section 3.2.1), and constrain the characterization of each experience to two normalized features, namely TotalVMAF (the sum of the VMAF values across all chunks in the experience) and TotalStall (the total stall duration divided by the total duration of the experience). Figure 3.4a depicts the 315 experiences as points in the two-dimensional input space formed by the TotalVMAF and TotalStall features. The colors of the points expose the ranges of the experience scores in the output space: we color the 1-20 (bad), 21-40 (poor), 41-60 (fair), 61-80 (good), and 81-100 (excellent) score ranges in blue, cyan, green, yellow, and red, respectively. This example sets H equal to 50 experiences.

From the simplicity perspective, the best strategy is random sampling (RS) that picks non-rated experiences from set K randomly. While simple, RS might cover the input space unevenly and represent some areas in the space insufficiently. In our example, Figure 3.4b shows that most of the 50 experiences selected by RS lie around the TotalStall = 0 or TotalVMAF = 1 lines and that only a handful of the selected experiences represent poor conditions with respect to both stalls and VMAF.



(a) All 315 experiences

(b) Selection by RS

(c) Selection by GS

(d) Selection by IGS

Figure 3.4: Selection of 50 experiences by the RS, GS, and IGS samplers from the set of 315 experiences.

One possibility for addressing the above weakness of RS is to adopt greedy sampling (GS) that accounts for the distances between experiences in the $|C|$-dimensional input space. With f_{ij} and f_{ik} referring to the values of feature i for rated experience e_j in array J and non-rated experience e_k in set K respectively, GS defines the distance between experiences e_j and e_k as $\sqrt{\sum_{i \in C} (f_{ij} - f_{ik})^2}$, computes for each e_k in K the minimum distance between this e_k and any experience e_j in J, determines the largest of these $|K|$ minimum distances, and iteratively moves the corresponding experience e_k from set K to array J. Whereas GS is likely to strike a better balance than RS in sampling the input space, GS remains oblivious of the output space and might cover it unevenly. Returning to our example, Figure 3.4c confirms that while the 50 experiences selected by GS cover the input space more evenly than the 50 RS-selected experiences in Figure 3.4b, the coverage of the output space remains insufficiently balanced, e.g., the experiences with excellent (red) scores dominate the experiences with bad (blue), poor (cyan), and fair (green) scores.

A logical fix for the demonstrated drawback of GS is to go for IGS that accounts for distances in the output space as well. IGS redefines the distance metric of GS as $D_{jk} = |s_j - Q(e_k)| \sqrt{\sum_{i \in C} (f_{ij} - f_{ik})^2}$ where Q denotes the current QoE model, and

the prepended $|s_j - Q(e_k)|$ factor is the distance in the one-dimensional output space between score s_j of rated experience e_j in array J and QoE value $Q(e_k)$ of non-rated experience e_k in set K. The prepended factor uses $Q(e_k)$ as an estimate for score s_k of experience e_k because s_k becomes known only after the viewer assesses e_k. Similarly to GS, IGS iteratively moves from set K to array J the experience e_k with the largest minimum D_{jk} distance. IGS is a form of active learning since it selects an experience for the next assessment based on the QoE model trained on the previously selected experiences. In our running example, Figure 3.4d shows that the 50 experiences selected by IGS provide a good coverage in the input space and cover the output space more evenly that the 50 GS-selected experiences in Figure 3.4c, e.g., by reducing the imbalance between the excellent (red) scores and bad (blue), poor (cyan), and fair (green) scores. From the simplicity perspective, IGS has polynomial time complexity of $\mathcal{O}(|K||J||C|)$ per iteration and is less attractive than RS.

Our RIGS sampler selects a series of H assessments by combining the strengths of RS and IGS. The selection of experiences by RS is always simple and quick. On the other hand, the accuracy advantages of IGS grow as it collects more samples. Early on in the sampling process, the extra work done by IGS to compute the distances between experiences does not yield significant payoffs because the current QoE model remains too inaccurate to make QoE values $Q(e_k)$ an informative prediction for scores s_k of yet non-rated experiences e_k. As the number of samples increases, the QoE model becomes more accurate, and its $Q(e_k)$ values steer IGS to select a more instructive sample for further improvement of the model accuracy. Hence, RIGS starts by quickly picking a series of initial experiences via RS and then switches to selecting the subsequent experiences via IGS. iQoE controls the switching by parameter h: the RIGS sampler selects the first h experiences from set K randomly (Line 11 of Algorithm 1), the QoE model gets updated by the XSVR modeler starting from iteration h of iQoE (Line 19), and RIGS selects experiences $h + 1$ through H according to the largest minimum D_{jk} distance (Line 13). In Section 3.4.2, we analyze sensitivity of iQoE to parameters h and H, show that the default setting of h to 10 experiences is the most beneficial for the final model accuracy, and evaluate RIGS against alternative samplers.

3.3.3. XSVR Modeler

We also aim for a simple and yet effective design of the XSVR modeler. To build accurate QoE models, XSVR simultaneously considers a broad pragmatic set of influence factors. While many additional influence factors might increase predictive power, e.g., electroencephalographic or other psychophysiological signals [227], we restrict the choice of influence factors to those measurable without special equipment in the viewer's regular settings of video watching and compose an eXtended (X) set as a superset of the influence factors in the 10 existing models discussed in Section 3.1. The X set, which contributes

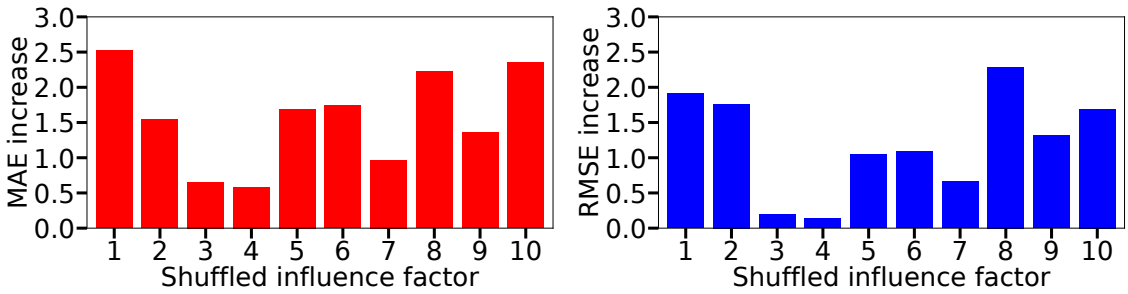


Figure 3.5: Importance of the 10 influence factors in XSVR for the atypical raters.

letter X to the XSVR name of our modeler, comprises the following 10 influence factors: (1) representation identifier, (2) stall duration, (3) bitrate, (4) chunk size, (5) frame width, (6) frame height, (7) indicator whether the bitrate is the highest in the bitrate ladder, (8) PSNR capped at 50 [290], (9) SSIM, and (10) VMAF. All 10 influence factors are easily measurable and, in particular, tracked by Waterloo-IV [278]. Because the 10 influence factors separately describe each of the $\mathcal{N}$ chunks in each experience, XSVR employs a total of $10\mathcal{N}$ features, which implicitly also represent inter-chunk influence factors such as bitrate changes and their magnitude.

The simplicity constraint calls for efficient approximation of the functional relation between QoE and $10\mathcal{N}$ features so that the memory consumption and training overhead of the QoE model on the client device are insignificant. Whereas construction of QoE models increasingly relies on ML techniques [229, 277, 291], we also adopt an ML-based solution but stay away from deep learning due to its high computational complexity. Specifically, the XSVR modeler relies on SVR [267] because of its memory efficiency and general effectiveness on small datasets with high-dimensional input spaces, i.e., in the same sort of situations as ours. Section 3.4.2 evaluates XSVR design choices, including its reliance on SVR vs. other simple ML-based solutions.

To understand how much the $10\mathcal{N}$ features contribute to the predictive power of XSVR, we apply the technique of permutation feature importance [292]. This techniques randomly shuffles the values of a feature and measures the impact of the disruption on the predictive power of the model [292]. The measurements reveal the importance of features for the predictive ability.

For each of the four atypical raters H1, H2, H31, and H32 in the Waterloo-IV dataset, we train XSVR separately and repeat the experiment 30 times. Figure 3.5 demonstrates that all 10 influence factors of the X set contribute positively to the predictive power of XSVR. Influence factors 8 (PSNR), 1 (representation identifier), 2 (stall duration), and 10 (VMAF) are the most important as their disruption increases average RMSE across the four raters by 2.30, 1.90, 1.79, and 1.62 and average MAE by 2.19, 2.48, 1.58, and 2.33, respectively. On the other hand, influence factors 3 (bitrate) and 4 (chunk size) are relatively the least important, as their random shuffling worsens average RMSE by 0.18 for either of them and average MAE by 0.63 and 0.61, respectively.

3.4. Evaluation

3.4.1. Subjective Studies

3.4.1.1. Methodology overview

We conduct the subjective studies in two phases. The first phase directly recruits 34 volunteers from around the globe and all walks of life. The number of 34 raters lies between 15 and 40, i.e., in the range that the ITU suggests for subjective tests in its Recommendation P.910 [36]. To generalize the conclusions to a broader population, the second phase scales the total number of raters in our studies to 120 by leveraging Microworkers, an online crowdsourcing platform [272].

To perform the studies, we develop a real website with access to 1,000-element experience set, extracted from Tears of Steel in its 4K version[1], to show to raters and deploy it on the Internet.

Website Design. Each rater accesses the first page of the website through a directly provided link and accepts terms and conditions before commencing a session. Then, the website creates a new folder tracking the session status and assigns a random unique identifier to the session. The subsequent page offers the rater to watch a reference experience. This reference experience has the maximum quality among all elements of the experience set and helps the rater to calibrate the highest expectations for the experiences watched during actual assessments.

After this introductory stage, the website takes the rater to a current status page where the rater can monitor the progress of the session, watch the reference experience, monitor own scoring history, pause the session, or watch a new experience. Selecting the latter option opens a playback window that automatically starts playing back the new experience to the rater. Because modern browsers require an explicit command from the viewer to enlarge video dimensions, we do not put the playback window into the full-screen mode automatically. Instead, the playback window contains a button for full-screen toggling. We remove the control bar from the playback window to prevent unwanted behaviours, such as skipping an experience to the end without watching the experience. Once the experience playback ends, the browser opens a scoring page where the rater can provide a score of the experience by sliding a bar along the 1-100 scale, watch the reference experience, monitor own scoring history, or rewatch the latest experience. Upon the score submission, the browser redirects the rater to a current status page which displays the updated status of the rater's session. This cycle repeats until the playback window finishes playing back the last experience of the series. At each iteration, the website saves information about the rater's session in .txt and .npy files and the current QoE model in the .pkl format. The website implementation and deployment use Python

[1] https://mango.blender.org/download/ accessed last on October 24, 2024

3.7 with Flask 1.1.2 for the backend, hyperText markup language (HTML), JavaScript, and cascading style sheets (CSS) for the frontend, and Apache 2.4.29 server on an Ubuntu 18.04.6 machine.

The rater's machine installs cookies to enable the rater to resume a previously paused session within three days. At the end of the assessment series, the website asks the rater to fill in a form with the following information: the rater's home country, the country where the rater takes the assessment series, gender, age, viewing device, level of satisfaction with the assessment series, and any optional suggestions.

Experience Set. To generate the 1,000-element experience set, we utilize the Park platform [293]. Park simulates ABR streaming over a network trace that specifies dynamic network bandwidth available for streaming. We experiment with throughputrule (TR) [294], BBA [167], and MPC [46] and select 34 real-world network traces from each of the federal communications commission (FCC) [295], Norway [296], and Oboe [178] datasets, i.e., 102 network traces in total. The used bitrate ladder consists of 13 levels where the resolution and bitrate range from 320×180 and 235 Kbps to 3,840×2,160 and 16,800 Kbps, respectively [297]. To engender experiences with diverse scenes and increase the rater's involvement, we adopt Tears of Steel without its closing credits as the source video. The running time of the video is about 10 minutes. We segment the video into 294 chunks by applying the FFmpeg and MP4Box tools and transcode each chunk as per the used bitrate ladder. Our application of Park to the segmented video produces 306 experiences. Under the constraints that an extracted experience has stalls only between its chunks and that the total stall duration of the experience does not exceed 50% of the original running time without stalls, we randomly extract from the 306 long experiences a set of 1,000 short experiences. Each of these short experiences consists of four chunks and has the total playback time of 8 s without stalls.

Atypical raters. In agreement with the definition of atypical viewers in the introduction, our subjective studies involve 12 atypical raters comprising 10% of all 120 raters. Five and seven of the 12 atypical raters are from the first and second phases of the studies, respectively. Hence, both direct recruitment and crowdsourcing contribute atypical raters to the studies.

Training and testing of the baselines and iQoE. The subjective studies configure the h and H parameters of iQoE to their default settings of 10 and 50 experiences, respectively, and consider the 10 models from Section 3.1 as baselines. We randomly pick 50 experiences from the experience set to train baselines, and reuse 10 of these 50 experiences as the iQoE's initial h experiences across all 120 raters. iQoE relies on RIGS to select the subsequent $H - h = 40$ experiences for each rater, and these 40 experiences are generally different across the 120 raters. Hence, 90 out of the 120 experiences assessed by a rater support model training, with each particular QoE model trained on 50 experiences. We use the MOSes and individual scores by the rater, respectively, to train

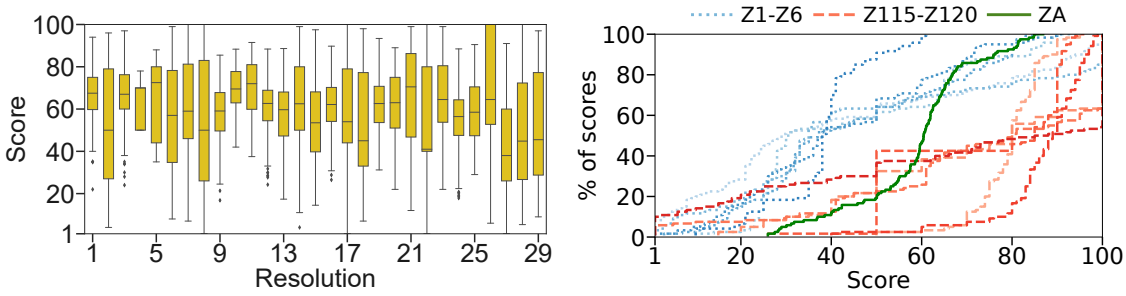
eight parameterized baselines and iQoE via regression. These parameterized baselines are models B, G, R, S, V, N, F, and A, and the regression returns values for their parameters, e.g., parameters κ, λ, and μ for the former five baselines and parameters α, β, γ, and δ for model F. The regression-based training produces QoE models that predominantly compute QoE values on the 1-100 scale. On rare occasions when a QoE model computes a value below 1 or above 100, we treat the spillover as a prediction error without adjusting the out-of-scale value. Models L and P come without a publicly available training module and compute QoE values on the 1-5 scale. We use these two models in their public configurations trained by the models' proposers and linearly map the computed QoE values into the 1-100 scale. We test all 10 baselines and iQoE on the same set of 30 experiences, which accurately represent the 1,000-element experience set. A shuffle of the 90 training and 30 testing assessments throughout the 120-assessment series ensures that the rater is unaware whether the current assessment is for training or testing.

Metrics. We measure accuracy of QoE model Q by means of MAE $= \sum_{e_k \in K} |Q(e_k) - s_k|/|K|$ and RMSE $= \sqrt{\sum_{e_k \in K}(Q(e_k) - s_k)^2/|K|}$, respectively, where K refers to a set of rated experiences, and s_k and $Q(e_k)$ denote the rater's score and QoE value of experience e_k, respectively. As the variance in the individual errors increases, RMSE exceeds MAE by a larger amount.

Ethical issues. The Ethical Board of our institute granted full approval to conduct the research. The subjects opted into the studies via informed consent on the front page of the website, with the consent required before any data collection could commence. The studies did not collect any personal identifiers and did not open any opportunities for linking the collected experimental results or demographic statistics with the subjects' actual identities.

3.4.1.2. Dataset

We collect and openly release, along with all accompanying code, a dataset consisting of the 1,000-element experience set and 14,400 individual scores provided by the 120 raters [298]. 115 of the raters answer all questions in the post-assessment survey. Among those who answer, 28% and 72% identify themselves as female and male, respectively, from locations in 47 countries on four continents (with 45 home countries). The age ranges from 20 to 63 years old. 64% of the respondents rank their participation in the studies as pleasant, with the other three options being slightly annoying, quite annoying, and very annoying. The answers to the post-assessment survey indicate that 94% and 6% of the respondents complete the assessment series on a personal computer and phone, respectively. 96% and 4% of the respondents view the experiences in Google Chrome and Mozilla Firefox, respectively. We also track the screen resolution of each respondent's viewing device, detect 29 different resolutions altogether, and label them with the following resolution identifiers: (1) 360×640, (2) 360×800,



(a) Screen resolution (b) 12 atypical raters

Figure 3.6: Extra insights into the collected dataset.

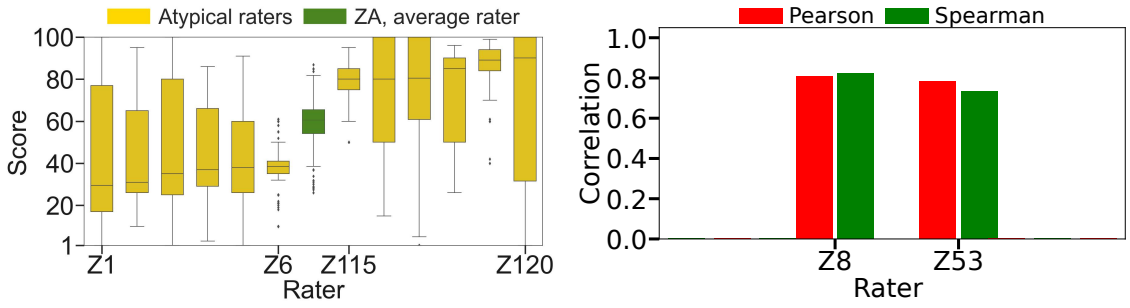
(3) 384×857, (4) 393×873, (5) 412×915, (6) 1,028×578, (7) 1,093×615, (8) 1,241×697, (9) 1,280×720, (10) 1,280×800, (11) 1,360×768, (12) 1,366×768, (13) 1,440× 900, (14) 1,455×819, (15) 1,512×982, (16) 1,536×864, (17) 1,536×960, (18) 1,595×897, (19) 1,600×900, (20) 1,680×1050, (21) 1,707×1067, (22) 1,728×1,117, (23) 1,856×1018, (24) 1,920×1,080, (25) 1,920×1,200, (26) 1,928×970, (27) 2,048×1,152, (28) 2,560×1,440, and (29) 3,840×2,160. Resolutions 12, 16, and 24 are the most popular and account for 22%, 17%, and 18% of all resolutions, respectively. Figure 3.6a depicts the respondents' scores grouped according to the screen resolution and reveals that the resolution affects the QoE perception relatively mildly, with somewhat lower scores for the three highest screen resolutions.

Figure 3.6b shows that 12 atypical raters Z1 through Z6 and Z115 through Z120 of our dataset differ significantly in their QoE perception from average rater ZA. Due to the scarcity of real data on individual QoE perception, the collected dataset constitutes a contribution of independent importance.

3.4.1.3. Results

To analyze the collected dataset and evaluate iQoE against alternatives, we examine the following questions: *(1)* How heterogeneous is the QoE perception in the dataset? *(2)* How consistent is the QoE perception over time? *(3)* How much of the rater's time does iQoE take to construct the personalized QoE model? *(4)* How does iQoE perform compared to the MOS-based baselines? *(5)* Is iQoE superior to using multiple reference groups? *(6)* How does iQoE perform compared to personalized versions of the baselines?

(1) Perception heterogeneity. We arrange the 120 raters in nondecreasing order of their median scores and accordingly label the raters as Z1 through Z120. Raters Z1 through Z6 and Z115 through Z120 comprise the 12 atypical raters. Figure 3.7a depicts the individual and median scores of the 12 atypical raters and average rater ZA. The median scores of the atypical raters consist of six poor scores between 29.5 and 38.5, three good scores of 80, 80, and 80.5, and three excellent scores of 85, 89, and 90.



(a) Atypical raters vs. average ZA

(b) Scoring consistency over time

Figure 3.7: Score distributions and consistency of the subjective studies.

In contrast, the median score of average rater ZA, i.e., median MOS, is a good score of 61. Figure 3.7a corroborates that *the QoE perception by atypical viewers differs dramatically from the average QoE perception by all viewers.*

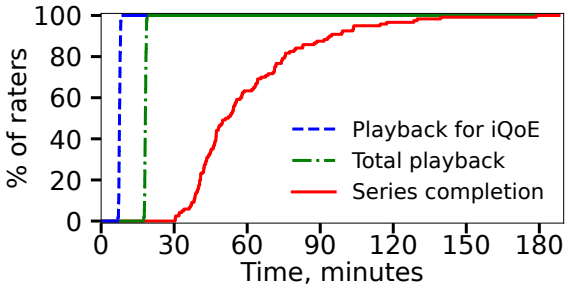


Figure 3.8: Playback and completion times of subjective studies.

(2) Scoring consistency over time. Raters Z8 and Z54 are the two raters who heed our request to repeat the same assessment series in the same settings, e.g., with respect to the viewing device and browser, 20 days after the original studies. For each of the two raters, Figure 3.7b presents the pearson linear correlation coefficient (PLCC) and spearman's rank correlation coefficient (SRCC) between the scores by the rater in the original and repeated series. For both raters, the scores exhibit high correlation in regard to either values or ranks: the pearson and spearman coefficients exceed 0.81 and 0.73 for raters Z8 and Z54, respectively. The detected consistency of QoE perception over time indicates that *personalized QoE models preserve their accuracy over time without a need for frequent retraining.*

(3) Time to construct the personalized QoE model. We analyze the collected dataset to estimate the amount of time it would take for a viewer to construct the personalized QoE model. For each rater, our subjective studies use 50 out of the 120 rated experiences to train the personalized QoE model. Figure 3.8 shows that the total playback time of the 50 experiences including the stalls varies across all 120 raters from 6.8 to 8.4 minutes. For the entire series of 120 rated experiences, the total playback time including the stalls ranges from 17.5 to 19 minutes, and the overall completion time spreads from 30.4 to 188 minutes, with the median value of 53 minutes. While the completion time excludes all intervals between an explicit pause and subsequent resumption of the series by the rater, the other contributors to the extra delay include downloading an entire

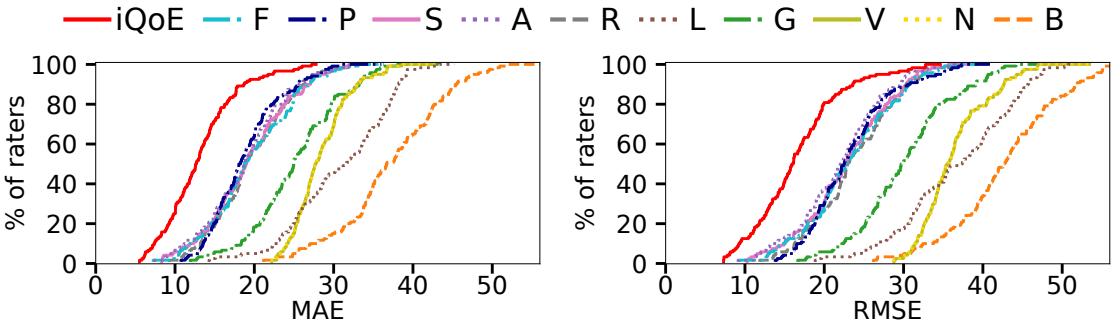


Figure 3.9: iQoE vs. MOS-based QoE modeling.

		A	S	R	P	F	G	N	V	L	B
All raters	MAE	1.54	1.58	1.61	1.63	1.64	2.17	2.41	2.41	2.67	3.23
	RMSE	1.42	1.47	1.53	1.57	1.52	2.03	2.43	2.43	2.51	2.89
Atypical raters	MAE	2.18	2.19	2.24	2.06	2.1	2.59	2.87	2.87	2.81	3.55
	RMSE	1.89	1.93	2.02	1.92	1.85	2.39	2.86	2.86	2.67	3.18

Table 3.1: Average iQoE gains over the 10 baselines.

experience before the browser starts its playback, rewatching of an experience by the rater, reflecting on an appropriate score for an experience, and being distracted by unrelated tasks without explicitly pausing the series. Because the ratio of the playback times is close to the $120/50 = 2.4$ ratio of the experience counts, we use this 2.4 factor to estimate the median completion time for the 50 assessments to be around 22 minutes, which is relatively low. Hence, our estimates indicate that *the amount of the viewer's time taken by iQoE to construct the personalized QoE model is affordable.*

(4) iQoE vs. MOS-based modeling. To evaluate iQoE against the 10 baselines, Figure 3.9 plots the distributions of their accuracy. iQoE significantly outperforms all baselines and provides MAE and RMSE values as low as 5.5 and 7.3, respectively. For 20% of all 120 raters, iQoE provides MAE and RMSE of at most 9.3 and 11.8, whereas model A is the best among all baselines with the corresponding MAE and RMSE of 14.3 and 17.6, meaning that the MAE and RMSE gain by iQoE over the best baseline is 1.53 and 1.48 in MAE and RMSE, respectively. Table 3.1 shows that the average gains by iQoE over the 10 baselines across all 120 raters range from 1.54 (model A) to 3.23 (model B) in MAE and from 1.42 (model A) to 2.89 (model B) in RMSE. The average gains for the 12 atypical raters are higher and vary from 2.06 and 1.85 (models P and F, respectively) to 3.55 and 3.18 (model B) in MAE and RMSE, respectively. Hence, while iQoE improves the model accuracy for all raters, iQoE provides larger QoE gains to the atypical raters. Our results show that *the average accuracy improvement of iQoE over the MOS-based baselines is at least 42% for all raters and at least 85% for the atypical raters.*

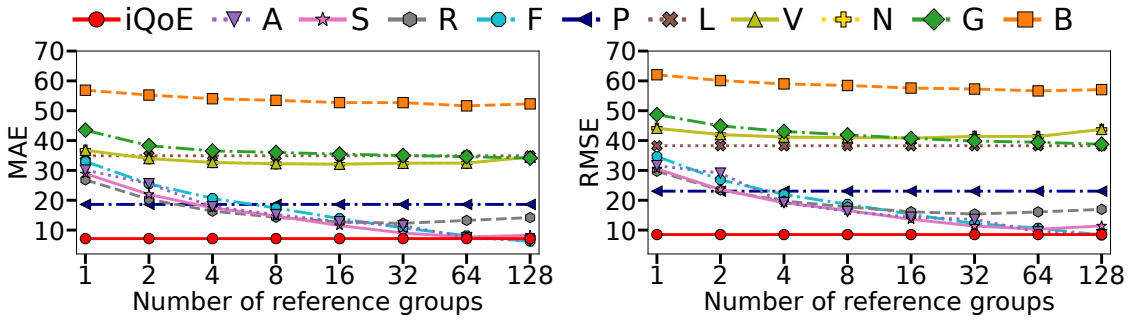


Figure 3.10: iQoE vs. multiple reference groups.

