Enabling Bandwidth-Efficient and Quality-Oriented Adaptive Streaming for Omnidirectional Videos

Abid Yaqoob

BSc, MSc

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (PhD)

to the

Dublin City University

School of Electronic Engineering

Supervisor:

Professor Gabriel-Miro Muntean

May 2022

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

Student ID: 18210719

Date: 2022/May/09

To my beloved parents

Acknowledgement

I would like to express my heartfelt gratitude to my wonderful parents for their unwavering love, care, and support. I would also want to convey my gratitude to my family for their constant support and devotion. Your prayer for me have kept me so far. Prof Gabriel-Miro Muntean, my vivacious supervisor, deserves special recognition for his unconditional support and confidence in my work. I feel your expertise and experience have really benefited me. Throughout my PhD, your compassion and encouragement were extremely helpful. Gabi, I appreciate you pushing me to perform to my full potential. I would especially like to thank my colleagues and friends including Mohammed Amine Togou, Fabio Silva, Ting Bi, Anderson Augusto Simiscuka for their assistance and support. I would want to thank my friends Inam Ul Ahad, Arif, Lehar Asip Khan, Syed Tajammul Ahmad, and Malek Eljadidi for putting up with me. You have been wonderful spiritual and professional mentors as well as friends. I thank, from the bottom of my heart, those who have supported me throughout my stay at DCU through their genuine friendship and care. I would like to extend a very special thank you to the technical personnel of the School of Electronic Engineering, Dublin City University for their invaluable assistance. Last, but not the least important, I gratefully acknowledge the examination board members Prof. Barry McMullin, Dr. Marissa Condon and Prof. Luigi Atzori for making this happen.

This work was funded by the European Regional Development Fund through the Science Foundation Ireland (SFI) Research Centres Programme under Grant numbers 12/RC/2289_P2 (Insight SFI Research Centre for Data Analytics) and 16/SP/3804 (ENABLE).

List of Publications

Journal Papers

1. **A. Yaqoob**, T. Bi and G. -M. Muntean, "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities," in IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2801-2838, Fourthquarter 2020, doi: 10.1109/COMST.2020.300699.
2. **A. Yaqoob** and G. -M. Muntean, "A Combined Field-of-View Prediction-Assisted Viewport Adaptive Delivery Scheme for 360° Videos," in IEEE Transactions on Broadcasting, vol. 67, no. 3, pp. 746-760, Sept. 2021, doi: 10.1109/TBC.2021.3105022.
3. **A. Yaqoob**, M. A. Togou and G. -M. Muntean, "Dynamic Viewport Selection-Based Prioritized Bitrate Adaptation for Tile-Based 360° Video Streaming," in IEEE Access, vol. 10, pp. 29377-29392, 2022, doi: 10.1109/ACCESS.2022.3157339.

Conference Papers

1. **A. Yaqoob**, T. Bi and G.-M. Muntean, "A DASH-based Efficient Throughput and Buffer Occupancy-based Adaptation Algorithm for Smooth Multimedia Streaming," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, pp. 643-649, doi: 10.1109/IWCMC.2019.876664.
2. **A. Yaqoob**, T. Bi and G. -M. Muntean, "A Priority-aware DASH-based Multi-View Video Streaming Scheme over Multiple Channels," 2020 International Wireless Communications and Mobile Computing (IWCMC), 2020, pp. 297-303, doi: 10.1109/IWCMC48107.2020.9148582.

3. **A. Yaqoob** and G. -M. Muntean, "A Weighted Tile-based Approach for Viewport Adaptive 360° Video Streaming," 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2020, pp. 1-7, doi: 10.1109/BMSB49480.2020.937951.
4. **A. Yaqoob**, C.H. Muntean, G. -M. Muntean, "Flexible Tiles in Adaptive Viewing Window: Enabling Bandwidth-Efficient and Quality-Oriented 360° VR Video Streaming," 2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2022.
5. T. Bi, **A. Yaqoob**, L. Zou and G. -M. Muntean, "A Study of Learning Experience during Olfaction-enhanced Adaptive Rich Media Delivery," 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2020, pp. 1-5, doi: 10.1109/BMSB49480.2020.937961.

Contents

Acknowledgements **i**

List of Publications **iii**

Contents **vii**

List of Acronyms **viii**

List of Tables **xiii**

List of Figures **xv**

Abstract **xx**

1 Introduction **1**
1.1 Research Motivation 1
1.2 Problem Statement 7
1.2.1 Bitrate Adaptation 7
1.2.2 Viewport Prediction 9
1.2.3 Tiles Selection 10
1.3 Contributions and Solutions Overview 11
1.4 Organization of the Thesis 15
1.5 Chapter Summary 15

2 Technical Background **17**
2.1 Video Streaming Evolution 18
2.1.1 HTTP Adaptive Video Streaming 19
2.2 Adaptive 360° Video Streaming 23
2.2.1 Capturing and Stitching 23
2.2.2 Projection Schemes 24
2.2.3 Encoding of 360° Video Content 26
2.2.4 Packaging and Transmission 27
2.2.5 360° Video Rendering and Displaying 29
2.3 Tile-based 360° Video Streaming 30
2.4 360° Video Standards and Technologies 34
2.4.1 Immersive Media Formats 34
2.4.2 Towards 6-Degrees of Freedom+ 36
2.4.3 Immersive Audio and Video Standards 38
2.4.4 QoE Metrics 40
2.5 Video Quality Assessment 40

2.5.1 Subjective Quality Assessment . . . 40
2.5.2 Objective Quality Assessment . . . 42
2.6 Chapter Summary . . . 46

3 Related Works 47
3.1 Adaptive Traditional Video Streaming Solutions . . . 48
3.1.1 Server-side Adaptive Streaming Solutions . . . 48
3.1.2 Client-side Adaptive Streaming Solutions . . . 50
3.2 Adaptive Multi-view Video Streaming Solutions . . . 59
3.2.1 Multipath-based Streaming Solutions . . . 59
3.2.2 Multi-view Video Streaming Solutions . . . 62
3.3 Adaptive 360° Video Streaming Solutions . . . 64
3.3.1 Viewport-Independent Streaming Solutions . . . 64
3.3.2 Viewport-Dependent Streaming Solutions . . . 66
3.3.3 Tile-based Streaming Solutions . . . 69
3.3.4 Viewport Prediction Solutions . . . 93
3.4 Chapter Summary . . . 101

4 Throughput and Buffer Occupancy-based Adaptation 102
4.1 Adaptive Video Streaming: Current Experience . . . 103
4.2 TBOA Streaming System Model . . . 105
4.3 Proposed TBOA Adaptation Algorithm . . . 106
4.4 Performance Evaluation . . . 109
4.4.1 Experimental Setup . . . 110
4.4.2 Experimental Results . . . 114
4.5 Conclusions . . . 129

5 Priority-aware DASH-based Multi-view Adaptation 130
5.1 Introduction . . . 131
5.2 Proposed MVP-DASH Framework . . . 132
5.2.1 System Architecture . . . 132
5.2.2 Problem Formulation . . . 133
5.3 Proposed MVP-DASH Approach . . . 134
5.3.1 Throughput Estimation Scheme . . . 134
5.3.2 Unified Segment Scheduling Strategy . . . 136
5.3.3 Bitrate Adaptation Approach . . . 138
5.3.4 Path Selection Method . . . 140
5.4 Performance Evaluation . . . 140
5.4.1 Single-Path Scenario . . . 141
5.4.2 Multipath Scenario . . . 144
5.5 Chapter Summary . . . 145

6 Fixed Tiling-based Adaptive Solutions for 360° Video Streaming 147
6.1 Introduction . . . 148
6.2 Proposed Fixed Tiling-based Architecture . . . 151
6.2.1 Fixed Tiling-based System Architecture . . . 151
6.2.2 Problem Statement . . . 152
6.3 Fixed Tiling-based Algorithms . . . 154
6.3.1 QoE Metric . . . 154
6.3.2 Fixed Tiling-based Streaming Tiles Selection Algorithms . . . 156

6.3.3 Fixed Tiling-based Prioritized Bitrate Adaptation 165
6.4 Experimental Evaluation 169
6.4.1 Experimental Setup 169
6.4.2 Experimental Results 174
6.4.3 Discussions 189
6.5 Conclusions 190

7 Dynamic Tiling-based Adaptive Solutions for 360° Video Streaming 192
7.1 Introduction 193
7.2 Proposed Dynamic Tiling-based Architecture 195
7.2.1 Dynamic Tiling-based System Architecture 195
7.2.2 Problem Definition 197
7.3 Proposed Dynamic Tiling-based Adaptation Algorithms 199
7.3.1 DFT Tiling Pattern Selection Algorithms 199
7.3.2 DFT Streaming Tiles Selection Algorithm 203
7.3.3 DFT Tiles Bitrate Adaptation Algorithm 206
7.4 Experimental Evaluation 208
7.4.1 Experimental Setup 208
7.4.2 Experimental Results 211
7.4.3 Discussion 219
7.5 Conclusions 220

8 Conclusions and Future Works 221
8.1 Conclusions 221
8.2 Future Works 226

Bibliography 229

List of Acronyms

2D Two-Dimensional.

3D Three-Dimensional.

3DoF 3-Degree of Freedom.

3GPP 3rd Generation Partnership Project.

5G 5th Generation.

6DoF 6-Degrees of Freedom.

A3C Asynchronous Advantage Actor-Critic.

ABC Always Best-Connected.

ABR Adaptive BitRate.

AC ACTOR-CRITIC.

ACR Absolute Category Rating.

AP Access Point.

AR Augmented Reality.

AVC Advanced Video Coding.

BK Bron-Kerbosch.

BSc Base Stations.

CMAF Common Media Application Format.

CMP Cubemap Projection.

CMVR Commercial Module VR.

CNN Convolutional Neural Network.

COTS Commercial-Off-The-Shelf.

CTU Coding Tree Unit.

DASH Dynamic Adaptive Streaming over HTTP.

DBSCAN Density-based Spatial Clustering of Applications with Noise.

DCR Degradation Category Rating.

DiffServ Differentiated Services.

DRL Deep Reinforcement Learning.

DVB Digital Video Broadcasting.

ERP Equirectangular Projection.

ETSI European Telecommunications Standards Institute.

EVS Enhanced Voice Services.

FCFS First Come First Served.

FoV Field of View.

GoP Group of Picture.

GPGU General-Purpose Computing on the Graphics Processing Unit.

GPM Grey Prediction Model.

HAS HTTP Adaptive Streaming.

HD High-Definition.

HDS HTTP Dynamic Streaming.

HEVC High-Efficiency Video Coding.

HLS HTTP Live Streaming.

HMD Head-Mounted Display.

HMM Hidden Markov Model.

HTTP HyperText Transfer Protocol.

IETF Internet Engineering Task Force.

IMEx Immersive Media Experience.

IMG Immersive Media Group.

IntServ Integrated Services.

IP Internet Protocol.

IRD Integrated Receiver/Decoder.

ISOBMFF ISO Base Media File Format.

ITU-T International Telecommunication Union-Telecommunications.

JVET Joint Video Exploration Team.

KAMA Kaufman's Adaptive Moving Average.

KNN K-Nearest-Neighbors.

LFU Least Frequently Used.

LR Linear Regression.

LRU Least Recently Used.

LSTM Long-Short Term Memory.

LTE Long Term Evolution.

MBS Macro-cell Base Station.

MCTS Motion-Constrained Tile Set.

MEC Mobile Edge Computing.

ML Machine Learning.

MLP Multi-Layer Perceptron.

MMT MPEG Media Transport.

MOS Mean Opinion Score.

MPC Model Predictive Control.

MPD Media Presentation Description.

MPEG Moving Picture Expert Group.

MPEG-I MPEG Immersive media.

MPTCP Multipath Transmission Control Protocol.

MSE Mean Square Error.

MSS Microsoft Smooth Streaming.

NATs Network Address Translations.

NDN Named Data Networking.

OMAF Omnidirectional Media Format.

OSMF Open Source Media Framework.

OTT Over-the-Top.

PSNR Peak Signal to Noise Ratio.

QERs Quality Emphasized Regions.

QoE Quality of Experience.

QoS Quality of Service.

QP Quantization Parameter.

RAP Random Access Point.

RL Reinforcement Learning.

RNN Recurrent Neural Network.

ROI Region of Interest.

RR Ridge Regression.

RTCP RTP Control Protocol.

RTP Real-time Transport Protocol.

RTSP Real-time Streaming Protocol.

RTT Round-Trip Time.

SBSs Small Base Stations.

SDOs Standards Developing Organizations.

SHVC Scalable High Efficiency Video Coding.

SRD Spatial Relationship Description.

SVA Streaming Video Alliance.

SVC Scalable Video Coding.

SVR Support Vector Regression.

TBOA Throughput and Buffer Occupancy-based Adaptation.

TCP Transmission Control Protocol.

TDMA Time Division Multiple Access.

TSP Truncated Square Pyramid.

UDP User Datagram Protocol.

VESA Video Electronics Standards Association.

VQEG Video Quality Experts Group.

VR Virtual Reality.

VRIF VR Industry Forum.

VVC Versatile Video Coding.

WLR Weighted Linear Regression.

XML Extended Markup Language.

List of Tables

1.1 Traditional, Multi-view, and 360° Video Streaming Approaches . . . 6

3.1 Summary of Related Surveys . . . 71
3.2 Summary of Fixed Viewport-based Adaptive 360° Video Streaming Solutions . . . 80
3.3 Summary of Marginal Region-based Viewport Based Adaptive 360° Video Streaming Solutions . . . 86
3.4 Summary of Extended Viewport-based Adaptive 360° Video Streaming Solutions . . . 90
3.5 Summary of Viewport Prediction Approaches for 360° videos . . . 98

4.1 Simulation Parameters . . . 109
4.2 Bitrate Ladder [Mbps] of Big Buck Bunny Video Sequence [1] . . . 111
4.3 Comparison of **TBOA** and Five Adaptive Solutions for Three Streaming Clients Using Bitrate Set $\mathcal{L}1$ . . . 123
4.4 Comparison of **TBOA** and Five Adaptive Solutions for Three Streaming Clients Using Bitrate Set $\mathcal{L}2$ . . . 123
4.5 Comparison of **TBOA** and Five Adaptive Solutions for Three Streaming Clients Using Bitrate Set $\mathcal{L}3$ . . . 124
4.6 Comparison of **TBOA** and Five Adaptive Solutions for Five Streaming Clients Using Bitrate Set $\mathcal{L}1$ . . . 126
4.7 Comparison of **TBOA** and Five Adaptive Solutions for Five Streaming Clients Using Bitrate Set $\mathcal{L}2$ . . . 126

4.8 Comparison of **TBOA** and Five Adaptive Solutions for Five Streaming Clients Using Bitrate Set $\mathcal{L}3$ 127

5.1 WiFi and LTE Channels Configurations 143

6.1 Notations Used in this Chapter 152
6.2 Average and Standard Deviations of Video Bitrates for the *Boxing*, *Conan*, *Football*, and *Spotlight* videos [Mbps] 168
6.3 Experimental Settings 169
6.4 Comparison Between Proposed and Existing Tile-based Adaptive 360° Video Streaming Solutions in terms of Prediction Mechanisms, Tiles Selection Decisions, Tiles Selection Scenarios, and Streaming Regions 173
6.5 Streaming Behaviors of **CFOV** and **SFOV** clients in terms of Selecting Different Tiles Selection Scenarios for Four 360° videos Prepared in 4x3, 6x4, and 8x6 Tiling Patterns and 1s, 2s, and 3s Segment Lengths 187

7.1 Tile-based Streaming Solutions for Adaptive 360° Video 194
7.2 Notations Used in This Chapter 196
7.3 Content Characteristics 209
7.4 Average and Standard Deviations of Segment Bitrates [Mbps] for the **Football**, **Performance**, **Spotlight**, **Surfing**, and **VR Interview** Videos 210
7.5 Simulation Settings 210
7.6 Streaming Behavior [%] of **DFT1** Client in terms of Tiling Pattern Selection, Tiles Selection, and Bitrate Adaptation Scenarios 212
7.7 Streaming Behavior [%] of **DFT2** Client in terms of Tiling Pattern Selection, Tiles Selection, and Bitrate Adaptation Scenarios 213

List of Figures

1.1 360° video viewing using a head-mounted display device. The current generation three degrees of freedom (3DoF) viewing allows the user to freely navigate through the scene in yaw, pitch, and roll directions. 3

1.2 From capturing to display: Adaptive streaming architecture of traditional, multi-view, and 360° videos. 6

2.1 Traditional Real-time Streaming Protocol (RTSP) streaming system. 19

2.2 Adaptive streaming over HyperText Transfer Protocol (HTTP). . . . 20

2.3 Moving Picture Expert Group (MPEG)-Dynamic Adaptive Streaming over HTTP (DASH) client-server model. 21

2.4 The hierarchical data model of MPD. 22

2.5 End-to-End 360° video streaming framework. 23

2.6 Different projection patterns of 360° video. 24

2.7 MPD samples for regular and 360° video sequences. 27

2.8 Adaptive streaming of 3x3 tiles of 360° video with three segments, each containing one frame. 28

2.9 The architecture of mobile edge-assisted rendering of 360° video. . . 29

2.10 Server architectures of traditional fixed tiling-based adaptive streaming. 31

2.11 Server architectures of dynamic tiling-based adaptive streaming. . . 31

2.12 Tiles arrangement and quality adaptation for fixed viewport-based adaptive streaming in an equirectangular space, where the yellow window represents the viewport region. 32

2.13 Tiles arrangement and quality adaptation for fixed viewport-based adaptive streaming in an equirectangular space, where the yellow window represents the viewport region. 33
2.14 Tiles arrangement and quality adaptation for extended viewport-based adaptive streaming in an equirectangular space, where the yellow window represents the viewport region. 33
2.15 Region-wise resampling and positioning for the Equirectangular Projection (ERP) (top) and the Cubemap Projection (CMP) (bottom) projections. 35
2.16 VR viewing arrangements in 3DoF, 3DoF, and 6DoF+ environments. 37

3.1 Architecture of server-side adaptive streaming solutions. 49
3.2 Classification of client-side adaptive streaming solutions. 50
3.3 Cubemap projection-based scalable video encoding for 360° videos [2]. 74
3.4 FoV-aware edge caching architecture for tile-based 360° video streaming [3]. 78
3.5 Different tile delivery solutions for viewport adaptive 360° video streaming [4]. 81
3.6 Saliency, motion maps, and viewing position based fixation prediction network in [5]. 99

4.1 Concept of buffer dynamics in adaptive video streaming. 104
4.2 HTTP DASH-based client side bitrate adaptive streaming model. . . 106
4.3 Playback buffer levels in the proposed client. 107
4.4 The wired and wireless network topology for simulation experiments. 111
4.5 Average video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link. 114
4.6 Average number of video quality switches achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link. 116
4.7 Per-segment video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link.117

4.8 Average QoE achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link. 118
4.9 Average video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link. 119
4.10 Average number of video quality switches achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link. 120
4.11 Per-segment video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link. 120
4.12 Average QoE achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link. 122
4.13 Average video bitrate, quality fluctuations, and QoE achieved by three streaming clients of **TBOA** and comparative streaming solutions for three bitrate sets. 124
4.14 Average video bitrate, quality fluctuations, and QoE achieved by five streaming clients of **TBOA** and comparative streaming solutions for three bitrate sets. 128

5.1 The client-server model for multi-view video streaming. 132
5.2 Network topology for multi-view video streaming over the wireless network. 141
5.3 Average video bitrates and number of quality fluctuations in a single-path scenario. 142
5.4 Time spent on playing HD segments in a single-path scenario. 142
5.5 Multipath wireless network topology. 143
5.6 Average video bitrates and number of quality fluctuations in a multi-path scenario. 145
5.7 Time spent on playing different quality segments in a multi-path scenario. 145

6.1 The proposed end-to-end 360° video streaming framework. 151
6.2 Tiles selection cases in CFOV1 for the (i)th segment. The yellow rectangle represents the primary predicted FoV, while the blue rectangle represents the secondary predicted FoV. The potential viewport tiles are represented in light red, while the marginal and background tiles are marked in light green and brown, respectively. 159
6.3 Tiles selection cases in SFOV1 for the (i)th segment. The black rectangle represents the predicted FoV, while the red rectangle represents the actual FoV for the $(i-1)$th segment. The potential viewport tiles are represented in red, whereas the marginal and background tiles are marked in green and light brown, respectively. 164
6.4 The per-frame SI.TI values for Boxing, Conan, Football, and Spotlight videos. 170
6.5 Bandwidth scenarios used in experiments. 171
6.6 Average tiles overlap achieved by **CFOV** and Spherical Walk methods for 1s, 2s, and 3s prediction horizons; results from four videos watched by 48 users. 175
6.7 Average QoE results achieved by nine streaming clients for 48 VR users with QoE weight coefficients set to C1; results from four videos prepared in three tiling patterns and three segment durations. 178
6.8 Average QoE results achieved by nine streaming clients for 48 VR users with QoE weight coefficients set to C2; results from four videos prepared in three tiling patterns and three segment durations. 181
6.9 Average QoE results achieved by nine streaming clients for 48 VR users with QoE weight coefficients set to C3; results from four videos prepared in three tiling patterns and three segment durations. 183
6.10 Average bandwidth Utilization for viewport tiles achieved by 9 streaming clients for 48 VR users; results from four videos prepared in three tiling patterns and three segment durations. 185

7.1 The proposed 360° client-server streaming architecture. 196

7.2 Tiling pattern selection in an equirectangular space for two consecutive segments. 201
7.3 Tiles selection cases in DFT for $\mathcal{T}_z(i)$ tiling pattern of (i)th segment. 204
7.4 Tiles selection cases in DFT for $\mathcal{T}_x(i+1)$ tiling pattern of $(i+1)$th segment. 205
7.5 Bandwidth traces employed in experiments. 208
7.6 Average Tiling Pattern Selection in the ATS method. 213
7.7 Average tiles overlap achieved by **DFT1**, **DFT2**, and Spherical Walk methods for the **Football**, **Performance**, **Spotlight**, **Surfing**, and **VR Interview** videos. These videos were prepared in 4x3, 6x4, and 8x6 tiling patterns, which were watched by 48 VR users. The recorded results are for 1s, 1.5s, and 2s segment durations. 214
7.8 Average QoE achieved by **DFT1**, **DFT2**, UVP, CTF, PBA, and AVR streaming clients for **Football** and **Performance** videos. These videos were prepared in 4x3, 6x4, and 8x6 tiling patterns, which were watched by 48 VR users. The recorded results are for 1s, 1.5s, and 2s segment durations while employing the bandwidth trace 1 and trace 2. 215
7.9 Average QoE achieved by **DFT1**, **DFT2**, UVP, CTF, PBA, and AVR streaming clients for the **Spotlight**, **Surfing**, and **VR Interview** videos. These videos were prepared in 4x3, 6x4, and 8x6 tiling patterns, which were watched by 48 VR users. The recorded results are for 1s, 1.5s, and 2s segment durations while employing the bandwidth trace 1 and trace 2. 217

Abstract

Enabling Bandwidth-Efficient and Quality-Oriented Adaptive Streaming for Omnidirectional Videos

Abid Yaqoob

Recently, omnidirectional (or 360°) video, Virtual Reality (VR), and multi-view videos have become wide-used streaming formats for immersive multimedia applications. The latest trends in terms of the adoption of immersive applications are due to the recent proliferation of smart displays and delivery techniques for personal and enterprise uses, but mostly because they offer a closer-to-real-life viewing experience. Unfortunately, compared to regular monoscopic videos, 360° videos have different requirements related to content preparation, packaging, transmission, specialized viewing equipment, and display characteristics (e.g., brightness, contrast, delay, frame rate, resolution, image quality, etc.). These aspects affect the remote transmission of such a massive amount of content and the related applications require substantial network and computational resources, which are challenging to support with the conventional transmission and rendering infrastructure. Employing state-of-the-art viewing region-based adaptive streaming techniques somehow lowers the delivery and processing requirements by performing selective transmission of certain video frame areas in response to users' viewing information. However, they still could end up with unsatisfactory Quality of Experience (QoE) and feeble bandwidth utilization since the accurate viewing region identification, selection, and extraction mechanisms are highly error-prone in response to highly variable viewing preferences. It is therefore essential to devise advanced streaming solutions that enable bandwidth-efficient and quality-oriented streaming over the existing best-effort networks. This work considers end-to-end video streaming in general but focuses explicitly on designing smart adaptive transmission solutions for 360° video. First, a throughput and buffer occupancy-based HTTP Adaptive Streaming (HAS) solution is proposed to provide improved QoE for both single and multiple clients, primarily when delivered under time-varying network conditions. Secondly, a priority-aware multi-view streaming mechanism is proposed to enable differentiated quality streaming with a focus on ensuring synchronous and high-quality timely playback of the multiple concurrent video streams. Next, this work covers a comprehensive survey on 360° video streaming techniques along with proposing fixed and dynamic tiling-based 360° video streaming frameworks. The proposed 360° video streaming solutions utilize content-dependent and/or content-independent information and are empowered by fixed or flexible tiling versions selection, systematic viewport identification and selection, and intelligent bitrate adaptation mechanisms to achieve improved data transmission and QoE goals. The proposed solutions were evaluated through extensive trace-driven simulation-oriented testing against the closest state-of-the-art works. The in-depth results analysis verifies the success of proposed solutions in terms of achieving improved streaming performance.

Chapter 1

Introduction

1.1 Research Motivation

There has been a considerable increase in the design, production, and use of innovative multimedia applications (e.g., 360° video, Augmented-, Virtual-, Mixed, Hyper-reality (AR/VR/MR/HR), etc.) supported by the recent technological advancements in networking (e.g., 3G, 4G, 4G+, 5G), display devices (e.g., smartphones, gaming consoles, laptops, Head-Mounted Display (HMD), etc.), and users' increasing interest to enrich their experience. A new generation of immersive multimedia applications has been introduced to improve the expressiveness of traditional streaming content by offering users more opportunities to interact with the content. This applies in particular to omnidirectional video, also known as 360° or spherical video, which allows users to switch their viewing angles during video watching. The 360° video user can perceive a sense of "presence there" by freely interacting with the captured content. Fig. 1.1[1] presents a spherical video viewing environment where the viewing position can be adjusted across three directions (i.e., yaw, pitch, and roll). With the increasing adoption of such interactive multimedia applications for personal and/or enterprise use-cases, mobile video traffic is projected to account for about 82% of global cellular traffic by 2022, along with a 12-fold increase in immersive multimedia traffic by 2022 compared to 2017 [6].

[1]The equirectangular image has been used in the thesis and is sourced from GoPro VR-Tahiti Surf video available at https://www.youtube.com/watch?v=MKWWhf8RAV8.

360° videos, already famous on popular video-sharing platforms such as YouTube, Facebook, and Vimeo, exhibit enormous potential for both enterprises and consumers by providing more engaging and unprecedented interaction between viewers and the content [7]. Users' engagement varies by experiencing 360° videos on different devices. For instance, the user changes the viewing position on a PC through the up/down/left/right keys or pan-and-zoom by using the mouse. The user can control the view on a smartphone or tablet by swiping the screen or turning the orientation. VR headsets or HMDs are more popular and key enablers for supporting wide 360° video adoption. Often the overall video scene is larger than the one visible to viewers. The video information displayed on the user device is known as the viewport. It depends on the user's horizontal and vertical viewing directions. These define the viewer's observable scene, which is known as Field of View (FoV)[2] and ranges between 90° and 120° according to the specification of the device. In this way, a user can visualize only a limited portion (e.g., 20%-30%) of the 360° content due to the short visual span of both the display devices and the human eye. Several HMDs are available in the market, e.g., Samsung Gear VR[3], Oculus Rift/Quest[4], HTC VIVE[5], Google Cardboard[6], Daydream[7], and PlayStation VR[8], etc. These HMDs differ in terms of their FoV, display resolution, viewing panels, screen refresh rate, and operating platforms. Moreover, they can be standalone/mobile VR devices that do not require any external components (i.e., Oculus Quest, Samsung Gear VR, and Google Cardboard) or tethered VR devices that require a PC or a PlayStation to operate (i.e., Oculus Rift, HTC VIVE, Razer OSVR). Modern HMDs can manage and support real-time immersive content due to the combination of powerful sensors, good computational support, excellent networking, and advanced display features. VR HMDs are likely to see a broad adoption shortly, as worldwide forecast shipment is expected to be about 100 million HMD units by the end of 2021, of which about 50% are anticipated to be mobile headsets [8].

[2]The terms FoV and viewport are used interchangeably in this thesis.
[3]Samsung Gear VR, http://www.samsung.com/global/galaxy/gear-vr
[4]Oculus Rift/Quest, http://www.oculus.com/
[5]HTC VIVE, https://www.vive.com/
[6]Google Cardboard, https://www.google.com/get/cardboard/
[7]Daydream, https://www.google.com/get/daydream/
[8]PlayStation VR, https://www.playstation.com/en-ca/explore/playstation-vr/

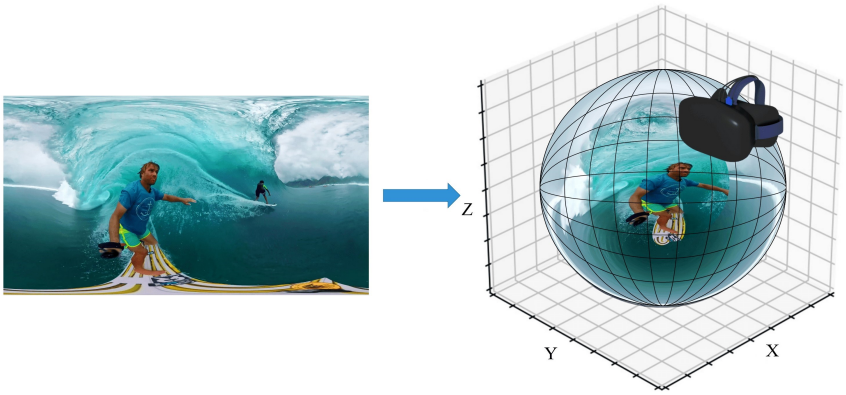


Figure 1.1: 360° video viewing using a head-mounted display device. The current generation three degrees of freedom (3DoF) viewing allows the user to freely navigate through the scene in yaw, pitch, and roll directions.

As traditional video content is being replaced by 360° video to provide more creative and entertaining opportunities, the latter offers immersive possibilities in different domains. 360° news production has evolved from its initial experimental phase to become a more critical part of several news organizations [9]. Many public service media organizations have already adopted 360° cinematic VR, and an increasing amount of new content is being created or consumed. A user can virtually attend 360° events from an attractive rotating seat, including live sports and live concerts; they can watch movies or visit relatives located in far-away locations. Small town classrooms can virtually tour amazing world sights, attend famous science laboratories, enter gigantic theme parks, and visit industrial sites. 360° videos can be used to train physicians, neurosurgeons, and paramedics. Doctors can analyze tumors in rare conditions that can be reconstructed virtually to minimize human error in a natural environment. These advancements will help achieve significant time and cost savings in areas such as training and teaching [10].

Compared to regular 2D videos, 360° videos have different streaming requirements related to content preparation, packaging, transmission, specialized equipment, and display factors (e.g., brightness, contrast, delay, frame rate, resolution, image quality, etc.). These requirements are not only limited to 360° video, but they also apply to some other video types, which include 3D television (3DTV), free-viewpoint video (FVV), and multi-view video (MVV). 3DTV enables a three-dimensional cinematic depth perception using 3D glasses. At the same time, FVV delivers 3D content

with advanced stereoscopic viewing that allows an interactive viewpoint choice to the viewer within a specific operating range. MVV provides an improved visual experience by offering multiple sequences to the end terminals. FVV and MVV share a similar kind of content capturing format by using multi-viewpoint camera systems to record the same scene from multiple positions and angles. The user can freely switch among multiple available views for a single file packaging system. However, this type of system suffers from high view switching delay. Contrary, multiple captured streams can be simultaneously delivered to the end-user, who can experience all the downloaded views in parallel regardless of viewing direction[9] MVVs are extensively used for a wide range of streaming applications (e.g., video conferences, surveillance, telepresence, etc.). The pre-processing, transmission, and display requirements of such videos depend on the number of video streams. The transmission of 360° videos over the current network infrastructure is challenging since it requires up to 6 times more bandwidth than traditional videos due to the larger file sizes of multiple viewing angles [12]. In addition, the entire captured 360° content has to be downloaded in higher resolution ($\geq$8k) to achieve acceptable user-perceived quality. Netflix's recommended connection speed for streaming 4K regular videos is 25 Mbps [13], but only less than 10% of global network connections have bandwidths higher than 25 Mbps [14]. As a result, the transmissions of MVVs and 360° videos over existing networks are affected by delays and loss and consequently suffer from severe video quality degradation.

In order to prevent or reduce the negative effect in terms of video quality, diverse solutions were proposed at different network layers, including using Error Correction Coding [15, 16, 17], Error Concealment [18, 19, 20], Bitrate Adaptation [21, 22, 23, 24], etc. However, the most successful of these are adaptive bitrate solutions, which perform dynamic adjustment of the transmitted video bitrate according to the current network conditions [25, 26, 27]. Among the adaptive solutions, HAS [28, 29] can be employed for streaming traditional as well as multi-view and 360° videos over best-effort networks. Diverse HAS solutions employ different objectives, including

[9]This thesis considers an independent file packaging system for MVVs [11] that transmits priority-oriented multiple concurrent streams to a remote user.

maximization of video quality, optimization of bandwidth utilization, prioritized and/or selective content transmission, etc., in mobile, wired, and wireless network environments.

By leveraging the client-centric HAS approach, the adaptive transmission of 360° video can be classified into three broad categories: (i) viewport-independent streaming; (ii) viewport-dependent streaming; (iii) tile-based streaming. Viewport independent streaming is similar to regular video streaming, where only the bitrate adaptation for the whole segment is performed based on the throughput and/or buffer information. This streaming technique is simple to implement; however, it results in substantial wastage of bandwidth resources [30] due to the higher quality full view data transmission. Viewport-dependent streaming technology allows the transmission of certain frame areas in accordance with the user's viewing orientation. When considering such streaming, first, a certain region is identified as the next potential viewport, and then the bitrate adaptation is performed for the selected region. Such solutions have lower bandwidth requirements; however, they are associated with very high storage, and processing overheads [31]. Tile-based adaptive streaming is an extension of viewport-dependent streaming, where the 360° video is divided into a number of horizontal and vertical rectangular areas, known as tiles [32, 33, 34]. Each tile can be regarded as a separate video clip. These smaller size video clips are then temporally split into several equal-duration segments to facilitate adaptive streaming. Both spatial (tiles selection) and temporal (bitrate selection) adaptation is performed by leveraging network and human viewing behavior information. The client selects the visible and non-visible tiles and their quality levels based on the predicted viewport, and the next segment's available connection speed. The tiles requests are performed in advance to ensure synchronous and timely playback of the content. The adaptive accurate tiles selection and bitrate adjustment mitigate the bandwidth requirements of ever-increasing 360° video services. Table 1.1 illustrates the significant differences between traditional, multi-view, and 360° video streaming approaches. Tile-based streaming requires a low number of content versions on the server-side. It incorporates lower storage and processing overhead compared to viewport-dependent streaming. For MVVs, the storage, processing, and transmis-

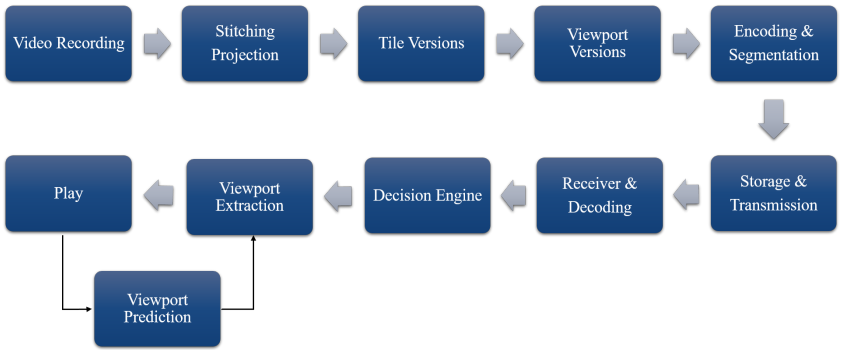


Figure 1.2: From capturing to display: Adaptive streaming architecture of traditional, multi-view, and 360° videos.

Table 1.1: Traditional, Multi-view, and 360° Video Streaming Approaches

	Regular Video Streaming	**MVV**	**360 Video Streaming**		
			Viewport-Independent Streaming	**Viewport-Dependent Streaming**	**Tile-based Streaming**
Adaptation	Bitrate	Bitrate	Bitrate	Bitrate Viewport Region Viewport Size Adaptation	Bitrate Viewport Region Tile-Size Adaptation
Bandwidth Requirements	Low	Medium to High	Ultra High	High	High
Latency Requirements	up to 300 ms	up to 300 ms	up to 300 ms	<20 ms	<20 ms
Projections	N/A	N/A	CMP, ERP	Pyramid, TSP Offset cubemap, Hybrid Multi-resolution ERP/CMP [35]	CMP, ERP, Pyramid TSP, Offset cubemap, Hybrid Multi-resolution ERP/CMP
Cache efficiency	High	Low to Medium	High	Low	Medium
Storage and Processing Overhead	Low	Medium to High	Medium	Very High	High
Viewing Orientation	N/A	N/A	N/A	Head Tracking	Head Tracking

sion requirements are somehow closer to the 360° videos.

The adaptive streaming architecture of traditional, multi-view, and 360° videos is shown in Fig. 1.2. The video recording, encoding, segmentation, storage, transmission, decoding, and playback steps are common in traditional, multi-view, and 360° video streaming. Similarly, the decision engine is the main part of all streaming solutions, as it mainly decides the video bitrates for each adaptation interval. The stitching and projection steps are performed in viewport-independent, viewport-dependent, and tile-based 360° video streaming. The tile versions and viewport versions creation steps are adopted in tile-based and viewport-dependent 360° video streaming, respectively. The viewport prediction and viewport extraction functions are performed in viewport-dependent and tile-based 360° video streaming. Considering this general streaming architecture, this thesis focuses on bandwidth-efficient and quality-oriented HTTP adaptive streaming of traditional, multi-view, and tile-based 360° video streaming.

1.2 Problem Statement

The growing popularity of smart TV sets, support of fast home pervasive broadband network access, and the latest cellular technologies (e.g., LTE/LTE-Advanced, 5/6G) are critical drivers for internet video. The increasing adoption of rich multimedia services requires support for high-quality delivery of large amounts of content data. Currently, efforts are being put to enable high-quality, rich multimedia delivery in existing network environments. One solution adopted by multimedia services providers is to use HAS in order to facilitate the transmission of quality-oriented streaming without sacrificing picture quality. However, the best-effort dynamic internet service leads to uncontrolled data transmission, which affects the video quality for receivers. It is essential to devise solutions that enable bandwidth-efficient streaming over best-effort packet networks. Despite many attempts, it is still challenging to achieve this, and additional solutions need to be proposed to satisfy different user requirements. In this context, this thesis addresses the following problems to ensure a bandwidth-efficient and quality-oriented HAS for both regular and rich multimedia content, such as 360 videos.

1.2.1 Bitrate Adaptation

The usefulness of multimedia services is strongly dependent on how efficiently the adaptive client can manage the available network resources. A critical challenge in HAS is to accurately adjust video bitrates in response to the dynamic network conditions. Single client-based adaptive solutions (e.g., throughput-based/buffer-based) work reasonably well by requesting the maximum allowable video bitrate in response to the measured throughput or buffer status. However, such solutions experience unfair visual quality, frequent rebuffering, and quality fluctuations when employed for multiple streaming clients under different segment durations [36, 37, 38]. Several existing adaptive solutions either increase video quality aggressively to achieve high-quality streaming or use a conservative approach to maintain continuous video playback. However, this is unacceptable because the former approach will result in playback interruptions and wastage of the available network resources, especially for

longer duration segments, while the later will result in poor quality and inefficient bandwidth utilization. The same is also true for MVV streaming, where the existing adaptive solutions assign uniform quality levels to all the streams, irrespective of their priority. The priority of the streams is crucial for multiple concurrent streams, where the user is particularly interested in a specific view to be visualized at higher quality. The unified bandwidth distribution among multiple concurrent streams can lower the user's perception levels for interesting views.

It is challenging to employ conventional bitrate adaptation solutions [39, 38, 40, 41, 42, 43] for tile-based 360° video streaming because they usually involve a large buffer capacity which is not suitable for 360° video streaming. They usually switch the video bitrate without respecting the tiles' spatial distribution and their viewing importance. For a tile-based bitrate adaptation algorithm, it is necessary to consider the priority of different streaming regions to adjust the bitrates accordingly since the segments are prepared in numerous tiles and encoding bitrates. A recent topic of interest for 360° video is to empower differentiated streaming of quality-variable tiles based on users' interests to avoid wasting network resources. A straightforward solution is to stream only the predicted viewport tiles in the highest allowable quality. However, this will lead to frequent viewport deviations or black dots in the viewport for which no content was requested [4, 44]. Another solution is to stream the viewport tiles at a higher quality and the remaining tiles at a lower quality to avoid viewport deviations. But due to the limitations of existing prediction mechanisms, this leads to severe quality degradation, especially for segments over 1s length [45, 46, 47]. Another concern is how to distribute the bitrate budget to different predicted regions of 360° video. The per-tile and per-area bitrate adaptations have different consequences in terms of achieving acceptable quality. For instance, per-tile quality increment starting from the center tile of the viewport can introduce lower quality levels for tiles within the same viewport [48], and per-area quality allocation may request the tiles in remaining regions (e.g., marginal and background tiles) in the lowest quality levels [49]. In addition, the non-uniform bitrate adaptation for the viewport region can lead to severe quality variations within the same viewport even for the content is being watched with slower head movements. Suppose a complex

adaptation technique can correctly calculate the tiles bitrate for the next segment in real-time. Still, it is essential to best match the quality that depends on multiple parameters such as visual quality, non-visual quality, and quality variations in both spatial and temporal domains. Therefore, for all these reasons, there is a need to design efficient bitrate adaptive solutions to provide bandwidth-efficient and quality-oriented HAS for regular, multi-view, and 360° videos.

1.2.2 Viewport Prediction

When a user wearing an HMD moves his/her head, the viewport position changes, and the updated content should be displayed in real-time. Quick and accurate response to viewer head movements substantially impacts positively viewer satisfaction while experiencing the immersive environment. Accurate viewport prediction is particularly important in a VR streaming environment, where an inaccurate viewport delivery can cause motion sickness [50], making the 360° video streaming experience undesirable to the users [51]. Viewport prediction can be considered as a sequence prediction problem, which focuses on predicting the viewport for future horizons based on the recorded head motion trajectories. Long-term accurate FoV prediction for the upcoming segments can support high-quality future media services [52]. Viewport prediction encourages downloading the predicted tiles in advance before the user moves. According to the likelihood of visual visits, the client can assign more bits to the viewport tiles [2] to save the communication and computation resources. In practice, long-term accurate viewport prediction ($\geq$2s) is highly error-prone [52, 53] and adds tremendous pressure on the prediction mechanisms since the accuracy of current prediction mechanisms is limited to 58$\sim$80% for the next 1s prediction horizon [54, 55]. Therefore, a short segment length (0.5s-2s) with a lower buffer level (2 segments) is usually employed for 360° videos. Longer segment duration and larger buffer levels, however, may not be sufficient for 360° video streaming since it becomes harder to predict and therefore fetch longer in the future. Consequently, if video tiles are requested based on a wrong prediction, the user's actual viewport may deviate from the predicted viewport. Therefore, it is indispensable to design an effective prediction mechanism that leads to a higher viewport match

even for $\geq 2s$ video segment length.

1.2.3 Tiles Selection

After predicting the viewport coordinates, the next significant problem is selecting the tiles to best match the viewing region. Tile-based streaming solutions often divide the streaming tiles into viewport, marginal, and background tiles. The tiles covering the visible region are referred to as viewport tiles which are streamed at the higher possible bitrates. To deal with the possible viewport prediction errors and given the fact the user can move his/her head at any time during the segment playback, tile-based delivery solutions add a marginal region around the viewport [45, 56]. The marginal tiles are streamed in equal or lower than viewport bitrates and provide a cushion to absorb the impact of quick head rotations. The remaining out of sight tiles are referred to as background tiles which are streamed in the lowest resolution. The dynamic selection of the viewport and marginal tiles before each segment request influences the overall streaming experience. Therefore, it is fundamental and challenging to truly decide the streaming tiles and relevant bitrates in a dynamic fashion for cost-effective streaming. The viewport tiles should be streamed in uniform quality levels to improve the sense of immersion and spatial presence. Under good prediction accuracy, introducing marginal tiles around the viewport in all directions can significantly lead to bandwidth wastage. It is important to carefully decide how much marginal region to include and towards which direction in order to achieve bandwidth-efficient streaming. Sometimes, adding a marginal region is not enough for dynamic motion content. Tile-based solutions also extend the viewport coverage to provide a wide-angle visible region to the user. However, how much to extend the viewport and under which variables (e.g., network conditions, viewport prediction, etc.) to avoid over- and under- extension for a favorable higher QoE is also challenging. In addition, mapping tiles to the viewport can introduce redundant pixels and lose many compression opportunities [57]. For instance, employing a smaller number of tiles (i.e., larger resolution tiles) can improve the compression performance in some cases. Yet, at the same time, it may include additional pixels that contribute to the waste of bandwidth resources [58]. Contrary, using a large

number of tiles (i.e., smaller resolution tiles) reduces the number of redundant pixels; however, it may introduce several visual distortions, e.g., flickering, floating, and blurring at the edges of the tiles [59]. Employing dynamic tiling to lower the number of redundant pixels along with providing a finer FoV boundary for different watching behaviors of the users is very important, although challenging tasks and need to be investigated.

1.3 Contributions and Solutions Overview

Focusing on achieving bandwidth-efficient and quality-oriented adaptive streaming, this thesis proposes the following novel solutions:

1. Proposal of a quality-aware and bandwidth utilization-concerned adaptive streaming solution for traditional videos: **Throughput and Buffer Occupancy-based Adaptation (TBOA)**. TBOA targets maximization of both viewer quality and bandwidth utilization in given network delivery conditions, while performing graceful rate adjustments and also being fair with competing streams. TBOA is specifically designed to work for both single and multiple streaming clients. The salient features of this work are as follows:
    - TBOA considers the average throughput measurement model to estimate the throughput for future segments in stable and fluctuating network environments. TBOA solution attentively employs both the throughput and buffer occupancy signals to achieve: (i) higher visual quality levels, (ii) lower video quality fluctuations, (iii) higher QoE levels, and (iv) fair, stable, and efficient utilization of the available network resources.
    - TBOA evaluations in single- and multi-client environments under dynamic network conditions and multiple segment durations show that TBOA achieves higher visual quality, QoE, and bandwidth utilization while reducing the video quality fluctuations and playback interruptions in comparison to the state-of-the-art bitrate adaptation algorithms.
2. Proposal of MVV streaming solution: **Multi-View Priority-aware DASH (MVP-DASH)**. MVP-DASH is a novel adaptive MVV streaming solution

that deploys priority-based bandwidth estimation, inter-stream synchronization, and unified bitrate adaptation for multiple concurrent streams delivered over multiple paths. MVP-DASH implements differentiated quality streaming to overcome the streaming challenges discussed in section 1.2.1. MVP-DASH utilizes the best available path for high priority streams while ensuring the segments of low priority streams are also delivered within the same transmission cycle to ensure playback continuity. In this context, MVP-DASH:

- Employs a priority-oriented bandwidth estimation mechanism to distribute the available network capacity to each stream based on relevant priority-related weights. This enables the MVP-DASH client to request high-priority streams with higher video bitrates while maintaining acceptable quality levels for low-priority streams to support the multi-view streaming system.
- Utilizes a unified segment scheduling mechanism to achieve synchronization between multiple streams. Employs a hybrid bitrate adaptation approach that selects suitable video bitrates based on the estimated throughput of each stream. In addition, a path selection mechanism requests higher priority video streams over the best available path and other streams over other paths to exercise differentiated quality streaming.
- Achieves higher average video bitrates and fewer video bitrate switches for high-priority segments in comparison with two alternative approaches.

3. Proposal of viewport prediction-assisted fixed tiling-based solutions: **Combined FoV (CFOV) and Single FoV (SFOV)**. CFOV and SFOV solutions are specifically designed to address the streaming challenges described in Section 1.2. These fixed tiling-based solutions include viewport prediction, tiles selection, and tiles bitrate adaptation modules to provide stable visual angles and higher quality levels recorded for different motion clips. CFOV solutions specifically utilize two low-complex viewport prediction mechanisms and implement combined FoV prediction-based streaming of 360° tiles. SFOV solutions introduce the idea of using both the prediction angles and prediction differences for accurate tiles selection during 360° video streaming. In com-

parison to existing algorithms, CFOV and SFOV implement attractive bitrate adaptation policies for tiles belonging to different streaming regions to achieve higher QoE levels:

- CFOV solutions are practical-oriented tile selection solutions for 360° videos. CFOV solutions reduce the complexity of tiles selection and improve the viewport availability by adopting two FoV prediction mechanisms. Specifically, CFOV2 considers different prediction combinations along with content information to select suitable tiles for different streaming regions.
- SFOV solutions predict the future viewport position by exploiting the viewing history of the user as well as content-related information (in SFOV2) in the context of prioritized remote 360° video transmission. The fixed tiling-based solutions predict both the horizontal and vertical movements of the user. SFOV solutions utilize the prediction angles and prediction differences to absorb the difference between real and predicted viewports.
- The proposed fixed tiling-based bitrate adaptation solution further refines the tiles selection outputted by CFOV and SFOV solutions based on the observed network capacity. It implements different bitrate adaptation policies to optimize the QoE objectives. In this context, active and improved region-wise bitrate allocations are performed for selected tiles without incurring unnecessary bandwidth consumption. The proposed bitrate adaptation algorithm selects the uniform bitrate for viewport tiles and other tiles with lower resolution or not at all to make efficient use of the available bandwidth.
- Extensive trace-driven simulations using real head motion traces of 48 VR users, with four different motion video clips, three tiling patterns, three segment lengths, and three dynamic bandwidth variations demonstrate that the proposed CFOV and SFOV solutions significantly improve the streaming performance compared to existing tile-based streaming approaches.

4. Dynamic tiling-based solution: **Dynamic Video Frames Tiling (DFT)**. DFT supports user interactivity and is capable of dynamically performing appropriate tiling pattern selection during 360° video streaming. DFT solution introduces three-tier adaptation: (i) adaptive tiling pattern selection, (ii) adaptive tiles selection, and (iii) adaptive bitrate selection. The main contributions of the DFT solution are as follows:
    - Two novel solutions named DFT1 and DFT2 are introduced for advanced tiling pattern adaptation. DFT1 considers a newly proposed visual-area-based weighted quality metric to decide the optimal tiling pattern for each segment. DFT2 adapts the tiling versions based on viewport prediction differences to improve the viewport availability for different motion content.
    - DFT solutions consider the following three cases for innovative tiles selection according to the user watching behavior: (i) *Case 1: Fixed viewport with no marginal tiles*; (ii) *Case 2: Fixed viewport with marginal tiles*; (iii) *Case 3: Extended viewport with no marginal tiles.* These tiles selection cases follow user head movements and provide a useful trade-off between viewport availability and corresponding quality levels.
    - DFT solutions actively allocate the video bitrate budget to different frame areas for maximizing the VR perception levels. DFT employs aggressive, weighted, or conservative bitrate adaptation policies to determine the quality levels for the tiles belonging to different regions based on exploration and exploitation of physical variables such as viewing areas, tiles distribution, and connection speed.
    - Testing involving 360° video streaming with five 4K videos prepared in three tiling patterns (4x3, 6x4, 8x6) and three segment durations (1s, 1.5s, 2s) demonstrate that DFT solutions improve the streaming performance under dynamic bandwidth conditions in comparison to some existing state-of-the-art adaptive tile-based solutions.

1.4 Organization of the Thesis

This thesis is organized as follows:

- **Chapter 1** - Presents the research motivations, identifies problems, and illustrates the main contributions of the research work presented in this thesis.
- **Chapter 2** - Introduces the technical background related to traditional video streaming techniques, 360° video streaming techniques, 360° video streaming framework, 360° video standards and technologies, and video quality assessment for the work presented in this thesis.
- **Chapter 3** - Provides state-of-the-art survey on the following topics: adaptive traditional video streaming solution, adaptive MVV streaming solutions, adaptive 360° video streaming solutions.
- **Chapter 4** - Explains the streaming architecture of TBOA and SD-DASH solutions and presents the experimental setup and performance evaluations, including a comparison with different other approaches.
- **Chapter 5** - Presents the illustration of the MVPDASH algorithm and performance evaluation under different network settings.
- **Chapter 6** - Presents the system architecture and proposed fixed tiling-based 360° streaming solutions WTV360 and CFOV along with detailed performance evaluations for these approaches.
- **Chapter 7** - Presents the detailed description of dynamic tiling-based 360° streaming solution DFT. The solutions were assessed in comparison to the fixed and dynamic tiling-based streaming solutions.
- **Chapter 8** - Presents the conclusion of this thesis with a detailed plan for future works.

1.5 Chapter Summary

This chapter presented the research motivation evidenced by the popularity and consumption of regular, multi-view, and 360° video streaming services over existing networks with limited bandwidth resources. The main research goals in terms of proposing bandwidth-efficient and quality-oriented HAS-compatible streaming so-

lutions for regular, multi-view, and 360° videos were elaborated. The main thesis contributions in terms of the proposed solutions that overcome the limitations of existing streaming systems were also described. In the next chapters, these contributions are described in detail.

Chapter 2

Technical Background

In this chapter, the technical background for this thesis is presented. First, the concepts related to traditional multimedia streaming are described, looking at various transmission protocols and adaptive video streaming technologies. The hierarchical streaming structure for the 360° video is presented next. Tile-based 360° video streaming technologies are covered in detail. Important technical standardization efforts that allow interoperability and flexibility for immersive media services are then discussed. Last but not least important, the most common QoE assessment metrics and QoE influencing factors are presented in detail. Understanding these technologies is essential to support the contributions of this work.