(5) iQoE vs. multiple reference groups. According to Section 3.2.2, multiple reference groups pose an alternative to the iQoE approach of allowing each viewer to act as a rater in the construction of the viewer's personalized QoE model. The alternative requires more reference groups, with the raters of each group having less heterogeneous QoE perception, and associates every viewer with the most representative group. For this series of experiments, we conduct subjective tests with eight extra raters on Microworkers so as to increase the total number of raters to 128, which is a power of two and supports recursive partitioning of the rater population into equal-sized groups all the way down to the single-rater groups. We rearrange the expanded set of 128 raters in nondecreasing order of their median scores and correspondingly relabel the raters as S1 through S128. Under the new labeling, atypical rater Z120 becomes atypical rater S128. In this experimental series, we use rater S128 as the viewer and consider eight different partitions of the 128-rater population into one, two, four, eight, 16, 32, 64, and 128 groups where the number of raters in each group equals 128, 64, 32, 16, eight, four, two, and one, respectively. The partitions and the viewer's association with the most representative group follow the order of the median scores. Specifically, as the number of reference groups increases from 1 to 128, we associate viewer S128 with reference groups {S1, ..., S128}, {S65, ..., S128}, {S97, ..., S128}, {S113, ..., S128}, {S121, ..., S128}, {S125, ..., S128}, {S127, S128} and {S128}, respectively. In the latter partition with the single-rater groups, the viewer and rater are the same, implying that atypical rater S128 acts as the sole rater in building the personalized QoE model.

Figure 3.10 explores how much an increase in the number of reference groups helps the baselines to bridge the accuracy gap with iQoE. Even in the single-rater partition, baseline models B, G, V, N, L, P, and R still fail to close the gap in either MAE or RMSE. Personalized models S, A, and F perform the best among the baselines. Because these models match the iQoE accuracy only when the viewer becomes the sole rater, Figure 3.10 supports the conclusion that *iQoE produces more accurate QoE models compared to the approach of multiple reference groups, unless the latter reduces itself to personalized QoE modeling.*

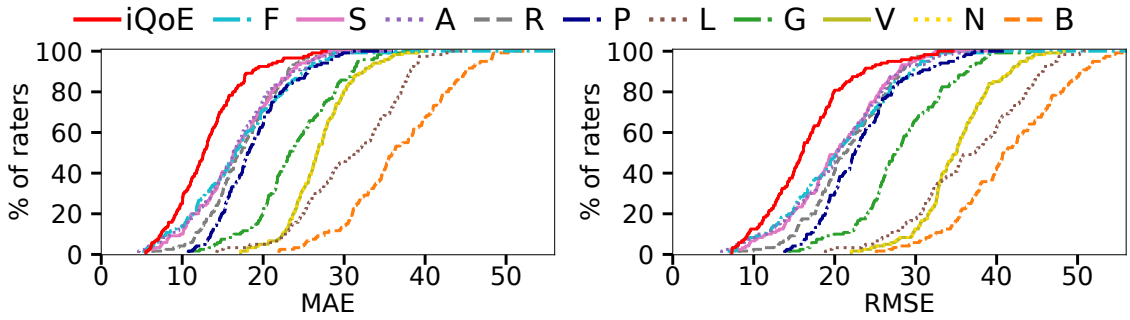


Figure 3.11: QoE vs. personalized baselines.

(6) iQoE vs. personalized baselines. To investigate the what-if scenario that personalizes the baseline QoE models, we return to our default setting with 120 raters Z1 through Z120 and train the baselines on the raters' individual scores rather than the MOSes. Figure 3.11 reports the accuracy distributions for the personalized baselines vs. iQoE. As Section 3.4.1 mentions earlier, the training modules of models L and P are not publicly available. Figure 3.11 includes the results for models L and P for completeness. Although the personalization of the baselines reduces the accuracy gap, iQoE still outperforms all personalized baselines. In particular, the average gain by iQoE over the best baseline, which is model A, across all 120 raters is 30% and 27% in MAE and RMSE, respectively. Thus, we conclude that *iQoE derives its accuracy advantage from both modeling personalization and its specific design which combines the RIGS sampler and XSVR modeler.*

3.4.2. Simulations

3.4.2.1. Methodology

While Section 3.4.1 leverages crowdsourcing to scale up the number of raters, the method of subjective studies imposes other limitations on the evaluation scope. For example, it is incredibly difficult for a real rater to evaluate a series of 1,000 experiences. Because simulations are a common way to enhance the scope of real-world experiments, this section develops and applies a new simulation technique of synthetic profiling where a large number of synthetic raters quickly evaluate experience series of an (almost) arbitrary length. Another aspiration for the synthetic raters is to accurately represent the QoE perception of real raters. The proposed simulation technique utilizes the proliferation of parameterized QoE models and refers to them as profiles. Whereas the structure of a profile is predetermined, the profile produces a different instance with different parameter values when trained on a different dataset, e.g., the individual scores of experiences by a real rater. Each instance is a specific QoE model, which automatically and quickly produces a specific QoE value when presented with values of the influence factors. Hence, we utilize each instance to act as a synthetic rater. The training of p profiles on the

individual scores by g real raters has the multiplicative effect of creating $p \times g$ synthetic raters. To denote a synthetic rater, we combine the label of the respective synthetic profile with the number in the name of the real rater on whose individual scores we train this synthetic rater. For example, synthetic rater F32 corresponds to synthetic profile F and real rater H32. In the simulations, each experience contains seven chunks, and the playback of each chunk takes 4 s without stalls.

Because a profile aggregates the QoE perception by the raters in the reference group behind the respective QoE model, the usage of multiple diverse profiles allows the technique of synthetic profiling to realistically enhance the heterogeneity of the QoE perception in the simulations. The heterogeneity of the profiles also creates complications because the models vary in their ranges of produced scores. To emulate scores given by real raters, we define a scoring function for each synthetic rater by mapping the respective QoE model into the same scoring scale. Without loss of generality, we employ the 1-100 scale and map QoE models Q into scoring functions S through the following equation:

$$S = 1 + \frac{99}{1 + e^{-(Q-\sigma)\rho}} \tag{3.3}$$

where least-squares minimization between Q values and assessment scores by the corresponding raters configures parameters σ and ρ.

This chapter applies synthetic profiling to $p = 8$ profiles and $g = 32$ real raters to create $p \times g = 256$ synthetic raters. We take the 32 real Waterloo-IV raters from Section 3.2.1, models B, G, R, S, V, N, F, and A from Section 3.1 as the eight profiles, and utilize the 256 synthetic raters to conduct large-scale simulations where each synthetic rater assesses 1,000 experiences. We train (create) every synthetic rater and test (collect QoE values from) the synthetic rater on 70%, i.e., 700, and remaining 30%, i.e., 300, of all 1,000 experiences, respectively. In contrast, we train iQoE in its default settings on only $H = 50$ experiences.

We implement iQoE in Python 3.7. The implementations of RS, GS, UC, QBC, IGS and RIGS utilize the modAL framework. The implementations of SVR, RF, GP, and XGB are from the scikit-learn library with their optimal hyperparameters determined by grid search. The experiments run on an Intel i7 machine that has six cores, 2.6-GHz CPUs, 16-GB RAM, and Windows 10, with each experiment repeated five times by shuffling the experience set to improve generalizability.

3.4.2.2. Results

To validate the technique of synthetic profiling, we examine the Pearson and Spearman correlations between the scores by the 256 synthetic raters and the 34 real raters from the first phase of the subjective studies in Section 3.4.1. The scores are also for the same 120 experiences as in Section 3.4.1. For each real rater, we find the

	MAE				RMSE			
	XSVR	XGB	RF	GP	XSVR	XGB	RF	GP
RIGS	**4.3±0.3**	5.3±0.2	6.6±0.3	9.9±0.3	**6.4±0.3**	7.4±0.2	8.2±0.4	11.9±0.4
IGS	4.6±0.2	6.4±0.1	7.5±0.2	10.6±0.1	6.5±0.3	8.5±0.2	9.1±0.3	12.4±0.2
RS	4.7±0.2	4.5±0.2	4.7±0.3	9.1±0.3	7.8±0.4	7.7±0.4	7.6±0.5	12.1±0.5
GS	5.7±0.8	8.1±0.6	9.3±0.7	10.9±0.4	7.9±1.0	10.7±0.6	11.6±0.7	12.6±0.4
UC	9.7±1.9	7.1±0.4	7.7±0.3	14.4±2.0	13.2±1.8	9.8±0.4	10.5±0.2	17.2±1.8
QBC	5.0±0.2	4.9±0.3	6.4±0.4	9.0±0.5	8.1±0.4	8.4±0.6	8.0±0.4	12.3±0.8

Table 3.2: Accuracy of sampler-modeler combinations.

most correlated synthetic rater, and Figure 3.12 plots the distribution of these closest correlations across all real raters. Both Pearson and Spearman coefficients are high, above 0.7 for 80% of the real raters, and suggest that *the synthetic raters represent real raters accurately.* For brevity, the rest of this section refers to synthetic raters as raters.

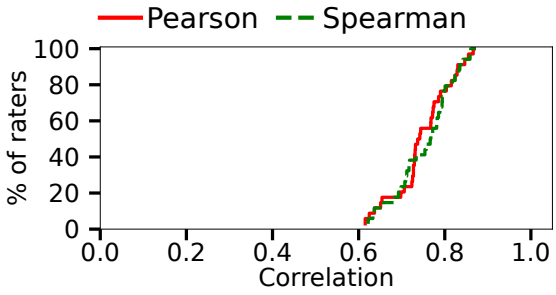


Figure 3.12: Synthetic vs. real.

Our large-scale simulations with 256 raters corroborate the conclusion of the subjective studies in Section 3.4.1 that the personalized QoE models built by iQoE greatly improve on the accuracy of the MOS-based baselines.

(1) iQoE design choices. We consider uncertainty clustering (UC) [299], query by committee (QBC) [300], IGS [301], GS [302], and RS as alternative samplers and extreme gradient boosting (XGB) [303], Gaussian processes (GP) [304], and RF [305] as alternative simple modelers.

Table 3.2 shows that, with 50 assessments by the 256 raters, the tandem of RIGS and XSVR is the most accurate among the 6×4 sampler-modeler combinations and enables iQoE to achieve average MAE and RMSE of 4.3 and 6.4, respectively, which are remarkably low for the 1-100 scale. With XSVR fixed as the modeler, Figure 3.13 shows that iQoE with its RIGS sampler consistently outperforms the RS+XSVR, GS+XSVR, UC+XSVR, QBC+XSVR and IGS+XSVR alternatives, e.g., reduces average RMSE to 7 after 39 assessments compared to 77, 57, 190, 96, and 43 assessments by the counterparts. The assessment effort decreases, respectively, by a factor of 1.97, 1.46, 4.87, 2.46, and 1.10. Figure 3.13 also plots the MAE and RMSE distributions for all individual raters after 50 assessments and also backs the choice of RIGS for iQoE. For example, while iQoE attains MAE of 6 for 85% of the raters, this percentage is 77%, 59%, 24%, 73%, and 79% for RS+XSVR, GS+XSVR, UC+XSVR, QBC+XSVR, and IGS+XSVR, respectively. Figure 3.14 examines different modelers in conjunction with RIGS. iQoE in its reliance on XSVR consistently delivers lower average MAE and RMSE than the RIGS+XGB,

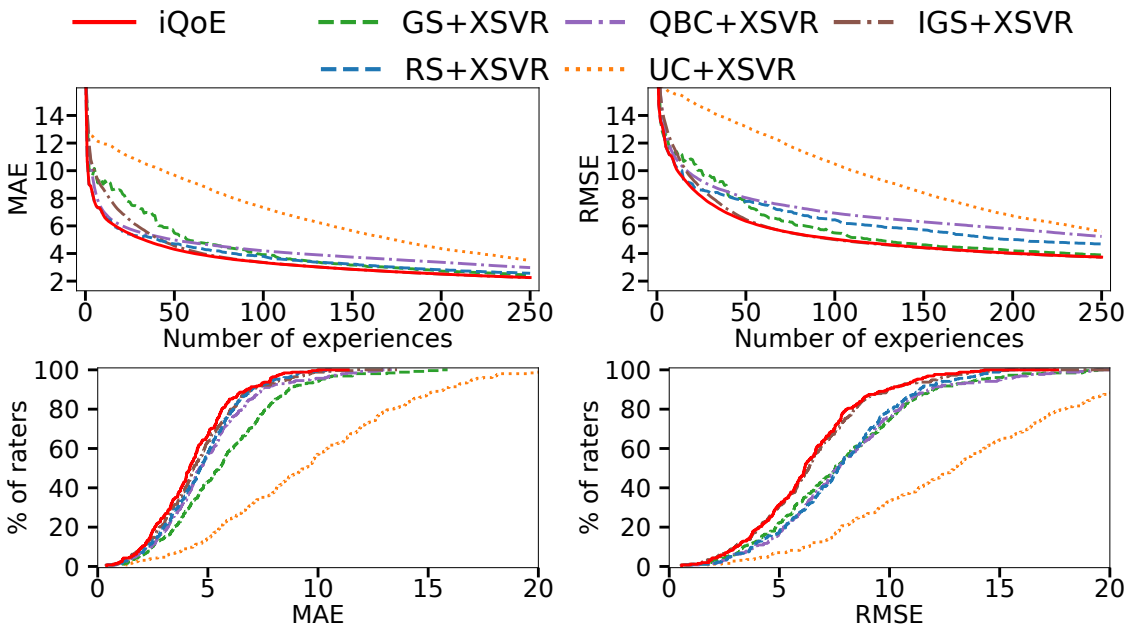


Figure 3.13: Evaluating the sampler design choice of iQoE.

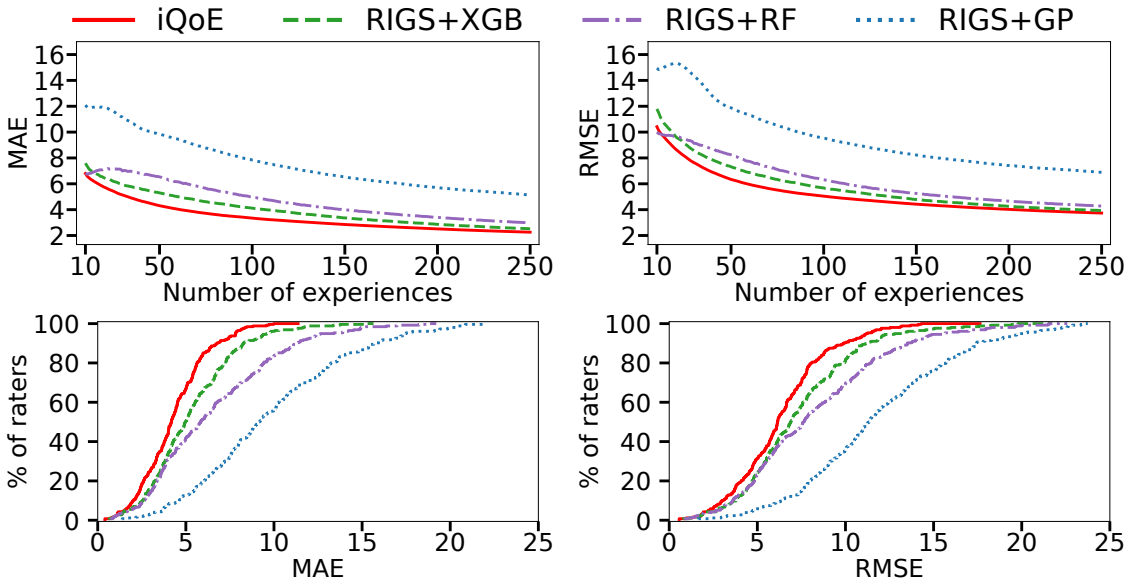


Figure 3.14: Evaluating the modeler design choice of iQoE.

RIGS+RF, and RIGS+GP alternatives. Overall, *the simulations support the adoption of RIGS and XSVR by iQoE.*

(2) Parameter Sensitivity. *Parameter h.* Figure 3.15 analyzes sensitivity of iQoE to its parameter h, which controls the switch from RS to IGS in RIGS. With $H = 50$ experiences, MAE and RMSE reach their minimum values around $h = 10$ experiences, justifying this default setting for h. With H set to 75 or 100 experiences, the impact

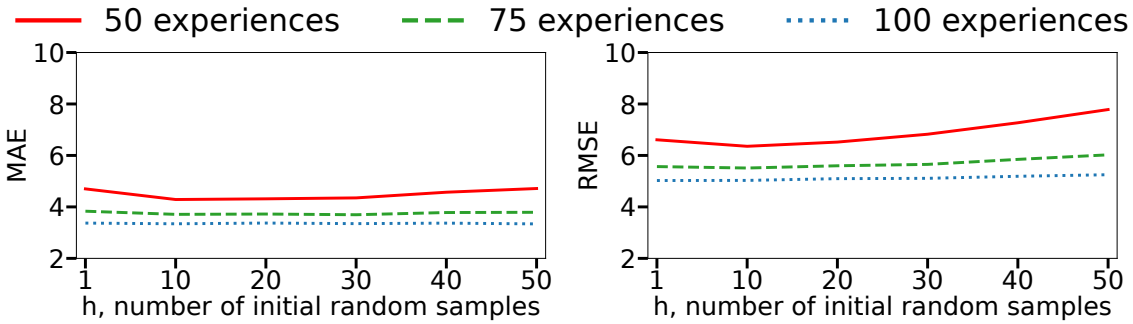


Figure 3.15: Sensitivity of iQoE to the h parameter.

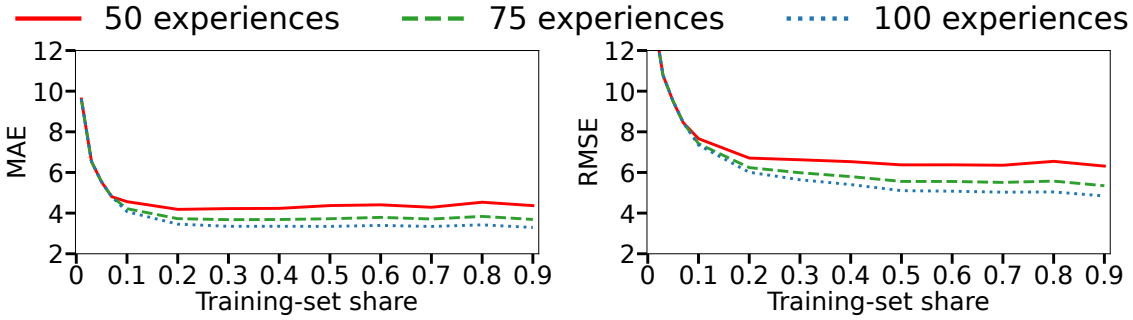


Figure 3.16: iQoE sensitivity to the training-set share.

of parameter h on the model accuracy is less pronounced. Despite the modest accuracy improvements, we consider RS a valuable contribution to RIGS because RS is also simpler than IGS.

Training Share of the Experience Set. By default, our evaluation uses 70% of the 1,000-element experience set for training, i.e., the training-set share equals 0.7. Figure 3.16 shows that average MAE and RMSE of the personalized QoE models are largely insensitive to the training-set share. Only when the training-set share decreases below 0.2 (i.e., 200 experiences), the inaccuracy starts to ramp up. This result opens the *possibility to train iQoE on a smaller experience set*, which has a positive effect of reducing the computational and storage overhead.

(3) iQoE generalizability. Not only size but also composition of the experience set affects the constructed QoE models. We create three experience sets by using BBA, TR, or MPC as the ABR algorithm in the Park platform. After training QoE models on each of the three sets, we test every QoE model on all three sets and ensure that the training and testing portions of the sets never overlap. This setup corresponds to a scenario where the current ABR algorithm of a QoE-based streaming system relies on a QoE model built with an outdated ABR algorithm. Figure 3.17 reports MAE and RMSE for all combinations of the training and testing ABR algorithms. Training with BBA yields excellent generalizability. With TR or MPC as the training ABR algorithm,

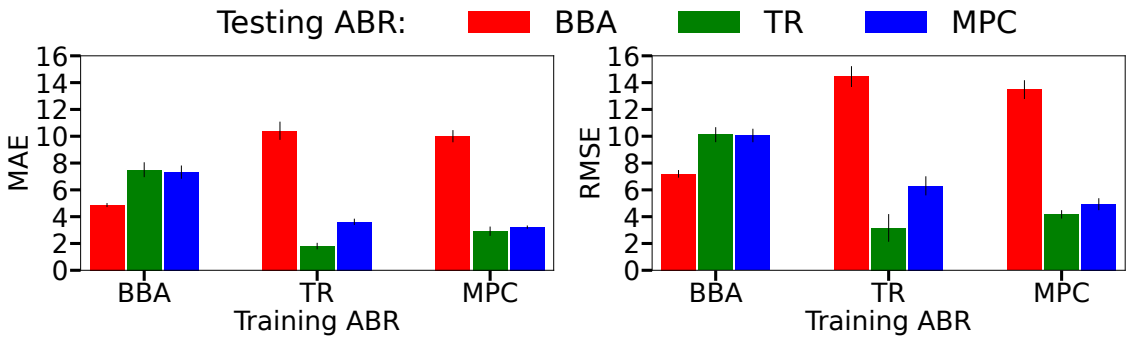
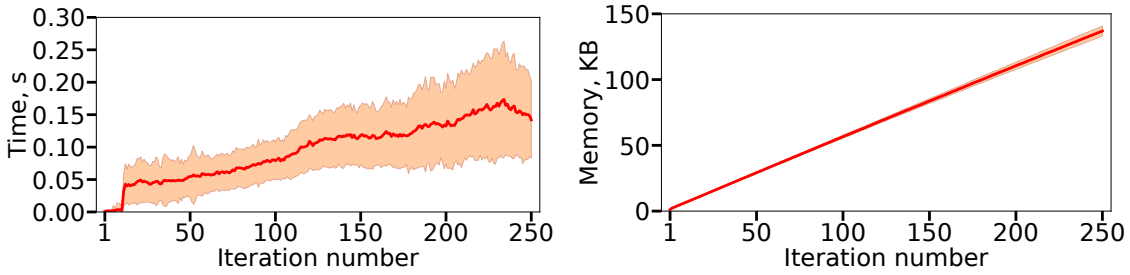


Figure 3.17: iQoE generalizability.



(a) Execution time

(b) Memory consumption

Figure 3.18: iQoE processing and memory overhead.

the generalizability is noticeably weaker. These results suggest that if the eventual ABR algorithm of the QoE-based streaming system is unknown at the time of constructing the QoE model, BBA constitutes a reasonable choice as the ABR algorithm for generating the experience set.

(4) iQoE overhead. iQoE employs an iterative design where each iteration entails selection of an experience by RIGS, assessment of the experience by the rater, and update of the QoE model by XSVR. We evaluate overhead for the automated part of the iQoE method per iteration, from obtaining the rater's score of an experience until selecting the next experience for the rater.

We measure execution time of every iQoE iteration for each rater. Considering all 256 raters, Figure 3.18a plots the average and standard deviation of the per-iteration execution time as a function of the iteration number. The execution time grows in a nearly linear pattern from iteration 11 to iteration 241, which reflects the linear increase in the number of the accumulated assessment scores. At iteration 11, the execution time exhibits a perceptible step-up because RIGS switches from RS to IGS. At iteration 241, the execution time slightly decreases due to issues related to XSVR training. At iteration 50, which corresponds to the default number of 50 experiences, the average execution time reaches 54 ms, and the respective wait of the rater for the next experience still remains insignificant. Because each experience lasts 28 s, 50 experiences take at least 23 minutes,

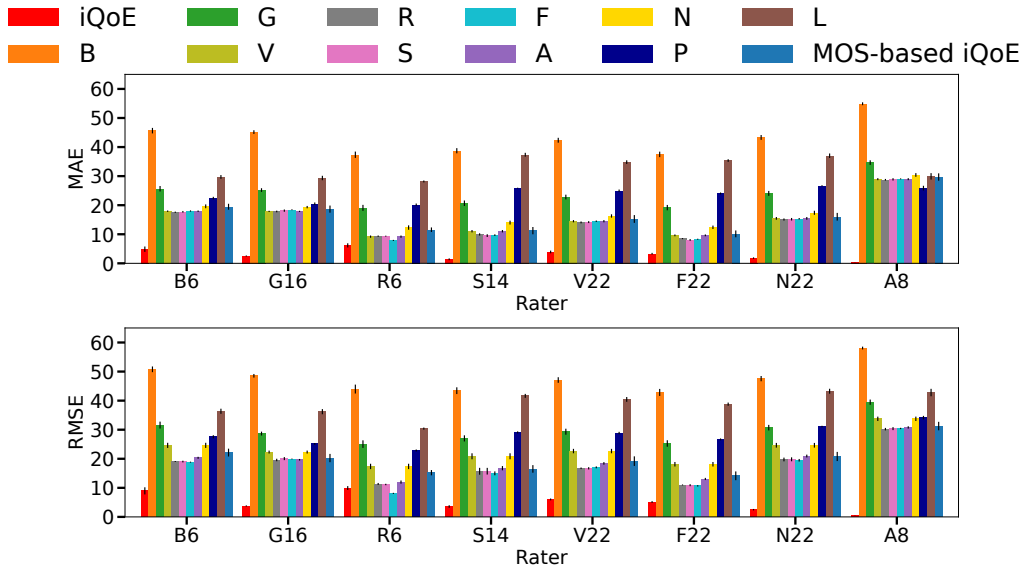


Figure 3.19: iQoE vs. baseline MOS-based QoE models in the simulations.

not accounting for unavoidable extra delays on the rater's side, such as additional time to provide the scores. The automated portion of the iQoE algorithm consumes a total of 1.82 s, which constitutes at most 0.13% of the total time to construct the QoE model.

Because memory pressure might degrade video streaming performance [306], we also track the memory consumed by the personalized QoE model during its construction. For all 256 raters, Figure 3.18b depicts the average and standard deviation of the memory consumption, which grows in a nearly linear fashion due to the increasing complexity of the refined QoE model. In the default setting with 50 experiences, the personalized QoE model consumes 20 KB on average, which is an extreme small amount by current standards. Overall, the results confirm that *the time and memory overhead of constructing an accurate personalized QoE model via iQoE is affordable.*

(5) Evaluation of iQoE vs. MOS-Based Modeling In this set of simulations, each of the 256 raters from Section 3.4.2 scores the 700 experiences in the training set. The respective 700 MOSes guide the construction of the eight MOS-based models, while models L and P use their default configurations. iQoE trains its personalized models on rater-specific sets of 50 experiences.

Figure 3.19 evaluates the iQoE and MOS-based models in worst-case scenarios for iQoE. From each of the eight rater profiles, we pick the rater whose QoE model has the largest MAE within that profile, e.g., rater B6 within profile B. Even in these worst-case scenarios for iQoE, the personalized QoE modeling decreases both MAE and RMSE substantially compared to the baseline MOS-based models. On average over all 256 raters, iQoE reduces MAE by a factor of 7.63, 7.38, 4.53, 3.57, 2.79, 2.60, 2.60, 2.55, 2.43, and 2.34 in comparison to MOS-based models L, B, P, G, F, N, V, S, R, and A, respectively. The

corresponding reduction factors for RMSE are 5.74, 5.81, 3.54, 3.04, 2.33, 2.24, 2.24, 2.14, 2.03, and 1.95. Hence, in the simulations, iQoE improves the model accuracy over the MOS-based models by a factor of at least 2.34 and 1.95 in MAE and RMSE, respectively.

Figure 3.19 also plots, as the rightmost bar in each bar group, the model accuracy of a MOS-based iQoE variant that constructs a QoE model based on the MOSes instead of the individual scores by a rater. Despite being trained on 50 experiences only, the MOS-based iQoE variant provides a similar model accuracy as the best of the 10 existing MOS-based QoE models.

3.5. iQoE Integration into a Video Streaming Platform

3.5.1. Integration into a Video SP

While the main focus of the chapter is on personalizing QoE models and making iQoE sample-efficient and accurate, we now discuss iQoE integration into video streaming systems. For concreteness, we start by considering a specific hypothetical video SP, or a platform for short.

Video streaming platform. The VoD platform extensively employs the cloud and, in particular, uses EC2 [63] to encode ingested video content into multiple representations and train its proprietary cloud-based ABR algorithm. The ABR algorithm leverages RL to maximize a one-size-fits-all QoE model combining VMAF, VMAF stability, and client-side stall duration as influence factors [208]. The platform quickly adapts to specific network conditions by means of transfer learning [307], keeps its vast library of videos in Amazon simple storage service (S3) [308], and utilizes Amazon relational database service (RDS) [309] to store the viewer's account information and large amounts of other structured data about viewing behaviors and preferences. By relying on Amazon Redshift [310], the platform analyzes the large-scale structured data to personalize content recommendations, trending lists, ad selection, etc. [57, 311–313]. For efficient low-latency distribution of the video content to viewers worldwide, the platform employs CDN services from Akamai [314].

The platform's client-side app. The platform has its own client-side application, or simply app, available on smartphones, tablets, laptops, and other device types. The standalone app provides the viewer with an interface to the platform's cloud-based services. The supported functionalities include authenticated access to the viewer's account [315, 316], retrieval of personalized content recommendations, viewing history, and other particulars of the viewer's profile. The app also allows the viewer to supply feedback, e.g., to rate videos and submit reviews, which the platform utilizes as an input to its cloud-based recommendation engine. During regular streaming of a video to the viewer, the app informs the cloud-based ABR algorithm about the client-side stall duration and

estimated throughput in real time by appending these data in the CMCD format [154] to the uniform resource locator (URL) of each HTTP secure (HTTPS) message requesting a video chunk from Akamai. The CDN scalably communicates the client-side data to the EC2 instance running the ABR algorithm of the session.

iQoE integration. iQoE represents an addition to the platform's vast personalization portfolio. The platform deploys iQoE as part of its regular updates of the client-side app and cloud-based infrastructure. Whereas the platform already stores historical streaming traces in S3 and RDS for the advanced data analytics in Redshift to identify popular content, preferred genres, etc., the platform reuses the historical traces to compile the experience set that iQoE utilizes later to construct personalized QoE models for viewers. The platform composes the experience set offline, characterizes the video chunks of each experience in the set with their precomputed values of QoE influence factors, e.g., VMAF, and stores the experience set in S3. With 100 GB allocated to the experience set, i.e., 10 times more than in Section 3.4, the storage requirement remains a tiny fraction of the space consumed by the platform's current personalization tasks.

Viewer's role in the iQoE integration. After the app incorporates iQoE, the app's interface offers the viewer the option to build a personalized QoE model via iQoE. If the viewer exercises this option, the app constructs the QoE model as described in Section 3.3, i.e., by downloading from the cloud-stored experience set, playing back, and collecting the score for each of the experiences chosen by iQoE for this viewer. The app uploads the constructed QoE model to the EC2 instance that trains a personalized ABR algorithm based on the personalized QoE model. The QoE and ABR personalization incur acceptable storage and bandwidth overheads: whereas the personalized QoE and ABR models, respectively, consume about 20 KB in the viewer's device and 3 MB in the cloud, the total amount of data communicated between the cloud and app during the QoE modeling is around 500 MB. The platform offers the viewer the trained personalized ABR algorithm as one of ABR options, including the one-size-fits-all ABR algorithm, for the viewer's subsequent streaming sessions. The app enables the viewer's profile to store up to four personalized ABR algorithms so as to accommodate different genres and contexts, e.g., streaming to a smartphone or HDTV device.

While the chapter deliberately differentiates between the typical and atypical viewers because the latter benefit more from QoE personalization and, thus, are more likely to adopt iQoE, the platform permits any viewer to take advantage of iQoE. To clarify the relative utility of iQoE for the viewer, the app's interface offers a similarity check between the viewer's personalized QoE model and the platform's one-size-fits-all QoE model.

Relationship with regular streaming. While the viewer dedicates time and effort to train the personalized QoE model via iQoE, the training of the ABR algorithm in EC2 occurs concurrently with regular streaming. Whenever the viewer wants to stream a

video, the viewer launches the regular streaming by selecting one of the ABR algorithms available in the viewer's profile.

Application-layer operation. iQoE and the platform as a whole operate on the application layer and communicate over HTTPS. Before or after incorporating iQoE, the platform does not explicitly deal with network resource allocation or its fairness. The different application-layer transmission patterns under the personalized ABR algorithms contribute to the increasing diversity in network traffic, e.g., caused by the bottleneck bandwidth and round-trip propagation time (BBR) [317] and CUBIC [318] congestion control algorithms which have different levels of aggressiveness.

Privacy. The platform's personalization of QoE modeling and ABR streaming strives for the same main goal as the platform's personalization efforts in general, i.e., enhancing the user experience. On the other hand, any personalization intrinsically raises concerns about privacy, albeit seemingly to a smaller extent for personalized QoE models than content preferences. The platform handles the extended set of concerns through its traditional methods of data protection and privacy control.

3.5.2. Extensions to Other Streaming Systems

Alternative implementations of ABR streaming. While Section 3.5.1 discusses integration of iQoE into a hypothetical video streaming platform that places the ABR logic in the cloud in alignment with current industry trends, iQoE also integrates easily with the traditional client-side ABR designs. The client-side ABR implementation diminishes the privacy concerns because the viewer retains the personalized QoE model. However, the local personalization of the RL-based ABR algorithm increases the load on the viewer's device. By adopting instead a control-theoretic ABR logic, e.g., BOLA [187], with the personalized QoE model as the optimization objective, the system decreases the client-side overhead.

Live streaming. In comparison to the VoD streaming in Section 3.5.1, live streaming imposes different requirements, such as lower end-to-end latency. The system design also changes, e.g., the client discards late frames instead of stalling the playback until the late chunk arrives. Live-streaming systems, e.g., those leveraging WebRTC [319], integrate iQoE by using QoE models with different influence factors, such as the frame rate [320]. Necessary modifications also include techniques suitable for measuring QoE factors in real time, e.g., PSNR instead of VMAF for video quality. In live streaming, the personalized QoE models offered by iQoE provide a promising foundation for not only ABR decisions but also dynamic construction of bitrate ladders [144].

Volumetric video streaming, VR, and AR. iQoE integration into these emerging applications is conceptually similar to the discussed above. The main difference arises due to the need for distinct QoE models that account for dissimilar application-specific influence factors. Motion-to-photon latency, viewport drift, and point density exemplify

the new relevant QoE influence factors [321,322]. The design of QoE models for volumetric video streaming, VR, and AR is a vibrant research problem without many definitive conclusions so far.

Fairness of network resource allocation. An intriguing possibility is to leverage iQoE to improve fairness of network sharing, especially because the resource allocation in the current Internet falls far short of theoretical ideals such as max-main fairness [323]. Whereas [260, 324, 325] apply max-min fairness to QoE rather than flow rates, the personalized QoE models built by iQoE represent an alternative to the one-size-fits-all QoE model as the basis for QoE fairness. As a word of caution, QoE fairness is a controversial objective because of diminishing the incentives for an application to achieve high QoE by utilizing the available network bandwidth more efficiently.

3.6. Related and Future Work

While [202, 297, 326, 327] show great heterogeneity of QoE perception among humans, our new dataset corroborates these findings. Unlike prior approaches to QoE personalization through indirect inference [227,269,287–289] or control knobs for a generic QoE model [188,202,271], iQoE is a novel personalization method that leverages a limited amount of explicit expressible feedback. Whereas [328–330] apply active learning to traditional MOS-based modeling, the focus of our work is on personalized QoE modeling. Future improvement of iQoE might benefit from additional influence factors that include personal traits [232, 331, 332], sensitivity [224], emotions [192, 333], and interests [181]. Although this chapter deals mostly with QoE, our plans are to expand the work into other aspects of video streaming such as power consumption on mobile devices [334, 335], efficiency and fairness of network utilization [186, 336], and cross-layer design [337]. Also, while iQoE does not seem to raise any privacy concerns, this question deserves a deeper investigation.

3.7. Conclusion

One-size-fits-all QoE models built by traditional MOS-based methods misrepresent the QoE perception by an atypical viewer. Seeking to empower the atypical viewers, this chapter proposes iQoE, a novel method that utilizes explicit, expressible, and actionable feedback from a viewer to construct a personalized QoE model for this viewer. iQoE combines the RIGS sampler with the XSVR modeler and exercises active learning so as to be sample-efficient and accurate. By allowing users to actively shape the functional form of their QoE models, iQoE aligns with the trend of user empowerment and fosters collaboration between users and SPs. We envision SPs implementing iQoE as an additional personalization tool within their service offerings. To validate the approach,

we use Microworkers to accomplish subjective studies with 120 raters who provide 14,400 individual scores. Based on the subjective studies, an iQoE session of about 22 minutes suffices for constructing an accurate personalized QoE model. Compared to the best of the 10 baseline models, iQoE delivers the average accuracy improvement of at least 42% for all viewers and at least 85% for the atypical viewers. The large-scale simulations support design choices and clarify performance properties of iQoE.

4 In-Band Quality Notification from Users to ISPs

ISPs manage network infrastructure using a tiered model that involves various relationships, such as peering agreements for direct traffic exchange and remote peering through IXPs. ISPs also buy and sell full or partial transit services to higher- or lower-tier ISPs, with lower-tier ISPs delivering direct connections to users, as described in Section 1.1 and depicted in Figure 1.2. Regardless of their level of direct interaction with users, all ISPs aim to enhance the QoE for end users to maintain a strong business reputation, attract and retain customers and peers, and remain competitive in the Internet connectivity market.

To mitigate QoE impairments at a granular level ISPs consider data flows, which are identified by the 5-tuple description constituted by source and destination IP addresses, port numbers, and protocol in use and follow a two-step approach. First, they detect problematic flows, through QoE inference, using techniques like per-flow deep packet inspection (DPI), which analyzes packet payloads passing through their routers. Then, they implement QoE-aware traffic management on a per-flow basis, such as rerouting or prioritization [52, 338].

This approach faces challenges with over-the-top (OTT) providers, represented by SPs in video streaming, which run their clients on end-user devices or use web browsers to engage directly with users through interactive interfaces and leverage vast cloud and edge infrastructures. Despite having the technical capability to assist ISP-side QoE inference [339, 340], they often opt not to cooperate, advocating instead for the enforcement of network neutrality regulations [15]. These regulations prevent ISPs from applying application-based traffic differentiation, except during temporary congestion, and generally limit their role to providing basic Internet connectivity. SPs fear that breaking neutrality could lead to economic flow differentiation, potentially harming their favorable market position.

This situation creates a tussle between the two parties [14], prompting SPs to increasingly raise fairness concerns to strengthen their position and adopt end-to-end traffic encryption which renders DPI-based QoE inference ineffective. For example,

YouTube utilizes the QUIC protocol [246] to encrypt its traffic. To overcome the problem researchers propose various alternative methods for extracting QoE insights from encrypted traffic [247,341–347], yet independent ISP-side solutions still struggle to achieve high performance.

Despite the ongoing disputes between ISPs and SPs, end users have a clear incentive to assist ISPs in improving QoE for specific applications [348]. There are indeed solutions for user-assisted QoE inference, such as application-layer traffic optimization (ALTO) [61] and proactive network provider participation for P2P (P4P) [62]. However, their adoption remains limited, primarily because they rely on out-of-band signaling of QoE information from the end user to the ISP.

Direct out-of-band communication occurs outside the primary data flow, but this mechanism suffers from the complexity of the Internet's structure, which consists of thousands of ISPs and multi-ISP paths [340]. These paths provide global connectivity for billions of end users and link various entities through diverse, typically bilateral, relationships. For example, ISP-level routing is asymmetric, and an SP operates its global private network by receiving client requests through one ISP and responding via another. When congestion occurs in a mid-path ISP and affects QoE, the end user cannot identify which ISP to contact via out-of-band signaling to resolve the issue [349]. Furthermore, the congested mid-path ISP is unaware of which users are impacted and cannot reach out via out-of-band communication, as NAT [350] alters the identifiers of network flows.

To tackle this challenge, this chapter proposes *in-band quality notification (IQN)*, a novel and practical mechanism for user-assisted ISP-side QoE inference that leverage in-band signaling of QoE from an automated end-user agent to server-to-client ISPs, without requiring any SP support. Its design takes inspiration from principles of SP independence, user simplicity, and Internet compatibility. The key idea is to leverage the SP client's interactive interface. IQN utilizes the interface to estimate the end user's QoE and signal the QoE estimates to the ISPs along the server-to-client delivery path. IQN relies on automated end-user and ISP agents, which run on the end-user device and in an ISP's routers, respectively. To communicate a QoE estimate, the end-user agent leverages the SP client's interface interactivity to programmatically issue a command that affects the pattern of packet transmission from the SP server to the SP client. Even though the SP server encrypts the traffic, the ISP agent along the server-to-client path detects and interprets the changed packet pattern as an IQN signal. Figure 4.1 extends Figure 1.2 by showing the IQN's working mechanism and integration inside the tiered ISPs structure in presence of congestion.

The IQN mechanism is policy-free because it communicates QoE estimates without dictating a policy on usage of this information by ISPs. Each ISP independently decides how to react to the notification. Similar to an SOS signal, IQN acts as a distress signal sent by the end-user agent through the SP service without knowing the intended ISP

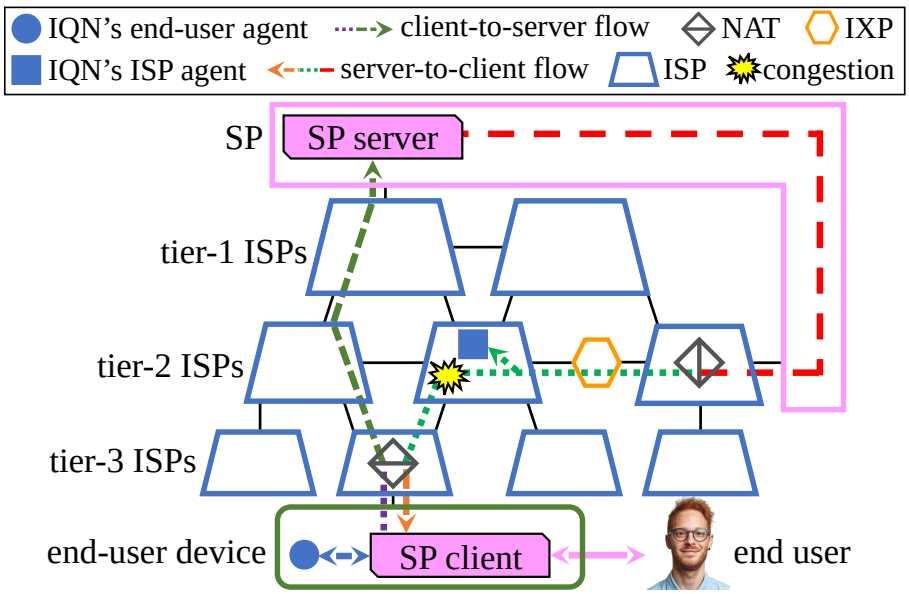


Figure 4.1: IQN signaling of QoE from the end user of an SP to server-to-client ISPs in the economically and technologically complex Internet ecosystem.

recipients of the signal. The end-user agent does not establish direct communication with any ISP and hopes that some ISP along the server-to-client path of the SP service detects the IQN signal in monitored traffic, associates any reported QoE impairment with ongoing local congestion, and takes action to improve QoE for the end user. Furthermore, like the SOS signal, IQN is a signal intended for transient situations rather than persistent use, ensuring compliance with network neutrality regulations.

We design IQN for voluntary installation on user devices, contributing to user empowerment through facilitated communication between ISPs and the users. This collaboration aims to enhance the QoE for services in exchange for the user's effort in installing the software, while leaving the service unchanged for those who choose not to install IQN. In our vision, ISPs or ISP consortium take the lead in developing and maintaining IQN's components, though this responsibility can also be handled by a third-party entity.

In addition to designing IQN, the chapter assesses the feasibility and performance of IQN signaling in a real Internet environment, through instantiation of a system prototype. After our preliminary studies show IQN's effectiveness for three major SPs and different QoE factors, we develop a prototype for inferring stalls of the QUIC-based YouTube service. This system, named *YouStall*, emits IQN signals by toggling the autoplay switch in YouTube's interface. The prototype incorporates an Amazon EC2 instance [63], which emulates an ISP router executing IQN's ISP agent to infer stalls and estimate their duration. IQN-assisted QoE inference at the router incurs less overhead compared to the traditional independent ISP-side QoE inference based on per-flow data-plane

DPI. Additionally, YouStall leverages IQN's end-user agent for user-side stall detection by periodically capturing screenshots and identifying the spinning icon in YouTube's playback area. The main objective of the stall detection is to avoid false positives, i.e., not to signal a stall spuriously.

Our experiments with YouStall on YouTube Live [351] streams of three genres corroborate the promise of IQN signaling. For example, while the average stall duration across the three genres exceeds 1.4 s, IQN-assisted ISP-side inference estimates the duration of significant stalls (those lasting at least 400 ms) with an average MAE and RMSE of 231 and 288 ms, respectively. The experiments also evaluate YouStall's parameter sensitivity and overhead.

4.1. Motivation and Principles

This chapter proposes a novel mechanism for ISP-side QoE inference. Although SPs, which directly interact with end users, are in the best position to assist ISPs in this task, real-world evidence shows little interest from SPs in cooperating with ISPs on QoE measurement and improvement. On the contrary, SPs often complicate independent QoE inference by an ISP through actions like end-to-end traffic encryption. Given this reality, we argue that an effective ISP-side QoE inference mechanism should not expect support from the SP service, leading to our first design principle:

Principle 1. *(SP independence) The mechanism should operate without any support from the SP.*

Since end users are the primary beneficiaries of QoE improvement, they have an incentive to assist ISP-side QoE inference [348]. However, this assistance should align with the low-effort role typical of casual users and avoid placing a significant burden on them:

Principle 2. *(User simplicity) The effort required by the mechanism from the end user should be minimal.*

While we allow for limited support from the end user, the complex structure of the Internet poses practical challenges for leveraging this support in ISP-side QoE inference. As Figure 4.1 illustrates, NAT might cause a network flow to change its identity as it traverses the Internet. Due to asymmetric inter-domain routing in the tiered Internet structure with multi-ISP paths [340], an SP server might receive client requests through one ISP and respond through another. Consequently, the ISP responsible for QoE-impairing congestion, and capable of resolving the issue, might be off the client-to-server path. This requires end-user support to reach ISPs along the server-to-client path. Hence, the QoE inference mechanism should account for such practical Internet constraints:

Principle 3. *(Internet compatibility) The mechanism should function effectively within the realities of the current Internet, such as multi-ISP paths, asymmetric routing, and NAT.*

4.2. In-Band Quality Notification

4.2.1. General IQN Mechanism

The design principles outlined in Section 4.1 guide our novel approach to QoE inference by an ISP. While mainstream efforts pursue independent ISP-side QoE inference, they struggle to accurately infer QoE from encrypted traffic [341,342]. As per Principle 1, SPs are unlikely to remove encryption or assist ISPs in inferring QoE through other means. Therefore, we explore *ISP-side QoE inference assisted by the end user.*

The key issue addressed in this chapter is QoE signaling from end users to ISPs. Although ALTO [61], P4P [62], and other cross-layer designs explicitly support such signaling, we attribute their low adoption to a mismatch between their out-of-band nature and the Internet's structural complexity. With multiple tiers of ISPs, multi-ISP paths, and congestion occurring before the last-hop ISP, the end user is typically unaware of which ISP is responsible for QoE-impairing congestion [349]. Similarly, an ISP is often unaware of the specific end users due to NAT and other modern Internet realities. Since the end user generally does not know which ISPs to notify about QoE impairments via an out-of-band mechanism, our research focuses on in-band signaling.

According to Principle 3, the signaling mechanism should be compatible with asymmetric routing and notify ISPs along the server-to-client path of the SP service. One possibility is for the SP server to reflect in-band signals received from the end user, similar to how network cookies facilitate network neutrality [348]. However, this support from the SP would violate Principle 1. Hence, we seek a mechanism for *in-band signaling to server-to-client ISPs without relying on SP support.*

End users typically exert low effort when consuming services, often limited to installing and running applications like an SP client. For example, explicit congestion notification (ECN) [352], which transmits in-band congestion signals from networks to end devices for transport-layer reactions, operates transparently to end users and their applications. In conformity with Principle 2, we aim to maintain this minimal level of effort, requiring *only the installation of an agent on the end-user device, which automatically signals QoE.*

The principle-guided approach leads to the design of IQN, a new mechanism for in-band signaling of QoE information from an automated end-user agent to server-to-client ISPs, without relying on SP support. IQN's key insight is that each major SP service provides a rich interactive interface, allowing the end user to issue commands through the SP client and receive responses from the SP server. IQN's end-user agent leverages

this interface to programmatically send a series of commands that induce the SP server to transmit a distinctive packet pattern, aligned with the SP service's internal logic, to the client. This packet pattern encodes QoE information, which the server-to-client ISPs infer by detecting the pattern in the network flow.

While this chapter primarily focuses on QoE signaling from the end user to server-to-client ISPs, which is our most innovative contribution, we also examine other, less novel but important, aspects of user-assisted ISP-side QoE improvement. For example, the end-user agent automatically detects QoE by leveraging again the SP client's interface.

For ISP-side mitigation of inferred QoE impairments, we consider well-established techniques such as traffic prioritization [52] and rerouting [338]. For example, when an ISP infers a QoE impairment in a network flow on a congested link, it assigns higher priority to that flow. If multiple server-to-client ISPs infer the same impairment, they respond independently based on their own understanding of potential local causes, without coordination. IQN acts like a distress signal to any ISP capable of resolving the transient QoE-impairing congestion, an emergency rather than a persistent issue.

4.2.2. IQN Instance in the YouStall System

To evaluate the feasibility and accuracy of the IQN mechanism in real Internet environments, we instantiate it in a system prototype. Our preliminary analysis of Amazon Prime Video, Netflix, and YouTube reveals that their client interfaces offer interactive features, such as an autoplay switch and audio language selection, that support effective IQN signaling. In line with the scenarios targeted by IQN, we develop the prototype for YouTube, which encrypts its end-to-end traffic in QUIC packets. The system notifies ISPs about playback stalls, a prominent QoE influence factor [353]. We refer to the prototype as YouStall.

User-side QoE signaling. To issue IQN signals via YouTube's interface, YouStall utilizes the autoplay switch, which determines whether the next video plays automatically after the current one finishes. We select this feature to avoid disrupting the user experience. The end-user agent emits an IQN signal by toggling the autoplay switch programmatically an even number of times, with intervals shorter than 5 ms between toggles. This sequence of n toggles occurs too quickly for human perception and concludes with both the cursor and autoplay switch in their original positions. By default, we set n to 4 as the smallest even number that is unlikely to occur naturally in user behavior while still triggering a distinctive pattern in server-to-client transmissions.

The end-user agent issues an n-toggle IQN signal whenever its QoE check, conducted at intervals of p (set to 200 ms by default), detects a playback stall. For each toggle in the IQN signal, YouTube's client transmits an HTTP version 3 (HTTP3) POST request to the YouTube server, which updates its autoplay settings and responds with an HTTP3 "200 OK" status over QUIC. Although a monitored network flow might include other QUIC

packets that contain encrypted "200 OK" confirmations, a quick series of such packets is a rare pattern.

ISP-side QoE inference. To identify this distinctive pattern in each monitored network flow, an ISP runs IQN's ISP agent on the data plane of its routers. The ISP agent identifies QUIC-encrypted candidate packets sized between 1,180 and 1,288 bytes, which might contain "200 OK" confirmations. Because some of these packets might be unrelated to an IQN signal, the ISP agent has to discern and filter them out. Additionally, the client-server-agent path might experience packet loss, reordering, and significant jitter due to in-network and end-host processing [354], causing variations in the count and timing of packets representing an IQN signal upon its arrival at the ISP agent.

The algorithm for stall inference operates in two modes: out-of-stall and in-stall. In the out-of-stall mode, the ISP agent monitors packet arrival times by sliding a stall-start window over them. It infers the start of a new stall when the window contains at least n packet arrivals. The algorithm registers the first arrival time as start time s of the stall and switches to the in-stall mode. In the in-stall mode, it slides a stall-end window over packet arrival times and infers the end of the stall when the window contains fewer than n packet arrivals. The ISP agent records the first of these times as end time e of the stall, estimates the stall duration as $e - s$, and then switches back to the out-of-stall mode.

We keep the algorithm simple by sizing both stall-start and stall-end windows relative to signaling interval p, instead of introducing new independent parameters. We size the stall-start window to $w_s = \frac{3p}{2}$, i.e., 50% longer than interval p. Despite potential timing distortions along the delivery path, we expect this duration to be long enough for n packets to arrive and inform the ISP agent about the onset of a stall. On the other hand, $w_s = \frac{3p}{2}$ is short enough to avoid inferring a stall spuriously due to the arrival of unrelated candidate packets. We empirically set the stall-end window to $w_e = 4p$ to reduce false negatives, which might occur if the end-user agent fails to emit an IQN signal during an actual stall. Because unrelated candidate packets are infrequent, they do not interfere with inferring the end of a stall when w_e is as long as $4p$. Additionally, setting $w_e = 4p$ ensures that this window is shorter than the typical gaps between actual stalls, thus preventing the inference of multiple stalls as one.