2.1 Video Streaming Evolution

The internet, including its component networks, relies on a best-effort data delivery approach. This approach is suitable for most content distribution services, but it is not the best choice for high-quality multimedia streaming applications. The real-time requirements are critical differences between multimedia and other packet data network traffic and require special attention. Both industrial and academic research development efforts have focused on proposing high-quality multimedia streaming from dedicated servers to remote users. As the goal was achieving high Quality of Service (QoS) levels, diverse Standards Developing Organizations (SDOs) such as International Telecommunication Union-Telecommunications (ITU-T), Internet Engineering Task Force (IETF), 3rd Generation Partnership Project (3GPP), European Telecommunications Standards Institute (ETSI), etc., have expanded their research activities to propose new technologies and protocols to support not only multimedia streaming, but also QoS-aware multimedia streaming.

Some of the early models supporting QoS-aware packet video delivery designed on top of the Internet Protocol (IP) were Integrated Services (IntServ) [60] and Differentiated Services (DiffServ) [61]. The IntServ architecture includes models for representing service types and quantifying resource management. Resources are explicitly reserved for meeting application specifications and are carried out by a signaling protocol known as Resource Reservation Protocol (RSVP) [62]. IntServ uses RSVP to represent the QoS requirements of an application's traffic along with the end devices in the network. The IntServ/RSVP model ensures guaranteed and predictive services based on a quantitative specification of resource reservations. However, the IntServ per-flow QoS-support approach is challenging to scale. In contrast to IntServ, DiffServ is a straightforward and lightweight solution that performs differentiation per class of traffic. DiffServ architecture aims to facilitate higher than best-effort QoS and scales well to extended networks. Nevertheless, service providers (SPs) QoS customization may affect the fine-grained end-to-end QoS.

Real-time Transport Protocol (RTP) [63] based on the simple User Datagram Protocol (UDP) [64] was proposed to standardize packet format and enable real-time mul-

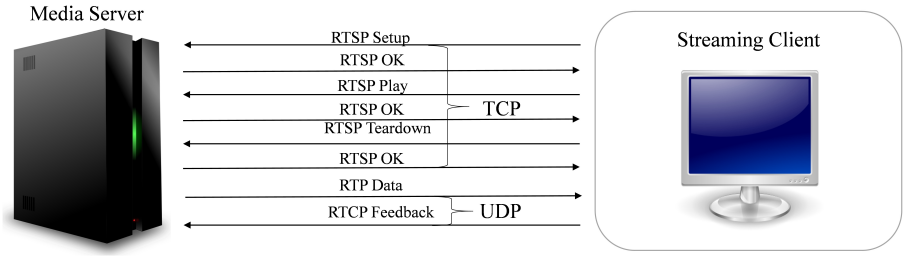


Figure 2.1: Traditional RTSP streaming system.

timedia streaming. Two fundamental principles used to design RTP are application-layer framing and integrated layer processing. In conventional non-HTTP-based streaming, the multimedia servers are controlled by a standard protocol known as Real Time Streaming Protocol (RTSP) [65], which first establishes a client-server connection before downloading the desired video file. The switching between multiple representations requires a connection re-establishment to download the desired quality from the beginning. The client issues specific commands such as play and pause to the server. The server responds to the user commands to deliver the content using RTP. Fig. 2.1 illustrates the RTSP streaming system where the state information of the streaming session is maintained throughout the session. RTP Control Protocol (RTCP) [66] is also a UDP-based protocol that monitors and controls the transmission statistics and QoS levels. It can achieve synchronization across multiple streams. Contrarily, such protocols have some problems in traversing firewalls and Network Address Translations (NATs) and require a dedicated network infrastructure, increasing the additional complexity and implementation costs. Transmission Control Protocol (TCP) [67] is a reliable transmission protocol that allows for efficient data transmission but suffers from variable delays in return for its reliability. TCP was proved to be beneficial by the researchers for delay-tolerant video transmission in the early 2000s. The rate fluctuations of TCP were compensated by introducing an application layer playout buffer.

2.1.1 HTTP Adaptive Video Streaming

The internet has evolved around HTTP. In early implementations, HTTP video streaming using HTTP over top of TCP allows progressive downloading, where a

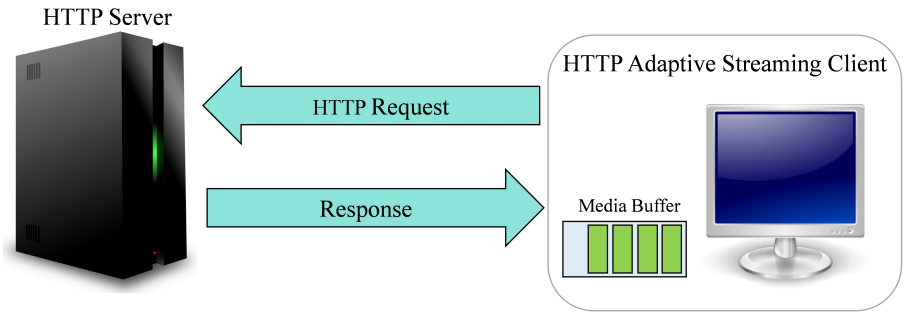


Figure 2.2: Adaptive streaming over HTTP.

constant quality video file is downloaded as quickly as TCP enables. The client plays the video before completing the download. A significant limitation is that the different types of clients receive the same video quality over various network connections, which may cause rebuffering or unwanted stalls. This technique is unsuitable for mobile environments, where bandwidth varies considerably more than the static environments [68]. The progressive download does not support live streaming [69] unless the video file is sliced into segments and the manifest supports it. This situation motivated the researchers towards the development of HAS to optimize user QoE. Fig. 2.2 shows an end-to-end adaptive streaming system, where the client uses the TCP as transport and HTTP as the application-layer protocol to pull the content from a server, which is the host of the media content.

Compared to HTTP video streaming, HAS achieves a better bandwidth utilization and decreases playback interruptions [70]. In HAS the client controls and manages the streaming session state using the standard HTTP protocol to reduce server-side network load. The streaming traffic, when sharing the network with other traffic, utilizes only a fair network bandwidth share (due to TCP's inter-flow friendliness). With HAS, the video bitrate of each chunk can be adapted according to the current network settings.

Commercially, after the first launch of HAS by Move Networks in 2006 [71], it was adopted by three large companies, such as Microsoft Corporation (2008), Apple Inc. (2009), and Adobe Systems Inc. (2010) as Microsoft Smooth Streaming (MSS) [72], HTTP Live Streaming (HLS) [73], and HTTP Dynamic Streaming (HDS) [74], respectively. These solutions were designed to meet different business requirements

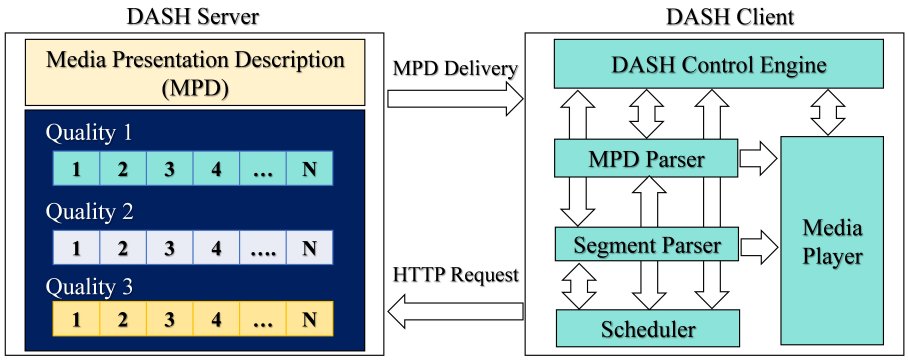


Figure 2.3: MPEG-DASH client-server model.

with various bitrate adaptation processes. Microsoft Smooth Streaming was developed for High-Definition (HD) media streaming to users. It utilizes different adaptation metrics (e.g., network throughput, CPU load, etc.). MSS opens two interchangeable TCP connections (i.e., one for video and the other for audio) with the server during a streaming session. HLS is also a widely embraced video streaming solution for both live and on-demand streaming applications. It uses both the device characteristics (e.g., memory, CPU, etc.) and network capabilities for adaptation decisions. HDS is regarded as the logical solution to stream HTTP content because it can facilitate intelligent multi-bitrate adaptive logic.

All existing HTTP streaming technologies share a similar technical background of content capturing, processing, and transmission. Despite wide commercial adoption, these solutions have several incompatibilities, such as not being well fit on existing HTTP infrastructures, lack of HTML5 support, QoE metrics, and support for interactive and multi-view videos, etc.

2.1.1.1 MPEG-DASH

The joint efforts of 3GPP and MPEG [75] resulted in the emergence of the codec agnostic DASH framework in 2012. DASH uses an application layer adaptation algorithm to provide a seamless video playback to a wide range of devices over heterogeneous networks. The adaptation algorithms try to prevent playback interruptions while improving the video quality by dynamically adjusting the bitrate to the ongoing network conditions. It also supports live and multi-view video streaming.

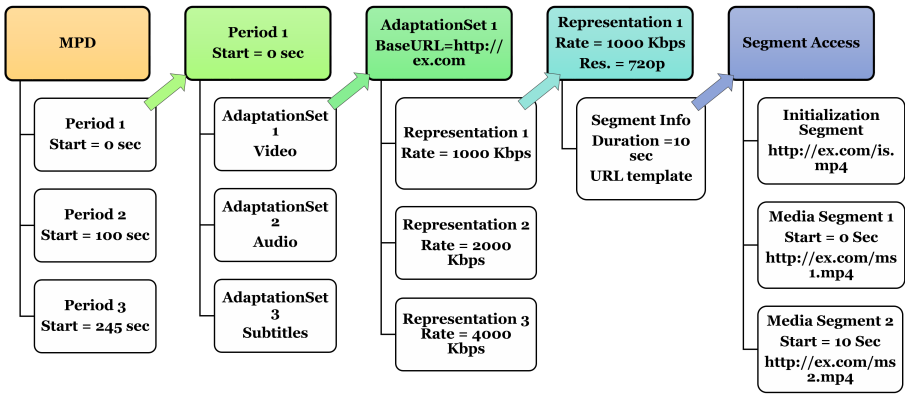


Figure 2.4: The hierarchical data model of MPD.

Fig. 2.3 shows an example of DASH-based adaptive streaming, connecting an HTTP video server to a DASH client. When employing a DASH-based approach, any video is prepared in several representations, encoded at different bitrates [69]. Each video representation is divided into several equal duration segments. Different bitrate segments are aligned such that the client can choose the appropriate video bitrate according to different network situations. The server stores the multimedia content information, and a client-side adaptation algorithm decides which bitrate to be selected for the next segment [76]. The adaptation algorithm is not part of the DASH standard.

The multimedia content stored on the server-side includes the pre-processed content and an Extended Markup Language (XML) file called Media Presentation Description (MPD) [77]. The MPD file contains the metadata information (e.g., mimeType codecs [78, 79], IP addresses, video profiles, download URLs, etc.). The video information is described as one or more *Periods* in MPD. Each *Period* has one or multiple *AdaptationSets*, each including several video versions called *Representations*. Each *Representation* is characterized by bitrate, height, and width of the video components and includes multiple video *Segments*. The *Segment* is the primary content unit and can be accessed and presented to the end-user. The MPD hierarchical structure is demonstrated in Fig. 2.4.

At the beginning of an adaptive streaming session, the client first requests the MPD file. Based on the underlying content information and user preferences, it selects the most fitting representations [80]. The client then sequentially requests the segments

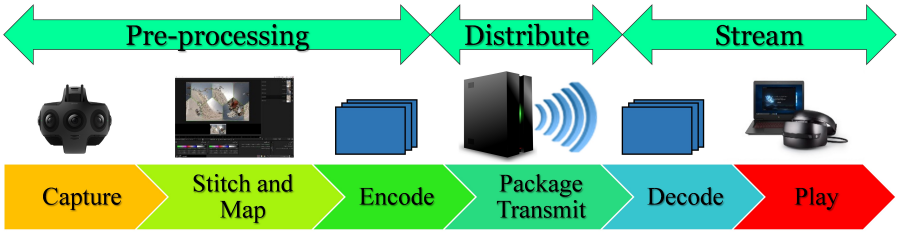


Figure 2.5: End-to-End 360° video streaming framework.

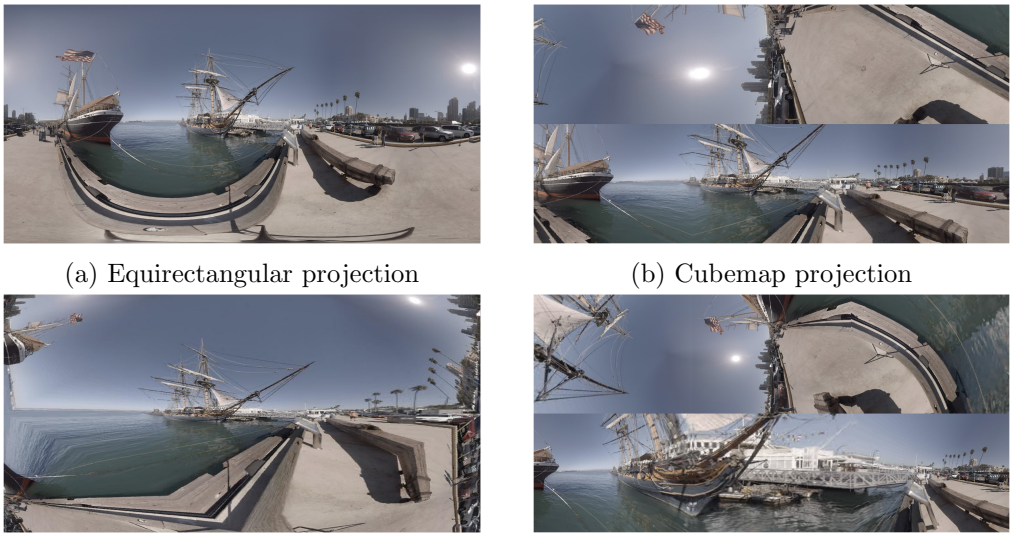
via HTTP GET requests and performs decoding and rendering to play the content. The client can switch to a different representation on a per-segment basis during the streaming session. DASH-based adaptation strategy should be carefully designed to decide when to adapt and which representation to request [70].

2.2 Adaptive 360° Video Streaming

Traditional HTTP streaming involves preparing the content at the server-side and then using the standard packaging and transmission formats to accomplish the successful delivery to the end-user. However, content creation, transmission, and consumption features are different in 360° video streaming architecture. Cameras with high resolutions are available for close-to and professional creation of 360° scenes. The increased efficiency of stitching software allows for better content preparation than ever. The encoding tools have improved the compression of full omnidirectional scenes. Networks deliver high bitrate content to remote end-users devices. Devices themselves (e.g., HMDs, mobile phones, tablets, etc.) are equipped with robust sensors, processing, and display components that enable them to receive, process, and display rich media content. However, supporting real-time streaming of 360° videos is still very challenging because they are associated with vast amounts of data, and its handling is highly time-sensitive. Fig. 2.5 illustrates the major stages in 360° video streaming. Next, these stages are discussed in detail.

2.2.1 Capturing and Stitching

The latest broad access to capturing and storage devices has shifted rich media production from studios to the large public. Regular users consume videos on smart



(a) Equirectangular projection (b) Cubemap projection

(c) Pyramid projection (d) Offset Cubemap

Figure 2.6: Different projection patterns of 360° video.

devices and are also involved in creating and distributing the 360° content. A multidimensional camera ring is usually used to record 360° scenes. Several companies, including GoPro[1], Insta360[2], Ricoh Theta [3], Samsung Gear[4], KODAK PIXPRO[5], etc., have started manufacturing Commercial-Off-The-Shelf (COTS) portable 360° cameras. These cameras usually include dual- or single-lens to generate a full 360° view of the environment. These cameras produce 2K-4K 360° video with a frame rate of 25-30 frames per second. The aspect ratio is recommended to be 4:3 in order to capture the maximum possible spatial area. Content stitching applications are used to attach different views to get a full 360° view. The captured 360° view is then represented as a Three-Dimensional (3D) sphere.

2.2.2 Projection Schemes

After capturing and stitching steps, the 360° sphere representation is transformed into a planer format. For this purpose, two main projection techniques are employed: (i) viewport-independent projection; (ii) viewport-dependent projection.

[1] https://gopro.com/
[2] https://www.insta360.com/
[3] https://theta360.com/
[4] https://www.samsung.com/global/galaxy/gear-360/
[5] https://kodakpixpro.com/cameras/360-vr/sp360

In *Viewport-Independent Projection*, the full 360° video is projected to a 2D plane in uniform quality, independent of the viewport. Examples include ERP [81] and CMP [81]. ERP is the most known mapping technique for 360° videos. ERP can be represented as flattening a sphere around the viewer onto a Two-Dimensional (2D) surface using yaw and pitch values. The yaw values range from -180° left to 180° right, while the pitch values range from 90° top to -90° bottom. Fig. 2.6a shows an ERP-generated frame. The world map is the most common example of this projection technique. Several 360° video streaming services, including YouTube, Youku[6], and Jaunt VR, use the ERP format. However, with this projection, the poles are represented with more pixels compared to the equator region, possibly consuming the limited bandwidth of the user in less interesting regions. Furthermore, this projection can generate several visual impairments due to lower compression efficiency.

A CMP is another popular mapping scheme. A six-sided cube combination is used to map the pixels of a sphere to the relevant pixels on the cube, as shown in Fig. 2.6b. This projection is widely used in gaming applications. Cubemap projection is more space-efficient and decreases the video size by 25% compared to an equirectangular approach [82]. It has been widely being used by Facebook for streaming 360° content, and the transformation code has been made open-source[7]. A significant inefficiency of this scheme is that unfair FoV boundaries can be observed. Thus, wasting the associated transmitted data.

In *Viewport-Dependent Projection*, the viewing areas are represented with higher fidelity than other areas. Examples of such projections are the pyramid projection [82], Truncated Square Pyramid (TSP) projection [83], offset cube map projection [31], etc. In a pyramid projection, the 360° sphere is projected onto a pyramid. Fig. 2.6c represents a pyramid projection where the base part is considered as the viewing region and coded with the highest available quality. Most of the projected area belongs to the user's viewing direction. This approach decreases the video size by 80% [82]. The main drawback associated with this projection is that if users

[6]https://www.youku.com/
[7]https://github.com/facebook/transform360

move their heads by 120°, the quality decreases aggressively as they rotate their heads by 180°. On the contrary, TSP projection lowers the amount of data over the edges and improves the streaming performance for high bitrate content. However, this projection approach involves sharp edges.

The offset cube map projection is shown in Fig. 2.6d. It is similar to a conventional cube map technique where the spherical pixels are projected to six cube sides, e.g., left, right, front, back, top, and bottom. The orientation-assisted higher quality viewing region compression offers smooth viewport quality and graceful quality variations to the edges. This projection has a strong storage overhead. With offset cube map projection, multiple quality representations, e.g., 88 versions, are required to serve different bandwidth profiles [31].

2.2.3 Encoding of 360° Video Content

The compression efficiency has improved significantly following the development of next-generation coding solutions. Currently, traditional and 360° videos use the same encoders, including Advanced Video Coding (AVC)/H.264, High-Efficiency Video Coding (HEVC)/H.265, VP9, VP10, AV1, etc. The current generation 4K 360° video requires 10-50 Mbps bitrate, while next-generation and 6-Degrees of Freedom (6DoF) 360° videos require 50-200 Mbps and 200-1000 Mbps bitrates, respectively [84]. Therefore, the efficient compression of 360° video is essential to be able to deliver it over the best-effort networks. AVC/H.264 standard uses a 16x16 macroblock structure for frame encoding. The size of encoded data is reduced through the motion prediction feature of the encoder. On the other side, HEVC/H.265 saves nearly 50% video bitrate compared to the AVC/H.264 with the same subjective quality. HEVC/H.265 supports the tile feature for efficient video streaming [85, 86], where a 64x64 Coding Tree Unit (CTU) structure is used to encode each tile. By employing this CTU structure, a higher compression ratio than AVC/H.264 is achieved. The video tiles are physically separated and concatenated in a regular stream to decode them by a single decoder. The next-generation, Versatile Video Coding (VVC) standard [87] is expected to increase compression efficiency by up to 30% compared to HEVC/H.265.

```
<!-- MPD file Generated with GPAC version 0.5.1-DEV-rev5379 on 2014-09-10T13:20:29Z -->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H9M56.465"
     profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
 <ProgramInformation moreInformationURL="http://gpac.sourceforge.net">
  <Title>dashed/BigBuckBunny_2s_onDemand_2014_05_09.mpd generated by GPAC</Title>
 </ProgramInformation>
 <Period duration="PT0H9M56.465">
  <AdaptationSet segmentAlignment="true" group="1" maxWidth="480" maxHeight="360" maxFrameRate="24" par="4:3" subsegmentStartsWithSAP="1">
   <Representation id="320x240 46.0kbps" mimeType="video/mp4" codecs="avc1.42c00d" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="45652">
    <BaseURL>bunny_45652bps/BigBuckBunny_2snonSeg.mp4</BaseURL>
    <SegmentBase indexRangeExact="true" indexRange="885-4504">
     <Initialization range="0-884" />
    </SegmentBase>
   </Representation>
   <Representation id="320x240 89.0kbps" mimeType="video/mp4" codecs="avc1.42c00d" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="89283">
    <BaseURL>bunny_89283bps/BigBuckBunny_2snonSeg.mp4</BaseURL>
    <SegmentBase indexRangeExact="true" indexRange="887-4506">
     <Initialization range="0-886" />
    </SegmentBase>
   </Representation>
   <Representation id="320x240 131.0kbps" mimeType="video/mp4" codecs="avc1.42c00d" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="131087">
    <BaseURL>bunny_131087bps/BigBuckBunny_2snonSeg.mp4</BaseURL>
    <SegmentBase indexRangeExact="true" indexRange="887-4506">
     <Initialization range="0-886" />
    </SegmentBase>
   </Representation>
   <Representation id="480x360 178.0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="178351">
    <BaseURL>bunny_178351bps/BigBuckBunny_2snonSeg.mp4</BaseURL>
    <SegmentBase indexRangeExact="true" indexRange="888-4507">
     <Initialization range="0-887" />
    </SegmentBase>
   </Representation>
   <Representation id="480x360 222.0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="221600">
```

(a) Sample MPD for traditional 2D video

```
<?xml version="1.0"?>
<!-- MPD file Generated with GPAC version 0.8.0-revUNKNOWN_REV at 2021-08-13T10:25:06.347Z-->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" mediaPresentationDuration="PT0H3M6.000S" maxSegmentDuration="PT0H0M1.000S"
     profiles="urn:mpeg:dash:profile:isoff-live:2011">
 <ProgramInformation moreInformationURL="http://gpac.io">
  <Title>ss.mpd generated by GPAC</Title>
 </ProgramInformation>

 <Period duration="PT0H3M6.000S">
  <AdaptationSet segmentAlignment="true" maxWidth="3840" maxHeight="1920" maxFrameRate="25" par="2:1" lang="und" startWithSAP="1">
   <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,0,0"/>
   <SegmentTemplate media="seg_dash_track1_$Number$.m4s" initialization="ss_set1_init.mp4" timescale="25000" startNumber="1" duration="25000"/>
   <Representation id="1" mimeType="video/mp4" codecs="hvc2.1.6.L186.80" width="3840" height="1920" frameRate="25" sar="1:1" bandwidth="7774352">
   </Representation>
  </AdaptationSet>
  <AdaptationSet segmentAlignment="true" maxWidth="448" maxHeight="320" maxFrameRate="25" par="448:320" lang="und" startWithSAP="1">
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,448,320"/>
   <SegmentTemplate media="seg_dash_track2_$Number$.m4s" initialization="seg_dash_track2_init.mp4" timescale="25000" startNumber="1" duration="25000"/>
   <Representation id="1_2" mimeType="video/mp4" codecs="hvt1.1.6.L186.80" width="448" height="320" frameRate="25" sar="1:1" bandwidth="15548704" dependencyId="1">
   </Representation>
  </AdaptationSet>
  <AdaptationSet segmentAlignment="true" maxWidth="512" maxHeight="320" maxFrameRate="25" par="8:5" lang="und" startWithSAP="1">
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,448,0,512,320"/>
   <SegmentTemplate media="seg_dash_track3_$Number$.m4s" initialization="seg_dash_track3_init.mp4" timescale="25000" startNumber="1" duration="25000"/>
   <Representation id="1_3" mimeType="video/mp4" codecs="hvt1.1.6.L186.80" width="512" height="320" frameRate="25" sar="1:1" bandwidth="15548704" dependencyId="1">
   </Representation>
  </AdaptationSet>
  <AdaptationSet segmentAlignment="true" maxWidth="448" maxHeight="320" maxFrameRate="25" par="448:320" lang="und" startWithSAP="1">
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,960,0,448,320"/>
   <SegmentTemplate media="seg_dash_track4_$Number$.m4s" initialization="seg_dash_track4_init.mp4" timescale="25000" startNumber="1" duration="25000"/>
   <Representation id="1_4" mimeType="video/mp4" codecs="hvt1.1.6.L186.80" width="448" height="320" frameRate="25" sar="1:1" bandwidth="15548704" dependencyId="1">
   </Representation>
  </AdaptationSet>
```

(b) Sample MPD for 360° video

Figure 2.7: MPD samples for regular and 360° video sequences.

2.2.4 Packaging and Transmission

The DASH framework does support not only traditional video streaming but also enables the quality efficient streaming of wide-angle panoramic, and 360° videos. DASH facilitates such Over-the-Top (OTT) services because of its compatibility and simplicity in the context of the current network infrastructure. In this context, Spatial Relationship Description (SRD) defines a set of media attributes representing the spatial relationships between different regions of a video. The MPD file is an extension of SRD because of the spatial relationship between associated video areas. It allows the DASH client to fetch the video streams at suitable bitrates related to the user's interest. The SRD syntax considers MPD authoring requirements and hybrid deployments with both legacy and SRD-aware DASH clients. Fig. 2.7 represents the MPD samples for regular and 360° videos. Fig. 2.7a represents MPD sample for a 2D video with 480x360 resolution and 24 fps. Fig. 2.7b shows an MPD sample for a 4K 360° video with 25 fps and an 8x6 tiling pattern. The supplemental property

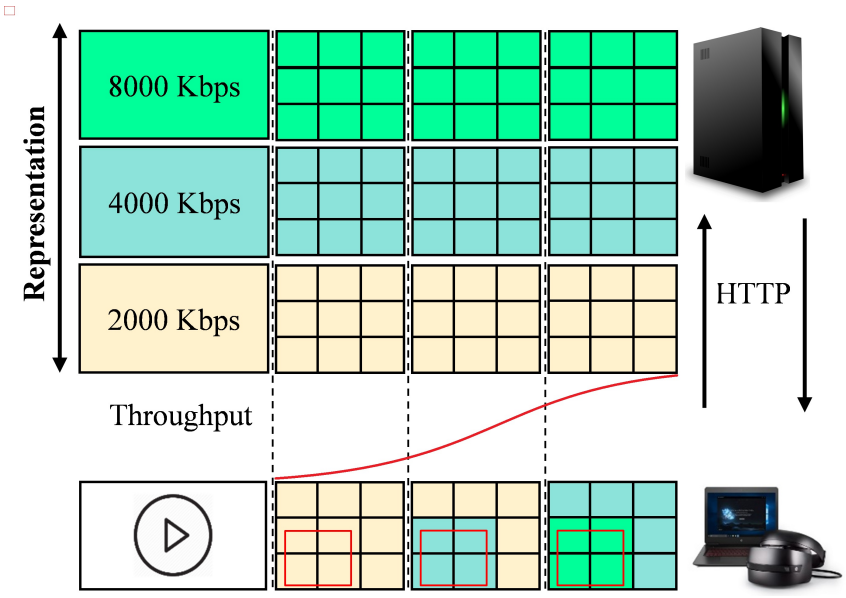


Figure 2.8: Adaptive streaming of 3x3 tiles of 360° video with three segments, each containing one frame.

descriptor "urn:mpeg:dash:srd" indicates the relevant positions of the tiled videos. Fig. 2.8 represents DASH-based adaptive streaming of 360° tiles. In this context, a client requires a high-quality subset of tiles according to the viewport position (red rectangle) and available bandwidth. The quality at which the client requests the tiles is dependent on network delivery.