Per-flow tracking of packet sizes and arrival times in IQN-assisted QoE inference imposes less overhead than the stateful packet processing required for traditional data-plane DPI, which advanced routers already support for independent ISP-side QoE inference and more complex tasks [355, 356]. Therefore, we expect the proposed method to scale to ISPs in the Internet core.

User-side QoE detection. To signal QoE, the end-user agent has to detect it first. One possibility is to leverage YouTube's "stats for nerds", a feature that provides end users with detailed QoE information. We demonstrate IQN's feasibility and effectiveness in more general settings, where an SP does not disclose detailed QoE metrics and instead

offers limited visual cues, such as the spinning icon in the center of YouTube's playback area during a stall, to indicate temporary QoE impairments.

YouStall's end-user agent detects stalls by periodically capturing a screenshot of the playback area at interval p. If the central part of two consecutive screenshots changes while the periphery remains the same, the end-user agent attributes the change to the spinning icon and infers a potential stall. To avoid sending spurious IQN signals, the end-user agent strives to minimize false positives by incorporating logic to filter out fictitious stall detections caused by user actions, such as clicking the progress bar. While the end-user agent does not detect stalls shorter than p, this feature aligns with YouStall's aim to inform ISPs about longer stalls that noticeably disrupt end-user QoE and are resolvable by ISPs.

ISP-side QoE improvement. We develop the YouStall prototype for IQN-assisted QoE inference primarily to evaluate the effectiveness of IQN signaling, which is our main innovation. QoE improvement via in-network techniques, such as flow prioritization and rerouting, is well-researched and offers well-known performance benefits. For example, [357] demonstrates that these techniques enable ISPs to effectively mitigate temporary QoE impairments, including stalls, when informed through out-of-band signaling. Enhancing YouStall with similar QoE-improving mechanisms is a direction for future work.

YouStall implementation. We implement YouStall in Python 3.10. YouStall's end-user agent utilizes Pillow [358] for image manipulation and sets the pause parameter of PyAutoGUI [359] to 0 for rapid toggling of the autoplay switch. The ISP agent employs pyshark [360] for real-time packet capture and analysis. We openly release YouStall's code and our experimental configurations on GitHub [361].

4.3. Evaluation

4.3.1. Experimental Setup

IQN's end-user agent runs on a Madrid-based Intel i7 machine (six cores, 2.6 GHz CPU, 16 GB RAM) with Linux Ubuntu 22.04.4 LTS. Located in Frankfurt, approximately 1,500 km away, an EC2 instance (one core, 2.4 GHz Intel Xeon CPU, 1 GB RAM) with Linux Ubuntu 24.04.4 LTS emulates IQN's ISP agent. We tunnel the end-user device's Internet traffic to the EC2 instance using OpenVPN [362]. An automated script plays YouTube Live [351] video streams from the news, music, and sports genres on the end-user device through the Google Chrome browser.

Since YouTube stalls are typically infrequent, our evaluation induces 100 stalls per genre by fixing the video resolution at 720p and using tcconfig [363] to limit the

YouTube traffic to 1 Mbps [364]. We collect ground truth on the stalls by employing Selenium [365] to track the ytp-spinner tag, which renders the spinning icon in YouTube's interface. We synchronize the timelines of the stalls and captured screenshots. To assess memory and central processing unit (CPU) overhead, we utilize psutil [366].

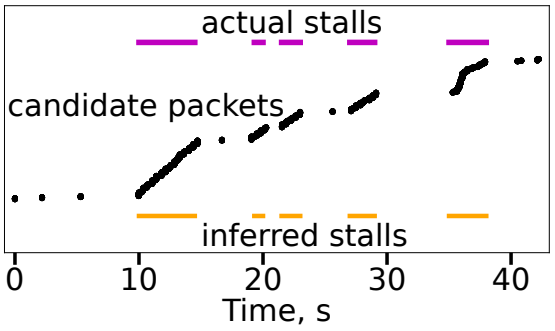


Figure 4.2: An operational example

4.3.2. Experimental Results

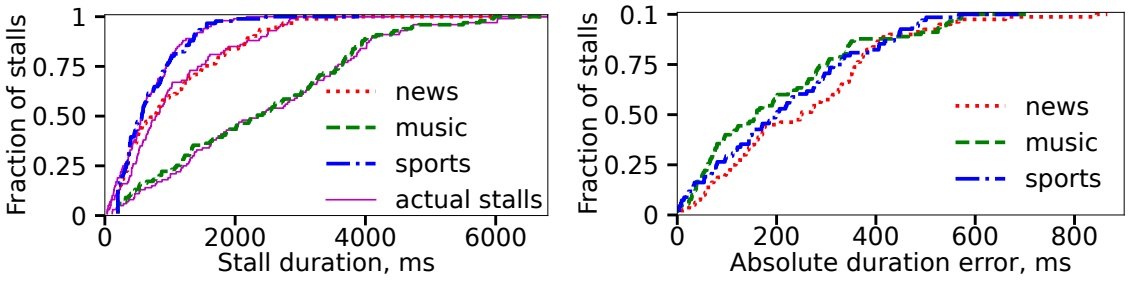
Figure 4.2 illustrates YouStall's operation. The horizontal magenta segments indicate actual stalls, which prompt IQN signaling from the end-user agent. Each black dot represents the arrival of a candidate packet at the ISP agent. Between stalls, the ISP agent observes infrequent candidate packet arrivals unrelated to IQN signaling. During a stall, the ISP agent detects a higher arrival rate of candidate packets and correctly infers the stall. Figure 4.2 depicts the inferred stalls as horizontal orange segments.

Figure 4.3a shows the inferred stall-duration distributions for the news, music, and sports genres, with the respective ground-truth distributions plotted as solid magenta lines. Although the actual stall-duration distributions for the three genres are distinct (with averages of 1.1, 2.6, and 0.7 s), the inferred and actual distributions for each genre are very similar, indicating that YouStall accurately infers stall-duration distributions.

Since we experiment with YouStall in real-world conditions, where the path between the two agents introduces substantial jitter of up to 1 s, evaluating the inference accuracy for each stall is challenging. We utilize the end-user and ISP-agent timelines to match actual and inferred stalls within a 1-s window. For significant stalls lasting at least 400 ms, we confidently identify 238 one-to-one mappings between inferred and actual stalls across the three genres (from a total of 300 actual stalls). The average MAE and RMSE for these 238 mappings are 231 and 288 ms, respectively. Figure 4.3b depicts the absolute error of the inferred stall duration for each genre. These results suggest that YouStall estimates the duration of significant stalls with relatively low error.

Based on the synchronized timelines of actual stalls and captured screenshots, Figure 4.4 presents the precision and recall of YouStall's user-side QoE detection. The sampling interval has minimal impact on these metrics. The end-user agent detects stalls with nearly 100% precision, effectively avoiding spurious IQN signals. However, recall is lower because YouStall, by design, does not detect stalls shorter than p. Because shortening p increases the number of detected stalls, we set the default sampling interval to the shortest examined value, $p = 200$ ms.

Figure 4.5 presents the overhead of YouStall's end-user agent, based on four runs



(a) Duration of inferred vs. actual stalls (b) Duration error for significant stalls

Figure 4.3: IQN signaling from the end user enables accurate QoE inference by the ISP along YouTube's server-to-client path.

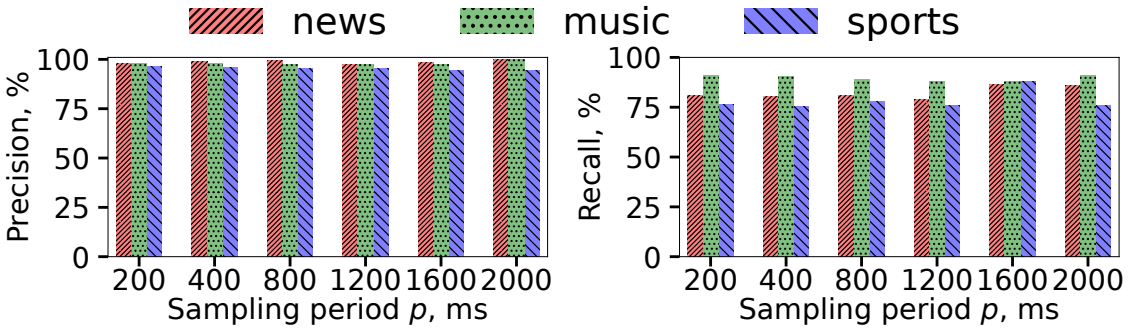


Figure 4.4: Precision and recall of user-side QoE detection.

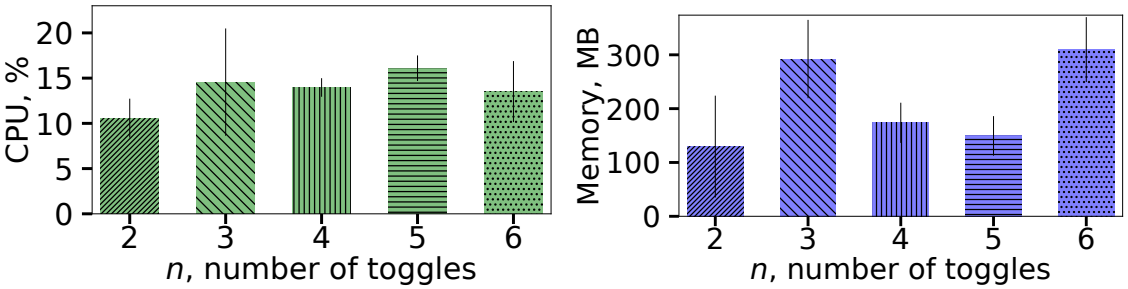


Figure 4.5: Overhead of YouStall's end-user agent.

with 10 actual stalls. The CPU and memory consumption are approximately 12% and 200 MB, respectively. This overhead is manageable and does not significantly depend on the number of toggles constituting an IQN signal.

4.4. Related Work

Independent ISP-side QoE inference. [341] identifies video segments in encrypted traffic by examining packet sizes and timing. Based on combinatorial matching, [342] infers video resolution when QUIC multiplexes video and audio content. [247, 343, 344] employ decision trees, random forests, and convolutional neural networks, respectively,

to infer various QoE metrics. Other solutions for QoE detection from encrypted traffic include [345–347]. Despite the growing effectiveness of learning methods, independent QoE inference by an ISP still struggles to achieve high performance. In contrast, IQN enables accurate user-assisted ISP-side QoE inference.

ISP-OTT cooperation. Collaboration with OTT providers, represented by SPs in video streaming, promises tangible benefits for ISPs. While [339] explores ISP-OTT cooperation in software-defined networking to jointly optimize intra-domain and inter-domain routing, [340] enables cooperation between OTT hypergiants and remote ISPs without requiring direct peering. Whereas practice shows that SPs are reluctant to assist ISPs in QoE inference, we design IQN to operate without any explicit assistance from the SP service.

QoE signaling from end users. While ALTO [61] and P4P [62] enable QoE signaling from the end user to an ISP, the out-of-band nature of these mechanisms is only weakly compatible with modern Internet realities, such as multi-ISP paths and NAT. In-band signaling via network cookies [348] differs from IQN by requiring support from the SP server.

In-band notification. Similarly to IQN, ECN [352] is an in-band mechanism for conveying congestion-related information. Whereas ECN sends congestion signals from networks to end devices for transport-layer reactions and does not distinguish between applications, IQN enables an end-user agent to notify networks about congestion-triggered application-specific QoE impairments. Unlike IQN, which functions as an emergency signal for transient QoE-impairing congestion, the in-band signaling in rate-delay (RD) network services [367] supports persistent differential treatment of application classes that prioritize either high throughput or low latency for their network flows.

User-side QoE detection. While VideoEye [368] detects QoE offline from recordings of the SP client's screen, Tero [369] extracts QoE from thumbnails in gaming footage. In contrast, our YouStall prototype captures screenshots to detect and signal QoE in real time. While [370] evaluates how user interactions with an SP client affect QoE of encrypted video sessions, YouStall leverages the SP client's interface to induce a distinctive pattern of packet transmission from the SP server.

4.5. Discussion

Interference by the SP provider. Beyond simply not assisting, SPs might deliberately disrupt IQN signaling. Such actions would frustrate IQN-agent developers and ISPs, potentially leading to the introduction of neutrality regulations for SPs, similar to those that currently govern ISPs.

SP-specific IQN instantiation. SP services differ significantly in their interface features and operation, potentially requiring a tailored IQN instance for each service.

Since the number of SPs is relatively small, maintaining separate IQN instances for all of them appears scalable.

Future work. The primary contribution of this chapter is IQN signaling, which enables accurate ISP-side QoE inference from encrypted traffic without any support from SPs. While our preliminary results substantiate IQN's promise, this pioneering work opens up numerous directions for further research, such as encoding additional QoE factors into IQN signals and improving QoE inference from observed packet patterns. We also plan to enhance the YouStall prototype with QoE-impairment mitigation mechanisms and evaluate their performance gains in real large-scale network environments.

4.6. Conclusion

ISPs aim to improve users' QoE by detecting and mitigating per-flow QoE impairments through traffic management, but the increasing use of flow encryption by SPs makes this difficult. This challenge is worsened by the tension between ISPs and SPs, with SPs advocating for network neutrality, which ISPs wish to relax. Existing out-of-band solutions are often ineffective due to the complexities of network infrastructure. To address this, this chapter introduces IQN, a novel QoE impairment signaling mechanism compatible with multi-ISP paths, asymmetric routing, and other Internet realities. IQN works without requiring support from SPs and induces SP servers to generate distinctive packet patterns encoding QoE information, enabling ISPs to infer QoE by monitoring these patterns in network traffic. The mechanism relies on an end-user agent, to trigger QoE signals through the SP's client interface, and an ISP agent, to detect them in traffic flows. IQN fits within the user empowerment landscape by enabling collaboration between ISPs and users, where the users install additional software to support IQN on their devices. We consider the ISP or an ISP consortium to be the most suitable entities to develop and manage IQN, though a third-party entity could also take on this role. This maintenance involves updating the agents in response to any changes made by the SP to its client interface. To validate IQN, we develop a prototype called YouStall and conduct experiments on YouTube Live streams, utilizing YouTube's autoplay switch for QoE impairment signaling and an Amazon EC2 instance for detection of these signals in QUIC traffic. The results show IQN's promise, with YouStall estimating the duration of significant stalls with an average MAE of 231 ms and RMSE of 288 ms. We also evaluate IQN's parameter sensitivity and overhead.

5 QoE in Video Streaming: Status Quo, Pitfalls, and Guidelines

QoE plays an important role in the design, operation, and evaluation of networked computer systems that serve humans. As explored in Section 2.5.1.3, QoE reflects the overall satisfaction of a user with an application, making it a subjective personal concept. Additionally, QoE depends on multiple factors related to the user's current context, such as network connectivity, device type, and content nature. The ITU, a United Nations agency, plays a key role in improving QoE across different systems and platforms by developing global telecommunication standards.

A comprehensive understanding of QoE is essential for developing and evaluating effective user empowerment strategies, which are inherently designed to enhance QoE. By gaining deeper insights into QoE, service providers can significantly improve the effectiveness of these strategies and foster new forms of active user engagement, as well as expand existing ones. This understanding enables a more seamless involvement of users into current video streaming systems and paves the way for innovative methods to implement these strategies effectively.

This chapter, aligned with the thesis's focus, examines QoE in ABR video streaming [25], which dominates a significant portion of Internet traffic [371, 372], as discussed in Chapter 2. Figure 1.1 illustrates how, in an ABR streaming session, the media server divides the video into chunks and encodes each into multiple bitrate-resolution pairs. The client's ABR algorithm selects the appropriate representation for each chunk to adapt to network conditions. High bitrates may cause delays and playback stalls, reducing QoE, while low bitrates result in lower video quality, also affecting QoE. Although this chapter focuses on ABR streaming, the insights are relevant to QoE in other networked systems.

While appealing as a basis for user-centered system design, operation, and evaluation, QoE raises a variety of practical complications. In particular, QoE subjectivity implies that direct assessment of QoE involves subjective tests where human raters provide scores for experiences presented to them. However, lab-based subjective assessments consume

significant amounts of time and effort, and online crowdsourcing alternatives mitigate the overhead concerns only to some extent [273, 373].

The overhead of subjective tests fuels the emergence and wide spread of QoE models. A QoE model automatically derives QoE from objective IFs, such as stall duration and bitrate changes across consecutive chunks [42]. Traditionally, as shown in Figure 1.3, the construction of a QoE model presents a series of experiences to a group of raters, averages the raters' individual scores to compute the MOS of each experience, and approximates QoE as a function mapping the considered IFs to MOS. Figure 2.7 illustrates the flowchart of this process. Although some accuracy concerns are discussed in Chapter 3, the key advantage of the traditional QoE modeling is that only a relatively small group of raters participates in subjective tests whereas the constructed QoE model automates QoE assessment for all users of the application without imposing any subjective-test overhead on a huge majority of them.

Despite offering the scalable automated support for QoE assessment, QoE models spawn new difficulties. Human perception of video is complex, and many IFs of different kinds are pertinent to QoE [69–72, 374]. Besides, practical considerations necessitate that the IFs of a QoE model are measurable by the entity that uses the QoE model, with these measurements being sufficiently accurate and incurring only low overhead. For example, although research indicates promise of electroencephalographic signals as IFs of QoE [375], a streaming provider is unlikely to deploy a large-scale application that attaches electrodes to the users' scalps. Instead, it is typical for an SP to directly measure stall duration and estimate available network bandwidth based on throughput observations in the client. QoE models also diverge with respect to the approximation function that maps the considered IFs to QoE. For instance, closed-form expressions and learning-based approaches are both common in QoE modeling. Consequently, there exist a large number of diverse QoE models, with the most relevant ones reviewed in Section 3.1.

The diversity of QoE models also arises due to different usages of the models. Timing and accuracy considerations might necessitate different models for design, operation, and evaluation of systems. In particular, a complex QoE model might be suitable for offline design or evaluation but not for real-time operation of an ABR algorithm. For example, whereas the PSNR [135] and VMAF [166] are metrics of video quality that dramatically differ in their computational requirements, [144] relies on PSNR to predict video quality during live streaming and leverages VMAF to evaluate the actually achieved video quality.

Besides, the multifaceted QoE problem involves separable tasks such as test conducting, model building, and model using. *Test conducting* performs subjective tests. *Model building* constructs QoE models based on subjective scores. *Model using* utilizes QoE models in system design, operation, or evaluation. A single work might handle multiple tasks. For instance, iQoE [2] both conducts subjective assessments and constructs personalized QoE models. SENSEI [224] and Ruyi [59] address all three tasks

of test conducting, model building, and model using. ARTEMIS [144] neither builds nor validates a QoE model and instead uses an existing QoE model to evaluate its proposal that dynamically configures the bitrate ladder of a live ABR streaming session. The separation of concerns in dealing with QoE has both positives and negatives. On the one hand, the focus on a single task enables its more thorough execution. On the other hand, the limited outlook might derail the overall effort, e.g., when an ABR algorithm uses a QoE model validated for dissimilar settings.

The importance, complexity, and separation of concerns put QoE in a precarious position. The widely recognized importance of QoE creates expectations to consider QoE in ABR video streaming, at least for evaluation if not for design and operation. However, QoE complexity makes comprehensive treatment of QoE difficult. Furthermore, the diversity of existing QoE models creates a false impression that one may easily introduce a new QoE model without a proper validation. Hence, QoE becomes both the holy grail and a free-for-all.

In this chapter, we review the current landscape of QoE in ABR video streaming and zoom in on a number of areas including: (a) scoring scale, interface design, and experience selection for subjective tests, (b) validation, value interpretability, and capping of the value range in QoE modeling, (c) mismatch between usage and construction of QoE models, (d) evaluation of QoE models via correlation vs. error metrics, and (e) QoE evaluation of ABR algorithms. We identify problems afflicting these areas and offer advice on how to rectify the situation. Our methodology leverages real data and arranges the advice in accordance with the classification of QoE-related tasks into test conducting, model building, and model using. In recommending the good practices for subjective assessments, construction and usage of QoE models, our overarching aspiration is to foster high standards in future work on QoE in ABR streaming and to provide valuable insights for enhancing user empowerment. While we expect the increased awareness and good practices to be the most valuable for newcomers to the field, this chapter serves as a wake-up call for the entire community to acknowledge and address the identified problems.

5.1. Background

QoE has its roots in QoS, an earlier notion from packet-switched computer networking. QoS characterizes network performance via such metrics as the transmission rate, packet loss, end-to-end delay, and delay jitter provided to applications [26]. Two main features differentiate QoE from QoS. First, QoE shifts the focus from objective system performance to the user's subjective perception of the performance. Second, while QoS is rather an umbrella term for multiple metrics, QoE constitutes a holistic concept capturing the user's overall satisfaction with the application. The evolution from network-centered QoS

to user-centered QoE not only fulfills the interests and needs of SPs but also is relevant to ISPs. For example, an ISP might utilize a QoE model as a basis for allocation of link capacities to video streams [260].

In subjective QoE tests of ABR video streaming, an experience refers to a sequence of chunks played back by the client to a rater who provides a score for the experience. When a subjective test collects scores for a series of experiences to support construction of a QoE model, the test also records the IF values of each rated experience. To keep the load on the raters manageable, the series of experiences should be relatively short. [37,297,376], and, to a smaller extent, [284] select the experiences and their IF values to be representative of real-world settings.

The scoring scale is an important element of subjective testing methodologies, including those standardized by ITU [40]. For instance, ACR is a popular method with a five-level scale where integers from 1 to 5 constitute bad, poor, fair, good, and excellent levels [36]. Another common scale consists of 100 levels where level ranges 1-20, 21-40, 41-60, 61-80, and 81-100 correspond to bad, poor, fair, good, and excellent QoE, respectively [2, 37, 39, 297]. While such discrete absolute scales are the most typical, alternative testing methods employ continuous scales for scoring an experience, assess QoE degradation rather than QoE itself, or perform pairwise comparison of experiences [40,41]. According to [377] and [378], usage of continuous vs. discrete scales exhibits no significant statistical differences in QoE assessment.

Building a QoE model based on the experiences' scores and IF values has many methods of different kinds at its disposal. Although classification techniques seem a natural fit for modeling of discrete QoE scores, regression methods dominate QoE modeling. In this chapter, we consider 10 existing QoE models and, for brevity, refer to them with the following single-letter labels: B [46], G [187], R [276], S [183], V [208], N [261], F [47], A [231], P [379], and L [277]. The first six of these QoE models rely on regression with similar linear target functions and account for video quality with a different IF. The target function of QoE model F is exponential. The construction of QoE models A, P, and L relies on machine learning and, specifically and respectively, on support vector regression, random forest, and long short-term memory.

When constructed, a QoE model avails itself to various usage in system design, operation, and evaluation. MPC [46] makes ABR decisions via model predictive control based on QoE model B (as labeled above). Pensieve [51] is an ABR algorithm that uses QoE model B as the optimization objective in actor-critic reinforcement learning. Although BBA [167] and TR [294] do not rely on any QoE model in their design or operation, usage of QoE models to evaluate QoE performance of such ABR algorithms is common as well.

5.2. Methodology

The nine subsequent sections identify and examine problems in dealing with QoE by progressively covering the tasks of test conducting, model building, and model using. In the process, we cite additional problem-specific related work and, when needed, utilize the aforementioned QoE models and ABR algorithms. The analysis in each of these sections offers advice on addressing the examined problem. Because our analyses heavily leverage two large real datasets of QoE perception by individual raters on the 1-100 scoring scale, we now describe these Waterloo-IV and iQoE datasets in more detail.

Waterloo-IV [278] is a dataset with 43,650 individual scores from lab experiments with 92 raters aged between 18 and 38 years old, with 29, 32, and 31 of the raters using phone, HDTV, and ultra HDTV devices, respectively. The presented experiences span all combinations of two codecs, nine network traces, and five ABR algorithms. 13 IFs characterize every chunk, which has the duration of 4 s. Each experience consists of seven chunks, i.e., the playback of an experience without stalls takes 28 s.

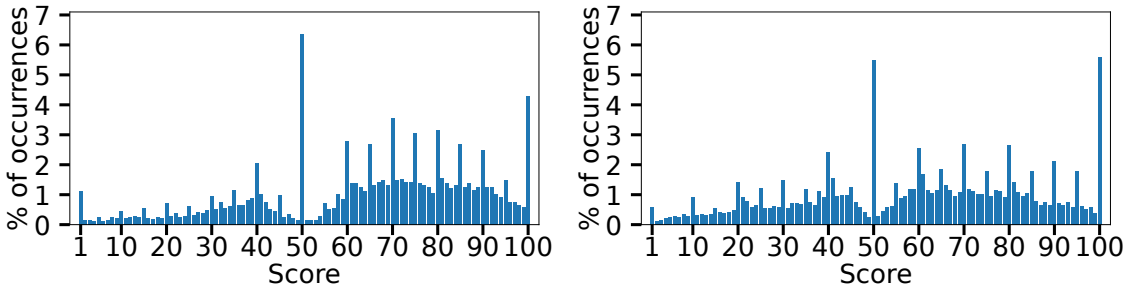
iQoE [298] refers to a dataset with 14,400 individual scores from online subjective tests with 120 raters aged between 20 and 63 years old. Among the raters who disclose their viewing device, six and 110 raters claim using a phone and personal computer, respectively. The iQoE dataset contains 1,000 experiences generated via simulations in Park [293] by utilizing one codec, 102 network traces, and three ABR algorithms. Each experience contains four chunks characterized by 10 IFs. Because the chunk duration is set to 2 s, each experience plays back for 8 s without stalls.

5.3. Scoring Scale in Subjective Assessments

Conducting a subjective test involves selecting a scale for scoring of experiences. The Waterloo-IV and iQoE datasets described in Section 5.2 employ the 1-100 scale. Compared to the five-level ACR scale, the 1-100 scale gives the raters an opportunity to express their QoE perception with a finer granularity, which might make the QoE assessment more accurate. On the other hand, the 100 levels increase the raters' uncertainty about which specific level to choose, and the increased cognitive load on the raters might degrade the assessment accuracy due to hasty or careless decisions.

To analyze how the number of scale levels affects QoE rating, we consider the distributions of individual scores in the Waterloo-IV and iQoE datasets. Because the experiences chosen by the datasets deliberately cover the entire 1-100 scale to support construction of accurate QoE models, we expect the popularity of the individual scores across the scale to be smooth if not uniform. With the uniform distribution, the popularity of each score would be 1%.

For the 32 HDTV raters of the Waterloo-IV dataset, Figure 5.1a depicts the



(a) Waterloo-IV HDTV (b) iQoE

Figure 5.1: Distributions of the individual scores in the datasets.

distribution of the score popularity that is neither uniform nor close to being smooth. Instead, there is a small number of scores that spike in popularity compared to the adjacent scores. In particular, the score of 50 dominates by grabbing 6.37% of all score occurrences and apparently drawing attention to itself at the expense of the other scores in the 41-59 range. The next nine popular scores, in decreasing order of popularity, are 100, 70, 80, 75, 60, 65, 85, 90, and 40 that capture 4.30%, 3.55%, 3.14%, 3.07%, 2.77%, 2.70%, 2.69%, 2.49%, and 2.05% of all score occurrences, respectively. The results suggest that, in agreement with prototype theory [380], the raters form their own new categories of scores where the prototype of each category is either a score divisible by five or the lowest score of 1. When presented with an experience, a rater determines a matching new category and reports the category prototype as the score for the experience.

Figure 5.1b plots the 14,400 individual scores in the iQoE dataset. Despite conducting the tests in online rather than lab settings, the qualitative results are remarkably consistent with those for Waterloo-IV. A small number of prototype scores gain disproportional attention, spiking high above the nearby scores. The scores of 50 and 100 stand out again by attracting 5.49% and 5.58% of all score occurrences, respectively. The other five scores exceeding the popularity threshold of 2% are 70, 80, 60, 40, and 90, with them getting 2.67%, 2.66%, 2.57%, 2.42%, and 2.10% of all score occurrences, respectively. The next four scores in order of decreasing popularity are 65, 85, 95, and 75. Similarly to the findings for Waterloo-IV, scores divisible by five emerge as the prototypes of the score categories newly formed by the raters.

Our analyses indicate that 100 levels are clearly excessive for subjective assessment of QoE in ABR streaming, at least by the factor of five given the popularity of scores divisible by five. Raters' responses in the iQoE post-assessment survey support this sentiment. Hence, we align our recommendation on the scoring scale with the perspective in [377, 378] that a small number of levels, e.g., five in the ACR scale, are sufficient for efficient accurate characterization of QoE:

(Test conducting) Use a scoring scale with a small number of levels, such as the five-level ACR scale.

5.4. Interface Design for Subjective Assessments

The prominence gained by score 50 in the Waterloo-IV and iQoE datasets deserves a separate discussion. 50 is by far the most popular score in comparison to all other intermediate scores on the 1-100 scale in Figure 5.1. In the iQoE dataset, this outcome might arise partly due to score 50 constituting, as Figure 14b in [2] shows, the initial position of the handle on the slider in each iQoE assessment. Because keeping the handle in the initial position before submitting the score of 50 is effortless, and the effort to change the score by dragging the handle from 50 to 51, 60, or 61 is about the same and not negligible, it seems logical that the popularity difference between scores 50 and 51 compared to scores 60 and 61 is significantly larger. Although another likely contributor to the dominance of intermediate score 50 is its central role on the 1-100 scale as the middle point in the ternary QoE perspective between the extreme scores of 1 and 100, our previous observation highlights the importance of designing an unbiased interface for subjective tests, e.g., by randomizing the initial position of the handle on the slider in different assessments:

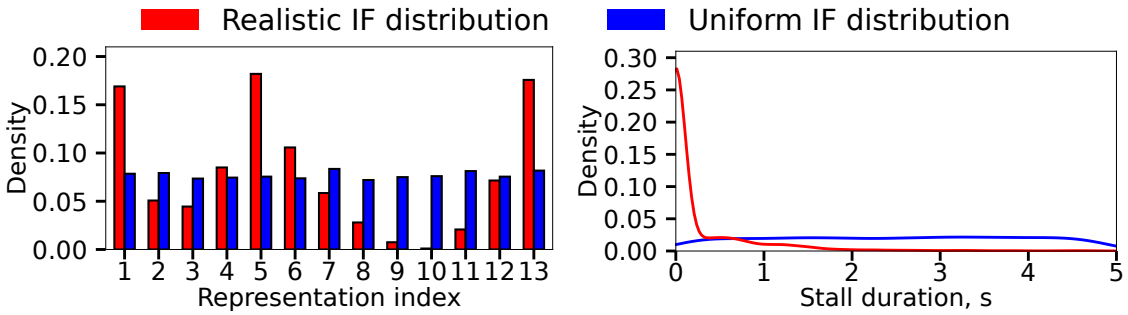
(Test conducting) Design an unbiased interface for subjective assessments, e.g., a randomized initial position of the slider handle.

5.5. Experience Selection for Subjective Tests

The outcome of subjective tests depends significantly on the experiences presented to the raters and, in particular, on the IF values of these experiences. Hence, the choice of the tested experiences and their IF values is an important task. However, the following three circumstances complicate the task. First, multiple IFs characterize an experience. Second, an IF might have many potential values, e.g., stall duration spread between 0 and 5 s. Third, a rater is capable of evaluating only a relatively short series of experiences.

Despite the importance, the selection of experiences routinely lacks in sufficient care. Specifically, it is common to select values for an IF across the experience series in a simplistic manner, such as by drawing the values uniformly or randomly from the range of the IF's possible values. While choosing the values for stall duration and frequency in the randomly uniform fashion, [41] employs other ad-hoc rules for video quality. [381] and [382] adopt similar approaches for their IFs of video quality and stalling. [290] restricts stalling to either beginning or middle of experiences.

To examine experience selection, we utilize the iQoE set of the 1,000 experiences generated through simulations on real network traces with three ABR algorithms and a bitrate ladder comprising 13 representations indexed from 1 to 13. Representation 1 has bitrate 235 Kbps and resolution 320×180. The bitrate and resolution in representation 13 are 16,800 Kbps and 3,840×2,160, respectively. Figure 5.2a shows that representations 1,



(a) Representation index

(b) Stall duration

Figure 5.2: Realistic, as per the iQoE dataset, and uniform selection of IF values for tested experiences.

5, and 13 are the most frequent, with each of them taking in this realistic experience set a larger share than 15%. Representations 8 through 11 are the least frequent, with their individual shares in the experience set falling below 3%. The plot contrasts this realistic distribution with the randomly uniform sampling of the representation index between 1 and 13. Figure 5.2a clearly illustrates that the randomly uniform selection of values for the representation index gives unrealistically high attention to unpopular representations 8 through 11 and unrealistically low attention to popular representations 1, 5, and 13.

We change the IF of interest to stall duration and compare its realistic value distribution in the iQoE experience set against the randomly uniform sampling of stall duration between 0 and 5 s. Figure 5.2b plots kernel density estimates for the two alternatives. In the realistic distribution, stall duration is predominantly below 0.5 s and rarely exceeds 2 s. Thus, the uniform selection of values for stall duration substantially exaggerates the real stalling behavior.

The above analysis illustrates that uniform and other simplistic approaches to experience selection are unrealistic, thereby endangering the validity of conducted subjective tests. Thus, we give the following advice:

(Test conducting) Realistically select experiences for subjective tests and, in particular, with respect to the IF values across the tested experiences.

5.6. Validation of QoE Models

Sections 5.3, 5.4, and 5.5 demonstrate that subjective tests require a substantial amount of thoughtfulness in their setup in order to appropriately collect scores needed for constructing a QoE model. On the other hand, a QoE model in ABR streaming is rarely a goal in itself and instead receives an auxiliary role in the design, operation, or evaluation of ABR algorithms. This might be a reason why various QoE modeling efforts

are insufficiently careful. It is not uncommon to propose a QoE model based on abstract considerations without a proper experimental validation.

QoE model B [46] is a prominent example of the validation concern. The model employs a linear approximation function and combines four IFs as a weighted sum with predefined weight values. [46] introduces QoE model B without conducting subjective tests and does not validate its choices of the linear function and specific IFs. A simple experiment only illustrates how three sets of weight values affect the QoE value produced by the model. Because [46] and [51] leverage QoE model B in their respective MPC and Pensieve algorithms, the success of these pioneering QoE-based ABR algorithms heightens attention to this QoE model and inspires numerous attempts to improve it. The improvements by QoE models G [187], R [276], S [183], V [208], and N [261] primarily target the usage of the bitrate as a proxy of video quality in QoE model B. For example, instead of the bitrate, PSNR and VMAF characterize video quality in QoE models R and V, respectively. However, the above extensions of QoE model B neither question nor validate its major underlying assumptions, such as the linearity of its approximation function.

The existence of the prominent family of QoE models that lack a proper validation leads us to dual recommendations which are both obvious and unfortunately relevant:

(Model building) When proposing a QoE model, validate it through subjective tests.

(Model using) Use validated QoE models only.

5.7. Value Interpretability of QoE Models

Lacking validation of a QoE model might have another negative side effect of the model values losing their interpretability. Due to the separation from subjective tests and their scoring scale, the QoE model is likely to yield values that defy interpretation by humans. For example, while Figures 8 through 15 in [51] evaluate different ABR algorithms via three variants of QoE model B, the values produced by the QoE_lin and QoE_hd variants range from -0.5 to 3 and from -1 to 15, respectively, and it remains unclear how these two empirical value ranges relate to the bad, poor, fair, good, and excellent levels of the common scoring scales.

The disconnection of QoE values from their interpretation also undermines their utility for comparison of different ABR algorithms. Although higher values produced by a QoE model typically indicate better quality of experience, the lacking interpretability of QoE values translates into lacking interpretability of their differences, e.g., of whether a QoE increase with a new ABR algorithm is not meaningful due to falling within the just-noticeable difference (JND), i.e., the maximum difference imperceptible by a human [383].

The above problematic example of QoE model B and its variants that produce both negative and positive values calls for a word of caution about reporting only relative

changes in QoE. Consider an ABR algorithm achieving a positive QoE value which lies arbitrarily close to zero. If another ABR algorithm surpasses this QoE value by a small amount, the relative QoE increase might be 100%, 1,000%, or higher even when the absolute increase is within the JND, i.e., meaninglessly small.

Our discussion in this section highlights dangers of segregating a constructed QoE model from subjective, humanly interpretable perception of QoE. Hence, we argue for QoE models that support interpretation of their values. Apart from the advantages for QoE evaluation of ABR algorithms, the value interpretability equips QoE models with other strengths, e.g., their direct applicability as synthetic raters [2]. Besides, we contend that the values produced by the QoE model should be positive numbers so as to facilitate their mathematical treatment, including meaningful relative comparisons. While both desired properties hold for the range of QoE values aligned with the five-level ACR scale, we advise the following:

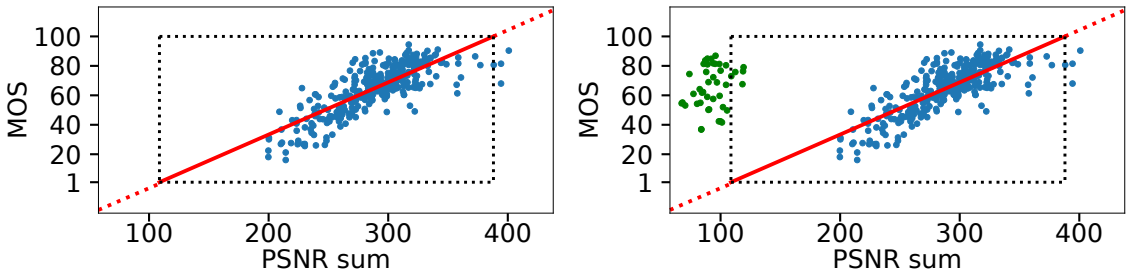
(Model building) Construct a QoE model producing positive interpretable values, e.g., in the range consistent with the five-level ACR scale.

5.8. Capping of the Value Range

Whereas Sections 5.6 and 5.7 expose general problems in the construction of QoE models, we now examine specific technical reasons why these problems arise. Even when a QoE model aspires to align its value range with a humanly interpretable scale, the common reliance on unconstrained regression does not assure such alignment, including on the data used to train the regression. We consider the 450 experiences assessed by the 32 HDTV raters in Waterloo-IV and retain only the experiences devoid of stalling. For ease of exposition, we characterize each of the remaining 326 experiences with a PSNR sum, a new single IF calculated as the sum of the PSNR values across all seven chunks in the experience.

Figure 5.3a presents a scatter plot of the PSNR sum and MOS for the 326 experiences as blue dots. The graph also depicts as a red line a QoE model constructed on this data via linear regression with the least squares fitting. The solid portion of the line represents the QoE values between 1 and 100, i.e., within the 1-100 scoring scale of Waterloo-IV. The respective range of the PSNR sum is from 108 to 388. However, four experiences in the training data have a larger PSNR sum than 388, and the QoE model returns 100.1, 102.3, 102.3, and 104.6 as the QoE values for these four experiences. Thus, due to the reliance on the unconstrained regression, the constructed QoE model produces values beyond the targeted scale even on the training dataset.

A simple way for a regression-based QoE model to address the problem of unconstrained regression is to cap the regression output to an intended range of values. For example, [273], [276], and [384] apply capping to prevent negative values, with [273]



(a) Construction on Waterloo-IV (b) Usage on iQoE data

Figure 5.3: A regression-based QoE model: (a) values beyond the scale and (b) mismatch between usage and construction.

and QoE model R [276] imposing the nonnegative limit on the outputs of Petrangeli model [385] and QoE model B [46], respectively. [386] and [387] align QoE values with the five-level ACR scale by restraining the values to the range from 1.05 to 4.9 for QoE model P and various models from [231, 290, 388, 389], respectively.

An alternative to the output capping is to use an approximation function with built-in adherence to the intended value range. For instance, QoE model L [277] utilizes a hyperbolic tangent function to guarantee values within the range between -1 and 1 and then linearly transforms the guaranteed range to match the five-level ACR scale. [47] configures the exponential function of QoE model F so that the regression always yields values between 1 and 5. [2] creates synthetic raters by adopting a sigmoid function that assuredly produces values between 1 and 100. [390] guarantees QoE values between 0 and 100 by using a sigmoid function as well.

While not advocating a specific method for ensuring that a QoE model produces values within the targeted range, we view such assurances as important for the interpretability of the QoE model and make the following recommendation:

(Model building) Construct a QoE model that assuredly returns values in the intended interpretable range.

5.9. Mismatch between Usage and Construction

Restricting the values produced by a QoE model to an intended range does not ensure their meaningful interpretation. Figure 5.3b enhances Figure 5.3a by adding a scatter plot of the PSNR sum and MOS for the 43 stalling-free iQoE experiences as green dots, where the PSNR sum again refers to the sum of the PSNR values across all chunks in the experience. However, unlike Waterloo-IV with its seven-chunk experiences, the iQoE dataset composes its experiences from four chunks, and the PSNR sum across the 43 stalling-free iQoE experiences varies from 67 to 119. Consequently, the linear QoE model trained on the Waterloo-IV data, i.e., the red line in Figure 5.3b, returns values within the

intended 1-100 range for only four of the 43 experiences. These QoE values are 1.7, 3.8, 4.0, and 4.1, clustering at the bottom of the range. The other 39 experiences receive QoE values smaller than 1 and as low as -14.4. Even with the regression output capped from below by 1, the QoE model characterizes the 43 experiences with values between 1 and 4.1, which is meaninglessly low because the iQoE dataset carefully assembles experiences to cover the entire QoE spectrum from bad to excellent levels.

The observed problem occurs due to the different settings during the usage and construction of the QoE model. While the training Waterloo-IV data contains seven-chunk experiences with the PSNR sum ranging from 200 to 401, the testing iQoE data employs four-chunk experiences with the PSNR sum varying from 67 to 119. Unfortunately, the mismatch between the usage and construction settings is not uncommon. For instance, [273] and [44] use QoE model P in settings that differ from those explicitly assumed in its construction, such as experiences that last less than a minute or contain more than five stalling events.

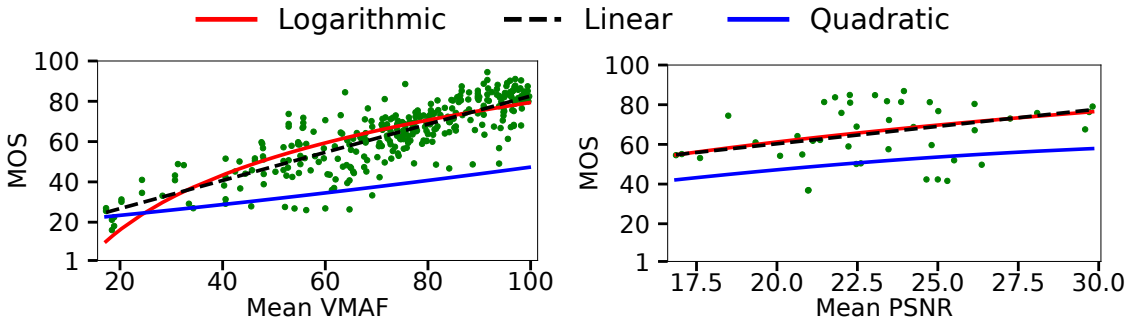
The mismatch problem becomes graver because many QoE models do not describe their construction settings fully, clearly, or at all. For example, QoE models P [379] and L [277] do not publicly release their training modules. Furthermore, [277] trains QoE model L on three datasets and only vaguely describes the roles played by two of them in the training. Similarly, [385] leaves the construction settings of its Petrangeli model unclear by simply referring to [384] and [391]. Thus, even an entity willing to use QoE models appropriately might be unable to do so because the models do not disclose their construction settings.

We suggest addressing the problem by quenching its fundamental source, i.e., by using a QoE model in settings covered during its construction. Although there are alternative heuristics, such as extrapolation or normalization of IF values, these heuristics rely on simplifying assumptions and have ad hoc applicability. The following dual advice promotes the general fundamental solution:

(Model building) Annotate the proposed QoE model with its construction settings. (Model using) Restrict the usage of QoE models to their annotated construction settings.

5.10. Correlation vs. Error

Evaluation of QoE models commonly utilizes metrics of correlation or error. Both kinds of metrics characterize the relationship between the ground-truth subjective scores and values produced by a QoE model. Quantifying the strength and direction of the relationship between these two variables, the correlation metrics include PLCC and SRCC, which deal with the two variables' values and their ranks, respectively. Both PLCC and SRCC vary from -1 (perfect negative relationship) through 0 (no relationship) to 1 (perfect positive relationship). On the other hand, MAE and RMSE are metrics of error



(a) On Waterloo-IV data (b) On iQoE data

Figure 5.4: Three QoE models constructed via logarithmic, linear, and quadratic regressions.

	Logarithmic	Linear	Quadratic
MAE	14.8	15.4	26.9
RMSE	20.5	19.5	30.2
PLCC	0.053	0.105	0.115

(a) On Waterloo-IV data

	Logarithmic	Linear	Quadratic
MAE	13.4	13.3	17.2
RMSE	15.8	15.9	20.7
PLCC	0.108	0.103	0.112

(b) On iQoE data

Table 5.1: MAE, RMSE, and PLCC performance of the three logarithmic, linear, and quadratic QoE models.

in regression problems and measure differences between the subjective scores and values returned by the QoE model. While MAE treats all individual differences equally, RMSE assigns larger weights to larger differences.

Similarly to Section 5.8, we utilize the 326 stalling-free Waterloo-IV experiences assessed by the 32 HDTV raters. This time, the only IF is the mean VMAF computed as the average of the VMAF values across all seven chunks in the experience. We apply the Nelder-Mead method [392] to build three regression-based QoE models that employ logarithmic, linear, and quadratic approximation functions. Specifically and respectively for the logarithmic, linear, and quadratic QoE models, we aim to minimize MAE, minimize RMSE, and maximize PLCC with (0, 0), (0, 0), and (2, 2, 1) as the initial simplex.

Figure 5.4a depicts the three QoE models along with their training Waterloo-IV data. All three models perform identically with respect to SRCC by achieving the same value of 0.184. Table 5.1a reports the MAE, RMSE, and PLCC performance of the three QoE models and, for each of the metrics, highlights in orange the cell with the best performance. The results reveal that each of the logarithmic, linear, and quadratic QoE models outperforms the other two counterparts in regard to MAE, RMSE, and PLCC by providing the best values of 14.8, 19.5, and 0.115, respectively. On the one hand, it is not surprising that the QoE model achieving the best value for a metric is the

model constructed to optimize this metric. On the other hand, it is remarkable that the performance of this QoE model is never the best in regard to the other metrics.

We also conduct a similar analysis for the logarithmic, linear, and quadratic QoE models trained on the 43 stalling-free iQoE experiences from Section 5.9. The only IF is the mean PSNR calculated as the average of the PSNR values across all four chunks in the experience. To build the logarithmic and linear QoE models, we apply the Nelder-Mead method to minimize MAE with (1, 0) as the initial simplex. For the quadratic QoE model, we strive to maximize PLCC with (2, 0, −1) as the initial simplex.

Figure 5.4b plots the training iQoE data and three regression-based QoE models. Again, the SRCC performance is the same across the QoE models, with all three models delivering the identical value of 0.127. In regard to MAE, RMSE, and PLCC, Table 5.1b confirms the qualitative conclusion reached above for Waterloo-IV: while each of the QoE model outperforms its counterparts in one metric, the performance of this QoE model is not the best with respect to the other metrics. Our findings manifest that the advantage of a QoE model in regard to one metric might be misleading and that substantiating the overall goodness of the QoE model necessitates its comprehensive evaluation via multiple metrics.

Although the above analyses on the Waterloo-IV and iQoE data indicate that error and correlation metrics, including their MAE, RMSE, and PLCC varieties, are important due to quantifying different relevant aspects of QoE models, it is not unusual for evaluations to omit some of the metrics. For example, [208], [389], and [393] consider correlation metrics only. While [41], [387], and [230] ignore MAE, [290] excludes RMSE. The evaluation in [277] employs only PLCC and RMSE.

On the question which metrics to use, we call for diversity of perspectives. In spite of existing arguments that error metrics are superior to correlation metrics in their utility for evaluation and understanding of QoE models [39], our position is that metrics of both types are pertinent because of their potential to unveil dissimilar conclusions. For the same reason, we advocate using multiple metrics of the same type, e.g., both MAE and RMSE as error metrics. Hence, our recommendation on metrics is as follows:

(Model building) For diversity of perspectives, evaluate QoE models via metrics of both error and correlation, including MAE, RMSE, and PLCC.

5.11. QoE Evaluation of ABR Algorithms

Moving the evaluation focus from QoE models to QoE achieved by ABR algorithms, we start by analyzing the usage of QoE models for ABR evaluation. We use 945 (i.e., 70%) of all 1,350 experiences in the Waterloo-IV dataset to train QoE models B, G, R, S, V, N, F, and A, i.e., eight parameterized models from Section 5.1. After the training, these QoE models produce values predominantly within the 1-100 range. The training dataset

	B	G	R	S	V	N	F	A	P	L
Pensieve	37.58	59.22	62.95	61.93	54.60	54.60	66.48	58.03	3.13	3.97
MPC	58.91	71.17	66.07	66.56	67.64	67.64	63.81	69.31	3.93	3.00
BBA	58.92	73.92	64.79	64.97	67.71	67.71	66.54	66.92	3.46	4.73
TR	53.62	63.74	67.90	68.74	69.04	69.04	66.17	69.44	3.88	3.32
Increase, %	0.02	3.86	2.77	3.28	1.96	1.96	0.09	0.19	1.29	19.1

Table 5.2: Average QoE performance of ABR algorithms on the Waterloo-IV dataset according to different QoE models.

of 945 experiences includes 189 (i.e., 70%) of the 270 experiences generated with each of Waterloo-IV's five ABR algorithms. To test the achieved QoE, we consider Pensieve, MPC, BBA, and TR as four well-known schemes among these five ABR algorithms. For each of the four tested ABR schemes, we evaluate the average QoE over the remaining 81 (i.e., 30%) of the 270 experiences generated with this ABR scheme. We conduct this QoE evaluation by separately using each of the 10 QoE models from Section 5.1, including models P and L which return values in the 1-5 range. Because QoE models P and L come without public training modules, our evaluation uses these two models in their publicly released configurations without any retraining.

Table 5.2 reports the QoE performance achieved by the four ABR algorithms according to the 10 QoE models. For each QoE model, the table highlights in orange and blue the cell with the best and second-best QoE value, respectively, and shows the relative improvement of the former over the latter in the bottom cell. Table 5.2 shows that TR provides the highest average QoE according to five of the 10 QoE models, with the relative QoE improvement over the second-best ABR algorithm ranging from 0.19% to 3.28%. BBA delivers the highest QoE according to four QoE models. MPC provides the best average QoE according to QoE model P only. Pensieve is never on top and ends up being the second-best ABR algorithm according to two of the 10 QoE models. The findings show that the choice of a QoE model for evaluation of ABR algorithms significantly affects which of the algorithms achieves the highest QoE. [394] tunes various ABR algorithms for four QoE models and reaches the same conclusion that the ability of an ABR algorithm to outperform its counterparts depends on the QoE model selected for QoE evaluation.

Nevertheless, a widespread practice is to evaluate QoE performance of ABR algorithms by using only one QoE model or a small set of similar QoE models. For example, [51] evaluates QoE under Pensieve vs. other ABR algorithms via three similar variants of QoE model B. The QoE evaluation of STALLION [174] employs a version of QoE model B that accounts for latency. [204] compares QoE of its Stick proposal and baseline ABR algorithms by utilizing differently parameterized instances of a single QoE model.

The concerns about using only one QoE model to evaluate QoE performance of an

ABR algorithm get exacerbated when the design or operation of the evaluated ABR algorithm relies on the very same QoE model. Instead of detecting any systematic error introduced into the ABR algorithm by the QoE model, such QoE evaluation espouses and exonerates the bias of this QoE model. Besides, the evaluation gives the ABR algorithm an unfair advantage in comparison with other ABR algorithms that do not employ this QoE model in their design and operation. This bias problem afflicts the evaluations in [46], [208], and [395].

Given the diversity of existing QoE models and the lack of a single, universally accepted QoE model, we argue that QoE evaluation of ABR algorithms should use multiple diverse QoE models. The alternative perspectives offered by multiple QoE models mitigate the biases of individual models and promote comprehensive evaluation of QoE achieved by ABR algorithms. Hence, our recommendation on the usage of QoE models for evaluation of ABR algorithms is as follows:

(Model using) Evaluate ABR algorithms via multiple diverse QoE models.