360° videos are bandwidth-hungry streaming applications, demanding low response and buffering time. This is not always feasible, especially when there are far-located data service points. Recently, several new solutions, i.e., micro data centers [88], fog computing [89], and mobile edge computing [90, 91], have emerged, intending to minimize the distance and number of network components between users and far located large processing centers (i.e., cloud). Fig. 2.9 provides an architecture of edge computing and edge cache-enabled wireless networks for 360° video delivery. These technologies augment and extend the storage and computational resources from the current cloud model to the edges of the networks with objectives of low latency, efficient mobility, and real-time interactive support. Thus, the network transmission performance improves the user's immersive experience and is considered an essential step toward achieving an experience-centric 360° video delivery.

Edge computing assisted remote rendering provides several important benefits to

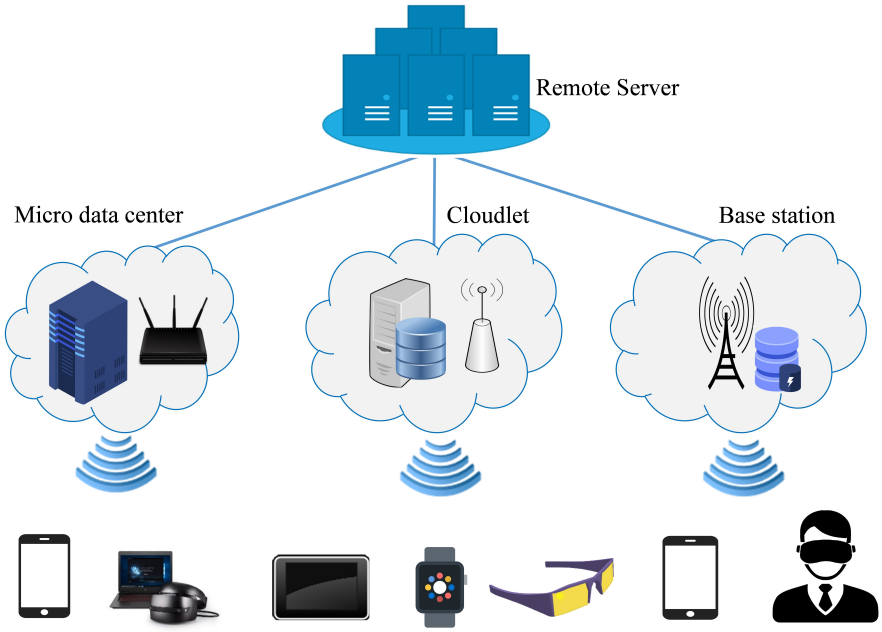


Figure 2.9: The architecture of mobile edge-assisted rendering of 360° video.

support high-resolution and high interactive VR video delivery over limited bandwidth networks, including:

1. *Latency Reduction:* The cloud alone cannot satisfy the requirements of all latency-sensitive applications, as it is usually far away from user devices. Edge computing enables collaborative computing to access a shared pool of servers. This design enables meeting the latency requirements of 360° video applications [92, 93, 51].
2. *Lower Energy Consumption:* Computation offloading to distributed computing clusters according to network architecture and resource provisioning significantly improves the energy performance of mobile devices [94, 95].
3. *Load Management:* Edge caching provides means to store the content near the users, i.e., Base stations, small cells, or end terminals, lowering the load on the core network [96].

2.2.5 360° Video Rendering and Displaying

360° video rendering and displaying are computationally expensive due to the considerable processing power requirements. The client-based rendering solution is the most commonly used option in web-based players like YouTube and Facebook, where

the viewing area is constructed from decoded frames and then presented to the end-user. The client-based rendering results in the processing of a larger amount of content that does not belong to the FoV of the user. Another 360° video rendering option is cloud/edge-based rendering [97]. There is no need for content processing on the client-side. The video data is processed on the cloud/edge, and only the requested FoVs are streamed to the end-users with no additional requirements of bandwidth or hardware resources. This is particularly useful in a multi-client streaming environment.

360° video has become an integral part of multimedia applications. The consumer is always interested in the interactive and immersive streaming experience. The term "streaming" refers to the continuous delivery of audio-visual content to the user. Several standard organizations and industry forums are contributing many helpful insights for the future of 360° videos. In general, the architecture of the 360° video is still in progress. QoE goals are somehow advanced for 360° content compared to regular videos. These goals aim to achieve an uninterrupted, highly immersive spatial presence in a VR environment.

2.3 Tile-based 360° Video Streaming

Tile-based adaptive streaming has become a widely embraced solution for both live [98, 99, 100, 101] and on-demand [102, 103, 104, 105] 360° streaming applications. Different tiling patterns are employed to achieve compression and bandwidth-efficient streaming goals. Based on the spatial partitioning of tiles, we have classified tile-based adaptive streaming into two main categories: (i) fixed tiling-based adaptive streaming; (ii) dynamic tiling-based adaptive streaming. In fixed tiling-based adaptive streaming, the server stores the content prepared in a single tiling pattern. The client requests the viewing tiles from the same tiling pattern stored at the server end for each adaptation interval. Fig. 2.10 shows a server architecture of a traditional fixed tiling-based adaptive streaming where the projected video is spatially partitioned into 12 tiles and temporally divided into S equal segments considering five different quality levels. In dynamic tiling-based adaptive streaming, multiple

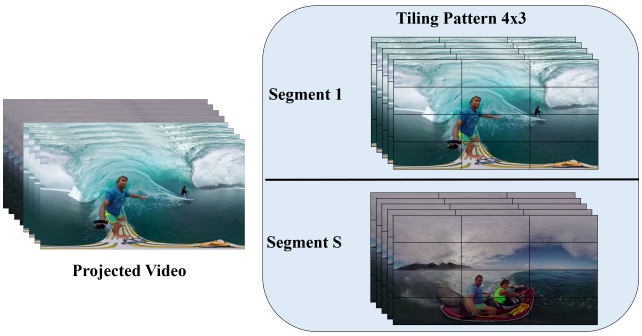


Figure 2.10: Server architectures of traditional fixed tiling-based adaptive streaming.

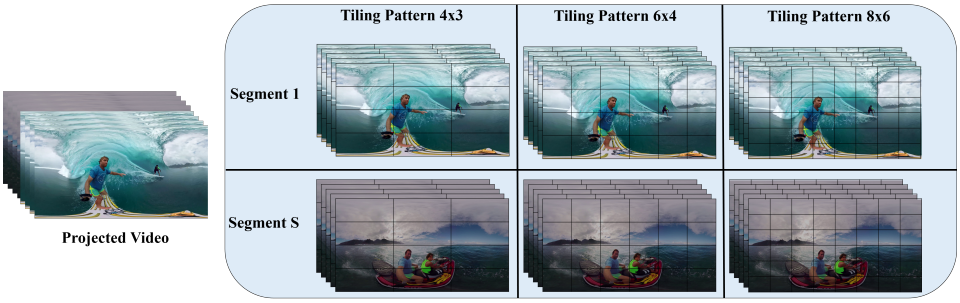
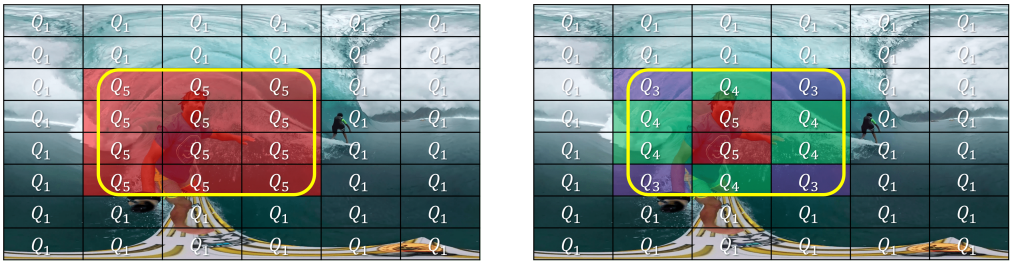


Figure 2.11: Server architectures of dynamic tiling-based adaptive streaming.

tiling patterns/versions are prepared on the server-side. Fig. 2.11 represents the server containing three different tiles sets, i.e., 12, 24, and 48, for a single streaming session. In this technique, the client first dynamically selects the viewing tiles from one of the available tiling patterns based on different decision factors. In addition to deciding the appropriate visible tiles and their corresponding bitrates, the adaptive client should be able to optimally decide which tiling pattern to fetch for the next segment to maximize the overall QoE.

The fixed tiling-based adaptive streaming is further divided into three main categories: (i) fixed viewport-based adaptive streaming; (ii) marginal region-based adaptive streaming; (iii) extended viewport-based adaptive streaming. In fixed viewport-based adaptive streaming, a fixed viewport field is employed to render the higher quality viewport (usually one-quarter of the whole stream) regarding the user's dynamic motion patterns. The basic idea is to map the tiles to the actual viewport coverage of the display terminal in higher quality levels. At the same time, the remaining portion of the sphere (three-quarters) can be streamed at a relatively lower quality. Fig. 2.12 illustrates the tiles arrangement and quality adaptation for the



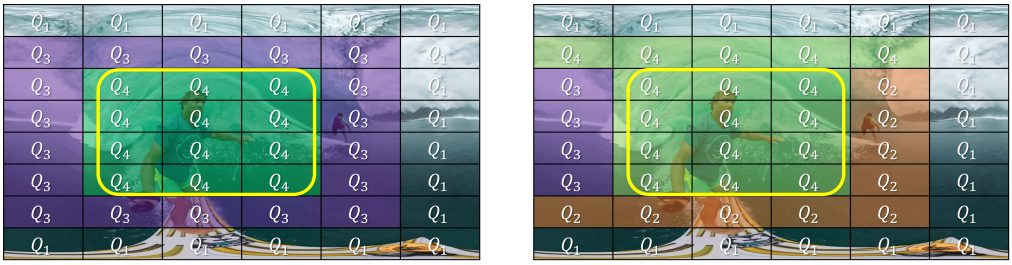
(a) Fixed viewport-based streaming in [48]. (b) Fixed viewport-based streaming in [106].

Figure 2.12: Tiles arrangement and quality adaptation for fixed viewport-based adaptive streaming in an equirectangular space, where the yellow window represents the viewport region.

fixed viewport-based adaptive streaming technique where the user can watch the content within the actual viewport. In [48], the tiles are arranged into viewport and background groups, and a uniform quality allocation is performed for two regions (Fig. 2.12a). Fig. 2.12b represents different priority levels and non-uniform quality allocation for fixed viewport tiles in [106].

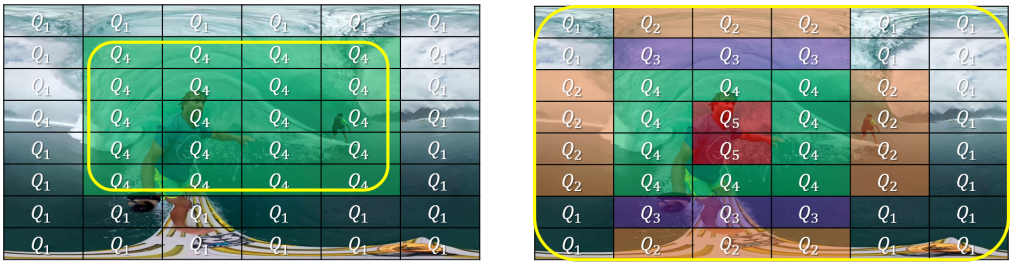
In marginal region-based adaptive streaming, a spatial extension is defined around the viewport to deal with possible head movement prediction errors. The coverage of the marginal regions could be in horizontal or vertical directions by defining mapping on the left and right or top and bottom sides of the viewport to manage the horizontal or vertical head movements, respectively. Fig. 2.13 shows marginal region-based adaptive streaming techniques where the 360° video tiles are divided into viewport, marginal, and background tiles. An impeccable solution is to surround the viewport region with marginal tiles in all directions to create a high-quality safety zone around it, as shown in Fig. 2.13a. However, this can lead to bandwidth wastage under medium to high prediction accuracy. A differentiated quality marginal region scheme is employed in [56], where the marginal tiles are divided into four sub-areas (Fig. 2.13b). The dynamic selection of the viewport and marginal tiles before each segment request influences the overall streaming experience. Therefore, it is necessary to carefully decide how much marginal region to include and in which direction to achieve bandwidth-efficient streaming.

The main idea of extended viewport-based streaming is to virtually extend the original viewport coverage to include a larger viewing region to balance the actual



(a) Marginal region-based streaming in [107]. (b) Marginal region-based streaming in [56].

Figure 2.13: Tiles arrangement and quality adaptation for fixed viewport-based adaptive streaming in an equirectangular space, where the yellow window represents the viewport region.



(a) Extended viewport-based streaming in [108]. (b) Extended viewport-based streaming in [48].

Figure 2.14: Tiles arrangement and quality adaptation for extended viewport-based adaptive streaming in an equirectangular space, where the yellow window represents the viewport region.

and predicted tiles' overlap and visual quality. Instead of adding a marginal region around the viewport, the viewport coverage is extended by 10%-30% or so, depending on the viewport prediction performance. Only the higher-quality viewport tiles or higher-quality viewport and remaining tiles can be rendered remotely in such a scenario. Fig. 2.14 shows the tiles classification for the extended viewport case. A 1°-30° viewport extension is considered in [108] based on the network conditions (Fig. 2.14a). In [48] the full-frame is considered as the extended viewport; however, the quality is assigned in a priority fashion from the center to the last tile (Fig. 2.14b). It is imperative for an extended viewport technique to determine how much to extend and under which variables (e.g., network conditions, viewport prediction, etc.) to avoid over- and under- extension for a favorable higher QoE.

2.4 360° Video Standards and Technologies

Standardization is a key issue for proper technical interoperability. The overall goal of standardization efforts is to specify the minimum essential to enable creative and competitive technologies and services. Several standardization efforts for omnidirectional videos are currently gaining momentum. To ensure universal media access and interoperability for production, distribution, sharing, and consumption [109], media is often encapsulated and signaled using standardized file formats and transport protocols. In this context, MPEG has developed several standards including MPEG-2 Transport Stream (TS) [110], MPEG-4 Part 14 (MP4) [111], MPEG-DASH, MPEG Media Transport (MMT) [112], etc., to ensure media interoperability. In this section, the standardization efforts for immersive media applications are described by considering the immersive media formats, 3-Degree of Freedom (3DoF)+ streaming, and industrial setups.

2.4.1 Immersive Media Formats

MPEG has developed a standard for immersive media, called MPEG Immersive media (MPEG-I), that includes an Omnidirectional Media Format (OMAF) [113] focusing on the specifications of omnidirectional media applications. OMAF is the first international standard on immersive media format and describes the means to allow the coding, presentation, and consumption of 360° videos. OMAF is compatible with existing standards, including coding (e.g., HEVC), file format (e.g., ISO Base Media File Format (ISOBMFF)), delivery signaling (e.g., DASH, MMT), and includes metadata information of encoding, projection, packing, and viewport orientation.

In an OMAF workflow, the media is captured via one or more fish-eye lenses where stitching and projection steps are shifted towards the capturing side. OMAF considers equirectangular projection and cube map projection due to their effectiveness. OMAF also relies on Region-Wise Packing (RWPK) [114], where the projected frame can be scaled, resampled, rotated, and mirrored according to the streaming requirements. The region-wise packing can be used to circumvent the weaknesses of the

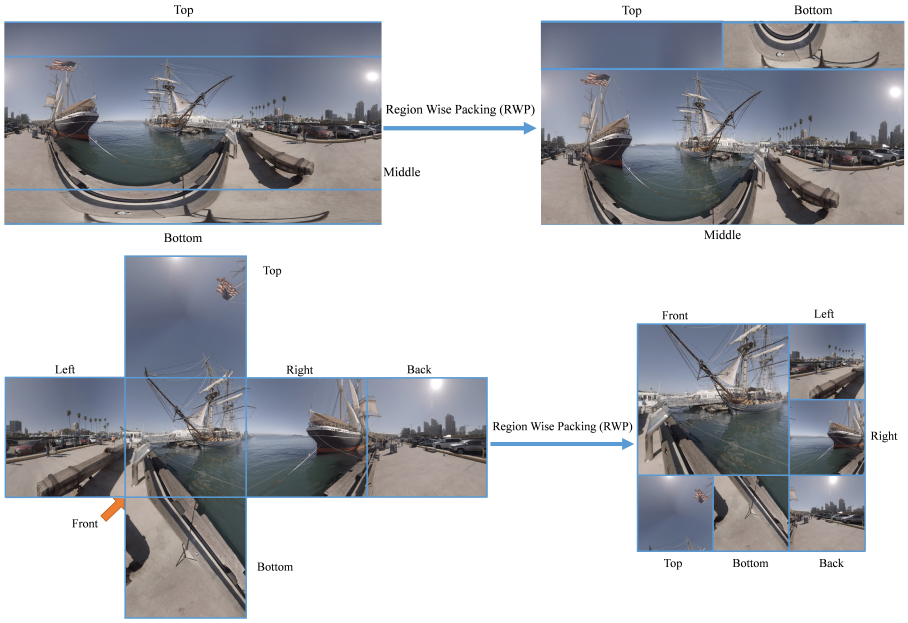


Figure 2.15: Region-wise resampling and positioning for the ERP (top) and the CMP (bottom) projections.

projection schemes and to reduce the computational complexity by downsampling the lesser important regions, i.e., top and bottom regions in case of ERP, and all the regions except the front face in case of CMP, as shown in Figure 2.15. The encoded frame contains the resulting packed regions. After encoding, the content is processed using the existing media file format and transport protocols together with some metadata to facilitate additional signaling for 6DoF navigation and selective viewport delivery to DASH clients [115].

The integration of OMAF with DASH comes with additional property descriptors included in the MPD to inform the client about 360° media properties. Several newly defined omnidirectional metadata are added in MPD, including Projection Format (PF) descriptor, RWPK descriptor, Content Coverage (CC) descriptor, Spherical Region-wise Quality Ranking (SRQR) descriptor, 2D Region-Wise Quality Ranking (2DQR) descriptor, and Fish-eye OMnidirectional Video (FOMV) descriptor. OMAF specifies nine media profiles, including three video profiles such as HEVC-based viewport-independent, HEVC-based viewport-dependent, and AVC-based viewport-dependent video profiles [116]. OMAF provides a consistent quality to the whole frame regardless of the viewport position for viewport-independent

streaming. The regular HEVC codec and DASH streaming format can be used for viewport-independent streaming. However, the adaptive viewport-based operation using HEVC/AVC codec is a technical development of OMAF that allows unconstrained use of rectangular RWP for enhanced quality of viewport regions.

In 2016, MPEG approved the Common Media Application Format (CMAF) [117] that aims to provide a uniform encoding format and media profiles to be used across multiple applications and devices. The CMAF makes it possible to request lower latency segments. ISOBMFF [118] is the most popular file format for timed data exchange, management, and presentation. The ISOBMFF files consist of a series of compliant and extensible file-level boxes. Each box represents a data structure comprised of four printer characters code. The ISOBMFF media and metadata streams kept in a track are distributed separately. The media data includes coded audio and video data. The metadata, similar to the conventional formats, represents media type, codec properties, timestamps, size, etc. ISOBMFF specifies additional metadata for omnidirectional content such as projection format, rotation, frames packaging, encoding, and delivery. The track describes the format or content of samples, including the coding and packaging format. After decoding for proper display, the video tracks that need post-processing are marked with the 'resv' entry type. ISOBMFF ensures the flexible aggregation of valuable information for easy access and inclusion in a transport manifest that supports efficient media consumption over the dynamic network.

2.4.2 Towards 6-Degrees of Freedom+

MPEG-I divides the standardization of immersive media into three phases. Phase 1a of MPEG-I aimed to complete essential monoscopic and stereoscopic 360° video services, also known as 3DoF, by 2017. In a 3DoF scenario, the user can freely move his head along three axes, i.e., yaw, pitch, and roll. If a user fastly moves his/her head while watching the stereoscopic video, the parallax error indicating the poor VR experience is observed. Phase 1b of the MPEG-I standard aims to support commercialized 3DoF+ services by the end of 2020. In 3DoF+, the user can freely look in any direction along with limited head movements in front/back,

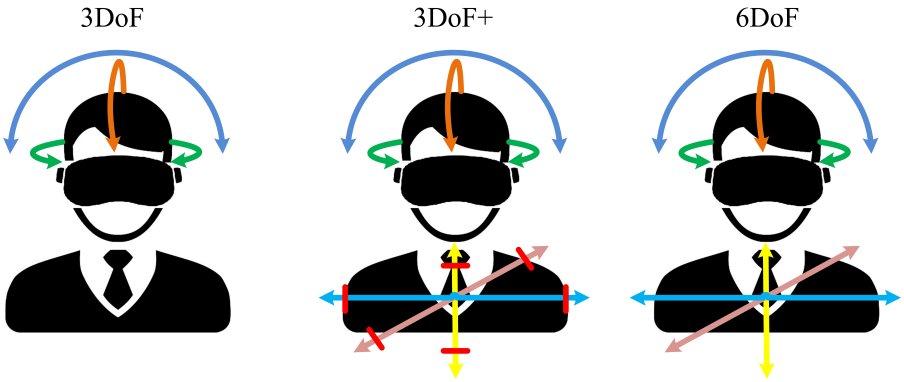


Figure 2.16: VR viewing arrangements in 3DoF, 3DoF, and 6DoF+ environments.

up/down, and left/right directions. 3DoF+ will advance the viewport quality to strengthen the sense of realism. If the view is missing in the original video, 3DoF+ headsets will be able to employ the reference intermediate view synthesizer [119] to synthesize the view accordingly. In March 2019, MPEG announced a Call for Proposals (CfPs) on 3DoF+ videos to develop advanced coding solutions and 3DoF+ metadata standardization. The purpose of the final phase, phase 2 of MPEG-I is to support more elaborated services such as 6DoF by 2022. 6DoF visual system will provide full support for orientation and position tracking. The user will have free movements, similar to the real-world environment. Fig. 2.16 represents different possible movements of a user concerning 3DoF, 3DoF+, and 6DoF.

Digital Video Broadcasting (DVB)[8] created the Commercial Module VR (CMVR) group to support commercial requirements for the efficient delivery of VR media over digital video broadcast networks. Currently, the CM-VR research group targets the panoramic/3DoF+ content as highlighted within MPEG-I. Furthermore, the DVB CM-VR Study Mission Group (DVB-VR-SMG) also aims to explore exceptional experiences, such as 6DoF. CM-VR group is also considering the work done by other organizations, e.g., MPEG, VRIF, and 3GPP, to ensure it is in synchronization with the latest technological developments.

[8]https://www.dvb.org/

2.4.3 Immersive Audio and Video Standards

Immersive audio and video is an enabling technology behind the media and entertainment industries. Several standard development organizations and industry forums contribute many helpful insights for the future of immersive multimedia streaming. The VR Industry Forum (VRIF)[9] is composed of a wide range of participants from different sectors to provide a high-quality audio-visual VR media experience to consumers. The main focus of the VRIF is to perceive 360° VR content accompanied by 3D spatial audio by pursuing VR guidelines. VRIF is focusing on building a content library that could benefit the industry to examine and promote VR implementations. 3GPP has documented the impact of VR by evaluating the relevance and potential of VR services for the industry. The document includes the audio-video media formats, delivery procedures, subjective tests, as well as latency and synchronization aspects [120]. 3GPP has put several efforts into immersive audio services with the primary intention to support codecs for VR and 3D Audio with Enhanced Voice Services (EVS) extensions. The study items include codecs for VR audio (CODVRA) and codec for Immersive Voice and Audio Services (IVAS) to support potential standardization in this domain.

The Streaming Video Alliance (SVA)[10]) association covers the video ecosystem for developing best practices and specifications to promote the online video value chain. The SVA study group on VR and 360° videos focus on understanding the market potential, existing VR technologies such as players and use-cases, and cataloging standardization efforts.

The Video Electronics Standards Association (VESA)[11], responsible for a lot of digital display standards, including VGA, DisplayPort, etc., has formed a special working group to explore the standardization of the emerging Augmented Reality (AR)/VR applications. The group is responsible for: (i) establishing communication connections and data transfer protocols for AR/VR services, (ii) analyzing existing VESA standards to suggest any modifications, and (iii) collaborating with other

[9] https://www.vr-if.org/
[10] https://www.streamingvideoalliance.org/
[11] https://www.vesa.org/

SDOs [121].

IEEE P2048 is one of the largest working groups involving over 200 companies and institutions for immersive media standardization. IEEE P2048.10/P2048.3 standards define the immersive audio/video file and streaming formats. IEEE P2048.9/P2048.2 deals with the taxonomy and quality metrics for several immersive audio/video service variants. These standards are required to reduce the confusion among variants of immersive media services. IEEE P2048.4 standard deals with methods for maintaining a person's meaningful representation in a VR environment. Furthermore, the IEEE P3333.3 standard provides technical guidance to reduce cybersickness or 3D sickness to protect the viewer's health and develop a healthy ecosystem.

JPEG Pleno [122] is an initiative by the Joint Photographic Experts Group (JPEG[12]) that targets to determine a standard system for recording, packaging, and transmission of immersive media content. It aims to provide an efficient compression format and tools to support advanced functionalities with limited resource utilization for the highest quality content representation. The supported features include data and metadata manipulation, low latency, scalability, editing, JPEG compatibility, random access, privacy protection, and security. JPEG XT [123] is an image format standard for coding 360° images. JPEG XT is backward compatible with the existing JPEG standards and involves multi-part specifications.

A further effort by JPEG is JPEG360 Ad Hoc Group which was established in 2017 to ensure full interoperability for enabling interactions with multi-sensor images captured using omnidirectional cameras. The main objectives include defining use cases and requirements for 360° applications, updating metadata descriptions, gathering evidence of existing solutions, and facilitating the processes leading to the evaluation of new proposals. The functional metadata requirements include storing multiple images within a single file, stitching software, projection type, coding format, pixel format, orientation, etc.

[12]https://jpeg.org/

2.4.4 QoE Metrics

With the rapid development of immersive video technology, there is much need for advancements in precise quality assessment methodologies. Towards this cause, MPEG Joint Video Exploration Team (JVET) has defined some common test conditions and reference configurations for performance evaluation of 360° video [124]. MPEG-JVET has further investigated several quality assessment metrics by considering different aspects of the 360° video. The 3GPP work item, FS_QoE_VR, aims at the identification of QoE metrics for VR content and device conditions and capabilities. The QUALINET[13] group focuses on QoE of multimedia systems and services. The Immersive Media Experience (IMEx) group of QUALINET focuses on a broad range of services, including mulsemedia, 360° VR, 3D audio, and future video coding solutions to develop methodologies and best practices for immersive media experience. The IMEx aims to identify application domains, use cases, software and hardware tools, and subjective quality assessment methodologies. It further supports standardization activities and liaisons with other SDOs for collaborative research and mutual benefits.

Video Quality Experts Group (VQEG) has initiated Immersive Media Group (IMG)[14], which targets the quality assessment of immersive media applications. The collaboration of the IMG group of VQEG and the IMEx group of QUALINET leads to the development of the Joint QUALINET-VQEG team on Immersive Media (JQVIM). This joint venture aims to collect and produce immersive media content, tools, datasets, as well as maintain standardization procedures and research activities for QoE assessment of AR/VR immersive media [109].

2.5 Video Quality Assessment

2.5.1 Subjective Quality Assessment

It is critical to quantify the playback performance of any video application, i.e., traditional, multi-view, and 360°, to determine the user's specific degree of perception

[13] http://www.qualinet.eu/

[14] https://www.its.bldrdoc.gov/vqeg/projects/immersive-media-group.aspx

and satisfaction, referred to as QoE. The quality assessment approaches are often classified into two categories: subjective and objective. Since users are the ultimate consumer of videos, subjective quality assessment is the actual and surest way to estimate the video streaming performance. Subjective tests are directly associated with human eyes and shed light on the impact of different aspects of video delivery. **Mean Opinion Score (MOS)** [125] is the most commonly used metric for subjective quality assessment, which is based on gauging users' ratings (usually between 1 and 5) considering different quality aspects for any displayed video sequence. Subjective quality assessment needs comprehensive manual efforts and is thus expensive, time-consuming, and error-prone. Furthermore, the test results can be distorted if the assessors, video sequences, or quality versions are insufficient. Therefore, ITU-T has issued several standards, recommendations, and data processing guidelines to follow the best test settings and procedures. For instance, **ITU-T P.910** [126] and **P.911** [127] provide the guidelines about non-interactive subjective tests for assessing the video quality. They describe several interesting aspects, such as the number of users and video sequences, video quality levels, sequence duration, screen brightness, user's distance from the screen, test environment, etc., to better engage the users. **ITU-R BT.500** [128] recommends the general methodologies for subjective assessment of television images. It outlines assessment features, viewing conditions, and environmental recommendations similar to the ITU-T P.910 and ITU-T P.911. It recommends consistent source signals, test sequences, observers, and rating calculations to avoid skewness in the results. Compared to ITU-T P.910, ITU-T P.911, and ITU-R BT.500, **ITU-T P.913** [129] covers new paradigms for non-interactive video, audio, or audiovisual quality assessment techniques for Internet and distribution quality television videos. ITU-T P.913 testing methodologies are particularly useful in assessing the quality and performance of display devices in multiple environments and situation-specific subjective assessment in the confounded viewing and listening environment. **ITU-T P.920** [130] describes test conditions and experimental procedures for performing interactive tests for audio-visual quality assessment in multimedia services. This methodology recommends at least 16 participants that need to rate audio-visual quality, the effort needed to interrupt, communication

difficulty, and acceptability of communication.

As 360° video is highly prevalent, it is imperative to determine the user's specific quality aspects with this type of video distribution. The significant difference between regular and 360° videos is that only viewport content is displayed in 360° videos. Therefore, spatial presence and cybersickness caused by experiencing 360° video via VR headsets are most important since these effects do not occur if a user watches regular videos on a desktop screen. Cybersickness is a potential barrier to achieving higher QoE levels and can cause fatigue, nausea, discomfort, and aversion [131]. **ITU-T P.919** [132] outlines content characteristics, such as content type, spatial and temporal information, number of clips, duration, etc., to subjectively evaluate the quality of 360° videos on HMDs. The general testing recommendations are about audio-visual quality measurements, cybersickness symptoms, streaming experience, and rating collection methods (e.g., Absolute Category Rating (ACR) and Degradation Category Rating (DCR)). Subjective quality assessment methods need comprehensive manual efforts and are thus expensive, time-consuming, and error-prone.

2.5.2 Objective Quality Assessment

Objective quality assessment is more manageable and practical compared to subjective quality evaluation. **Peak Signal to Noise Ratio (PSNR)** [133] is the most frequent and extensively used objective quality assessment metric in the absence of subjective assessment. PSNR is a full reference method calculated based on frame-by-frame processing of the original and distorted video sequences for video streaming applications. PSNR calculates the synchronization between transmitted and received frames and is defined in eq. 2.1.

$$\mathrm{PSNR_{dB}} = 20 \log_{10} \frac{\mathrm{MAX_I}}{\sqrt{\mathrm{MSE}}} \tag{2.1}$$

where MAX_I represents the highest possible pixel value of the image (e.g., 255), and MSE represents the mean square error between the original and the distorted images. Several approaches appeared in the literature defining PSNR. For network-

based video transmission, the PSNR metric is calculated as follows:

$$\mathrm{PSNR}_{\mathrm{dB}} = 20 \log_{10} \frac{\mathrm{MAX_Bitrate}}{\sqrt{(\mathrm{EXP_Thr} - \mathrm{CRT_Thr})^2}} \tag{2.2}$$

where $MAX_Bitrate$ represents the maximum bitrate of the processed content, and EXP_Thr and CRT_Thr represent the expected average and current average throughput during the multimedia transmission.

Structural Similarity Index Measure (SSIM) [134] is a full-reference perception-based metric that computes the spatial closeness between the initial uncompressed and the distorted images. The SSIM metric has a reasonable association with the MOS and incorporates luminance masking, contrast masking, and structural similarities phenomenons for the perceptual score calculation of images. For video quality assessment, SSIM is calculated as a frame-by-frame comparison of original and encoded videos, similar to PSNR. SSIM metric returns a score value between -1 and +1. The observers can not distinguish between original and distorted images for scores equal to or close to 1.

It is natural to employ conventional objective assessment metrics, such as PSNR and SSIM for 360° content due to the similar encoding structure and 2D plane projection formats. However, the sampling density in existing projection methods is not uniform at each pixel position. Also, these metrics were designed to calculate the compression overhead or transmission-based quality loss, i.e., packet loss.

Spherical-PSNR (S-PSNR) [135] calculates color samples by comparing pixels of two spherical videos on pre-defined positions. The relative frequencies of visible areas are calculated based on the ground truth head movement traces instead of equally weighting the polar and equator pixels. By utilizing interpolation algorithms, S-PSNR can achieve an objective quality measurement of 360° videos supporting various projection schemes. **Spherical-SSIM (S-SSIM)** [136] metric computes the similarity between impaired and original 360° videos. In S-SSIM, a reprojection is incorporated to calculate the similarity between the extracted viewports.

Weighted Spherically PSNR (WS-PSNR) [137] measures the PSNR of two same projection and same resolution images in representation space (projected rep-

resentation) and weighs the pixels by multiplying with errors in observation space (spherical representation). Therefore, the mapping area of the pixels on the visible spherical portions impacts WS-PSNR. This metric, thus, avoids the propagation error and results in reliable quality evaluation results by equally sampling the pixels from the representation space to the relevant pixels in the observation space.
Weighted Spherical-SSIM (WS-SSIM) [138] metric measures the structural similarity between the windows on the projected area by considering the weights similar to [137]. WS-SSIM was reported to offer results closer to human perception than other quality evaluation metrics.

The quality metrics designed for 360° videos include remapping based on the corresponding projection format. The **Craster Parabolic Projection-PSNR (CPP-PSNR)** metric [139] performs the PSNR assessment by remapping the pixels to a two-dimensional cross projection format without changing the spatial resolution. With CPP projection, the resolution of a 360° video could hardly be decreased. CPP-PSNR downsamples the projected formats to match the pixel density of a sphere. However, it is also essential to consider the subsampling locations and pixel relationship of the sphere and the projected images implemented within the current encoding infrastructure.

ProbGaze-PSNR and **ProbGaze-SSIM** metrics [140] compute the PSNR and SSIM based on the spatial dimension of tiles and the gaze point within the viewport. ProbeGaze versions of the objective metrics consider the weight of peripheral tiles. They can estimate the quality changes on peripheral tiles when a user suddenly changes the viewing position compared to the center-based (which considers only the weight of the viewport tiles) and average-based (which considers the average weight of all the tiles) versions of PSNR and SSIM metrics.

Measuring visual quality alone for video streaming services is insufficient for a complete adaptive framework. It is also essential to find the impact of other impairments, e.g., playback startup delay, background quality, playback rebuffering, spatial and temporal quality variations, etc.) which strongly influence the adaptive streaming performance.

Playback startup delay impacts the performance of HAS applications as the user

needs to wait for a certain amount of video data to be buffered before the playback begins [141]. The practical value of initial delay depends on several factors, such as available connection speed, amount of data to be transferred or segment duration, as well as decoding time. Usually, the user has to wait for more than necessary for a larger data size, e.g., 360° video. In addition, the initial delay can make the user feel more uncomfortable for a video with a shorter duration than a loner duration video. Achieving a quick startup time can lead to playback buffer depletion. Therefore, the adaptation client must be carefully designed to provide a worthwhile trade-off between the initial delay and the playback buffer risk.

Stalling/rebuffering is the interruption in playback due to the buffer depletion [142]. A user can quit/change the video streaming for a stalling duration of more than 2s. In a stable/fluctuating network environment, requesting a video segment with a bitrate higher than the available network throughput will drop the buffer levels, which eventually leads to stalling [143]. The stalling in HAS impacts the user experience in two dimensions, i.e., the number and duration of stalls. Sometimes a user can feel more uncomfortable for several short-duration stalls compared to a single longer-duration playback interruption. Compared to classical HTTP video streaming, HAS is more manageable to align the segment size to the available network conditions. However, under prompt bandwidth drops, the HTTP client should be able to display the content to improvise the perceived quality. Similarly, for 360° streaming services, the viewport content should be displayed all the time irrespective of the user's head turns. For an immersive streaming experience and providing a higher quality display, it is also necessary to avoid viewport deviations. Therefore, for any video streaming application, the client should be able to achieve an essential balance between video quality and playback buffer levels.

Quality fluctuation factor refers to the bitrate/quality switches in an adaptive streaming session [70]. The adaptation algorithm at the server/client-side decides the quality level for the next segments in response to the ongoing network conditions, buffer levels, or both. Non-adaptive streaming methods do not undergo any quality fluctuations as they adhere to only a single quality level throughout the streaming, leading to stalling or poor quality levels. On the other hand, the HAS client can ex-

perience severe quality fluctuations depending on the decision variables. A user can adjust to immediate quality increments and graceful quality decrements. Therefore, the frequency and magnitude of the quality fluctuations impact the overall streaming performance. In traditional streaming applications, the quality is adapted across the segments, referring to temporal quality fluctuations; however, in addition to temporal quality fluctuations, the 360° client needs to adapt the quality for different streaming regions, referring to the spatial quality oscillations. Therefore, the 360° client should be able to manage the inter- and intra- frames quality variations to provide an enduring streaming experience.

Accurate QoE assessment is a significant factor in optimizing the traditional and 360° video streaming services and is a fundamental one for adaptive delivery solutions. More structured research towards designing widely acceptable accurate QoE assessment models and metrics is highly needed, and this is challenging.

2.6 Chapter Summary

This chapter presented the evolution of video streaming technologies, including non-adaptive, adaptive, and MPEG-DASH streaming frameworks. Several current and emerging 360° streaming techniques were presented. The major 360° video streaming stages, including content creation, projection, encoding, packaging, transmission, and rendering, were described. The ongoing standardization efforts for 360° media services that ensure interoperability and immersive media deployment on a massive scale were briefly described. Finally, video quality assessment approaches for traditional and 360° videos were explained. This technical background is the basis for understanding the technical aspects of this work.

Chapter 3

Related Works

This chapter presents a comprehensive survey of the latest research on the following topics: (i) Adaptive Traditional Video Streaming Solutions, (ii) Adaptive Multi-View Video Streaming Solutions, and (iii) Adaptive 360° Video Streaming Solutions. In this context, current and emerging adaptive solutions for traditional video streaming, including server-side and client-side (e.g., throughput-based, buffer-based, and hybrid), are discussed. The benefits and limitations of several adaptation schemes for multi-view videos over single and multiple paths are discussed. In addition, the most relevant works about viewport-independent, viewport-dependent, and tile-based streaming (e.g., surveys and studies on 360° video streaming, fixed viewport-based, marginal region-based, extended viewport-based, and dynamic tiling-based streaming), as well as viewport prediction techniques are discussed and compared.

3.1 Adaptive Traditional Video Streaming Solutions

The primary goal of an Adaptive BitRate (ABR) algorithm is to accurately adjust the video bitrates in order to improve the streaming experience. Significant efforts have been put into designing diverse adaptive video streaming solutions. Considering the location of the adaptation logic, the adaptive schemes from the literature can be divided into two categories: (i) server-side adaptive streaming solutions and (ii) client-side adaptive streaming solutions. Next, these adaptive schemes are discussed in detail.

3.1.1 Server-side Adaptive Streaming Solutions

Server-side HTTP adaptive streaming solutions perform the bitrate switching and selection decisions at the server-side with/without any cooperation from the client-side. Most server-side adaptive schemes require the client to send feedback signals, such as the estimated available bandwidth and buffer occupancy, before selecting the available representation for the next segment. The bitrate decision engine on the server-side then selects the appropriate bitrate from the available bitrates to send for the next segment, as shown in Fig. 3.1.

Muntean et al. [27] proposed the Quality-oriented Adaptive Scheme (QoAS) to provide improved perceptual quality for streamed multimedia content to end-users. The QoAS system consists of server-side and client-side components to perform quality-oriented adaptation decisions. The client includes a Quality of Delivery Grading Scheme (QoDGS) component, which calculates the quality scores based on a number of parameters, i.e., packets lost, packet delay, and delay jitter. The QoDGS unit transmits the short-term or long-term quality variations scores based on the variations of parameters to the server. The server arbitration scheme (SAS) at the server-side performs the quality selection among the lowest, below medium, medium, above medium, and the highest quality levels based on the feedback reports to minimize the impact of quality degradation.

De Cicco et al. [144] designed a stream-switching Quality Adaptation Controller (QAC) mechanism using feedback control theory for live adaptive video streaming.

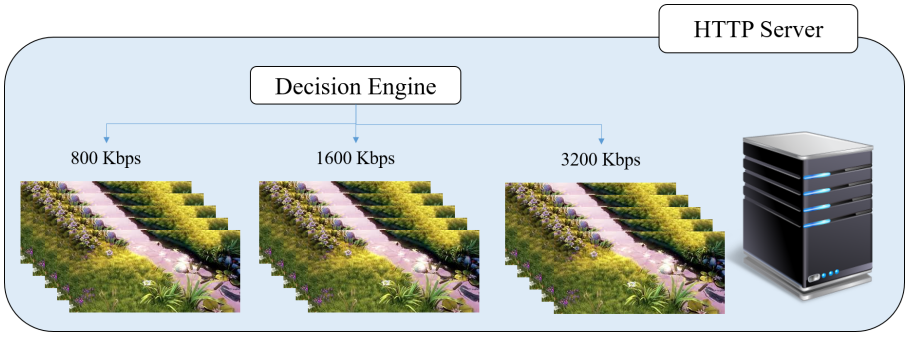


Figure 3.1: Architecture of server-side adaptive streaming solutions.

QAC architecture controls the length of a sender buffer, which temporarily stores the selected video segments at the server-side. It aims to avoid the buffer failure events at the client-side by throttling the bitrate level in accordance with the available bandwidth. The performance comparison with the Akamai streaming prototype shows that the QAC achieves a fair bandwidth share for greedy traffic flow and concurrent stream transmission.

Yuan and Muntean [145] presented an intelligent Prioritized Adaptive Scheme (iPAS) for heterogeneous multimedia, i.e., voice, video, best-effort, and background, transmission over IEEE 802.11 networks. Firstly, the proposed solution assigns different priority levels to multimedia streams based on application type, device characteristics (i.e., resolution, battery level), and performance parameters (i.e., delay, jitter, and packet loss). Afterward, a bandwidth share percentage for each stream is computed by a Stereotype-based Bandwidth Allocation module based on their determined priority levels. Finally, the model-based bandwidth estimation (MBE) module allocates a specific amount of the bandwidth to each stream based on the measured available bandwidth. The iPAS achieves a fair-share bandwidth among multiple streams and provides higher throughput performance than 802.11 DCF and 802.11e EDCA, as validated by the simulation and real-life experiments.

Zou et al. [146] introduced DOAS, a Device-Oriented Adaptive multimedia Scheme for Long Term Evolution (LTE) networks. This scheme was built on top of the LTE downlink scheduling mechanism. The DOAS server performs the adaptation decisions based on mobile device classification and the QoS parameters to use the network resources efficiently. The device classification module classifies the devices

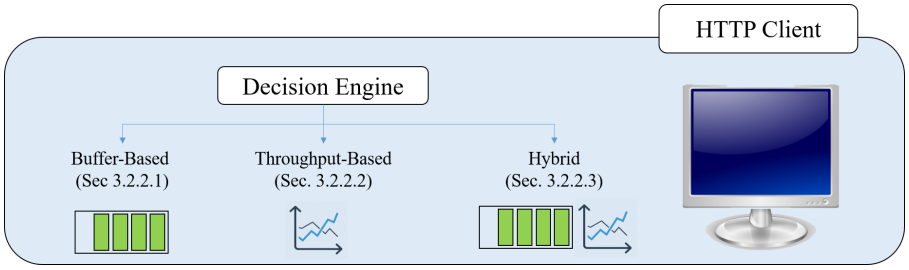


Figure 3.2: Classification of client-side adaptive streaming solutions.

into five classes based on their resolution. The adaptation module then selects the suitable quality levels for each device class based on the QoS information of the current channel. However, this work was compared with a non-adaptive streaming scheme.

Bruneau-Queyreix et al. [147] proposed a multi-server-based adaptative streaming system called MS-Stream to enhance the user's perceived quality. MS-Stream system includes a multiple-server streaming protocol to simultaneously retrieve content from multiple servers, content adaptation, a server-switching mechanism using MPD extension, and an intra-description switching algorithm for server selection. However, this work does not include any prediction mechanism for on-demand or user satisfaction-based online description placement in the network.

Despite the success of server-side bitrate adaptation approaches, these solutions can not be directly applied in existing streaming architectures without custom server software. Moreover, these approaches result in high computation overhead and struggle to perform appropriate coding selection when serving a large number of mobile users. In this context, the server and edge-computing technologies can be an alternative solution, where both the server and network components can improve the bitrate adaptation performance for multiple users.

3.1.2 Client-side Adaptive Streaming Solutions

Client-side adaptive streaming schemes have attracted essential consideration as the client-driven HAS increases scalability by removing the session maintenance from the server-side [148]. The bitrate adaptation unit residing at the client-side takes one or more metrics as input to decide the quality level for the next segment. The

main goal of these adaptation approaches is to maximize the video quality while minimizing playback interruptions, video quality fluctuations, and startup time. Client-side adaptation approaches can maintain playback buffer levels by adapting to bandwidth variations and, therefore, can be classified into throughput-based, buffer-based, and hybrid adaptive solutions, also represented in Fig. 3.2. The most representative adaptive approaches are discussed next.

3.1.2.1 Throughput-based Adaptive Streaming Solutions

In throughput-based adaptive streaming technology, the bitrate adaptation module performs the adaptation decisions solely based on the estimated network throughput. **Liu et al. [149]** presented a throughput-based adaptation and detected the throughput variations by considering segment fetch time (SFT), which represents the period between starting and receiving instants of HTTP GET requests. The decision engine decides the bitrate of the upcoming segments based on the measured SFT. In another work, [150], the authors considered both the parallel and sequential segment fetching approaches in content distribution networks. The authors compared both the actual and expected SFT to select the quality levels of future segments. However, the implemented conservative and aggressive bitrate switch-up decisions can significantly lower end-user satisfaction.

Jiang et al. [41] identified some bitrate selection issues when several HAS clients share a common congested bandwidth link. They studied the design space of ABR algorithms related to three metrics: efficiency, stability, and fairness. They introduced an algorithm named FESTIVE, which explores a robust mechanism for segment scheduling, throughput estimation, and bitrate selection. The FESTIVE algorithm contains a randomized scheduler to schedule the downloading of the next video chunks. They performed an Open Source Media Framework (OSMF)-based implementation [151] to improve the above-mentioned three metrics. The authors identified that a practical adaptive bitrate approach must try to avoid three main aspects when multiple clients share a full bandwidth link with capacity W, and every client x plays video bitrate $b_{x,t}$ at time t:

- *Inefficiency*: The multiple clients sharing a bottleneck link must be able to se-

lect the highest possible representations to improve their streaming experience. The inefficiency is given as:

$$Inefficiency = \frac{|\sum_x b_{x,t} - W|}{W} \tag{3.1}$$

A lower inefficiency value shows that the multiple users have a possible bitrate for maximum experience.

- *Unfairness*: The available network resources should be shared equally among multiple clients. In a multi-client environment, unfairness is given as [152]:

$$Unfairness = \sqrt{1 - JainFair} \tag{3.2}$$

A smaller value of unfairness indicates a fair share of network resources among competing clients.

- *Instability*: Unnecessary bitrate switches can negatively influence the streaming experience. An instability metric is defined as:

$$\frac{\sum_{d=0}^{k-1} |b_{x,t-d} - b_{x,t-d-1}| * w(d)}{\sum_{d=1}^{k} b_{x,t-d} * w(d)} \tag{3.3}$$

Li et al. [153] introduced the Probe AND Adapt (PANDA) algorithm to determine the network state considering an average target data bitrate for future bitrate selection. PANDA aims to minimize bitrate oscillations by correctly probing the network when several HAS clients compete for a congested bandwidth channel. The performance evaluations against FESTIVE [41] and Microsoft Smooth Streaming [72] players show that PANDA has the best adaptive behavior among these solutions, achieving the highest efficiency, fairness, and stability under different bandwidth and player settings.

Sun et al. [154] proposed the Cross Session Stateful Predictor (CS2P), a data-driven throughput estimation scheme to overcome the inaccurate HAS traffic prediction problem. The authors first applied clustering approaches to the streaming sessions sharing similar characteristics and then predicted corresponding throughput samples for each cluster using different Hidden Markov Model (HMM). An experimental evaluation with a large-scale dataset reveals that CS2P efficiently estimates

the available network throughput to improve the overall video bitrate adaptation. Similar to CS2P, some other solutions such as CFA [155] and Pytheas [156] also involve a data-driven controller to estimate the available throughput. However, these works do not support system heterogeneity and involve additional training complexity, making them less attractive.

In a live streaming environment, the end-to-end delay degrades the streaming experience. To overcome this, **Xiao et al. [157]** proposed a DASH-enabled content delivery solution to mobile clients. The authors analyzed that the overall streaming quality depends not only on the local measurements of the throughput but also on the server's network capacity. The authors utilized a server-initiated push mechanism to stream the DASH content to mobile clients to lower the end-to-end latency. They also leveraged HTTP/2's stream termination feature to perform intermediate quality adjustments. The segment scheduling based on the estimated user's Quality of Experience (QoE), energy cost, and available resources offers an improved streaming experience with limited battery consumption to mobile users. While there is evidence that the performance is improved, this work was evaluated via simulations in a controlled LAN environment only.

Throughput-based adaptive streaming solutions struggle to accurately measure the available network throughput. An erroneous approximation results in fetching the incorrect quality segments and limits the overall QoE [158]. For example, An under-estimation of throughput may bring lower-quality segments, while an over-estimation can result in rebuffering. In addition, throughput-based solutions could introduce severe buffer depletion by looking only at the throughput signals. Employing throughput-based adaptation for 360° video without correct throughput estimation may encounter instability and poor QoE, especially under highly dynamic wireless and cellular networks [2].

3.1.2.2 Buffer-based Adaptive Streaming Solutions

Buffer-based adaptive clients request the upcoming segments considering the current buffer levels during the video playback to overcome the limitations of incomplete network information. **Mueller et al. [159]** presented a buffer-based approach in

combination with a performance metrics toolset and a compensation algorithm in a multi-client cache-enabled environment. The client-centric model effectively detects the bitrate switching period and compensates these switches by choosing appropriate video bitrates, resulting in up to 20% quality increments. Despite the limited simulation analysis, the oscillations compensation approach described is fascinating.

Huang et al. [42] proposed a Buffer Based Adaptation (BBA) approach and deployed it in Netflix browser-based video player. They carried out extensive evaluations and found that it is risk-free to completely rely on buffer estimation for playback adaptation during steady buffer levels. However, when the buffer grows during the startup phase, it is necessary to consider the throughput estimation to adapt the bitrates for the next segments. Compared to Netflix's default algorithm, the proposed approach drops the rebuffering events by 10-20%. However, BBA considers a large buffer size, usually in minutes; therefore, it does not guarantee promising performance for short videos.

Spiteri et al. [39] proposed a Buffer Occupancy-based Lyapunov Algorithm (BOLA) that considers bitrate adaptation as an optimization problem associated with playback quality and rebuffering time. BOLA aims to avoid rebuffering by maintaining the buffer close to a set target level. BOLA avoids the frequency of stalling events for a sudden drop in buffer level by requesting the lowest available video bitrate. The authors implemented BOLA using an open-source DASH reference player[1] and showed that it offers an adequate enhancement to the video quality with less probability of rebuffering.

To ease the buffer utilization for adaptation decisions, **Beben et al. [160]** proposed an enhanced Adaptation and Buffer Management Algorithm (ABMA+) that determines the download time of future representations based on the probability of rebuffering events. ABMA+ ensures a smooth playback by selecting the maximum bitrates based on a pre-computed buffer size and segment download time. This buffer-based strategy results in an acceptable deployment with a low computational cost.

Sieber et al. [161] proposed a Scalable Video Coding (SVC)-based adaptation

[1] https://reference.dashif.org/dash.js/

approach, namely Bandwidth Independent Efficient Buffering (BIEB), to improve the quality selection. BIEB fetches the video chunks based on the quality layers distribution to maintain a consistent buffer level for lower playback interruptions. However, BIEB does not consider stalls or quality switches in the QoE model during unstable network conditions. While video quality variations rate and playback stalls negatively impact the user's satisfaction [162]. Furthermore, SVC-based streaming approaches involve additional coding and processing overheads.