The shift from QoS to QoE aspires to provide, among other goals, a holistic metric of the user's overall satisfaction. The lack of consensus on the most appropriate QoE model indicates that this aspiration still falls short of its fulfillment. [46], [51], and [178] evaluate QoE performance of ABR algorithms by not only using QoE models but also assessing individual IFs employed by the QoE models. Furthermore, [187], [183], and [177] relinquish QoE models altogether and appraise QoE in the QoS style by evaluating only individual IFs. In this regard, we again follow the spirit of comprehensive evaluation through diversity of perspectives and advise that QoE models and individual IFs meaningfully complement each other in QoE evaluation:

(Model using) To evaluate QoE provided by ABR algorithms, complement usage of QoE models with appraisal of individual IFs.

The constellation of problems that plague objective QoE evaluation of ABR algorithms brings usage of subjective tests back into the spotlight. Despite the larger overhead, direct assessment of QoE via subjective tests is attractive due to its higher accuracy. Although conducting large-scale subjective assessments is not always feasible, we strongly recommend considering this option for QoE evaluation of ABR algorithms:

(Test conducting) Use subjective tests to evaluate QoE achieved by ABR algorithms.

5.12. Conclusions

This chapter reviewed the current landscape of QoE in ABR video streaming. A thorough understanding of QoE is essential for creating effective user empowerment strategies that enhance user satisfaction. By gaining deeper insights, service providers can refine these strategies, encourage active user participation, and seamlessly incorporate user involvement into video streaming systems. Based on two large real datasets of QoE perception by individual raters, we identified and examined various QoE-related pitfalls in

test conducting, model building, and model using. Our analyses also derived the following guidelines for improving the status quo:

- **Test conducting:** We recommended scoring scales with a small number of levels (such as the five-level ACR scale), unbiased interface design (e.g., with a randomized initial position of the slider handle), realistic selection of IF values across the tested experiences, and usage of subjective tests to not only build QoE models but also evaluate QoE performance of ABR algorithms.

- **Model building:** This chapter argued that a proposed QoE model should be validated via subjective tests and annotated with its construction settings, that the QoE model should produce positive interpretable values in the intended range, and that evaluation of the QoE model should utilize metrics of both error and correlation (such as MAE, RMSE, and PLCC).

- **Model using:** We suggested usage of validated QoE models and only in their annotated construction settings, as well as evaluation of ABR algorithms via multiple diverse QoE models and individual IFs.

The chief aspiration of this chapter was to improve awareness of various problems in the current treatment of QoE and to indicate a way forward. We hope that our observations will help to foster high standards in future work on QoE and user empowerment in ABR video streaming.

6 Conclusions

6.1. Summary

This thesis aims to advance technological innovation and address the gap in the literature on user empowerment in adaptive video streaming within the existing best-effort Internet architecture. User empowerment, which enables users to actively improve their streaming experience, is increasingly gaining traction, particularly in content personalization, and has strong potential to enhance users' QoE. However, it remains relatively underexplored in video streaming. Given its capacity to significantly boost QoE—leading to higher revenue and stronger market positions for service providers—further research in this area is both critical and timely. Effective solutions must balance QoE gains for users with minimal required effort, ensuring accessible, privacy-compliant interactions that encourage widespread adoption. For service providers, cost efficiency and seamless integration into current streaming systems are crucial for practical deployment. Additionally, a thorough understanding and use of QoE and QoE models are essential for designing these solutions effectively.

The primary contribution of this thesis is the proposal of two innovative approaches, *iQoE* and *IQN*, which enable active user participation through collaboration with SPs and ISPs, respectively. Chapter 3 introduces iQoE, which allows viewers to create personalized QoE models through brief subjective assessment sessions, where they act as the sole evaluators of their streaming experience. iQoE focuses on atypical viewers—those with QoE perceptions that significantly differ from the median—as they benefit the most from engaging with the SP, though the method is applicable to all users. Integrated as a personalization tool within video streaming systems, iQoE leverages an iterative active learning framework with a novel RIGS sampler and an XSVR modeler. This approach demonstrates a substantial QoE improvement for all users and even more pronounced gains for atypical viewers, requiring a manageable amount of user engagement. In contrast, IQN, presented in Chapter 4, provides an alternative to out-of-band solutions for communicating users' QoE impairments to ISPs. By estimating and signaling QoE

issues through the SP's client interface, IQN modifies packet traffic to create an in-band signal interpretable by ISPs, even in the presence of SP encryption. IQN is intended for voluntary adoption by installing additional software on user devices. The thesis confirms efficacy of IQN through a prototype called YouStall, which alerts ISPs about stalls on YouTube Live. Evaluation results show good detection accuracy, with a MAE of 231 ms and a RMSE of 288 ms whereas the average stall duration exceeds 1.4 seconds across 300 instances.

Both approaches leverage deep and intricate relational and technical interactions among video streaming stakeholders, encompassing concepts, algorithms, and processes unique to the streaming environment. To fully understand and contextualize these contributions, a thorough exploration of the video streaming landscape, including its nuances and state-of-the-art advancements, is essential. For this reason, Chapter 2, preceding the descriptions of iQoE and IQN, provides a foundational overview aimed at setting the stage for the subsequent chapters. It offers insights into promising areas for user empowerment by examining the video streaming landscape and analyzing the pipeline for long-form 2D video delivery over the best-effort Internet, supported by CDNs and client-side ABR algorithms, which represent the prevailing industry approach. This analysis adopts a novel end-to-end perspective and includes a survey of recent state-of-the-art work in the field, organized under a new classification system based on problem-solving methodologies. It integrates comprehensive tutorial material and discussions on real-world applications, current trends, and promising future directions.

Given the profound impact of QoE on the proposed approaches and, more generally, on developing user empowerment strategies, this thesis concludes with an in-depth analysis of QoE-related challenges and recommended solutions. Chapter 5 examines QoE and QoE model applications within adaptive video streaming, identifying common pitfalls and misuses while proposing strategies to address them. These issues arise because QoE is often treated as a "free-for-all", receiving only superficial consideration in its critical role in designing and evaluating video streaming applications, with many core QoE elements handled without sufficient rigor. To address these challenges, the chapter provides guidelines organized by test design, model building, and model usage.

While the primary focus of this thesis is on adaptive streaming, the insights and considerations offered also apply to other video streaming systems.

6.2. Future Directions

All the contributions presented in this thesis provide a strong foundation for future research directions and for addressing the limitations of the current work. Enhancements to both iQoE and IQN could involve refining core components and extending applications across a wider range of platforms, impairments, and user scenarios. For instance, iQoE,

which currently employs a streamlined active learning approach, could benefit from more sophisticated samplers and modelers while remaining true to active learning principles. This might include exploring advanced deep learning architectures that leverage recent software and hardware improvements or experimenting with broader sets of influencing factors. Other learning methods, such as transfer or meta-learning, also show promise for boosting iQoE's performance and responsiveness. For IQN, expanding the capabilities of the YouStall prototype to detect a wider range of QoE impairments, such as video quality degradation, would significantly increase its utility. Further testing on diverse platforms with unique interfaces and protocols (e.g., those based on TCP rather than QUIC) would also ensure greater applicability. Finally, closing the loop by exploring ISP reactions and their impacts would provide valuable insights, allowing for the development of targeted corrective actions and enabling a more comprehensive evaluation of IQN's influence on user experience.

Moreover, the video streaming landscape explored in this thesis is rapidly evolving, with dominant technologies shifting quickly to meet new performance expectations and tackle emerging challenges. In this context, maintaining the relevance of the survey presented in Chapter 2 requires periodic updates.

Furthermore, additional improvements could be achieved by intensifying dedicated research efforts to extend studies beyond the dominant adaptive video streaming. Formats such as video conferencing, 360-degree video, and short-form videos are gaining traction, and the in-depth analysis of QoE pitfalls presented in Chapter 5 would benefit from extending this analysis to these additional formats while enhancing the data pool through the integration of more datasets. This expansion would facilitate a broader understanding of QoE across various streaming modalities and improve the adaptability of the proposed rectifying solutions.

References

[1] L. Peroni and S. Gorinsky, "An End-to-End Pipeline Perspective on Video Streaming in Best-Effort Networks: A Survey and Tutorial," in *arXiv:2403.05192*, 2024.

[2] L. Peroni, S. Gorinsky, F. Tashtarian, and C. Timmerer, "Empowerment of Atypical Viewers via Low-Effort Personalized Modeling of Video Streaming Quality," *PACMNET*, vol. 1, no. CoNEXT3, pp. 1–27, 2023.

[3] L. Peroni, S. Gorinsky, and F. Tashtarian, "In-Band Quality Notification from Users to ISPs," in *CloudNet*, 2024.

[4] L. Peroni and S. Gorinsky, "Quality of Experience in Video Streaming: Status Quo, Pitfalls, and Guidelines," in *COMSNETS*, 2024.

[5] Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017-2022," February 2019, White Paper C11-741490-00. https://branden.biz/wp-content/uploads/2018/12/Cisco-Visual-Networking-Index_Forecast-and-Trends_2017_2022.pdf.

[6] Statista, "Video Streaming (SVoD) - Worldwide," March 2024, Report, https://www.statista.com/outlook/dmo/digital-media/video-on-demand/video-streaming-svod/worldwide.

[7] Leichtman Research Group, "Internet-Delivered TV Services 2024," 2024, Report, https://leichtmanresearch.com/wp-content/uploads/2024/03/LRG-Press-Release-03-20-2024.pdf.

[8] A. Durrani and S. Allen, "Top Streaming Statistics in 2024," August 2024, Forbes, https://www.forbes.com/home-improvement/internet/streaming-stats/.

[9] Cisco, "Cisco Annual Internet Report (2018-2023)," March 2020, White Paper C11-741490-00. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[10] Conviva, "Conviva's State of Streaming Q4 2019," February 2020, Report, https://www.conviva.com/state-of-streaming/convivas-state-of-streaming-q4-2019/.

[11] Conviva, "Conviva's State of Streaming Q2 2021," August 2021, Report, https://www.conviva.com/state-of-streaming/convivas-state-of-streaming-q2-2021/.

[12] A. Fagerjord and L. Kueng, "Mapping the Core Actors and Flows in Streaming Video Services: What Netflix Can Tell Us about These New Media Networks," *JOMBS*, vol. 16, no. 3, pp. 166–181, 2019.

[13] C. Fang, H. Yao, Z. Wang, P. Si, Y. Chen, X. Wang, and F. R. Yu, "Edge Cache-Based ISP-CP Collaboration Scheme for Content Delivery Services," *IEEE Access*, vol. 7, pp. 5277–5284, 2019.

[14] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," *IEEE/ACM ToN*, vol. 13, no. 3, p. 462–475, 2005.

[15] V. Stocker, G. Smaragdakis, and W. Lehr, "The State of Network Neutrality Regulation," *SIGCOMM CCR*, vol. 50, no. 1, p. 45–59, 2020.

[16] H. Habibi Gharakheili, A. Vishwanath, and V. Sivaraman, "Perspectives on Net Neutrality and Internet Fast-Lanes," *SIGCOMM CCR*, vol. 46, no. 1, p. 64–69, 2016.

[17] C. Grece, "Trends in the VOD Market in EU28," 2021, European Audiovisual Observatory, https://www.oficinamediaespana.eu/images/media_europa/TrendsintheVOMarketinEU28pdf.pdf.

[18] E. Carroni and D. Paolini, "Business Models for Streaming Platforms: Content Acquisition, Advertising and Users," *IEP*, vol. 52, pp. 1–13, 2020.

[19] Y. Yang, "Business Model Innovation and User Experience Optimization of New Media Live Streaming Platforms," *IJGEM*, vol. 3, pp. 462–469, 2024.

[20] Netflix, "A Cooperative Approach To Content Delivery. A Netflix Briefing Paper." 2021, https://openconnect.netflix.com/Open-Connect-Briefing-Paper.pdf.

[21] C. Zhou, M. Xiao, and Y. Liu, "ClusTile: Toward Minimizing Bandwidth in 360-degree Video Streaming," in *INFOCOM*, 2018.

[22] S. Loreto and S. P. Romano, *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser*. O'Reilly Media, 2014.

[23] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE COMST*, vol. 17, no. 1, pp. 27–51, 2015.

[24] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *SIGCOMM CCR*, vol. 44, no. 3, pp. 66–73, 2014.

[25] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP," *IEEE COMST*, vol. 21, no. 1, pp. 562–585, 2019.

[26] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, "Quality of Service Terminology in IP Networks," *IEEE Communications Magazine*, vol. 41, no. 3, pp. 153–159, 2003.

[27] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM ToN*, vol. 9, no. 6, pp. 733–745, 2001.

[28] S. Hasan and S. Gorinsky, "Obscure Giants: Detecting the Provider-Free ASes," in *IFIP Networking*, 2012.

[29] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. Vazirani, "How Many Tiers? Pricing in the Internet Transit Market," in *SIGCOMM*, 2011.

[30] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and K. C. Claffy, "Mapping Peering Interconnections to a Facility," in *CoNEXT*, 2015.

[31] I. Castro, J. C. Cardona, S. Gorinsky, and P. Francois, "Remote Peering: More Peering without Internet Flattening," in *CoNEXT*, 2014.

[32] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a Large European IXP," in *SIGCOMM*, 2012.

[33] G. Antichi, I. Castro, M. Chiesa, E. L. Fernandes, R. Lapeyrade, D. Kopp, J. H. Han, M. Bruyere, C. Dietzel, M. Gusat, A. W. Moore, P. Owezarski, S. Uhlig, and M. Canini, "ENDEAVOUR: A Scalable SDN Architecture for Real-World IXPs," *IEEE JSAC*, vol. 35, no. 11, pp. 2553–2562, 2017.

[34] K. Brunnström et al., "Definitions of Quality of Experience," 2013, Qualinet White Paper, https://www.qualinet.eu/wp-content/uploads/2021/04/QoE_whitepaper_v1.2.pdf.

[35] International Telecommunication Union, "Vocabulary for Performance, Quality of Service and Quality of Experience," November 2017, Recommendation P.10. https://www.itu.int/rec/T-REC-P.10-201711-I/en.

[36] ——, "Subjective Video Quality Assessment Methods for Multimedia Applications," 2023, Recommendation P.910, https://www.itu.int/rec/T-REC-P.910-202310-I/en.

[37] Z. Duanmu, A. Rehman, and Z. Wang, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE TBC*, vol. 64, no. 2, pp. 474–487, 2018.

[38] Z. Duanmu, W. Liu, Z. Li, D. Chen, Z. Wang, Y. Wang, and W. Gao, "Assessing the Quality-of-Experience of Adaptive Bitrate Video Streaming," *arXiv*, no. 2008.08804, 2020.

[39] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," *IEEE TIP*, vol. 15, no. 11, pp. 3440–3451, 2006.

[40] International Telecommunication Union, "Methodologies for the Subjective Assessment of the Quality of Television Images," 2023, Recommendation BT.500-15, https://www.itu.int/rec/R-REC-BT.500-15-202305-I/en.

[41] N. Eswara, K. Manasa, A. Kommineni, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "A Continuous QoE Evaluation Framework for Video Streaming Over HTTP," *IEEE TCSVT*, vol. 28, no. 11, pp. 3236–3250, 2018.

[42] U. Reiter et al., "Factors Influencing Quality of Experience," in *Quality of Experience: Advanced Concepts, Applications and Methods*. Springer, 2014.

[43] K. Bouraqia, E. Sabir, M. Sadik, and L. Ladid, "Quality of Experience for Streaming Services: Measurements, Challenges and Insights," *IEEE Access*, vol. 8, pp. 13 341–13 361, 2020.

[44] B. Taraghi, M. Nguyen, H. Amirpour, and C. Timmerer, "INTENSE: In-Depth Studies on Stall Events and Quality Switches and Their Impact on the Quality of Experience in HTTP Adaptive Streaming," *IEEE Access*, vol. 9, pp. 118 087–118 098, 2021.

[45] International Telecommunication Union, "Reference Guide to Quality of Experience Assessment Methodologies," 2016, Recommendation G.1011, https://www.scribd.com/document/487719261/T-REC-G-1011-201607-I-PDF-E.

[46] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *SIGCOMM*, 2015.

[47] T. Hossfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience," in *Data Traffic Monitoring and Analysis*. Springer, 2013.

[48] W. Robitza et al., "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P.1203 – Open Databases and Software," in *MMSys*, 2018.

[49] H. Zhang, L. Dong, G. Gao, H. Hu, Y. Wen, and K. Guan, "DeepQoE: A Multimodal Learning Framework for Video Quality of Experience (QoE) Prediction," *IEEE TMM*, vol. 22, no. 12, pp. 3210–3223, 2020.

[50] S. Periaiya and A. T. Nandukrishna, "What Drives User Stickiness and Satisfaction in OTT Video Streaming Platforms? A Mixed-Method Exploration," *INT J HUM-COMPUT INT*, vol. 40, no. 9, pp. 2326–2342, 2024.

[51] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *SIGCOMM*, 2017.

[52] A. Nikravesh, D. K. Hong, Q. A. Chen, H. V. Madhyastha, and Z. M. Mao, "QoE Inference Without Application Control," in *Internet-QoE*, 2016.

[53] Microsoft, "Rate My Call in Skype for Business Server," Microsoft Learn, https://learn.microsoft.com/en-us/skypeforbusiness/manage/health-and-monitoring/rate-my-call.

[54] Zoom, "Enabling or Disabling Sending Feedback to Zoom," Zoom Support, https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0057689.

[55] Netflix, "How to Rate TV Shows and Movies," Netflix Help, https://help.netflix.com/node/9898.

[56] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender Systems Leveraging Multimedia Content," *ACM CSUR*, vol. 53, no. 5, pp. 1–38, 2020.

[57] X. Amatriain and J. Basilico, "Recommender Systems in Industry: A Netflix Case Study," in *Recommender Systems Handbook*. Springer, 2015.

[58] A. Gnolek and A. Witt, "Twitch State of Engineering 2023," September 2023, Twitch Technology Blog, https://blog.twitch.tv/en/2023/09/28/twitch-state-of-engineering-2023/.

[59] X. Zuo, J. Yang, M. Wang, and Y. Cui, "Adaptive Bitrate with User-Level QoE Preference for Video Streaming," in *INFOCOM*, 2022.

[60] C. Roth and H. Koenitz, "Bandersnatch, Yea or Nay? Reception and User Experience of an Interactive Digital Narrative Video," in *TVX*, 2019.

[61] S. Shalunov, R. Wendy, R. Woundy, S. Previdi, S. Kiesel, R. Alimi, R. Penno, and Y. R. Yang, "Application-Layer Traffic Optimization (ALTO) Protocol," 2014, IETF RFC 7285.

[62] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," in *SIGCOMM*, 2008.

[63] Amazon Web Services, "Amazon Elastic Compute Cloud (Amazon EC2)," 2023, AWS Website, https://aws.amazon.com/ec2/.

[64] Wowza Media Systems, "Video Streaming Latency Report," September 2019, Report, https://www.wowza.com/wp-content/uploads/Streaming-Video-Latency-Report-Interactive-2019.pdf.

[65] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Standards and Design Principle," in *MMSys*, 2011.

[66] International Organization for Standardization, "Information-Technology – Multimedia Application Format (MPEG-A) – Part 19: Common Media Application Format (CMAF) for Segmented Media," 2020, Standard ISO/IEC 23000-19:2020. https://www.mpeg.org/standards/MPEG-A/19/.

[67] J. Kua, G. Armitage, and P. Branch, "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP," *IEEE COMST*, vol. 19, no. 3, pp. 1842–1866, 2017.

[68] Y. Sani, A. Mauthe, and C. Edwards, "Adaptive Bitrate Selection: A Survey," *IEEE COMST*, vol. 19, no. 4, pp. 2985–3014, 2017.

[69] N. Barman and M. G. Martini, "QoE Modeling for HTTP Adaptive Video Streaming – A Survey and Open Challenges," *IEEE Access*, vol. 7, pp. 30 831–30 859, 2019.

[70] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE COMST*, vol. 18, no. 1, pp. 401–418, 2016.

[71] T. Zhao, Q. Liu, and C. W. Chen, "QoE in Video Transmission: A User Experience-Driven Strategy," *IEEE COMST*, vol. 19, no. 1, pp. 285–302, 2017.

[72] A. Barakabitze *et al.*, "QoE Management of Multimedia Streaming Services in Future Networks: A Tutorial and Survey," *IEEE COMST*, vol. 22, no. 1, pp. 526–565, 2019.

[73] S. Afzal, V. Testoni, C. E. Rothenberg, P. Kolan, and I. Bouazizi, "A Holistic Survey of Multipath Wireless Video Streaming," *JNCA*, vol. 212, no. 103581, 2023.

[74] B. Zolfaghari *et al.*, "Content Delivery Networks: State of the Art, Trends, and Future Roadmap," *ACM CSUR*, vol. 53, no. 2, pp. 1–34, 2020.

[75] X. Li, M. Darwich, and M. Bayoumi, "Chapter Four - A Survey on Cloud-Based Video Streaming Services," ser. Advances in Computers. Elsevier, 2021, vol. 123, pp. 193–244.

[76] B. Sredojev, D. Samardzija, and D. Posarac, "WebRTC Technology Overview and Signaling Solution Design and Implementation," in *MIPRO*, 2015.

[77] B. Jansen, T. Goodwin, V. Gupta, F. Kuipers, and G. Zussman, "Performance Evaluation of WebRTC-Based Video Conferencing," *SIGMETRICS PER*, vol. 45, no. 3, pp. 56–68, 2018.

[78] X. Zuo, Y. Li, M. Xu, W. T. Ooi, J. Liu, J. Jiang, X. Zhang, K. Zheng, and Y. Cui, "Bandwidth-Efficient Multi-video Prefetching for Short Video Streaming," in *MM*, 2022.

[79] J. He, M. Hu, Y. Zhou, and D. Wu, "LiveClip: Towards Intelligent Mobile Short-Form Video Streaming with Deep Reinforcement Learning," in *NOSSDAV*, 2020.

[80] Z. Li, Y. Xie, R. Netravali, and K. Jamieson, "Dashlet: Taming Swipe Uncertainty for Robust Short Video Streaming," in *NSDI*, 2023.

[81] J. Guo and G. Zhang, "A Video-Quality Driven Strategy in Short Video Streaming," in *MSWiM*, 2021.

[82] Y. Zhou, L. Tian, C. Zhu, X. Jin, and Y. Sun, "Video Coding Optimization for Virtual Reality 360-Degree Source," *IEEE JSTSP*, vol. 14, no. 1, pp. 118–129, 2020.

[83] M. Graf, C. Timmerer, and C. Mueller, "Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP: Design, Implementation, and Evaluation," in *MMSys*, 2017.

[84] M. Hosseini and V. Swaminathan, "Adaptive 360 VR Video Streaming: Divide and Conquer," in *IEEE ISM*, 2016.

[85] J. Pan, S. Paul, and R. Jain, "A Survey of the Research on Future Internet Architectures," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, 2011.

[86] D.-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *CNIS*, vol. 17, no. 1, pp. 1–14, 1989.

[87] T. Achterberg and R. Wunderling, *Mixed Integer Programming: Analyzing 12 Years of Progress.* Springer, 2013.

[88] T. Glad and L. Ljung, *Control Theory.* CRC Press, 2017.

[89] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems.* Morgan and Claypool, 2010.

[90] P. I. Frazier, "A Tutorial on Bayesian Optimization." arXiv:1807.02811, 2018.

[91] R. Bellman, "Dynamic Programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[92] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* MIT Press, 2018.

[93] I. Grondman *et al.*, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE SMCC*, vol. 42, no. 6, pp. 1291–1307, 2012.

[94] E. Alpaydin, *Introduction to Machine Learning.* MIT Press, 2020.

[95] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications," *ACM CSUR*, vol. 51, no. 5, pp. 1–36, 2018.

[96] A. Nikos and G. Lee, *Cloud Computing: Principles, Systems and Applications.* Springer, 2010.

[97] M. A. Joshi, M. S. Raval, Y. H. Dandawate, K. R. Joshi, and S. P. Metkar, *Image and Video Compression: Fundamentals, Techniques, and Applications.* CRC Press, 2014.

[98] A. Mackin, F. Zhang, and D. R. Bull, "A Study of Subjective Video Quality at Various Frame Rates," in *ICIP*, 2015.

[99] H. Dominguez, O. Vergara, V. Sanchez, E. Casas, and K. Rao, "The H.264 Video Coding Standard," *IEEE Potentials*, vol. 33, no. 2, pp. 32–38, 2014.

[100] Bitmovin, "Bitmovin Video Developer," September 2020, Report https://bitmovin.com/video-developer-report#pdf.

[101] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE TCSVT*, vol. 22, no. 12, pp. 1649–1668, 2012.

[102] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders," in *PCS*, 2013.

[103] Y. Chen *et al.*, "An Overview of Core Coding Tools in the AV1 Video Codec," in *PCS*, 2018.

[104] A. V. Katsenou, F. Zhang, M. Afonso, and D. R. Bull, "A Subjective Comparison of AV1 and HEVC for Adaptive Video Streaming," in *ICIP*, 2019.

[105] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE TCSVT*, vol. 17, no. 9, pp. 1103–1120, 2007.

[106] J. Famaey, S. Latré, N. Bouten, W. Van de Meerssche, B. De Vleeschauwer, W. Van Leekwijck, and F. De Turck, "On the Merits of SVC-Based HTTP Adaptive Streaming," in *IM*, 2013.

[107] B. Bross *et al.*, "Overview of the Versatile Video Coding (VVC) Standard and its Applications," *IEEE TCSVT*, vol. 31, no. 10, pp. 3736–3764, 2021.

[108] P. Topiwala, M. Krishnan, and W. Dai, "Performance Comparison of VVC, AV1, and HEVC on 8-Bit and 10-Bit Content," in *Applications of Digital Image Processing XLI*, vol. 10752, article 107520V, 2018.

[109] K. Choi, J. Chen, D. Rusanovskyy, K.-P. Choi, and E. S. Jang, "An Overview of the MPEG-5 Essential Video Coding Standard [Standards in a Nutshell]," *IEEE SPM*, vol. 37, no. 3, pp. 160–167, 2020.

[110] D. Grois, A. Giladi, K. Choi, M. W. Park, Y. Piao, M. Park, and K. P. Choi, "Performance Comparison of Emerging EVC and VVC Video Coding Standards with HEVC and AV1," *SMPTE Motion Imaging Journal*, vol. 130, no. 4, pp. 1–12, 2021.

[111] G. Meardi *et al.*, "MPEG-5 Part 2: Low Complexity Enhancement Video Coding (LCEVC): Overview and Performance Evaluation," in *Applications of Digital Image Processing XLIII*, vol. 11510, article 115101C, 2020.

[112] J.-S. Lee and T. Ebrahimi, "Perceptual Video Compression: A Survey," *IEEE J-STSP*, vol. 6, no. 6, pp. 684–697, 2012.

[113] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu, "Region-of-Interest-Based Rate Control Scheme for High-Efficiency Video Coding," in *ICASSP*, 2014.

[114] A. K. Katsaggelos, R. Molina, and J. Mateos, *Super Resolution of Images and Video*. Springer, 2007.

[115] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep Learning for Image Super-Resolution: A Survey," *IEEE TPAMI*, vol. 43, no. 10, pp. 3365–3387, 2021.

[116] C. Lottermann, S. Gül, D. Schroeder, and E. Steinbach, "Network-Aware Video Level Encoding for Uplink Adaptive HTTP Streaming," in *ICC*, 2015.

[117] S. Wilk, R. Zimmermann, and W. Effelsberg, "Leveraging Transitions for the Upload of User-Generated Mobile Video," in *MoVid*, 2016.

[118] M. H. Park, J. Choi, and J. K. Choi, "A Network-Aware Encoding Rate Control Algorithm for Real-Time Up-Streaming Video Services," *IEEE COMML*, vol. 21, no. 7, pp. 1653–1656, 2017.

[119] H. Yeo, H. Lim, J. Kim, Y. Jung, J. Ye, and D. Han, "NeuroScaler: Neural Video Enhancement at Scale," in *SIGCOMM*, 2022.

[120] D. Ray, J. Kosaian, K. V. Rashmi, and S. Seshan, "Vantage: Optimizing Video Upload for Time-Shifted Viewing of Social Live Streams," in *SIGCOMM*, 2019.

[121] Y. Chen *et al.*, "Higher Quality Live Streaming Under Lower Uplink Bandwidth: An Approach of Super-Resolution Based Video Coding," in *NOSSDAV*, 2021.

[122] M. Siekkinen, E. Masala, and J. K. Nurminen, "Optimized Upload Strategies for Live Scalable Video Transmission from Mobile Devices," *IEEE TMC*, vol. 16, no. 4, pp. 1059–1072, 2017.

[123] H. Pang *et al.*, "Content Harvest Network: Optimizing First Mile for Crowdsourced Live Streaming," *IEEE TCSVT*, vol. 29, no. 7, pp. 2112–2125, 2019.

[124] J. Chen, B. Balasubramanian, and Z. Huang, "Liv(e)-ing on the Edge: User-Uploaded Live Streams Driven by "First-Mile" Edge Decisions," in *EDGE*, 2019.

[125] K. Du, A. Pervaiz, X. Yuan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-Driven Video Streaming for Deep Learning Inference," in *SIGCOMM*, 2020.

[126] J. Kim, Y. Jung, H. Yeo, J. Ye, and D. Han, "Neural-Enhanced Live Streaming: Improving Live Video Ingest via Online Learning," in *SIGCOMM*, 2020.

[127] S. Zhu and Z. Xu, "Spatiotemporal Visual Saliency Guided Perceptual High Efficiency Video Coding with Neural Network," *Neurocomputing*, vol. 275, pp. 511–522, 2017.

[128] C. Cai, L. Chen, X. Zhang, and Z. Gao, "End-to-End Optimized ROI Image Compression," *IEEE TIP*, vol. 29, pp. 3442–3457, 2020.

[129] Z. Luo, Z. Wang, J. Chen, M. Hu, Y. Zhou, T. Z. J. Fu, and D. Wu, "CrowdSR: Enabling High-Quality Video Ingest in Crowdsourced Livecast via Super-Resolution," in *NOSSDAV*, 2021.

[130] M. Xu, T. Xu, Y. Liu, and F. X. Lin, "Video Analytics with Zero-Streaming Cameras," in *USENIX ATC*, 2021.

[131] H. Le, L. Zhang, A. Said, G. Sautiere, Y. Yang, P. Shrestha, F. Yin, R. Pourreza, and A. Wiggers, "Mobilecodec: Neural Inter-Frame Video Compression on Mobile Devices," in *MMSys*, 2022.

[132] A. S. Kaplanyan *et al.*, "DeepFovea: Neural Reconstruction for Foveated Rendering and Video Compression Using Learned Statistics of Natural Videos," *ACM TOG*, vol. 38, no. 6, pp. 1–13, 2019.

[133] Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. H. Xu, and R. Netravali, "Reducto: On-Camera Filtering for Resource-Efficient Real-Time Video Analytics," in *SIGCOMM*, 2020.

[134] H. Li, Y. Cheng, Z. Zhang, Q. Zhang, A. Arapin, N. Feamster, and A. Mazumdar, "Optimizing Real-Time Video Experience with Data Scalable Codec," in *EMS*, 2023.

[135] Q. Huynh-Thu and M. Ghanbari, "Scope of Validity of PSNR in Image/Video Quality Assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.

[136] Z. Wang, A. C. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE TIP*, vol. 13, pp. 600–612, 2004.

[137] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *CACM*, vol. 60, no. 6, pp. 84–90, 2017.

[138] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video Transcoding: An Overview of Various Techniques and Research Issues," *IEEE TMM*, vol. 7, no. 5, pp. 793–804, 2005.

[139] T. Zhang, F. Ren, W. Cheng, X. Luo, R. Shu, and X. Liu, "Towards Influence of Chunk Size Variation on Video Streaming in Wireless Networks," *IEEE MC*, vol. 19, no. 7, pp. 1715–1730, 2020.

[140] H. Yuan, C. Guo, J. Liu, X. Wang, and S. Kwong, "Motion-Homogeneous-Based Fast Transcoding Method from H.264/AVC to HEVC," *IEEE TMM*, vol. 19, no. 7, pp. 1416–1430, 2017.

[141] R. A. Shah, M. N. Asghar, S. Abdullah, M. Fleury, and N. Gohar, "Effectiveness of Crypto-Transcoding for H.264/AVC and HEVC Video Bit-Streams," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 21 455–21 484, 2019.

[142] K. Park and M. Kim, "EVSO: Environment-Aware Video Streaming Optimization of Power Consumption," in *INFOCOM*, 2019.

[143] A. Erfanian, H. Amirpour, F. Tashtarian, C. Timmerer, and H. Hellwagner, "LwTE: Light-Weight Transcoding at the Edge," *IEEE Access*, vol. 9, pp. 112 276–112 289, 2021.

[144] F. Tashtarian, A. Bentaleb, H. Amirpour, S. Gorinsky, J. Jiang, H. Hellwagner, and C. Timmerer, "ARTEMIS: Adaptive Bitrate Ladder Optimization for Live Video Streaming," in *NSDI*, 2024.

[145] D. K. Krishnappa, M. Zink, and R. K. Sitaraman, "Optimizing the Video Transcoding Workflow in Content Delivery Networks," in *MMSys*, 2015.

[146] C. Chen, Y.-C. Lin, S. Benting, and A. Kokaram, "Optimized Transcoding for Large Scale Adaptive Streaming Using Playback Statistics," in *ICIP*, 2018.

[147] D. Lee, J. Lee, and M. Song, "Video Quality Adaptation for Limiting Transcoding Energy Consumption in Video Servers," *IEEE Access*, vol. 7, pp. 126 253–126 264, 2019.

[148] L. Costero, A. Iranfar, M. Zapater, F. D. Igual, K. Olcoz, and D. Atienza, "MAMUT: Multi-Agent Reinforcement Learning for Efficient Real-Time Multi-User Video Transcoding," in *DATE*, 2019.

[149] T. Huang, R.-X. Zhang, and L. Sun, "Deep Reinforced Bitrate Ladders for Adaptive Video Streaming," in *NOSSDAV*, 2021.

[150] T. L. A. Bubolz *et al.*, "Quality and Energy-Aware HEVC Transrating Based on Machine Learning," *IEEE TCSI*, vol. 66, no. 6, pp. 2124–2136, 2019.

[151] M. Grellert, T. Oliveira, C. R. Duarte, and L. A. da Silva Cruz, "Fast HEVC Transrating Using Random Forests," in *VCIP*, 2018.

[152] P. Agrawal *et al.*, "FastTTPS: Fast Approach for Video Transcoding Time Prediction and Scheduling for HTTP Adaptive Streaming Videos," *Cluster Computing*, vol. 24, no. 3, pp. 1605–1621, 2021.

[153] D. García-Lucas *et al.*, "Cost-Efficient HEVC-Based Quadtree Splitting (HEQUS) for VVC Video Transcoding," *Signal Processing: Image Communication*, vol. 94, article 116199, 2021.

[154] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, and R. Zimmermann, "Common Media Client Data (CMCD): Initial Findings," in *NOSSDAV*, 2021.

[155] T.-Y. Huang, C. Ekanadham, A. J. Berglund, and Z. Li, "Hindsight: Evaluate Video Bitrate Adaptation at Scale," in *MMSys*, 2019.

[156] M. Pathan and R. Buyya, *A Taxonomy of CDNs*. Springer, 2008.

[157] B. M. Maggs and R. K. Sitaraman, "Algorithmic Nuggets in Content Delivery," *SIGCOMM CCR*, vol. 45, no. 3, pp. 52–66, 2015.

[158] J. Dilley, B. M. Maggs, J. Parikh, H. Prokop, R. K. Sitaraman, and B. Weihl, "Globally Distributed Content Delivery," *IEEE IC*, vol. 6, no. 5, pp. 50–58, 2002.

[159] C. Wang, A. Jayaseelan, and H. Kim, "Comparing Cloud Content Delivery Networks for Adaptive Video Streaming," in *CLOUD*, 2018.

[160] C. T. Association, "CTA-5006: Web Application Video Ecosystem – Common Media Server Data," November 2022, CTA Specification, https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5006-final.pdf.

[161] K. Bilal and A. Erbad, "Edge Computing for Interactive Media and Video Streaming," in *FMEC*, 2017.

[162] L. Skorin-Kapov and M. Varela, "A Multi-Dimensional View of QoE: The ARCU Model," in *MIPRO*, 2012.

[163] T. Hoßfeld and C. Keimel, "Crowdsourcing in QoE Evaluation," in *Quality of Experience: Advanced Concepts, Applications and Methods*, ser. TLABS, 2014, pp. 315–327.

[164] International Telecommunication Union, "Mean Opinion Score Interpretation and Reporting," July 2016, Recommendation P.800.2. https://www.itu.int/rec/T-REC-P.800.2.

[165] A. Raake *et al.*, "IP-Based Mobile and Fixed Network Audiovisual Media Services," *IEEE SPM*, vol. 29, no. 6, pp. 163–163, 2012.

[166] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a Practical Perceptual Video Quality Metric," June 2016, Netflix Technology Blog, https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

[167] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *SIGCOMM*, 2014.

[168] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment Aware Rate Adaptation Algorithm for Dynamic Adaptive Streaming over HTTP," in *ICCW*, 2015.

[169] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, "ABMA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming," in *MMSys*, 2016.

[170] K. Dong, J. He, and W. Song, "QoE-Aware Adaptive Bitrate Video Streaming over Mobile Networks with Caching Proxy," in *ICNC*, 2015.

[171] K. Miller, A.-K. Al-Tamimi, and A. Wolisz, "QoE-Based Low-Delay Live Streaming Using Throughput Predictions," *ACM TOMM*, vol. 13, no. 1, article 4, 2016.

[172] A. H. Zahran, D. Raca, and C. J. Sreenan, "ARBITER+: Adaptive Rate-Based Intelligent HTTP Streaming Algorithm for Mobile Networks," *IEEE TMC*, vol. 17, no. 12, pp. 2716–2728, 2018.

[173] R. Song, X. Zeng, X. Wang, and R. Han, "PREPARE – Playback Rate and Priority Adaptive Bitrate Selection," *IEEE Access*, vol. 7, pp. 135 352–135 362, 2019.

[174] C. Gutterman, B. Fridman, T. Gilliland, Y. Hu, and G. Zussman, "STALLION: Video Adaptation Algorithm for Low-Latency Video Streaming," in *MMSys*, 2020.

[175] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming with FESTIVE," *IEEE/ACM ToN*, vol. 22, no. 1, pp. 326–340, 2014.

[176] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming at Scale," *IEEE JSAC*, vol. 32, no. 4, pp. 719–733, 2014.

[177] C. Wang, A. Rizk, and M. Zink, "SQUAD: A Spectrum-Based Quality Adaptation for Dynamic Adaptive Streaming over HTTP," in *MMSys*, 2016.

[178] Z. Akhtar, S. Rao, B. Ribeiro, Y. S. Nam, J. Chen, J. Zhan, R. Govindan, E. Katz-Bassett, and H. Zhang, "Oboe: Auto-Tuning Video ABR Algorithms to Network Conditions," in *SIGCOMM*, 2018.

[179] T. Kimura, T. Kimura, A. Matsumoto, and K. Yamagishi, "Balancing Quality of Experience and Traffic Volume in Adaptive Bitrate Streaming," *IEEE Access*, vol. 9, pp. 15 530–15 547, 2021.

[180] A. H. Zahran *et al.*, "OSCAR: An Optimized Stall-Cautious Adaptive Bitrate Streaming Algorithm for Mobile Networks," in *MoVid*, 2016.

[181] G. Gao *et al.*, "Optimizing Quality of Experience for Adaptive Bitrate Streaming via Viewer Interest Inference," *IEEE TMM*, vol. 20, no. 12, pp. 3399–3413, 2018.

[182] Y. Li, S. Wang, X. Zhang, C. Zhou, and S. Ma, "High Efficiency Live Video Streaming with Frame Dropping," in *ICIP*, 2020.

[183] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in Situ: A Randomized Experiment in Video Streaming," in *NSDI*, 2020.

[184] L. Sun, T. Zong, S. Wang, Y. Liu, and Y. Wang, "Towards Optimal Low-Latency Live Video Streaming," *IEEE/ACM ToN*, vol. 29, no. 5, pp. 2327–2338, 2021.

[185] Y. Qin *et al.*, "A Control Theoretic Approach to ABR Video Streaming: A Fresh Look at PID-Based Rate Adaptation," in *INFOCOM*, 2017.

[186] Y. Qin, S. Hao, K. R. Pattipati, F. Qian, S. Sen, B. Wang, and C. Yue, "Quality-Aware Strategies for Optimizing ABR Video Streaming QoE and Reducing Data Usage," in *MMSys*, 2019.

[187] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," *IEEE/ACM ToN*, vol. 28, no. 4, pp. 1698–1711, 2020.

[188] C. Qiao, J. Wang, and Y. Liu, "Beyond QoE: Diversity Adaptation in Video Streaming at the Edge," *IEEE/ACM ToN*, vol. 29, no. 1, pp. 289–302, 2021.

[189] L. De Cicco, G. Cilli, and S. Mascolo, "ERUDITE: A Deep Neural Network for Optimal Tuning of Adaptive Video Streaming Controllers," in *MMSys*, 2019.

[190] T. Kimura, T. Kimura, and K. Yamagishi, "Context-Aware Adaptive Bitrate Streaming System," in *ICC*, 2021.

[191] P. K. Yadav, A. Shafiei, and W. T. Ooi, "QUETRA: A Queuing Theory Approach to DASH Rate Adaptation," in *MM*, 2017.

[192] S. Hu *et al.*, "Affective Content-Aware Adaptation Scheme on QoE Optimization of Adaptive Streaming over HTTP," *ACM TOMM*, vol. 15, article 100, pp. 1–18, 2019.

[193] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," in *PV*, 2013.

[194] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, "Neural Adaptive Content-Aware Internet Video Delivery," in *OSDI*, 2018.

[195] Y. Zhang *et al.*, "Improving Quality of Experience by Adaptive Video Streaming with Super-Resolution," in *INFOCOM*, 2020.

[196] Y. Liu, B. Jiang, T. Guo, R. K. Sitaraman, D. Towsley, and X. Wang, "Grad: Learning for Overhead-Aware Adaptive Video Streaming with Scalable Video Coding," in *MM*, 2020.

[197] S. Altamimi and S. Shirmohammadi, "QoE-Fair DASH Video Streaming Using Server-Side Reinforcement Learning," *ACM TOMM*, vol. 16, article 68, pp. 1–21, 2020.

[198] J. Yin, Y. Xu, H. Chen, Y. Zhang, S. Appleby, and Z. Ma, "ANT: Learning Accurate Network Throughput for Better Adaptive Video Streaming." arXiv:2104.12507, 2021.

[199] Y. Xu, X. Li, Y. Yang, Z. Lin, L. Wang, and W. Li, "FedABR: A Personalized Federated Reinforcement Learning Approach for Adaptive Video Streaming," in *IFIP Networking*, 2023.

[200] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, and R. Zimmermann, "Meta Reinforcement Learning for Rate Adaptation," in *INFOCOM*, 2023.

[201] L. Meng, F. Zhang, L. Bo, H. Lu, J. Qin, and J. Han, "Fastconv: Fast Learning Based Adaptive Bitrate Algorithm for Video Streaming," in *GLOBECOM 2019*, 2019, pp. 1–6.

[202] L. Huo *et al.*, "A Meta-Learning Framework for Learning Multi-User Preferences in QoE Optimization of DASH," *IEEE TCSVT*, vol. 30, no. 9, pp. 3210–3225, 2020.

[203] T. Feng, H. Sun, Q. Qi, J. Wang, and J. Liao, "Vabis: Video Adaptation Bitrate System for Time-Critical Live Streaming," *IEEE TMM*, vol. 22, no. 11, pp. 2963–2976, 2020.

[204] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, X. Yao, and L. Sun, "Stick: A Harmonious Fusion of Buffer-Based and Learning-Based Approach for Adaptive Streaming," in *INFOCOM*, 2020.

[205] T. Huang, X. Yao, C. Wu, R.-X. Zhang, Z. Pang, and L. Sun, "Tiyuntsong: A Self-Play Reinforcement Learning Approach for ABR Video Streaming," in *ICME*, 2019.

[206] Z. Meng *et al.*, "PiTree: Practical Implementation of ABR Algorithms Using Decision Trees," in *MM*, 2019.

[207] M. Grüner, M. Licciardello, and A. Singla, "Reconstructing Proprietary Video Streaming Algorithms," in *USENIX ATC*, 2020.

[208] T. Huang *et al.*, "Quality-Aware Neural Adaptive Video Streaming with Lifelong Imitation Learning," *IEEE JSAC*, vol. 38, no. 10, pp. 2324–2342, 2020.

[209] Y. Sani, D. Raca, J. J. Quinlan, and C. J. Sreenan, "SMASH: A Supervised Machine Learning Approach to Adaptive Video Streaming over HTTP," in *QoMEX*, 2020.

[210] B. Xu, H. Chen, and Z. Ma, "Karma: Adaptive Video Streaming via Causal Sequence Modeling," in *MM*, 2023.

[211] M. Dasari, K. Kahatapitiya, S. R. Das, A. Balasubramanian, and D. Samaras, "Swift: Adaptive Video Streaming with Layered Neural Codecs," in *NSDI*, 2022.

[212] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE TPAMI*, vol. 38, no. 2, pp. 295–307, 2015.

[213] L. Sun, M. Ma, W. Hu, H. Pang, and Z. Wang, "Beyond 1 Million Nodes: A Crowdsourced Video Content Delivery Network," *IEEE MultiMedia*, vol. 24, no. 3, pp. 54–63, 2017.

[214] R. Viola *et al.*, "Predictive CDN Selection for Video Delivery Based on LSTM Network Performance Forecasts and Cost-Effective Trade-Offs," *IEEE BC*, vol. 67, no. 1, pp. 145–158, 2021.

[215] M. K. Mukerjee, D. Naylor, J. Jiang, D. Han, S. Seshan, and H. Zhang, "Practical, Real-Time Centralized Control for CDN-Based Live Video Delivery," in *SIGCOMM*, 2015.

[216] D. S. Berger, R. K. Sitaraman, and M. Harchol-Balter, "AdaptSize: Orchestrating the Hot Object Memory Cache in a Content Delivery Network," in *NSDI*, 2017.

[217] K. Bilal, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani, "Collaborative Joint Caching and Transcoding in Mobile Edge Networks," *JNCA*, vol. 136, pp. 86–99, 2019.

[218] E. Ghabashneh and S. Rao, "Exploring the Interplay Between CDN Caching and Video Streaming Performance," in *INFOCOM*, 2020.

[219] A. O. Al-Abbasi, V. Aggarwal, T. Lan, Y. Xiang, M.-R. Ra, and Y.-F. Chen, "FastTrack: Minimizing Stalls for CDN-Based Over-the-Top Video Streaming Systems," *IEEE TCC*, vol. 9, no. 4, pp. 1453–1466, 2021.

[220] A. Zhang *et al.*, "Video Super-Resolution and Caching – An Edge-Assisted Adaptive Video Streaming Solution," *IEEE BC*, vol. 67, no. 4, pp. 799–812, 2021.

[221] W. Shi, Q. Li, C. Wang, G. Shen, W. Li, Y. Wu, and Y. Jiang, "LEAP: Learning-Based Smart Edge with Caching and Prefetching for Adaptive Video Streaming," in *IWQoS*, 2019.

[222] V. Kirilin, A. Sundarrajan, S. Gorinsky, and R. K. Sitaraman, "RL-Cache: Learning-Based Cache Admission for Content Delivery," *IEEE JSAC*, vol. 38, no. 10, pp. 2372–2385, 2020.

[223] Y. Zhu *et al.*, "Measuring Individual Video QoE: A Survey, and Proposal for Future Directions Using Social Media," *ACM TOMM*, vol. 14, no. 2s, article 30, pp. 1–24, 2018.

[224] X. Zhang, Y. Ou, S. Sen, and J. Jiang, "SENSEI: Aligning Video Streaming Quality with Dynamic User Sensitivity," in *NSDI*, 2021.

[225] C. Moldovan and F. Metzger, "Bridging the Gap Between QoE and User Engagement in HTTP Video Streaming," in *ITC*, 2016.

[226] X. Zhang, P. Schmitt, M. Chetty, N. Feamster, and J. Jiang, "Enabling Personalized Video Quality Optimization with VidHoc." arXiv:2211.15959, 2022.

[227] S. Porcu, A. Floris, and L. Atzori, "Towards the Prediction of the Quality of Experience from Facial Expression and Gaze Direction," in *ICIN*, 2019.

[228] F. Laiche, A. B. Letaifa, I. Elloumi, and T. Aguili, "When Machine Learning Algorithms Meet User Engagement Parameters to Predict Video QoE," *Wireless Personal Communications*, vol. 116, no. 3, pp. 2723–2741, 2021.

[229] International Telecommunication Union, "Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport, Amendment 1," 2017, Recommendation P.1203. https://www.itu.int/rec/T-REC-P.1203.

[230] N. Eswara *et al.*, "Streaming Video QoE Modeling and Prediction: A Long Short-Term Memory Approach," *IEEE TCSVT*, vol. 30, no. 3, pp. 661–673, 2020.

[231] C. G. Bampis and A. C. Bovik, "Feature-Based Prediction of Streaming Video QoE: Distortions, Stalling and Memory," *Signal Processing: Image Communication*, vol. 68, pp. 218–228, 2018.

[232] Y. Gao, X. Wei, and L. Zhou, "Personalized QoE Improvement for Networking Video Service," *IEEE JSAC*, vol. 38, no. 10, pp. 2311–2323, 2020.

[233] T. Huang, R.-X. Zhang, C. Wu, and L. Sun, "Optimizing Adaptive Video Streaming with Human Feedback," in *MM*, 2023.

[234] A. Norkin, J. Sole, M. Afonso, K. Swanson, A. Opalach, A. Moorthy, and A. Aaron, "SVT-AV1: Open-Source AV1 Encoder and Decoder," March 2020, Netflix Technology Blog, https://netflixtechblog.com/svt-av1-an-open-source-av1-encoder-and-decoder-ad295d9b5ca2.

[235] L. Guo, A. K. G. Valliammal, R. Tam, C. Pham, A. Opalach, and W. Ni, "Bringing AV1 Streaming to Netflix Members' TVs," November 2021, Netflix Technology Blog, https://netflixtechblog.com/bringing-av1-streaming-to-netflix-members-tvs-b7fc88e42320.

[236] A. Norkin, J. De Cock, A. Mavlankar, and A. Aaron, "More Efficient Mobile Encodes for Netflix Downloads," November 2016, Netflix Technology Blog, https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909.

[237] I. Katsavounidis, "Dynamic Optimizer – A Perceptual Video Encoding Optimization Framework," March 2018, Netflix Technology Blog, https://netflixtechblog.com/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f.

[238] C. G. Bampis, L.-H. Chen, and Z. Li, "For Your Eyes Only: Improving Netflix Video Quality with Neural Networks," November 2022, Netflix Technology Blog, https://netflixtechblog.com/for-your-eyes-only-improving-netflix-video-quality-with-neural-networks-5b8d032da09c.

[239] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN," *SIGCOMM CCR*, vol. 48, no. 1, p. 28–34, 2018.

[240] A. Mavlankar, Z. Li, L. Krasula, and C. Bampis, "All of Netflix's HDR Video Streaming Is Now Dynamically Optimized," November 2023, Netflix Technology Blog, https://netflixtechblog.com/all-of-netflixs-hdr-video-streaming-is-now-dynamically-optimized-e9e0cb15f2ba.

[241] F. Li, J. Chung, and M. Claypool, "Three-year Trends in YouTube Video Content and Encoding," in *SIGMAP*, 2021.

[242] Youtube, "Choose Live Encoder Settings, Bitrates, and Resolutions," 2024, YouTube Help, https://support.google.com/youtube/answer/2853702?hl=en.

[243] J. Ozer, "Which Codecs Does YouTube Use?" August 2021, Streaming Learning Center, https://streaminglearningcenter.com/codecs/which-codecs-does-youtube-use.html.

[244] S. C. Madanapalli, A. Mathai, H. H. Gharakheili, and V. Sivaraman, "ReCLive: Real-Time Classification and QoE Inference of Live Video Streaming Services," in *IWQOS*, 2021.

[245] D. Giordano, S. Traverso, L. Grimaudo, M. Mellia, E. Baralis, A. Tongaonkar, and S. Saha, "YouLighter: An Unsupervised Methodology to Unveil YouTube CDN Changes," in *ITC*, 2015.

[246] A. Langley *et al.*, "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *SIGCOMM*, 2017.

[247] C. Gutterman, K. Guo, S. Arora, T. Gilliland, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman, "Requet: Real-Time QoE Metric Detection for Encrypted YouTube Traffic," *ACM TOMM*, vol. 16, no. 2s, pp. 1–28, 2020.

[248] M. Licciardello, M. Grüner, and A. Singla, "Understanding Video Streaming Algorithms in the Wild," in *PAM*, 2020.

[249] R. Vanam and S. Sethuraman, "Improving Compression Efficiency Using an Encoder-Aware Motion Compensated Temporal Filter," March 2023, Amazon Science Blog, https://www.amazon.science/publications/improving-compression-efficiency-using-an-encoder-aware-motion-compensated-temporal-filter.

[250] Achraf Souk, "Using Multiple Content Delivery Networks for Video Streaming - Part 1," October 2019, AWS Technology Blog, https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-1/.