While video quality variation rate and playback stalls negatively impact the user's satisfaction, **Tian and Liu [36]** proposed a control-theoretic approach using a PID controller that enforces a buffer set-point to keep the buffer to an optimum level. The proposed design balances the display smoothness and bandwidth utilization by imposing a target buffer level. The video adaptation algorithms simply increase the bitrate in response to the buffer level and available throughput measured using the Support Vector Regression (SVR) [163]. An aggressive switch-down is implemented in response to sudden congestion levels. The proposed design was implemented in both single and multiple server environments. However, the proposed solution does not ensure fairness among multiple competing clients, resulting in lower perception levels for some clients.

Buffer-based adaptation approaches try to keep the buffer level to stable states while avoiding the risk of buffer underflow/underflow. However, buffer-based solutions can undergo inefficient bandwidth utilization in a multi-client streaming environment due to the unawareness of segment download time for each competing client. Buffer-based approaches usually adopt a larger buffer space for multi-client streaming. However, an ample buffer space could not be feasible for a smooth 360° video streaming because of the high uncertainty of long-term viewport prediction. Usually, a small buffer (<3s) is reasonable under short-term viewport prediction [164]. However, there is a high chance of the playback stalling with a short buffer space. Therefore, short-duration segments can also be used for tile-based streaming to lower the risks of playback buffering. However, short segments have lower coding efficiency compared to long segments, especially for tile-based streaming [165].

3.1.2.3 Hybrid Adaptive Streaming Solutions

In this category of adaptation, the client determines the video bitrate of upcoming segments considering both the throughput and playback buffer signals. **Yin et al. [166]** presented a control-theoretic approach called Model Predictive Control (MPC) that utilizes the set of well-defined parameters for estimating the available network and buffer resources to optimally adjust the bitrate decisions for high QoE. The proposed QoE model employs the average video quality R_k, average bitrate switches, rebuffering events, and initial delay T_s components.

$$QoE_1^K = \sum_{k=1}^{K} q(R_k) - \lambda \sum_{k=1}^{K-1} |q(R_{k+1}) - q(R_k)| - \mu \sum_{k=1}^{K} (d_k(R_k)/C_k - B_k)_+ - \mu_s T_s \tag{3.4}$$

where C_k and B_k represent the available bandwidth and buffer occupancy for the kth chunk, respectively. The components' weights (i.e., λ, μ, and μ_s) depend on user interest and can be adjusted accordingly. MPC considers throughput estimation using a harmonic mean approach and can explicitly manage the complex control objectives. This work studied only a single-player, so there was no fairness consideration.

Miller et al. [167] proposed a hybrid adaptation approach that involves the throughput and buffer measurements to achieve different objectives, i.e., minimize the startup time, maximize the video quality, minimize the playback interruptions, and video quality fluctuations. The proposed solution employs three thresholds for the buffer levels, such that $0 < B_{min} < B_{low} < B_{high}$. The target interval B_{tar} lies between B_{low} and B_{high}. However, the algorithm tries to stay at the optimum interval $B_{opt} = \frac{B_{low}+B_{high}}{2}$. By controlling B_{low} and B_{high} thresholds, the proposed solution tries to stabilize the buffer level and bitrate variations in response to the unknown TCP throughput. When tested for single and two client scenarios under restricted bandwidth settings, the algorithm exhibits smooth and fair behavior. However, this work does not involve a comparison with other approaches.

Mok et al. [37] proposed a QoE-aware DASH-based streaming model to improve the video playing QoE. The authors conducted subjective experiments with 11 video

clips and found that users are more reluctant to switch abruptly between best and worst quality levels. The proposed model is based on the same assumption to introduce intermediate quality levels instead of switching to the minimum quality level. It includes two main modules named QDASH-abw and QDASH-qoe with minimal systems modification. QDASH-abw is responsible for measuring the network condition on the server-side by the RTT variations to support the highest quality selection for the next segment. QDASH-qoe calculates the most appropriate quality level that the network can support. However, QDASH was tested for short duration H.264/AVC video clips.

Vergados et al. [38] proposed a fuzzy logic-based adaptation solution named FDASH to control the rebuffering events and video streaming quality. FDASH considers three buffering states, i.e., falling, steady, and rising, as input to the fuzzy controller falling, which outputs the resolution of the next segment as reduce, small reduce, no change, small increase, or increase. The proposed solution was evaluated for single and two client scenarios with cross-traffic interference in wired and wireless network environments. The experimental results show the success of FDASH in achieving higher video bitrates and stable buffer levels while avoiding playback interruptions and quality fluctuations against [149, 150, 36, 167]. However, this work was evaluated for only one video stream and considered only a fixed wireless connection with no background traffic for two client scenarios. Moreover, this work does not consider QoE metric [166] or fairness [41] among competing clients.

Similar to [38], **Sobhani et al. [168]** proposed a fuzzy logic-based adaptation approach to actively respond to the available network and player states. The proposed solution employs Kaufman's Adaptive Moving Average (KAMA) [169] method to maximize the network utilization for streaming decisions. The buffer state was measured using Grey Prediction Model (GPM) [170] to help the adaptation decision for improving the buffer state. Based on the available measurement, the fuzzy controller decides to increase, decrease, large decrease, or keep the same bitrate for the next segment. The emulation performance under competing flows reveals that the proposed system has better fairness (50% on average) and better-perceived quality (17% maximum) compared to four alternative baseline approaches. However, this

work considers only one video sequence with a 6s segment duration.

Wang et al. [171] introduced a Spectrum-based Quality Adaptation (SQA) mechanism to solve the inconsistencies of throughput prediction and buffer level estimation. Both throughput and buffer level feedback signals were used for appropriate quality selection. Initially, SQUAD fetches the lowest quality segments to lower the startup time and then increments the corresponding quality levels considering the throughput and buffer measurements. The authors showed that SQUAD offers significantly improved performance regarding video quality switching frequency and magnitude. Unfortunately, none of the solutions discussed provides a good balancing between video quality and bandwidth utilization.

Zhou et al. [172] proposed Throughput Friendly DASH (TFDASH) algorithm to achieve fairness, stability, efficiency, and higher video quality for multiple competing clients watching the Big Buck Bunny video sequence in dynamic network environments. The proposed system employs Kalman filter [173] for bandwidth estimation and Logarithmic Increase Multiplicative Decrease (LIMD)-based bandwidth probing to achieve maximum bandwidth utilization. In addition, the dual-threshold buffer ensures stable playback. In addition, a dual-threshold buffer model is employed to prevent buffer underflow and buffer overflow events. The proposed probability-driven bitrate adaptation approach performs the adaptation decisions based on several time-varying factors visual quality levels, buffer levels, and quality switching frequency and amplitude. The proposed model achieves maximum and fair bandwidth utilization by avoiding unnecessary OFF periods compared to FESTIVE [41] and PANDA [153] clients.

In hybrid adaptation approaches, several system parameters collaborate to help amenable HAS clients achieve higher streaming performance. In a single-client environment, such algorithms work reasonably well. However, multiple competing clients quickly choke the entire network. When the client buffer reaches a maximum level, the client enters an ON-OFF phase, during which it may not correctly estimate the available bandwidth because every client will adjust the video bitrate without respecting others. This leads to bandwidth under-utilization and unequal bandwidth distribution among competing clients [43]. Therefore, an efficient throughput and

buffer awareness-based adaptation solution should be designed to satisfy bandwidth utilization, perceived quality, viewing smoothness, and uninterrupted playback experience for single and multiple clients.

3.2 Adaptive Multi-view Video Streaming Solutions

The desire to deliver increasing amounts of rich media content at higher quality over the existing heterogeneous network environments has fuelled research in the area of adaptive multimedia streaming over multiple paths. Some related works about technologies, applications, and limitations have been demonstrated in this section.

3.2.1 Multipath-based Streaming Solutions

For all innovative rich media solutions, an important challenge in the current network environment remains the delivery of an increased amount of content. Multiple network interfaces on mobile terminals can be leveraged to improve the wireless availability and communication from an always-connected state to the Always Best-Connected (ABC) state. In this context, the direct employment of the Multipath Transmission Control Protocol (MPTCP) [174] helps but is not ideal because it requires kernel stack modification at both the sender and receiver terminals. Additionally, the MPTCP traffic may not pass through middleboxes as it is restricted by several network operators [175]. Other solutions such as CMT-QA employ specifically multiple network technologies, including cellular (e.g., LTE) and wireless broadband (e.g., WiFi), in order to enable concurrent multipath content delivery [176]. Researchers are always devoted to optimizing multiple paths to achieve high-quality video transmission and resiliency in the case of failure.

Chen et al. [177] proposed a multi-source player, called MSPlayer, to achieve high-quality video transmission over multiple links and resiliency in the case of failure. The MSPlayer requests video chunks from multiple YouTube video servers utilizing both WiFi and 4G interfaces simultaneously without involving kernel stack modifications or middleboxes similar to MPTCP. The client-driven bitrate allocation

of future video segments depends on the estimated network conditions. After bitrate selection, the video segments are alternatively downloaded over the two available networks. MSPlayer provides stable buffer levels by quickly filling the playout buffer with high-quality data segments. However, downloading segments over different paths may cause out-of-order delivery.

Xu et al. [178] analyzed the real-time quality of the data transmission paths by exploring the interaction between the data link layer and the transport layer and proposed a fairness-driven efficient flow control mechanism for parallel data transfer over multiple wireless networks. The proposed design performs the SNR measurement at the data link layer and the bitrate and bandwidth estimation at the transport layer to know about the channel quality and sending behavior. The cross-layer fairness-driven solution then enables the parallel data distribution by maintaining an important balance between path efficiency and fairness. The performance evaluation of the cross-layer fairness solution has been carried out against CMT-QA [176] considering average throughput, fairness, and Peak Signal to Noise Ratio (PSNR) metrics. Simulation results show that the cross-fairness solution attains higher fairness levels but obtains lower average throughput and PSNR in comparison to CMT-QA.

Kim and Chung [179] employed both the WiFi and LTE network interfaces to download partial segments from multiple video sources. The authors implemented a partial segment request strategy to avoid out-of-order-related problems. The size of the partial segments is computed based on the throughput measurements of each path. The partial segments transmitted over various paths are combined before they are presented to users. The aggregated bandwidth of the paths is smoothed by taking the transmission difference to avoid bandwidth fluctuations. The buffer change rate is determined with every video quality fluctuation, which is then used to select the next quality level. The proposed system achieves a smooth bandwidth measurement to improve the video quality with a lower number of quality fluctuations.

Elgabli et al. [180] considered two paths for SVC-based prioritized adaptive video delivery. The segments belonging to each layer can be transferred from one of the available routes based on the quality, chunk deadlines, and path preferences. The proposed model tries to achieve maximum playback quality and smoothness between

chunks of different layers. With real bandwidth traces, the proposed solution has been shown to outperform different throughput- and buffer-based adaptation algorithms implemented using MPTCP. However, [179, 180] did not consider applying maximum contribution on each path.

Silva et al. [181] examined the Round-Trip Time (RTT) of each subflow and then delivered the prioritized Augmented Reality (AR)/Virtual Reality (VR) content over the lowest RTT subflow to improve transmission performance in single- and multi-homed environments. The proposed solution performs the subflows selection based on the linear regression analysis of each subflow in terms of latency measurements and utilizes the delivery feature of MPTCP. For each transmission interval, the priority packets are always transmitted over the least latency subflows. The proposed system is implemented in Network Simulator 3 by modifying the existing MPTCP NS3 implementation [182]. However, the real-time transmission of prioritized AR/VR content using MPTCP requires additional proof of concept.

To overcome the limitations of a single interface for seamless HAS, **Go et al. [183]** considered networking cost constraints to schedule all the segments in a block with the same selected video bitrates across multiple networks. The proposed system systematically calculates the video quality, the number of segments, the amount of data on each network, and the request-response interval during each adaptive interval. Moreover, the proposed client dynamically adjusts the quality allocations based on the battery performance and the networking cost to balance the perceived video quality and energy consumption. The experimental evaluation of the MPEG-DASH-based streaming strategy in an Android mobile device under WiFi and cellular networks provides seamless video playback with low energy consumption. However, they did not analyze the impact of perceived video quality.

Evensen et al. [184] extended the HTTP-based streaming system, called DAVVI, to achieve multi-channel support over 3G and WiFi networks. The video segments are dynamically divided into subsegments based on the quality of each channel so that the maximum load can be applied to each channel. Using multiple network interfaces for multimedia delivery requires sophisticated mechanisms for path quality measurements and data scheduling to avoid packet loss and out-of-order delivery

issues that can adversely affect user QoE. However, the existing solutions are limited in terms of measuring the real-time information of the paths due to the highly dynamic and complex nature of heterogeneous wireless networks.

Multipath-based adaptive video streaming improves the performance of time-sensitive multimedia applications. However, these solutions must try to avoid out-of-order or asynchronous transmission for multiple concurrent streams. In addition, the bandwidth estimation and bandwidth allocation should be in favor of the adaptive client. For this purpose, the path selection strategy should be carefully designed to achieve the immediate and high-quality availability of chunks for different content types.

3.2.2 Multi-view Video Streaming Solutions

Delivering multiple video streams with different priority levels over existing networks is challenging when trying to achieve high user-perceived quality. Many solutions for the adaptive delivery of multi-view video content have been proposed. **Zhang et al. [185]** presented a priority-aware two stream-based adaptive solution that uses different video bitrates for each stream. The proposed scheme implements an integrated bandwidth approach to enable a higher video bitrate for the high-priority stream and terminates the low-priority stream when there is insufficient bandwidth. A unified segment scheduling approach is proposed to decide the request time for each segment for achieving the synchronization among different priority streams. An independent playback buffer for each video stream enables the out-of-order delivery problem. The multi-view streaming solution has been shown to achieve a higher bandwidth utilization than the single stream-based adaptive streaming solutions [41, 167]. However, only a single path configuration is adopted in this work.

Yun and Chung [186] proposed a DASH-based streaming framework for multi-view videos, which includes a buffer-based server-push scheme and a parallel transmission mechanism to lower the switching time between different transmitted views. A view switcher module detects the navigation events at the client-side and sends the signals to the buffer controller and segment schedular. The buffer is drained quickly, and a parallel transmission scheme is then applied to fill it with updated

views with the maximum allowable bitrate budget. However, this work does not illustrate the impact of view switching distortion on perceived video quality.

Zhang et al. [187] proposed a navigation-aware multi-view video system to reduce the view switching time and navigation distortion for interactive multimedia applications. Different views recorded from different cameras are pre-encoded into a different set of bitrates. According to the priority of each view, the client selects the best possible quality levels within the navigation window. These solutions [187, 186] mainly work in two domains: 1) view adapt, 2) bitrate adapt, where the client downloads only a single stream at a time. Employing SVC-based coding for such solutions [186, 187] can further reduce view switching delay by constantly streaming the base layer for all recorded views.

Bilal et al. [188] proposed a cloud-based multi-view broadcasting solution where the users can experience the recorded events from multiple viewing angles. The proposed system identifies the user interaction and delivers the requested transcoded views with the best possible resource allocations, subject to computational and communication constraints. The authors proposed a fairness-driven heuristic approach to achieve this optimization. However, this work focuses only on scheduling and resource allocation. An important research challenge is how the different users' watching behaviors impact the overall system's performance.

Rahman and Chung [11] introduced a HAS-based multi-view conference streaming solution where multiple streams of the presenter, audience and presentation screen are transmitted concurrently over multiple paths. The proposed scheme assigns equal priority levels to all three streams. It employs a unified bandwidth approach to assign a unified quality to the segments of all the streams. For each segment of the multiple streams, the path is decided by considering the network throughput and bitrate of the segments. Unfortunately, this work does not consider the priority of the different streams which is crucial for multiple video concurrent video streaming.

Taking advantage of multipath network characteristics and priority features [180, 185] for rich multimedia content can substantially improve the streaming performance. The priority-based solutions either adapt the bitrate according to direct

user interactions [186, 187, 188] or consider a predefined priority level for streams [185]. The solutions [186, 187, 188] covered in this section consider a single file packaging system for multi-view videos. In contrast, [185, 11] adopts an individual file packaging system for each stream and therefore requires a separate buffer space for each stream. For a single file packaging system, there is no concern about out-of-order delivery; however, for individual file packaging systems, this is the main challenge. Therefore, the multi-view streaming system should try to achieve accurate bitrate estimation and allocation to multiple streams along with synchronous transmission over-employed network channels.

3.3 Adaptive 360° Video Streaming Solutions

In the present information era, there is increasing use of 360° video streaming services. However, streaming redundant information outside the viewport wastes significant network resources. The bitrate requirements become even harder to meet when the same 360° content is streamed to multiple users over the bandwidth-constrained networks. Streaming techniques for 360° videos have progressed from a full-view equal quality (viewport-independent) mode to a full-view non-equal quality mode (viewport-dependent) or equal-quality FoV-only mode. Adaptive 360° video streaming leverages the HTTP Adaptive Streaming (HAS) framework for bitrate adaptation. Next, the most representative streaming schemes for 360° videos are discussed.

3.3.1 Viewport-Independent Streaming Solutions

Viewport-independent streaming is the most straightforward way to stream 360° content because the whole frame is streamed in an equal quality similar to traditional videos. The 360° sphere is projected/mapped using viewport-independent projection formats, e.g., ERP or CMP. The projected video after encoding is transmitted to the 360° client that does not require any orientation information from HMD sensors. The receiving terminal should be able to support the projection formats. Following this, the DASH client performs bitrate adaptation similar to a traditional video, i.e., the representations of the same projection format for upcoming segments are

requested based on network characteristics. Hence, a minimal DASH extension (e.g., projection-related metadata) is required to support equal quality streaming.

Afzal et al. [189] performed an experimental characterization of 2285 360° YouTube videos across 14 categories by directly fetching the complete frame regardless of the current viewing direction of a user. Through video characterization examination, the authors showed that 360° videos have less motion and lower bitrate variations in comparison to traditional 2D videos. Moreover, there are significant opportunities to reduce the transferable data size and bitrate by employing the viewport-aware transmission.

360° video or cinematic VR is having an increasing presence in humanitarian presence due to its low-cost production [190]. Many public service organizations have already seen the potential of this format, evidenced by the creation of applications or simply by the rate of consumption of 360° videos. **Pérez-Seijo and García-Orosa [191]** employed 360° video storytelling in organizational communication. The authors conducted a diverse study on migrant crises with the help of different NGOs by employing five 360° videos. The authors found that the immersive storytelling features of the 360° videos can promote the social awareness mission of NGOs to bring societies closer to realities.

360° video is viral in worldwide sports. 360° video adds numerous advantages to almost every sport. The omnidirectional capturing around an athlete or sports ground brings the most beneficial experiences by providing the users with new viewing angles. **Gänsluckner et al. [192]** presented a blended learning-based 360° climbing course of providing the participants with an interactive learning experience. The authors showed that for learning climbing techniques, 360° videos are much better than regular videos. **Hebbel-Seeger [193]** considered the use of 360° videos for education and training processes in sports. The authors determined that 360° video has high potential in the athletic training process, and there is a need to adopt modern storytelling means to attract user attention.

Viewport-independent streaming is also employed for technology-enhanced learning. In education, such streaming techniques help to present the scenarios that are complex to describe with conventional videos, words, or even images [194]. 360°

recording of a classroom is thought to be a compelling tool for pre-service teachers in exploring different activities performed by students. 360° video supports innovative activities in entertainment, medical, and news production fields. The traditional adaptive video streaming solutions presented in Section 3.2 can be straightforwardly applied for viewport-independent streaming. The implementation simplicity has become a pleasant introduction to viewport-independent streaming. However, it has 30% less coding efficiency compared to viewport-dependent streaming [195]. Moreover, it requires extensive bandwidth and decoding resources for invisible areas. These resources could be saved and adequately used for visible content by employing viewport-dependent streaming.

3.3.2 Viewport-Dependent Streaming Solutions

In viewport-dependent streaming, the end devices receive only certain video frame areas, which contain the visual information equal to or greater angle of the viewport. The end devices should detect the related viewport in response to the user head movement and send the interaction signals to the cloud or edge servers to precise the player information. Such solutions are adaptive as they dynamically perform their area selections and quality adjustments, reducing the transmitted bitrate during 360° video streaming. In this regard, several adaptation sets associated with the user's orientation are prepared at the server-side. A client decides which adaptation set to fetch according to the network and estimated viewing position. Facebook predefines five viewport versions separated by about 30° for the entire 360° video [82]. The main benefit of this scheme is that the transmitted versions can be decoded by employing a single decoder on the client-side. However, these adaptive solutions require smart mechanisms of viewing region identification, synchronization with user head movement, quality adjustment, etc., to provide a smooth playback experience. Several works [35, 31, 196] focus on providing better coding efficiency and resource management without affecting the viewport quality.

Sreedhar et al. [35] implemented and compared multi-resolution variants of ERP and CMP projections and the existing variants of pyramid projection, e.g., rhombic pyramid, square pyramid, and truncated pyramid. The coding performance of these

projection schemes is evaluated by using eight different motions and different spatial resolution test sequences. However, the length of the test sequences is very short (e.g., 49 frames). The test frames are encoded using four different QP values, i.e., 22, 26, 30, and 34. The results showed that the proposed multi-resolution variants for viewport-dependent projection schemes give the best RD performance compared to the pyramid formats.

Corbillon et al. [196] proposed a geometric layout to produce different quality versions for viewport-dependent streaming. The authors employed the Quality Emphasized Regions (QERs) approach to optimize the display quality when a limited number of representations are available to stream. Compared to the traditional approaches, the server generates multiple representations not only based on the bitrate but also based on the underlying QERs. The center of each QER is called the quality emphasis center, where the surrounding pixels are encoded with higher bitrates compared to the far-away pixels. The client requests representation for the next segment for each possible viewing orientation. In case of a viewport mismatch, the client can still observe the low-quality viewport representation instead of observing black dots. They further investigate the impact of different projection schemes, i.e., equirectangular, cubemap, pyramid, and dodecahedron. They showed that cube map projection offers the best quality arrangements on the video quality for the given bitrate budget. Moreover, they observed the impact of segment duration on 360° video and showed that for a shorter segment duration, i.e., 1s to 2s, the user could experience the full video quality three-quarters of the streaming time.

Biatek et al. [197] presented a scalable coding technique that extracts a 2D view from the entire 360° view and encodes it as High-Efficiency Video Coding (HEVC) base layer to be compatible with legacy Integrated Receiver/Decoder (IRD). After geometrical transforms, the base layer is turned into a reference picture to encode the enhancement layer. The proposed design involves real-time parallel decoding implementation using OpenHEVC. The viewing area was divided into the center and adjacent tiles based on the eye-tracking results. The tiles were arranged using a The authors show that the scalable coding achieves up to 15% coding gain and can be integrated with a real terrestrial broadcast environment.

Zhou et al. [31] analyzed different projection schemes for viewport-dependent adaptive streaming using Oculus HMD. They compared the standard cube map projection with the Offset cube map projection used by the Oculus. Offset cube map projection is a direction-oriented projection scheme that allocates more pixels towards the direction of viewing. This strategy could enhance visual quality by consuming 5.6% to 16.4% less video bitrate. For 360° video streaming, the Oculus framework adapts the size and quality of the viewport based on the available network resources and future viewing positions. This two-dimensional adaptation strategy could download over 57% of additional chunks spending 20% extra network bandwidth compared to an ideal downloading procedure.

The equal-quality streaming of the entire spherical frames is not a smart idea due to limited network resources. **Naik et al. [198]** performed two subjective experiments by smartly employing asymmetric qualities for both background and foreground views of stereoscopic videos. They used different video sequences and QPs settings for both experiments. They employed different sizes for the foreground and background regions in both experiments. The experimental evaluations show that a differentiated quality streaming approach could save up to 15% and 41% bitrate for both background and foreground tiles, respectively.

In viewport-dependent adaptive streaming, the client performs adaptation based on the network characteristics as well as the viewing orientations of the user. Therefore, the DASH manifest should also include the viewing position information in addition to the projection metadata. The multi-view adaptive bitrate selection solutions described in Section 3.3.2 can be applied for viewport-dependent streaming with modifications. The viewport-dependent streaming techniques have substantial storage requirements because several different content versions are stored at the server-side to support a specific suitable viewing angle. Moreover, these approaches seem to be less cache-efficient and need resource-intensive encoding, which can be expensive, particularly for commercial and live streaming services. When the users change their viewing position, the viewport resolution can be adjusted by replacing the old variant with a new high-resolution option only at the next available Random Access Point (RAP). Accordingly, there is a need to optimize the RAP period to

optimize the viewport resolution adaptation. Furthermore, low latency and active viewport switching are necessary when considering viewport-dependent streaming.

3.3.3 Tile-based Streaming Solutions

In traditional HAS, any video is segmented into small chunks for adaptive transmission to utilize the available bandwidth wisely. In 360° videos, the chunks are further partitioned into equal/non-equal rectangular tiles to precisely adjust the quality of the viewing tiles. The exiting tile-based adaptive solutions were categorized in six wide categories: *surveys and studies on 360° video streaming*, *fixed viewport-based adaptive 360° video streaming solutions*, *marginal region-based adaptive 360° video streaming solutions*,*extended viewport-based adaptive 360° video streaming solutions*, and *dynamic tiling-based adaptive 360° video streaming solutions*.

3.3.3.1 Surveys and Studies on 360° Video Streaming

Several useful surveys and studies exist, which try to elaborate on in-depth technical aspects from capturing to display and QoE assessment for 360° videos. **El-Ganainy and Hefeeda [199]** discussed the different representation formats, QoE models, and streaming solutions to overcome the bandwidth limitations for VR content. The authors surveyed uniform and variable quality mapping approaches for 360° videos. They highlighted the encoding performance of different tiled/RoI-based streaming solutions. However, the discussed partial and full delivery streaming systems do not involve the proper orientation prediction mechanisms.

Different from the content acquisition and actual display, **He et al. [200]** focused on reviewing some potential network-related challenges in AR/VR streaming. The authors discussed viewport prediction approaches, resource allocation, compression overhead optimization approaches, QoE-aware and bandwidth optimized viewport-dependent solutions, and network-assisted solutions. However, this work covers only a few works in respective domains.

Xu et al. [201] surveyed various relationship aspects between 360° content perception, compression, and quality assessment from the end-user perspective. The authors covered human visual attention-based datasets for 360° content and presented

several heuristic-based and data-driven approaches for user attention modeling. For compression overview, both the spherical and visual attention-based approaches for 360° are presented. In addition, they overviewed different subjective and objective quality assessments and coding techniques for 360° content. However, this work does not provide comprehensive future works on visual attention modeling, quality assessment, compression, etc. Moreover, it does not consider the network-assisted adaptation and rendering of the 360° video.

Zink et al. [202] provided a short survey on 360° video streaming systems with a focus on content preparation, distribution, and content experience techniques. The authors looked at DASH-based representation, projection schemes, cinematic rules, and multiple camera capturing and storage techniques for content preparation. They discussed application-level multicast techniques for efficient delivery of 360° videos in an edge-assisted 360° data distribution network model. Finally, they provide a general review of the QoE evaluation to promote the cybersickness-free viewing experience. However, they did not offer the impact of viewport prediction approaches for low-latency 360° videos.

d. A. Azevedo et al. [203] reviewed the visual artifacts that modify the spherical video signals during different stages, i.e., content acquisition, encoding, transmission, decoding, and consumption, in an end-to-end 360° streaming framework. The authors described the most common visual artifacts in terms of four categories, i.e., spatial, temporal, stereoscopic, and navigation artifacts. They explained how these distortions impact the overall visual quality assessment in both subjective and objective domains. The quality assessment approaches that consider the spherical representation and the underlying geometry can avoid several compression distortions. They further indicated that a new quality metric should be designed by having a clear view of the different discomfort levels in 360° content.