[251] Dario Fontanel *et al.*, "On the Importance of Spatio-Temporal Learning for Video Quality Assessment," 2023, Amazon Science Blog, https://www.amazon.science/publications/on-the-importance-of-spatio-temporal-learning-for-video-quality-assessment.

[252] Yixu Chen *et al.*, "Subjective and Objective Video Quality Assessment of High Dynamic Range Sports Content," 2023, Amazon Science Blog, https://www.amazon.science/publications/subjective-and-objective-video-quality-assessment-of-high-dynamic-range-sports-content.

[253] Joshua P. Ebenezer *et al.*, "No-Reference Video Quality Assessment Using Space-Time Chips," 2023, Amazon Science Blog, https://www.amazon.science/publications/no-reference-video-quality-assessment-using-space-time-chips.

[254] Twitch, "Broadcast Guidelines," 2024, Twitch Help, https://help.twitch.tv/s/article/broadcast-guidelines?language=en_US.

[255] ——, "Enhanced Broadcasting with Multiple Encodes," 2024, Twitch Help, https://help.twitch.tv/s/article/multiple-encodes?language=en_US.

[256] Akrum Elkhazin *et al.*, "How VP9 Delivers Value for Twitch's Esports Live Streaming," December 2018, Twitch Technology Blog, https://blog.twitch.tv/en/2018/12/19/how-v-p9-delivers-value-for-twitch-s-esports-live-streaming-35db26f6322f/.

[257] Jeff Gong *et al.*, "Live Video Transmuxing/Transcoding: FFmpeg vs TwitchTranscoder, Part I," October 2017, Twitch Technology Blog, https://blog.t

witch.tv/en/2017/10/10/live-video-transmuxing-transcoding-f-fmpeg-vs-twitch-transcoder-part-i-489c1c125f28/.

[258] W.-S. Wung, G.-T. Ting, R.-T. Hsu, C. Hsu, Y.-C. Tsai, C. Wang, Y.-T. Liu, H. Chen, and P. Huang, "Twitch's CDN as an Open Population Ecosystem," in *AINTEC*, 2021.

[259] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An Experimental Evaluation of Rate-Adaptive Video Players over HTTP," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 271–287, 2012.

[260] V. Nathan, V. Sivaraman, R. Addanki, M. Khani, P. Goyal, and M. Alizadeh, "End-to-End Transport for Video QoE Fairness," in *SIGCOMM*, 2019.

[261] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," in *MM*, 2016.

[262] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for Social Good: A University Campus Prototype," in *MM*, 2021.

[263] M. Rudow, F. Y. Yan, A. Kumar, G. Ananthanarayanan, M. Ellis, and K. V. Rashmi, "Tambur: Efficient Loss Recovery for Videoconferencing via Streaming Codes," in *NSDI*, 2023.

[264] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction," in *SIGCOMM*, 2016.

[265] R. Shirey, S. Rao, and S. Sundaram, "Optimizing Quality of Experience for Long-Range UAS Video Streaming," in *IWQOS*, 2021.

[266] D. Hocevar, "A Comparison of Statistical Infrequency and Subjective Judgment as Criteria in the Measurement of Originality," *JPA*, vol. 43, no. 3, pp. 297–299, 1979.

[267] M. Awad and R. Khanna, *Efficient Learning Machines*. Springer, 2015.

[268] H. Zhu, T. Li, C. Wang, W. Jin, S. Murali, M. Xiao, D. Ye, and M. Li, "EyeQoE: A Novel QoE Assessment Model for 360-Degree Videos Using Ocular Behaviors," *ACM IMWUT*, vol. 6, no. 1, pp. 1–26, 2022.

[269] K.-T. Chen, C.-C. Tu, and W.-C. Xiao, "OneClick: A Framework for Measuring Network Quality of Experience," in *INFOCOM*, 2009.

[270] X. Huang, C. Zhou, W. Wu, M. Li, H. Wu, and X. Shen, "Personalized QoE Enhancement for Adaptive Video Streaming: A Digital Twin-Assisted Scheme," in *GLOBECOM*, 2022.

[271] M. Nguyen, E. Çetinkaya, H. Hellwagner, and C. Timmerer, "WISH: User-Centric Bitrate Adaptation for HTTP Adaptive Streaming on Mobile Devices," in *MMSP*, 2011.

[272] Weblabcenter, "Microworkers," 2009, https://www.microworkers.com/.

[273] A. Seufert, F. Wamser, D. Yarish, H. Macdonald, and T. Hoßfeld, "QoE Models in the Wild: Comparing Video QoE Models Using a Crowdsourced Data Set," in *QoMEX*.

[274] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MM*, vol. 18, no. 4, pp. 62–67, 2011.

[275] B. Rainer and C. Timmerer, "Quality of Experience of Web-Based Adaptive HTTP Streaming Clients in Real-World Environments Using Crowdsourcing," in *VideoNext*, 2014.

[276] I. de Fez, R. Belda, and J. C. Guerri, "New Objective QoE Models for Evaluating ABR Algorithms in DASH," *Computer Communications*, vol. 158, pp. 126–140, 2020.

[277] H. T. T. Tran, D. V. Nguyen, N. P. Ngoc, and T. C. Thang, "Overall Quality Prediction for HTTP Adaptive Streaming Using LSTM Network," *IEEE TCSVT*, vol. 31, no. 8, pp. 3212–3226, 2021.

[278] Z. Duanmu, W. Liu, Z. Li, D. Chen, Z. Wang, Y. Wang, and W. Gao, "The Waterloo Streaming Quality-of-Experience Database-IV," 2020, IEEE Dataport, https://dx.doi.org/10.21227/j15a-8r35.

[279] Z. Duanmu, A. Rehman, and Z. Wang, "The Waterloo Streaming Quality-of-Experience Database-III," 2020, IEEE Dataport, https://dx.doi.org/10.21227/xzt6-p944.

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

[281] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. C. Bovik, "Towards Perceptually Optimized Adaptive Video Streaming – A Realistic Quality of Experience Database," *IEEE TIP*, vol. 30, pp. 5182–5197, 2021.

[282] Z. Shang, J. P. Ebenezer, Y. Wu, H. Wei, S. Sethuraman, and A. C. Bovik, "Study of the Subjective and Objective Quality of High Motion Live Streaming Videos," *IEEE TIP*, vol. 31, pp. 1027–1041, 2022.

[283] V. Hosu, F. Hahn, M. Jenadeleh, H. Lin, H. Men, T. Szirányi, S. Li, and D. Saupe, "The Konstanz Natural Video Database," 2017, Dataset, http://database.mmsp-kn.de.

[284] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A Subjective and Objective Study of Stalling Events in Mobile Streaming Videos," *IEEE TCSVT*, vol. 29, no. 1, pp. 183–197, 2019.

[285] S.-M. Choi, S.-K. Ko, and Y.-S. Han, "A Movie Recommendation Algorithm Based on Genre Correlations," *Expert Systems with Applications*, vol. 39, no. 9, p. 8079–8085, 2012.

[286] R. Zhou, S. Khemmarat, and L. Gao, "The Impact of YouTube Recommendation System on Video Views," in *IMC*, 2010.

[287] C. Keighrey, R. Flynn, S. Murray, S. Brennan, and N. Murray, "Comparing User QoE via Physiological and Interaction Measurements of Immersive AR and VR Speech and Language Therapy Applications," in *MM Workshops*, 2017.

[288] R. K. P. Mok, E. W. W. Chan, X. Luo, and R. K. C. Chang, "Inferring the QoE of HTTP Video Streaming from User-Viewing Activities," in *W-MUST*, 2011.

[289] S. Wang, S. Yang, H. Su, C. Zhao, C. Xu, F. Qian, N. Wang, and Z. Xu, "Robust Saliency-Driven Quality Adaptation for Mobile 360-Degree Video Streaming," *IEEE TMC*, vol. 23, no. 2, pp. 1312–1329, 2024.

[290] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A Quality-of-Experience Index for Streaming Video," *IEEE JSTSP*, vol. 11, no. 1, pp. 154–166, 2017.

[291] R. Ul Mustafa, S. Ferlin, C. Esteve Rothenberg, D. Raca, and J. J. Quinlan, "A Supervised Machine Learning Approach for DASH Video QoE Prediction in 5G Networks," in *Q2SWinet*, 2020.

[292] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[293] H. Mao et al., "Park: An Open Platform for Learning-Augmented Computer Systems," in *NeurIPS*, 2019.

[294] K. Spiteri, R. K. Sitaraman, and D. Sparacio, "From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player," *ACM TOMM*, vol. 15, no. 2s, p. 1–29, 2019.

[295] Federal Communications Commission, "Raw Data-Measuring Broadband America," 2020, Dataset, https://www.fcc.gov/oet/mba/raw-data-releases.

[296] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," in *MMSys*, 2013.

[297] Z. Duanmu, W. Liu, Z. Li, D. Chen, Z. Wang, Y. Wang, and W. Gao, "Assessing the Quality-of-Experience of Adaptive Bitrate Video Streaming," *arXiv*, no. 2008.08804, 2020.

[298] L. Peroni, S. Gorinsky, F. Tashtarian, and C. Timmerer, "iQoE Dataset and Code," 2023, GitHub, https://github.com/Leo-rojo/iQoE_Dataset_and_Code.

[299] R. Shraga, G. Katz, Y. Badian, N. Calderon, and A. Gal, "From Limited Annotated Raw Material Data to Quality Production Data: A Case Study in the Milk Industry," in *CIKM*, 2021.

[300] R. Burbidge, J. J. Rowland, and R. D. King, "Active Learning for Regression Based on Query by Committee," in *IDEAL*, 2007.

[301] D. Wu, C.-T. Lin, and J. Huang, "Active Learning for Regression Using Greedy Sampling," *Information Sciences*, vol. 474, pp. 90–105, 2019.

[302] H. Yu and S. Kim, "Passive Sampling for Regression," in *ICDM*, 2010.

[303] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *KDD*, 2016.

[304] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2005.

[305] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE TPAMI*, vol. 20, no. 8, pp. 832–844, 1998.

[306] T. Waheed, I. A. Qazi, Z. Akhtar, and Z. A. Qazi, "Coal Not Diamonds: How Memory Pressure Falters Mobile Video QoE," in *CoNEXT*, 2022.

[307] Z. Xu, D. Yang, J. Tang, Y. Tang, T. Yuan, Y. Wang, and G. Xue, "An Actor-Critic-Based Transfer Learning Framework for Experience-Driven Networking," *IEEE/ACM ToN*, vol. 29, no. 1, pp. 360–371, 2021.

[308] Amazon Web Services, "Amazon Simple Storage Service (Amazon S3)," 2023, AWS Website, https://aws.amazon.com/s3/.

[309] ——, "Amazon Relational Database Service (Amazon RDS)," 2023, AWS Website, https://aws.amazon.com/rds/.

[310] ——, "Amazon Redshift," 2023, AWS Website, https://aws.amazon.com/redshift/.

[311] Geeksforgeeks, "System Design of Youtube – A Complete Architecture," October 2023, Geeksforgeeks Website, https://www.geeksforgeeks.org/system-design-of-youtube-a-complete-architecture/.

[312] ——, "System Design Netflix – A Complete Architecture," October 2023, Geeksforgeeks Website, https://www.geeksforgeeks.org/system-design-netflix-a-complete-architecture/.

[313] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, and J. Basilico, "Deep Learning for Recommender Systems: A Netflix Case Study," *AI Magazine*, vol. 42, no. 3, p. 7–18, 2021.

[314] Akamai, "Content Delivery Network (CDN)," 2023, Akamai Website, https://www.akamai.com/solutions/content-delivery-network.

[315] Netflix Inc., "Edge Authentication and Token-Agnostic Identity Propagation," February 2021, Netflix Technology Blog, https://netflixtechblog.com/edge-authentication-and-token-agnostic-identity-propagation-514e47e0b602.

[316] Twitch, "How Twitch Addresses Scalability and Authentication," March 2019, Twitch Technology Blog, https://blog.twitch.tv/en/2019/03/15/how-twitch-addresses-scalability-and-authentication/.

[317] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-Based Congestion Control: Measuring Bottleneck Bandwidth and Round-Trip Propagation Time," *Queue*, vol. 14, no. 5, pp. 20–53, 2016.

[318] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *ACM OSR*, vol. 42, no. 5, pp. 64–74, 2008.

[319] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Analysis and Design of the Google Congestion Control for Web Real-Time Communication (WebRTC)," in *MMSys*, 2016.

[320] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Favor: Fine-Grained Video Rate Adaptation," in *MMSys*, 2018.

[321] Y. Liu, B. Han, F. Qian, A. Narayanan, and Z.-L. Zhang, "Vues: Practical Mobile Volumetric Video Streaming through Multiview Transcoding," in *MOBICOM*, 2022.

[322] A. Zhang, C. Wang, B. Han, and F. Qian, "YuZu: Neural-Enhanced Volumetric Video Streaming," in *NSDI*, 2022.

[323] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1992.

[324] W. Jiang, P. Ning, Z. Zhang, J. Hu, Z. Ren, and J. Wang, "Practical Bandwidth Allocation for Video QoE Fairness," in *WASA*, 2021.

[325] Y. Yuan, W. Wang, Y. Wang, S. S. Adhatarao, B. Ren, K. Zheng, and X. Fu, "Joint Optimization of QoE and Fairness for Adaptive Video Streaming in Heterogeneous Mobile Environments," *IEEE/ACM ToN*, vol. 32, no. 1, pp. 50–64, 2024.

[326] T. Hoßfeld, R. Schatz, and S. Egger, "SOS: The MOS Is Not Enough!" in *QoMEX*, 2011.

[327] T. Hoßfeld, P. E. Heegaard, M. Varela, and S. Möller, "QoE Beyond the MOS: An In-Depth Look at QoE via Better Metrics and Their Relation to MOS," *Quality and User Experience*, vol. 1, no. 2, pp. 1–23, 2016.

[328] V. Menkovski, G. Exarchakos, and A. Liotta, "Tackling the Sheer Scale of Subjective QoE," in *MobiMedia*, 2012.

[329] M. J. Khokhar, T. Spetebroot, and C. Barakat, "An Online Sampling Approach for Controlled Experimentation and QoE Modeling," in *ICC*, 2018.

[330] H.-S. Chang, C.-F. Hsu, T. Hoßfeld, and K.-T. Chen, "Active Learning for Crowdsourced QoE Modeling," *IEEE TMM*, vol. 20, no. 12, pp. 3337–3352, 2018.

[331] Y. Zhu, A. Hanjalic, and J. A. Redi, "QoE Prediction for Enriched Assessment of Individual Video Viewing Experience," in *MM*, 2016.

[332] Y. Wang, P. Li, L. Jiao, Z. Su, N. Cheng, X. S. Shen, and P. Zhang, "A Data-Driven Architecture for Personalized QoE Management in 5G Wireless Networks," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 102–110, 2017.

[333] Y. Hao, J. Yang, M. Chen, M. S. Hossain, and M. F. Alhamid, "Emotion-Aware Video QoE Assessment Via Transfer Learning," *IEEE MM*, vol. 26, no. 1, pp. 31–40, 2019.

[334] J. Meng, Q. Xu, and Y. C. Hu, "Proactive Energy-Aware Adaptive Video Streaming on Mobile Devices," in *USENIX*, 2021.

[335] J. Zhang, Z.-J. Wang, S. Guo, D. Yang, G. Fang, C. Peng, and M. Guo, "Power Consumption Analysis of Video Streaming in 4G LTE Networks," *Wireless Network*, vol. 24, no. 8, p. 3083–3098, 2018.

[336] J. J. Quinlan, A. H. Zahran, K. K. Ramakrishnan, and C. J. Sreenan, "Delivery of Adaptive Bit Rate Video: Balancing Fairness, Efficiency and Quality," in *LANMAN*, 2015.

[337] A. Zhou, H. Zhang, G. Su, L. Wu, R. Ma, Z. Meng, X. Zhang, X. Xie, H. Ma, and X. Chen, "Learning to Coordinate Video Codec with Transport Protocol for Mobile Video Telephony," in *MOBICOM*, 2019.

[338] M. Apostolaki, A. Singla, and L. Vanbever, "Performance-Driven Internet Path Selection," in *SOSR*, 2021.

[339] Y. Zhao, A. Saeed, M. H. Ammar, and E. W. Zegura, "Unison: Enabling Content Provider/ISP Collaboration Using a vSwitch Abstraction," in *ICNP*, 2019.

[340] C. Munteanu, O. Gasser, I. Poese, G. Smaragdakis, and A. Feldmann, "Enabling Multi-Hop ISP-Hypergiant Collaboration," in *ANRW*, 2023.

[341] S. Xu, S. Sen, and Z. M. Mao, "CSI: Inferring Mobile ABR Video Adaptation Behavior Under HTTPS and QUIC," in *EuroSys*, 2020.

[342] Y. Zhao, H. Wu, L. Chen, S. Liu, G. Cheng, and X. Hu, "Identifying Video Resolution from Encrypted QUIC Streams in Segment-Combined Transmission Scenarios," in *NOSSDAV*, 2024.

[343] M. H. Mazhar and Z. Shafiq, "Real-Time Video Quality of Experience Monitoring for HTTPS and QUIC," in *INFOCOM*, 2018.

[344] M. Shen, J. Zhang, K. Xu, L. Zhu, J. Liu, and X. Du, "DeepQoE: Real-Time Measurement of Video QoE from Encrypted Traffic with Deep Learning," in *IWQoS*, 2020.

[345] S. Wassermann, M. Seufert, P. Casas, L. Gang, and K. Li, "ViCrypt to the Rescue: Real-Time, Machine-Learning-Driven Video-QoE Monitoring for Encrypted Streaming Traffic," *IEEE TNSM*, vol. 17, no. 4, pp. 2007–2023, 2020.

[346] T. Sharma, T. Mangla, A. Gupta, J. Jiang, and N. Feamster, "Estimating WebRTC Video QoE Metrics Without Using Application Headers," in *IMC*, 2023.

[347] Tisa-Selma, A. Bentaleb, and S. Harous, "Inferring Quality of Experience for Adaptive Video Streaming over HTTPS and QUIC," in *IWCMC*, 2020.

[348] Y. Yiakoumis, S. Katti, and N. McKeown, "Neutral Net Neutrality," in *SIGCOMM*, 2016.

[349] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*. Morgan Kaufmann, 2021.

[350] C. Lan, J. Sherry, R. A. Popa, S. Ratnasamy, and Z. Liu, "Embark: Securely Outsourcing Middleboxes to the Cloud," in *NSDI*, 2016.

[351] Google LLC, "YouTube Live," 2024, Youtube Website, https://www.youtube.com/live.

[352] K. K. Ramakrishnan, S. Floyd, and D. L. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," 2001, IETF RFC 3168.

[353] R. Schatz, T. Hoßfeld, and P. Casas, "Passive YouTube QoE Monitoring for ISPs," in *IMIS*, 2012.

[354] V. Arun, M. Alizadeh, and H. Balakrishnan, "Starvation in End-to-End Congestion Control," in *SIGCOMM*, 2022.

[355] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "nDPI: Open-Source High-Speed Deep Packet Inspection," in *IWCMC*, 2014.

[356] J. Hypolite, J. Sonchack, S. Hershkop, N. Dautenhahn, A. DeHon, and J. M. Smith, "DeepMatch: Practical Deep Packet Inspection in the Data Plane Using Network Processors," in *CoNEXT*, 2020.

[357] T. Tran, D. Gageot, C. Neumann, G. Bichot, A. Tlili, K. Boutiba, and A. Ksentini, "On the Benefits and Caveats of Exploiting Quality on Demand Network APIs for Video Streaming," in *NOSSDAV*, 2024.

[358] J. A. Clark, "pillow 10.3.0," Python Sofware Foundation, https://pypi.org/project/pillow/, 2024.

[359] A. Sweigart, "Pyautogui 0.9.54," Python Software Foundation, https://pypi.org/project/PyAutoGUI/, 2024.

[360] KimiNewt, "pyshark 0.6," Python Software Foundation, https://pypi.org/project/pyshark/, 2024.

[361] L. Peroni, S. Gorinsky, and F. Tashtarian, "IQN and YouStall: Code and Experimental Configurations," GitHub, 2024, https://github.com/Leo-rojo/IQN_YouStall_Code_CloudNet_2024.

[362] OpenVPN Inc, "OpenVPN," https://openvpn.net/, 2024.

[363] T. Hombashi, "tcconfig 0.7.0a1," Python Sofware Foundation, https://pypi.org/project/tcconfig/0.7.0a1/, 2024.

[364] J. Liu, D. Lerner, J. Chung, U. Paul, A. Gupta, and E. Belding, "Watching Stars in Pixels: The Interplay of Traffic Shaping and YouTube Streaming QoE over GEO Satellite Networks," in *PAM*, 2024.

[365] Python Software Foundation, "selenium 4.22.0," https://pypi.org/project/seleniu m/, 2024.

[366] G. Rodola, "psutil 6.0.0." Python Software Foundation, https://pypi.org/project/p sutil/, 2024.

[367] M. Podlesny and S. Gorinsky, "Leveraging the Rate-Delay Trade-Off for Service Differentiation in Multi-Provider Networks," *IEEE JSAC*, vol. 29, no. 5, pp. 997–1008, 2011.

[368] S. Xu, E. Petajan, S. Sen, and Z. M. Mao, "What You See Is What You Get: Measure ABR Video Streaming QoE via On-Device Screen Recording," in *NOSSDAV*, 2020.

[369] C. Alvarez and K. Argyraki, "Learning a QoE Metric from Social Media and Gaming Footage," in *HotNets*, 2023.

[370] I. Bartolec, "Performance Estimation of Encrypted Video Streaming in Light of End-User Playback-Related Interactions," in *MMSys*, 2021.

[371] Conviva, "Conviva's State of Streaming Q2 2022," September 2022, Report, https: //www.conviva.com/wp-content/uploads/2022/09/Q2-SoS.pdf.

[372] Sandvine, "The Global Internet Phenomena Report January 2023," January 2023, Report, https://www.sandvine.com/global-internet-phenomena-report-2023.

[373] F. Chen, C. Zhang, F. Wang, and J. Liu, "Crowdsourced Live Streaming Over the Cloud," in *INFOCOM 2015*.

[374] L. Skorin-Kapov, M. Varela, T. Hoßfeld, and K.-T. Chen, "A Survey of Emerging Concepts and Challenges for QoE Management of Multimedia Services," *ACM TOMM*, vol. 14, no. 2s, p. 1–29, 2018.

[375] P. Davis, C. D. Creusere, and J. Kroger, "EEG and the Human Perception of Video Quality: Impact of Channel Selection on Discrimination," in *GlobalSIP*, 2013.

[376] D. Z. Rodríguez, R. L. Rosa, and G. Bressan, "Video Quality Assessment in Video Streaming Services Considering User Preference for Video Content," in *ICCE*, 2014.

[377] Q. Huynh-Thu, M.-N. Garcia, F. Speranza, P. Corriveau, and A. Raake, "Study of Rating Scales for Subjective Quality Assessment of High-Definition Video," *IEEE/ACM ToN*, vol. 57, no. 1, pp. 1–14, 2011.

[378] T. Tominaga, T. Hayashi, J. Okamoto, and A. Takahashi, "Performance Comparisons of Subjective Quality Assessment Methods for Mobile Video," in *QoMEX*, 2010.

[379] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, "A Bitstream-Based, Scalable Video-Quality Model for HTTP Adaptive Streaming: ITU-T P.1203.1," in *QoMEX*, 2017.

[380] E. Rosch, "Principles of Categorization," in *Cognition and Categorization*. Lawrence Erlbaum, 1978.

[381] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. de Veciana, "Video Quality Assessment on Mobile Devices: Subjective, Behavioral and Objective Studies," *IEEE JSAC*, vol. 6, no. 6, pp. 652–671, 2012.

[382] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "Modeling the Time – Varying Subjective Quality of HTTP Video Streams With Rate Adaptations," *IEEE TIP*, vol. 23, no. 5, pp. 2206–2221, 2014.

[383] J. Y. Lin, L. Jin, S. Hu, I. Katsavounidis, Z. Li, A. Aaron, and C.-C. J. Kuo, "Experimental Design and Analysis of JND Test on Coded Image/Video," *SPIE Applications of Digital Image Processing XXXVIII*, vol. 9599, pp. 324–334, 2015.

[384] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. D. Turck, "Design and Optimisation of a (FA)Q-Learning-Based HTTP Adaptive Streaming Client," *Connection Science*, vol. 26, no. 1, pp. 25–43, 2014.

[385] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," *ACM TOMM*, vol. 12, no. 2, pp. 1–24, 2015.

[386] H. Bermúdez-Orozco, J.-M. Martinez-Caro, R. Sanchez-Iborra, J. Arciniegas, and M.-D. Cano, "Live Video-Streaming Evaluation Using the ITU-T P.1203 QoE Model in LTE Networks," *Computer Networks*, vol. 165, 2019.

[387] D. Nguyen, N. Pham Ngoc, and T. C. Thang, "QoE Models for Adaptive Streaming: A Comprehensive Evaluation," *Future Internet*, vol. 14, no. 5, pp. 1–21, 2022.

[388] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and Validating User Experience Model for DASH Video Streaming," *IEEE ToB*, vol. 61, no. 4, pp. 651–665, 2015.

[389] Z. Duanmu, W. Liu, D. Chen, Z. Li, Z. Wang, Y. Wang, and W. Gao, "A Knowledge-Driven Quality-of-Experience Model for Adaptive Streaming Videos," *arXiv*, no. 1911.07944, 2019.

[390] A. V. Ivchenko, P. A. Kononyuk, A. V. Dvorkovich, and L. A. Antiufrieva, "Study on the Assessment of the Quality of Experience of Streaming Video," in *SYNCHROINFO*, 2020.

[391] J. De Vriendt, D. De Vleeschauwer, and D. Robinson, "Model for Estimating QoE of Video Delivered Using HTTP Adaptive Streaming," in *IM*, 2013.

[392] F. Gao and L. Han, "Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters," *Computational Optimization and Applications*, vol. 51, pp. 259–277, 2012.

[393] Z. Duanmu, W. Liu, D. Chen, Z. Li, Z. Wang, Y. Wang, and W. Gao, "A Bayesian Quality-of-Experience Model for Adaptive Streaming Videos," *ACM TOMM*, vol. 18, no. 3s, pp. 1–24, 2023.

[394] Y. Liu and J. Y. B. Lee, "A Unified Framework for Automatic Quality-of-Experience Optimization in Mobile Video Streaming," in *INFOCOM*, 2016.

[395] B. Alt, T. Ballard, R. Steinmetz, H. Koeppl, and A. Rizk, "CBA: Contextual Quality Adaptation for Adaptive Bitrate Video Streaming," in *INFOCOM*, 2019.