Table 3.1 includes some of the recently proposed surveys on state-of-the-art 360° videos. The studies presented here mainly focus on opportunities and challenges related to content preparation, compression techniques, quality assessment, and AR/VR applications. However, network-assisted cooperative and live streaming opportunities and immersive streaming standards and technologies are not covered

Table 3.1: Summary of Related Surveys

Works	Year	Scope
[199]	2016	Focuses on streaming of 360, panoramic, and multi-view videos
[200]	2018	Targets underlying network-related issues to AR/VR systems instead of display and streaming strategies
[201]	2019	Considers both 360°image/video with a focus on processing, quality assessment, and compression techniques
[202]	2019	Deals with the different technologies and challenges considering 360° data model
[203]	2019	Analysis on visual distortion in 360° videos

in detail. Moreover, the viewport prediction techniques and adaptive tiles selection are not presented. In addition, the adaptive streaming concept for 360° videos is also not explored well.

3.3.3.2 Fixed Viewport-based Adaptive 360° Video Streaming Solutions

Fixed viewport-based tiling solutions mainly try to overcome the bandwidth utilization for massive video content by avoiding the higher quality rendering of the whole omnidirectional scene. In this streaming category, several solutions employ probability-based, layers-assisted, straightforward machine learning, or network-assisted techniques to facilitate a smooth and higher-quality streaming experience.

Hosseini and Swaminathan [49] proposed a priority-based adaptation algorithm by dividing tiles into three zones, Z_1 (viewport center tile), Z_2 (surrounding tiles), and Z_3 (background tiles). The authors constructed a 3D geometry with six 3D meshes to smoothly represent the tiles in 3D space. The proposed algorithm firstly assigns the lowest quality to all the tiles. Then, it adjusts the central tile's quality to the highest level and repeats the same procedure for the surrounding and background tiles while respecting the available bandwidth budget. The authors presented a hierarchical resolution degradation for encoding five 360° videos into four quality levels. The probability-based streaming results in a bandwidth saving of 72% compared to a baseline approach, as shown by the experiments. However, the presented VR setup provides a 2K viewing resolution. Moreover, the videos were encoded using H.264/AVC encoder, which can decrease the coding performance for 360° videos.

Chen et al. [106] introduced different priority levels within the viewport, i.e., the highest priority for the center tile, medium priority for the top, bottom, left, and right tiles, and the lowest priority for the four diagonal neighbor tiles. Different encoding versions are assigned to the viewport tiles based on their priority levels to reduce the bandwidth and storage capacity. However, no viewport prediction mechanism was considered in this work, and the evaluations were performed using fixed network connections.

Hooft et al. [48] proposed the uniform quality selection approach for the fixed viewport region named Uniform ViewPort quality (UVP). UVP classifies the tiles into the viewport and non-viewport regions. It initially selects the lowest resolution for all the tiles; then, it uniformly increases the quality of the viewport and background tiles while respecting the available bandwidth budget. The tiles are arranged based on a spherical walk prediction approach that extrapolates the head motion trajectory as the walk on the spherical surface to predict the upcoming viewing points. However, the proposed solution was tested for only three videos.

Xie et al. [164] proposed an optimization framework for tile-based streaming to minimize the pre-fetched tile error and improve the smoothness of tile borders with different associated bitrates. The authors proposed a viewport orientation prediction model that employs Linear Regression (LR)-based [105] orientation prediction and Gaussian distribution based prediction errors in yaw, pitch, and roll directions to compute the viewing probability of each tile on the sphere. Two QoE functions are defined with the objectives to minimize the expected quality distortion ($\Phi(X)$) and spatial quality variance ($\Psi(X)$) of the viewport when considering the viewing probability of the tiles. These functions are defined as follows:

$$\Phi(X) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} D_{i,j} \cdot x_{i,j} \cdot p_{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j} \cdot s_i} \tag{3.5}$$

$$\Psi(X) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j} \cdot p_i \cdot (D_{i,j} - s_i \cdot \Phi(X))^2}{\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j} \cdot s_i} \tag{3.6}$$

A target-buffer-based adaptation approach is proposed to enable a smooth playback under short-term viewport prediction. At adaptation step k, when the kth set of

segments are downloaded completely, the buffer occupancy b_k is given by:

$$b_k = b_{k-1} - \frac{R_k \cdot T}{C_k} + T \tag{3.7}$$

where R_k is the bitrate, T is the segment duration, and C_k represents the network capacity. To prevent running out of chunks, the buffer occupancy is controlled by setting a target buffer level B_{target}, that is, $b_k = B_{target}$. The average spatial quality variance is 0.97, which is smaller than other tile-based strategies. The proposed probabilistic adaptive framework achieves around 39% gains on perceptual quality with 46% on average lower spatial quality variance. However, the authors tested their streaming solution for only one video sequence with 2K resolution.

Based on the given viewport data, **Ghosh et al. [47]** encoded the visible tiles in higher resolution while the rest tiles were in lower resolution according to the time-varying network constraints. The authors designed a low complex solution to reduce the probability of watching low-quality video parts. Two straightforward baseline algorithms are considered for performance comparison; the first algorithm simply downloads the entire segment in uniform quality, while the second algorithm downloads only the Field of View (FoV) tiles in the highest possible quality levels. The numerical evaluations show that streaming variable quality levels for the visible and non-visible regions can boost the performance in terms of formulated QoE metric by up to 20%. However, no practical experimental evaluation was considered for the proposed framework.

Scalable video coding is an alternative strategy to achieve viewport adaptation. The base layer is always required and can be pre-fetched at the client-side to avoid re-buffering events. Enhancement layers up-sample the viewport and can be requested when sufficient bandwidth is available. Moreover, SVC facilitates efficient in-network caching support to reduce the distribution cost when multiple clients request the same content [204]. **Nguyen et al. [205]** proposed SVC-based streaming using an HTTP/2 network protocol to overcome the randomness of both the network channels and head movements. In case of significant bandwidth drops, the proposed system utilizes the stream termination feature to terminate the delivery of the ongoing tile.

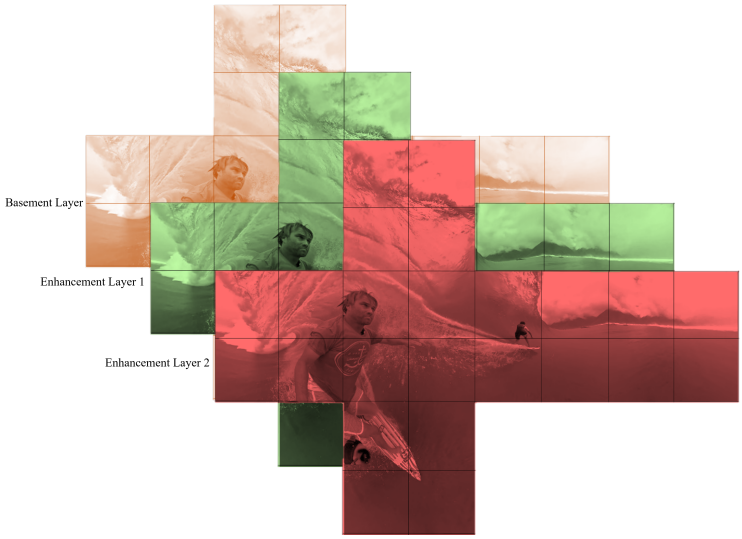


Figure 3.3: Cubemap projection-based scalable video encoding for 360° videos [2].

The quality reassignment feature of the proposed system reallocates the tiles' quality to deal with the viewport estimation errors. The proposed throughput-aware late tile termination and viewport-aware tile layer updating features can improve the viewport quality by 17%, as demonstrated by the experiments. However, only two head movement traces were employed in the experiments.

The quality expectations for the 360° video are similar to those for the traditional HAS, i.e., no stalls at all if possible [206]. **Nasrabadi et al. [2]** employed a cubemap projection-based scalable video coding scheme to solve the rebuffering issues for 360° video streaming. Each face of the cube was divided into two horizontal and two vertical tiles, and encoding was performed using one base layer and two enhancement layers, as shown in Fig. 3.3. The viewport prediction is performed using a weighted linear regression approach [105]. A buffer-based adaptation logic is then proposed to reduce the impact of rebuffering events. The experimental evaluations using four streams of different spatiotemporal complexities demonstrate that compared to the non-scalable coding, layer-assisted tiles coding results in fewer quality fluctuations and rebuffering events while offering improved quality. However, this method can suffer from severe quality fluctuations because it does not involve any mechanism to deal with viewport prediction errors.

Song et al. [103] proposed a two-layer (i.e., BL and EL)-based fast-FoV switching strategy for 360° video streaming to provide a persistent QoE to the end-user.

The proposed system includes fast view-switching, tiles quality merging, and pre-download mechanisms to quickly adapt to the user's movements, buffer level, and bandwidth fluctuations. An open Group of Picture (GoP) technique is proposed for merging tiles into a composite stream to reduce the decoding time at the client-side. The authors showed that during quick head rotations, the proposed solution is able to improve the availability of high-quality chunks by up to 90%. However, no optimal viewport prediction solution was considered in this work.

Ogasawara and Bandai [207] proposed tiles popularity-aware buffer-based adaptation algorithm using the hierarchical structure of SVC over Named Data Networking (NDN). The proposed client initially fetches the base layer for all the tiles during each predictable interval and then gradually updates to the enhancement layer considering the current buffer level. The major drawback of this work is that it does not involve any viewport prediction algorithm and updates the bitrate based on the allowable bandwidth budget. Moreover, the fixed network evaluations were performed for only one 60s long video clip.

Son et al. [208] proposed a Motion-Constrained Tile Set (MCTS)-based HEVC and Scalable High Efficiency Video Coding (SHVC) codecs for independent tile extraction and transmission in viewport-based mobile VR streaming. The proposed encoder generates a bitstream at the server-side that contains a base layer for the whole frames and an enhancement layer for certain frame areas. The proposed extractor extracts the relative RoI bitstream to present to the user on the client-side. The proposed system lowers the computation complexities on the decoder while achieving more than 47% bandwidth saving. However, the proposed design underperforms compared to the original HM and SHM encoders because MCTS restricts the temporal motion information.

The transmission of high-resolution content to multiple users should try to balance the expensive bandwidth, minimal latency, and high transmission reliability requirements. **Zhang et al. [204]** introduced a cooperative streaming scheme using the SVC quality adaptation methodology to improve the bandwidth sharing among multiple users watching the 360° content in a proximity MANET environment. The proposed heuristic approach selects the optimal subsets of tiles based on the proba-

bility of being watched and the aggregated group-level preference while meeting the constraints of the available network resources. However, the proposed solution does not consider optimizing resource allocation in a wireless multicast environment.

Ahmadi et al. [209] introduced a DASH-based weighted tile approach to optimize the coding performance of tiles requested by the subgroups of users. The proposed multicast streaming solution assigns appropriate weights to the tiles based on the probability of being watched by the users. It then selects the bitrate of tiles for each subgroup, respecting the available bandwidth and tile weights. The authors evaluated the proposed solution using 23 videos prepared in a 4x8 tiling pattern. However, there could be substantial spatial quality variations due to the per tile quality allocation, leading to poor streaming performance. Additionally, the discrete optimization problem is needlessly large and may not guarantee positive performance.

Kim and Yang [210] extended the MPEG-DASH Spatial Relationship Description (SRD) feature to lower the view switching delay for high resolution 360° content. The authors designed and implemented a multi-layers rendering enabled 360° VR player based on their previous work [211] to support high-resolution and low latency streaming for highly unpredictable head motion data. The player enables the parallel rendering of 16K/60 FPS content using General-Purpose Computing on the Graphics Processing Unit (GPGU). The proposed solution can achieve a view switching latency of 450ms for 16K 360° videos by implementing the parallel decoding of HEVC bitstream. However, this work does not involve any view prediction mechanism for the practical system implementation.

Machine Learning (ML)-based approaches for traditional video streaming [40, 212] effectively adjust the video bitrate and achieves long-term QoE rewards. Different from the traditional video content, 360° video includes several new aspects such as spherical layout, viewport prediction, capturing to display distortions, etc. Applying existing ML adaptation policies directly to 360° video streaming may lower streaming performance. **Fu et al. [213]** proposed a sequential reinforcement learning approach for 360° video streaming, called 360SRL, that makes adaptation decisions based on the several states instead of solely relying on estimated bandwidth. 360SRL extract multiple features, i.e., available throughput, buffer level, sizes of next tiles,

qualities of previous tiles, visible probability of previous and next tiles, and size and quality level of the last segment, by employing Convolutional Neural Network (CNN) and fully connected layers to decide the quality reward. 360SRL uses a tile-based streaming simulator to boost the training phase. The trace-driven evaluation demonstrates that 360SRL outperforms baseline adaptation approaches by achieving 12% QoE improvement. However, this work does not highlight the impact of different segment durations. Moreover, the viewport traces [214] and video content [215] were used from different datasets.

Zhang et al. [216] proposed a DRL model for 360° videos, named DRL360, that dynamically learns to allocate the bitrate for the future set of tiles using ACTOR-CRITIC (AC) network considering viewport prediction accuracy and network conditions. DRL360 employs a Long-Short Term Memory (LSTM) model for viewport and bandwidth prediction based on historical information. The proposed system aims to optimize multiple QoE objectives, i.e., average visual quality, rebuffering events, and average viewport temporal variations. Real-word and trace-driven evaluations show that the proposed scheme adapts well to a broad set of dynamic features and offers a 20% to 30% improved QoE reward compared to the legacy methods. However, the QoE metric does not include the impact of spatial quality variations inside the viewport.

Emerging immersive and interactive user applications with higher bandwidth, QoS, and computing requirements, are among the applications which would benefit the most from the 5th Generation (5G) networks [217]. The traditional network architecture with centralized cloud-based computing and storage is inadequate for real-time high-bitrate content delivery. Edge caching and Mobile Edge Computing (MEC) are regarded as pivotal enablers for 360° video services [218]. The massive video content can be transferred to the edge nodes and downstream clients to meet the high-resolution levels and stringent latency requirements by managing short-range communication. In edge computing [219, 92], the processing and storage tasks are shifted from the core network to the edge nodes such as Base Stations (BSc), cloudlets, micro data centers, set-top boxes, headsets, etc., with significant advantages in comparison to traditional networks.

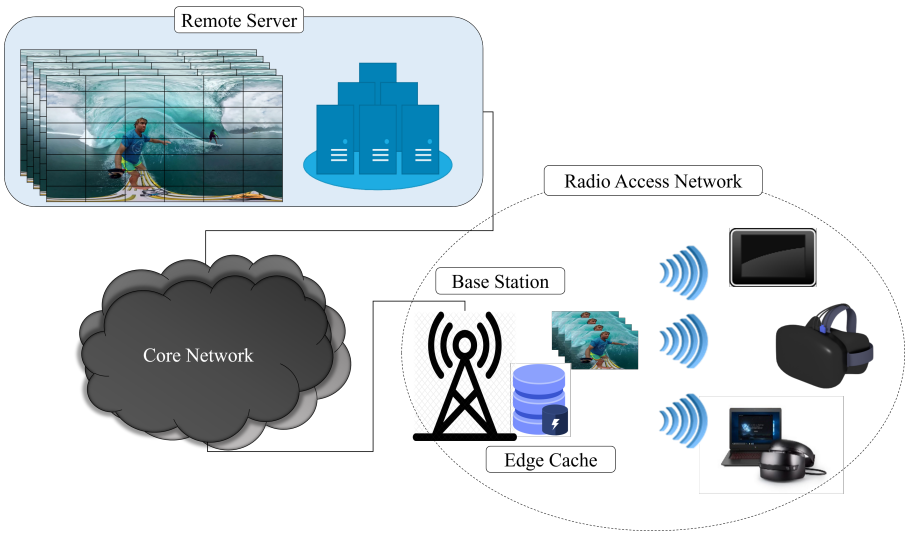


Figure 3.4: FoV-aware edge caching architecture for tile-based 360° video streaming [3].

In contrast to [220], **Lo et al. [221]** considered end-terminal heterogeneity, i.e., HMD with PC, HMD with a mobile phone, and standalone HMD, for the edge-assisted rendering of 360° video. In the proposed system, the optimization algorithm dynamically decides which client should be served by edge nodes according to the video quality, HMD types, and bandwidth constraints for the enhanced QoE of VR users. The edge server transcodes the HEVC tiles stream into the viewport video stream and transmits it to multiple clients. The performance is evaluated in terms of bandwidth consumption, system latency, Viewport PSNR, and power consumption. The powerful GPUs with more detailed optimization for the proposed system can improve rendering and streaming performance.

Caching solutions for conventional video content [222, 223] cannot be readily deployed for caching of 360° video. To facilitate the transmission of 360° video in an edge-cache-enabled network, a proxy cache between the two transmission ends is deployed to improve the content available. Edge caching can substantially decrease duplicate transmissions and make the content servers more scalable [224]. **Mahzari et al. [3]** introduced a popular content (e.g., FoV) caching policy for 360° video based on the watching behaviors of other users. A probability model is employed to collect common FoV patterns from different users, which are then added to the local cache to minimize the response time. Fig. 3.4 illustrates an FoV-aware edge

caching design deployed in this work. Experimental evaluation with open-source head movement traces of 156 users [225] shows superior performance in terms of cache usage when compared to Least Frequently Used (LFU) and Least Recently Used (LRU) caching policies with at least 40% and 17% improvements, respectively. However, this work considers a Weighted Linear Regression (WLR) model for FoV prediction. Employing cluster-based viewport prediction (e.g., K-Nearest-Neighbors (KNN), Density-based Spatial Clustering of Applications with Noise (DBSCAN)) approaches can further improve the prediction accuracy for a multi-client streaming environment.

Similarly, **Papaioannou and Koutsopoulos [226]** proposed a tile resolution and demand statistics-based caching policy to improve the viewport coverage with the minimum error between requested and cached tiles versions. The proposed system systematically decided which tiles to cache at which resolutions in response to the viewing statistics. The experimental evaluation with different caching and transmission delays resulted in improvements in the cache hit ratio, especially for the layered-encoded tiles. However, an important concern is how the proposed system will organize the users' head movement traces in a multicast streaming scenario, where users could have similar or different tile occurrence statistics in the viewport. Moreover, the resolutions of employed videos were not stated in this work.

Zhang et al. [227] proposed an SVC encoding-based cache replacement strategy to improve the cache hit ratio in an edge computing environment. The cache value for each segment is calculated based on three factors: the popularity of the video chunks, FoV information, SVC layer of the video chunks. When the cache capacity exceeds the limits, the proposed system evicts the lower popularity chunks, non-FoV chunks, and higher-layer chunks from the cache. The proposed system replaces the chunks with larger data sizes when the cache is full. The cache replacement strategy has been shown to improve the cache hit ratio and bandwidth utilization for tiled video segments. However, this fixed viewport-based streaming solution was evaluated using a single tiling pattern (i.e., 4x6).

Fixed viewport-based adaptive streaming solutions facilitate the transmission of quality-variable tiles based on the user's interest without sacrificing picture qual-

Table 3.2: Summary of Fixed Viewport-based Adaptive 360° Video Streaming Solutions

Works	Design Type	Prediction Model	Dataset	Tiling Pattern	Resolution	Segment Duration	Experimental Duration
[49]	Probability-based	Quaternion-based	5 Videos, 1 Users	6 tiles	720p-4K	-	Video duration
[106]	Probability-based	-	5 Videos [215]	3x3, 4x4, 5x5	2K	1s	20s
[48]	Probability-based	Spherical Walk	3 Videos, 48 Users [225]	11, 22, 42, 44, 84, 86, 8x8, 1612/16	4K	1.067s	Video duration
[164]	Probability-based	LR	1 Video, 5 Users	6x12	2K	1s	3m
[47]	Probability-based	-	1 Video, 1 User	4x8	-	2s	4m
[205]	Layer-assisted	-	1 Video, 2 Users	24 tiles	2k	1s	70s
[2]	Layer-assisted	WLR	4 Videos, 5 Users	6 and 24 tiles	4K	32 frames	Video duration
[103]	Layer-assisted	Linear	3 Videos, 1 User	20-30 FoV tiles	8K	-	30s
[207]	Layer-assisted	-	1 Video, 10 Users [229]	25	4K	1s	1m
[208]	Layer-assisted	-	4 Videos	3x3	8K	-	10s
[204]	Multicast	Zigzag	1 Video, 3 Users [229]	8x4	4K	1s	91s
[209]	Multicast	-	23 Video, 200 Users	4x8	4K	1s	≤1m
[210]	Layer-assisted	-	6 Videos	20x15, 20x20	16K	-	Video duration
[213]	Machine learning	Forwarding projection	10 Videos [215], viewing traces [214]	3x3	4K	2s	1m
[216]	Machine learning	LSTM	16 Videos [225]	4x6	4K	1s	Video duration
[221]	Edge-assisted	-	10 Videos, 50 Users [215],	5x5	4K	2s	1m
[3]	Edge-assisted	WLR	10 Videos, 50 Users [215],	6x4	4K	1s	30s
[226]	Edge-assisted	-	4800 Videos, 70 users	6 tiles	-	1s	1m
[227]	Edge-assisted	Last known	5 Video, 10 Users [229]	4x6	4K	1s	30s

ity. Table 3.2 illustrates the most significant works employing fixed viewport for adaptive 360° video streaming. Most of the probability-based solutions [49, 48, 164], layer-assisted solutions [205, 2, 103, 197], machine learning solutions [213, 216], and network-assisted solutions [221, 228, 3, 227], focus on higher quality viewport availability regarding different user movements. However, such solutions may face severe quality degradation when tested for a longer segment duration. Moreover, defining variable quality levels within the viewport [49, 106] can lead to severe spatial quality oscillations even for the perfect prediction results. Therefore, designing solutions that introduce marginal areas around the user's viewport can absorb the discrepancy between observed and predicted viewports to support a promising VR experience. In addition, the higher uniform quality allocation to the viewport tiles and relatively lower uniform quality selection for the marginal tiles can provide efficient bandwidth utilization for both stable and fast motion content.

3.3.3.3 Marginal Region-based Adaptive 360° Video Streaming Solutions

It is very important to define the useful marginal mapping around the viewport to deal with inaccurate viewport predictions. Many tile-based adaptive schemes focus on how to define the marginal area and how to adapt the quality in relation to the

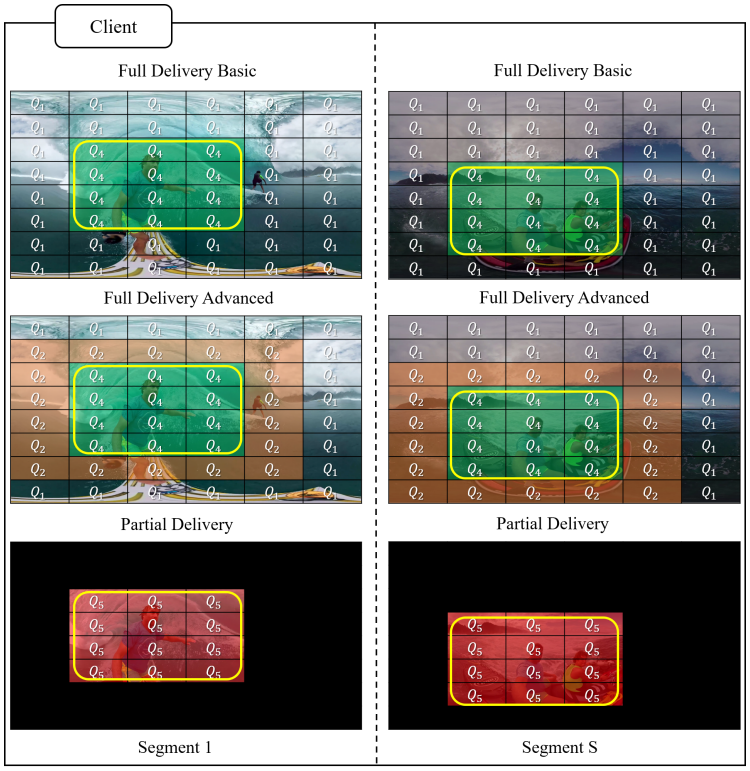


Figure 3.5: Different tile delivery solutions for viewport adaptive 360° video streaming [4].

viewport tiles.

Graf et al. [4] investigated bandwidth savings for 360° video using three tile delivery options, i.e., full delivery basic, full delivery advanced, and partial delivery, also illustrated in Fig. 3.5. The viewport and non-viewport tiles are requested in the highest and lowest possible qualities in full delivery basic technique. In the full delivery advanced approach, the tiles surrounding the viewport are requested at a higher rate than the far away tiles. In a partial delivery system, only the highest quality viewport tiles are streamed for each segment. The evaluations of these three schemes by employing five different tiling patterns, e.g., 1x1, 3x2, 5x3, 6x4, and 8x5, against monolithic streaming show that the full delivery basic streaming achieves a bandwidth saving of around 65% compared to full delivery advanced and partial delivery strategies under different bandwidth settings. Moreover, a 6x4 tiling scheme implements a valuable trade-off between bandwidth consumption and coding efficiency. However, only two video clips with a static viewport were considered for the evaluations.

Petrangeli et al. [107] proposed a region-based adaptive streaming framework involving tiles quality selection and viewport prediction modules facilitated by a network module to improve the streaming performance of tiled videos in a high latency network environment. The authors divided the tiles of the equirectangular frames into three regions (i.e., viewport, adjacent, and outside) and assigned variable bitrates to the visible and non-visible regions by incorporating the viewer's head movements. The viewport traces were collected using the Gear VR framework for one 360° video watched by ten users. A linear extrapolation of the recent (100ms) watching the history of the user was employed to predict the future fixation point. The authors considered three encoding levels and three segment durations for tiled and non-tiled versions of the same video. The client sends a single call for a segment, and the server transmits the k tiles using a First Come First Served (FCFS) policy. With HTTP/2's priority feature, the high-priority tiles could be fetched urgently to improve the streaming performance. The urgent transmission of high-priority tiles facilitated by HTTP/2 can improve the throughput performance compared to HTTP/1.1 under variable network delay conditions. However, this work considered only 60s long single 360° video for the performance evaluation.

Ben Yahia et al. [230] divided the equirectangular frame into viewport, marginal, immediate background, and far background regions. The proposed model involves two viewport prediction intervals, i.e., before and during the delivery of the same segment. The client assigns variable weights to different priority regions and can update the resource allocation based on updated prediction results. Similar to [107], the authors employed HTTP/2 network protocol to facilitate the tiles' re-transmission. With HTTP/2, the server can terminate or re-transmit the upsampled priority region. The proposed client includes upgrade and downgrade quality algorithms to increase/decrease the quality of tiles based on updated prediction results. However, the updated prediction results are of no use for shorter video segments because the segments may have already been received. Moreover, the experiments were performed using fixed network bandwidth.

Zou et al. [231] introduced a CNN-based prediction mechanism and then distributed the communication resources for the quality selection of predicted tiles. The

proposed solution maps the spherical representation to the planer projection to calculate the viewing probability of each tile. The tiles are then divided into viewport, marginal, and background tiles groups. The marginal tiles surround the viewport in all directions, similar to [107]. A server-side bitrate adaptation mechanism then assigns higher quality to the marginal tiles than the invisible tiles to minimize the quality distortion of the viewport and marginal parts. The CNN-based viewport prediction model outputs the lower prediction error for a prediction window set to 1s, and the proposed solution achieves a higher streaming performance.

Jiang et al. [232] leveraged Reinforcement Learning (RL) for the bitrate selection of viewport, marginal, and background tiles based on historical bandwidth, buffer space, tile sizes, and viewport prediction errors. The architecture of the proposed system consists of state buffer, viewport prediction (VPP), and tiles bitrate selection (TBS) agents. The state buffer provides the user viewing patterns and network states to the VPP and TBS agents. The VPP agent then estimates the next viewport position by employing an LSTM model. The Asynchronous Advantage Actor-Critic (A3C) [233] algorithm trains the TBS agent to perform suitable bitrate decisions. However, this work was evaluated using one 360° video prepared in a 12x6 tiling pattern and 1s segment duration.

Nguyen et al. [234] analyzed the influence of adaptation interval delay and buffering delay on viewport-dependent adaptive streaming. The authors proposed a server-side bitrate computation strategy to minimize the impact of response delay. Following the client's response, the proposed system estimates the available network throughput and the future viewport position. The decision engine on the server-side then streams the viewport and marginal tiles to meet the delay constraints. The experimental results reveal that small adaptation and buffering delays are inevitable for viewport-dependent adaptive streaming. However, this work involves only a single 360° video prepared in an 8x8 tiling pattern watched by a single VR user. In another work, **Wang et al. [235]** proposed a new adaptation mechanism that dynamically decides the viewport bitrate considering both the predicted points and viewport errors observed during the previous segment. The proposed system includes some marginal tiles to limit the viewport quality loss events. Depending

on the viewport's prediction performance, the marginal area could be around the viewport in all directions or towards some sides. The experimental evaluations with five different motion content demonstrate an increase in the viewport quality without acquiring excessive bandwidth utilization for non-viewport regions. However, the experiments were performed for only 60s long streaming sessions. In addition, similar to [234], this work considers a single viewport trace for each video.

Yuan et al. [236] focused on how to predict the viewing positions and how to select the best bitrate combination for different tiles groups. Before sending the GET request, the client anticipates the potential viewing patterns using a Gaussian prediction model. A simple and effective buffer-based quality-aware bitrate adaptation algorithm is also proposed to allocate different quality levels to the viewport, marginal, and outside tiles. In the case of fast viewport movement (e.g., viewport switch), a Zipf model is employed to reassign the qualities to different priority tiles. The experimental evaluations using three 4K test sequences prepared in a 6x4 tiling pattern under staged bandwidth variations show that the proposed solution favors the high visible quality levels with considerable navigation smoothness. However, very short duration (10s) simulations were performed for each video content.

Shi et al. [51] considered the high-quality rendering of 360° video without focusing much on viewport prediction. The proposed MEC-VR system employs a remote server to dynamically adapt the viewport coverage by using an adaptive cropping filter that adds some extra area outside the viewport according to the observed system latency. The latency-based viewport coverage adjustment allows the client to accommodate and compensate for abrupt head movements. The evaluation setup involves an edge node connected to an LTE network to allow a high refresh rate rendering for both dynamic and live VR content. However, this work does not consider potential viewport prediction errors for the FoV and marginal area selection for different content characteristics, which can significantly lower the performance.

Bao et al. [237] proposed a multicast framework based on motion prediction and channel conditions of concurrent viewers to deliver only the likely to be watched blocks of 360° video. A neural network model is used to predict the real-time motion of concurrent users. Then, the display area calculation is dynamically performed

for each user, and the server transmits the multicast and unicast versions of the same content to serve a number of users. The unicast and multicast areas are merged at the client-side to serve each independent user. The proposed client adds marginal tiles in both horizontal and vertical directions in response to the pixel loss ratio to increase the viewport match performance. However, the proposed solution does not consider optimizing resource allocation in wireless multicasting. Moreover, the authors performed the experiments by considering viewport motions known in advance, which does not reflect the actual multicast streaming framework.

Marginal region-based adaptive streaming solutions also presented in Table 3.3 involve extra high-quality pixels around the viewport to accommodate the variable viewing directions. However, one primary concern is how much and in which direction to include the marginal region for each adaptation interval under bandwidth-constrained networks. Several solutions [4, 107, 230, 231] simply employ a fixed marginal area around the viewport in all directions. It can compensate for the highly dynamic viewing nature of the user; however, a significant waste of the bandwidth can be observed under good prediction performance. Some more advanced solutions select marginal region based on viewport prediction performance for the single [235] or multi-users [237] environments. However, the bitrate of the marginal region could be increased, especially when employing larger resolution tiles. Another concern is the quality adaptation for the marginal area. Therefore, instead of the immediate lowest quality, a graceful/intermediate quality degradation can improve the bandwidth utilization in comparison to the non-marginal-based streaming solutions. Another possible approach could be selecting the weighted quality adjustment for the viewport and marginal regions depending on the number of tiles in these regions.

3.3.3.4 Extended Viewport-based Adaptive 360° Video Streaming Solutions

In extended viewport-based adaptive streaming, the viewport coverage is virtually extended to deal with different head movements. Next, several streaming solutions involving extended viewport concepts for single- and multi-user streaming environ-

Table 3.3: Summary of Marginal Region-based Viewport Based Adaptive 360° Video Streaming Solutions

Works	Design Type	Marginal Region	Prediction Model	Dataset	Tiling Pattern	Resolution	Segment Duration	Experimental Duration
[4]	Probability-based	Fixed	-	2 Videos	1x1, 3x2, 5x3, 6x4, 8x5	2K, 4K, 8K	1s, 2s, 4s	Video duration
[107]	Probability-based	Fixed	Position-based	1 Video, 10 Users	6 tiles	8K	1s, 2s, 4s	60s
[230]	Probability-based	Fixed	Generic	3 Videos, 3 Users [229]	6x4	4K	1s	1m
[231]	Probability-based	Fixed	CNN-based	3 Videos, 10 Users [238]	8x8	4K	1s	Video duration
[234]	Probability-based	Dynamic	Last known	1 Video, 1 User	8x8	4K	1s	60s
[235]	Probability-based	Dynamic	LR	5 Videos, 7 Traces	8x8	4K	1s	59.7s
[236]	Probability-based	Dynamic	Gaussian model	3 Videos, 1 Trace	4x6	4K	2s	10s
[232]	Machine learning	Dynamic	LSTM-based	1 Video, 59 Users [229]	12x6	4K	1s	70s
[51]	Edge-assisted	Dynamic	-	6 Videos, 1 Users [229]	Dynamic Cropping	4K, 8K	Live	60s
[237]	Multicast	Dynamic	Neural Network	16 Videos, 153 Users	Dynamic Cropping	4K	1s	Video duration

ments are discussed.

Hooft et al. [48] proposed a quality adaptation approach by considering the extended viewport (full-frame) region. The proposed approach mainly focuses on the quality enhancement of the center/viewpoint tile and is therefore named Centre Tile First (CTF). The proposed solution selects the entire frame at the lowest quality level when the available bandwidth is insufficient. If a sufficient amount of bandwidth is available, CTF increases the quality starting from the viewport center to the last tile. The proposed method was evaluated considering the weighted viewport quality, which assigns higher weights to the center tile quality and gradually lowers the weights towards the end tiles. It was shown to outperform the uniform viewport quality allocation solution UVP for the weighted viewport quality metric. However, when tested using average viewport quality, UVP performs better than CTF.

Kan et al. [239] designed a Deep Reinforcement Learning (DRL)-based adaptive streaming framework that adaptively adjusts the streaming policy based on the physical variables. The authors considered a scalable viewport concept where the entire frame is divided into multiple viewports based on the viewing probability of tiles. The proposed solution performs the bitrate decision with the A3C algorithm of DRL due to its effectiveness in making agents more and more intelligent. A uniform bitrate allocation is performed for the tiles belonging to each viewport. The performance evaluation reveals that the proposed systems ac QoE metrics, including average visual quality, average quality fluctuations, and rebuffering. However, a tile can be part of multiple viewports. Therefore, significant-quality variations can be observed for scalable viewport under dynamic network conditions.

He et al. [240] proposed a network response-based joint adaptation solution for the viewport coverage and viewport bitrate under dynamic and congested network conditions. The proposed algorithm firstly measures the RTT time and then computes the FoV size and allocates the necessary bitrate based on the delay measurements. NS-3-based simulation outcomes confirm that adaptable viewport coverage offers improved quality streaming instead of relying on predicted points and bandwidth for quality adjustments. However, the proposed system only adapts in response to the predefined delay settings, and under complex network conditions, the performance still needs to be investigated. Moreover, this work does not provide information about viewport prediction, dataset, tiling pattern, content resolution, etc.

Similarly, **Hu et al. [108]** The authors proposed an edge-based live streaming system, called MELiveOV, that enables a capillary distribution of processing tasks of high-resolution omnidirectional content to the 5G enabled edge servers. Similar to [240], the authors considered dynamic viewport size selection based on network performance. Under bad network conditions, a smaller viewport (90°) is requested in higher bitrates, while under ideal network conditions a larger viewport (120°) is selected for streaming. The end-to-end live streaming system includes a content creation module, transmission module, and viewport prediction module. The mobile edge-assisted streaming design reduces the bandwidth requirements up to 50%. However, this work was evaluated in comparison to only a viewport-independent streaming approach.

D'Acunto et al. [241] developed an SRD video player[2] by employing the SRD feature of the MPEG-DASH to facilitate the smooth streaming of zoomable and pannable videos. Multiple resolution layers are prepared on the server-side in advance. The player calculates the available resolution layers and downloads the most suitable navigation or zoom interaction representations. The low-resolution layers are always downloaded in advance to avoid rebuffering events when the user navigates through the scene. The current viewing region is then upsampled and presented to the user to support a high-quality zooming feature. The player aims to seamlessly switch between spatial subparts in time and space domains. However,

[2]https://github.com/tnomedialab/dash-srd.js

this work does not offer any implementation results and verify interactive video spatial and temporal smoothness.

In a multi-user streaming environment, the viewport coverage must be carefully decided to ensure the availability of visual content because of the users' continuous interaction through end devices. Several approaches employ cooperative viewport extension to serve a group of viewers to improve transmission efficiency. To leverage the collaborative transmission opportunities for on-demand streaming applications, **Maniotis et al. [242]** proposed a tile-level video popularity-aware caching and transmission in cellular networks containing a Macro-cell Base Station (MBS) and multiple Small Base Stations (SBSs). The proposed system employs advanced coding schemes to create extended viewport versions for each user and to enable cooperative caching of tiles in each SBSs. This cooperation allows storing only the extended viewport versions at SBSs. The users residing within each SBS coverage can fetch the viewport tiles meeting a time delivery constraint. In contrast, the other tiles can be fetched over the backhaul link, lowering the end terminals' computation burden. However, this work considers larger resolution tiles (4x3), and the viewport coverage could be extended to the entire frame. Moreover, the content was requested for previously computed users' interests.

Li and Gao [243] proposed the MultiUser Virtual Reality (MUVR) framework, where an edge cloud adaptively memorizes and reuses the redundant VR frames to reduce the computation and transmission load. The proposed solution aims to maximize cache utilization and minimize data transmission. The authors utilized the image compression and clipping concepts to reduce the transmission rate of the enlarged viewport. MUVR provides a two-level cache design, such as a small local cache at each user-end and a sizeable central cache at the edge. This cache design reduces the memory requirements by generating the background view for all users, and reusing frames whenever possible. The empirical evaluation using the Android platform and Unity VR application engine demonstrates that the proposed framework reduces the frame-associated data and computation load by more than 95% and 90%, respectively. However, this work does not state how much to extend the viewport based on prediction performance for multiple users.

Guo et al. [244] envisioned random motion patterns and erratic channel conditions for each user and exploited multicast opportunities to avoid redundant data transmissions. The authors considered two non-convex problems for each user: (i) under the given video quality constraints, minimization of average transmission time and energy, and (ii) under the given transmission time and energy budget, maximization of video quality. The proposed solution computes the actual viewport tiles for the current user and adds more tiles to the viewing region based on the common interest of other users. The authors considered 100° viewport coverage and an extra 15° in both horizontal and vertical directions. However, the comparison metrics considered in this work are not enough.

Similarly, **Long et al. [245]** considered the transmission time, video quality smoothness, and power constraints to optimize the aggregated utility of multiple users in a single-server multi-users wireless network environment. To reduce the transmission complexity, the authors prepared the tiles in multiple resolutions with different quality levels and divided the tiles set into disjoint subsets for each group of users. For each user, tiles containing the current FoV and the FoVs with lower watching probability are transmitted using Time Division Multiple Access (TDMA). However, this work does not consider the fairness among multiple VR clients.

Table 3.4 presents the summary of extended viewport-based adaptive 360° video streaming solutions. Such solutions can improve the 360° video transmission efficiency, especially when multiple competing clients look for slightly similar content. Single-user-based streaming solutions decide the viewport extension based on viewport estimation performance [239] or network performance [240, 108]. In contrast, the multi-users-based streaming solutions [242, 243, 244, 245] decide the viewport extension based on common Region of Interest (ROI). However, selecting the popular content based on multiple users' interests could end up in caching or streaming full frame in higher quality. Moreover, most of the multi-user streaming scenarios [242, 243, 204, 245] do not involve any prediction mechanism and consider viewport positions known in advance. Therefore, it is highly necessary to decide the optimal viewport extension in response to the different watching behavior of the users.

Table 3.4: Summary of Extended Viewport-based Adaptive 360° Video Streaming Solutions

Works	Design Type	Extension	Prediction Model	Dataset	Tiling Pattern	Resolution	Segment Duration	Experimental Duration
[48]	Probability-based	Fixed	Spherical Walk	[225]	11, 22, 42, 44, 84, 86, 8x8, 1612/16	4K	1.067s	Video duration
[239]	Machine learning	Dynamic	Neural Network	1 Video, 1 Users [238]	16x8	4K	1s	80s
[240]	Probability-based	Dynamic	-	-	-	-	-	-
[108]	Edge-assisted	Dynamic	LSTM	4 Videos, 1 User	4x6	4K	Live	Video duration
[242]	Edge-assisted	Dynamic	-	3 Videos	4x3	1280x720	1s	Video duration
[243]	Edge-assisted	Dynamic	-	3 Videos, 4 Users	Dynamic Clipping	-	-	Video duration
[244]	Multicast	Fixed (15°)	-	1 Video	36x2	-	0.1s	-
[245]	Multicast	Fixed (10°)	-	1 Video	18x36	-	-	Video duration

3.3.3.5 Dynamic Tiling-based Adaptive 360° Video Streaming Solutions

Most of the tile-based adaptive streaming solutions try to accomplish cost-effective streaming by adopting a fixed tiling pattern, which may introduce redundant pixels and lose many compression opportunities [57]. The polar regions have higher viewing distortions and less viewing probability than the equator regions when transforming a spherical video to a two-dimensional plane format, i.e., equirectangular projection. Therefore, encoding polar areas with more pixels possibly consume the user's limited bandwidth in less exciting regions. On one hand, fixed tiling encodes polar and equator regions in similar bitrate levels leading to unattractive viewport boundaries and losing many compression opportunities for smaller tiles compared to larger tiles. On the other hand, a tile-based adaptive mechanism with dynamic tiling lowers the amount of redundant data and provides a finer FoV boundary for different watching behaviors of the users. Many tile-based solutions employ dynamic tiling versions in response to the viewing preferences to enable higher perceived quality levels.

Nguyen et al. [246] proposed a dynamic tiling-based adaptation method for 360° video streaming. The authors employ four tiling patterns, i.e., 4x3, 6x4, 8x4, and 8x8. The tiles are classified into viewport and non-viewport regions for each tiling pattern. Next, based on the network measurements, the highest and lowest quality levels are allocated to the viewport and non-viewport tiles, respectively. The average visual distortion in terms of Mean Square Error (MSE) is computed for each tiling pattern. The tiling pattern giving the minimum quality objective, i.e., visual distortion, is selected for streaming. However, this work involves complex computations as quality allocations for each tiling pattern are carried out. Moreover, this work does not involve any viewport prediction mechanism, and experiments were

performed using fixed network connections.

Zhou et al. [247] proposed a cluster-based approach, namely ClusTile, where the tiles with the minimal bandwidth requirements are streamed to overcome the coding efficiency and computation overheads. ClusTile dynamically performs the tiles selection and bitrate adjustments for each segment. The authors took five videos from [229] and prepared them into 1920960 and 38401920 resolutions. For each resolution, the sizes of the basic tile are 64x64 and 128x128 pixels, respectively. The proposed system requests different basic tiles to cover the actual viewing region. ClusTile achieves a bandwidth saving of up to 72% and 52% compared to the traditional and advanced tile-based streaming approaches, respectively. The disadvantage of this approach is that cluster-based tiles selection may lead to inappropriate tiles selection when a discrepancy occurs between actually viewed and downloaded tiles during the streaming session.

Ozcinar et al. [58] transformed the observed viewport trajectories to the viewport-based visual attention maps to dynamically compute the tiling patterns and encoding versions for each of the frames of 360° video. The transformed polar regions are allocated with lower bitrates compared to the equator regions. A non-uniform bitrate is assigned to the variable size tiles belonging to different priority regions for each selected pattern. The bitrate selection entirely depends on the estimated viewport and network conditions. The authors prepared seven different spatial and temporal motion content into ten tiling patterns. However, the proposed framework struggles to optimize the viewport quality as a large portion of bandwidth transfers non-viewport tiles. Moreover, the employed videos have a duration of 10s only.

Quan et al. [248] analyzed user QoE by computing pixel-wise motion and depth information using a CNN model and used it to dynamically group tiles to provide an essential balance between the video quality and encoding efficiency. The proposed system includes an edge computing server, an edge distribution server, and a VR client. The tiles are grouped using an optical flow metric based on the PSNR metric at the edge computing server-side. The authors proposed an RL-based adaptation agent that intelligently adapts each tile's quality to the dynamic environment on the client-side. The validation of the proposal using real LTE bandwidth traces

demonstrates superior performance in terms of perceived quality while also saving bandwidth resources. However, the complexity of tiles selection based on the calculated reward can be enhanced for multi-users or live streaming environments.

Kattadige and Thilakarathna [57] proposed a visual attention-based geometrical approach to decide the tiling version for each segment. First, the frames are sampled, and viewport maps are generated for the user. Next, the visual frames are partitioned into three main blobs, i.e., the viewport and non-viewport blob, a buffering blob just outside the first region, and the remaining blob, based on the visual attention maps of the user. The partitions are further refined in postprocessing, and non-uniform bitrate is allocated to different salient regions. The proposed solution was compared with three fixed tiling patterns, 4x6, 6x6, and 10x20, and was shown to outperform in terms of pixels and bandwidth savings. However, the impact of different resolution tiles on perceived quality was not covered.

In a subjective experiment with 50 users, **Wang et al. [235]** showed that most of the users observed significant quality degradation when mixing 1920x1080 resolution tiles with 960x540 resolution tiles. However, the users noticed a small difference when mixing 1920x1080 resolution tiles with 1600x900 resolution tiles. This mixing effect leads to even severe quality degradation for high motion content. Therefore, in addition to dynamically performing the tiles selection [58, 57], there should be an appropriate choice of streaming resolutions to attain a perfect trade-off between the streaming quality, spatial quality variance, viewport prediction errors, and bandwidth efficiency.

Tile-based adaptive streaming helps support the transmission of quality-variable tiles based on the user's interest without sacrificing picture quality. Such solutions are associated with many challenges related to viewport prediction, tiles selection, bitrate adjustments in spatial and temporal domains, synchronization with user head movements, etc. Most proposed schemes [47, 33, 249, 2] use different resolutions for viewport and adjacent tiles. This can reduce the bandwidth cost for efficient streaming. However, the different resolution tiles can significantly lower the perceived video quality in case of wrong viewport prediction. In addition, several fixed viewport-based [164, 205, 204, 213, 3], marginal region-based [230, 231, 232, 236, 234], and

extended viewport-based [239, 242, 244] adaptive streaming solutions were tested using a single tiling pattern and single segment duration. How the tiling size and segment duration impact the user-perceived quality levels is an important question that needs to be explored. Similarly, dynamically selecting the size of the marginal region and the extended viewport based on different viewport preferences is very important, although challenging and needs to be investigated.

3.3.4 Viewport Prediction Solutions

The level of satisfaction of a user viewing 360° video content is more sensitive to disturbance when using a headset than when a traditional display is employed. The immersive experience is negatively influenced by imperfect viewport prediction. For instance, frequent viewport deviations introduce higher viewport disruptions and lower perceived quality levels. Viewport prediction performs an essential role in the optimized streaming of 360° video. Viewport prediction approaches are often classified into *content-agnostic approaches* that predict the future viewing position based on the historical information, and *content-aware approaches*, which require video content information to anticipate the future viewports.

3.3.4.1 Content-Agnostic Solutions

One of the essential characteristics of HMD is to respond quickly to the viewer's head movement. Wearable HMDs equipped with position sensors allow the clients to update a viewing scene corresponding to their viewing orientations. The HMDs process the interaction signals when users change their viewport and can detect the related viewport to precise the player information. Several existing content-agnostic approaches predict future viewing position using various prediction methods such as, speed-based [250], LR [105, 251], clustering [252, 253, 254], straightforward ML [238, 232, 255], and encoder-decoder architecture [55]. Next, these prediction approaches are discussed in detail.

Sanchez et al. [250] considered two prediction variants, i.e., angular velocity and angular acceleration, to measure the impact of delay and visual fidelity loss in MPEG-OMAF-based viewport-dependent streaming. Instead of predicting the

head pose for the first frame, the angular velocity and angular acceleration-based predictions are carried out for the middle frame and are kept constant for the whole segment. Then an alpha-correction factor is applied to limit the prediction inaccuracy. Based on the prediction results, different Quantization Parameter (QP) values are assigned to each HEVC tile. Unfortunately, the acceleration-based methods have limited prediction accuracy when predicting the viewport longer in the future (e.g., more than a 2s interval). Consequently, if video tiles are requested based on an inaccurate prediction, the user's actual viewport may be covered by black tiles for which no content was requested.

Qian et al. [105] employed average, linear regression, and weighted linear regression methods for four 360° videos. The viewport prediction was carried out for 0.5s, 1s, and 2s prediction horizons using past 1s history samples. The authors showed that the average method performs worst when predicting the next 2s in the future. The weighted linear regression with different weights was shown to provide long-lasting results for different motion content. The authors measured bandwidth savings for 1s streaming session by considering three scenarios: perfect knowledge of viewport; extending the FoV coverage by 10° in all directions; extending the FoV for 4s streaming session. The client could achieve about 80% bandwidth savings when the viewing positions are perfectly known. The bandwidth savings drop by 5% when the FoV coverage is extended. The bandwidth saving was measured in terms of the number of requested tiles. However, the experiments were performed locally with no functional client-server architecture. In another work, **Qian et al. [251]** proposed a practical view-based streaming system named Flare for commodity devices to minimize the mismatch between actual and predicted viewport. The authors collected 1300 head motion traces from 130 users for 10 360° videos. Four off-the-shelf methods i.e., naive, LR, Ridge Regression (RR), and SVR were employed for Flare to predict for next 0.2s, 0.5s, 1.0s, and 3.0s using 0.1s, 0.25s, 0.5s, and 1.5s history samples, respectively. The authors suggested using the LR (for ¡1s prediction window) and RR (≥1s prediction window) methods for the Flare to make it more robust and lightweight. Flare enables an incorrect prediction to be replaced by the latest prediction with the viewport trajectory prediction. Flare was successfully

tested on Samsung Galaxy S7, Samsung Galaxy S8, and Samsung Galaxy J7 devices. However, Flare was compared with conventional bitrate adaptation approaches, i.e., Festive and BBA.

The cross-users watching behavior can improve the prediction performance for 360° videos. **Ban et al. [252]** took advantage of users' attention distribution in 360° video to improve the viewport prediction performance. They analyzed the current user's watching behavior using the LR method and then combined it with other users' similar ROI using the KNN algorithm to fetch the viewport tiles for the next segment. Experimental evaluation on real datasets reveals that 20% improved prediction accuracy can result in up to 30% more quality gain than the LR -based streaming approach. However, the performance was evaluated using a single viewport trace for three videos from the open-source dataset [225].

Petrangeli et al. [253] employed the concept of vehicular trajectory prediction for viewport prediction in 360° space. First, the given trajectories are clustered based on the distance between two trajectories. After identifying the main clusters, the trending trajectory is computed for each cluster based on the average trajectory of users. Then the affinity score between the current user trajectory and each cluster's trend trajectory is computed to predict the potential viewport. The trend trajectory with the highest affinity score will be the potential predicted trajectory for the current user. The trajectory-based clustering approach results in a lower great circle distance and a higher viewport overlap for different prediction horizons. However, they examined three different trajectories for Euler angles (Φ, Θ, Ψ), which might lead to unsatisfactory performance.

Rossi et al. [254] considered clustering the users who have a meaningful viewport overlap for a frame or series of frames. The geodesic distance between two viewpoints measures the viewport overlap. A graph is then built by connecting the nodes if the corresponding viewports have a significant overlap. The clustering algorithm based on the Bron-Kerbosch (BK) [256] algorithm recognizes substantial groups of users who are watching the same 60% of 3s long chunks of spherical video. This method provides clusters with compatible and significant geometric viewports overlap compared to the benchmarks.

LR methods result in poor prediction accuracy for a long-term prediction horizon [238]. LSTM [257] is a Recurrent Neural Network (RNN) architecture that is suitable for sequence modeling and pattern exploiting. In order to achieve higher precision than LR in FoV prediction, **Jiang et al. [232]** developed a viewpoint prediction method using an LSTM model with 128 neurons. The LSTM layer takes the normalized watching samples as input, and then a transform layer is applied to convert the normalized values to the predicted orientations. The FoV coverage was extended in both horizontal and vertical directions to deal with the potential prediction errors. The authors analyzed the 360° dataset [229] and observed that users have fast head turns in the horizontal direction but almost stable motions in the vertical direction. The experimental comparison with LR and average approaches reveals that the LSTM-based viewport predictor generates lower prediction errors considering both horizontal and vertical head movements.

Bao et al. [238] conducted a subjective experiment with 150 users to analyze their viewing behavior across 16 video clips related to sports, landscape, and entertainment categories. They showed that the angles (Φ, Θ, Ψ) representing the user motion in 3D space have a strong auto-correlation and negligible cross-correlation. Therefore, these angles can be measured separately. The authors developed two separate LSTM neural network models to separately predict the viewing angles Θ and Φ as well as the viewing deviations. The authors showed that the user head movements could be well predicted for a short horizon, i.e., 100-500ms. The prediction results are then used for a targeted area streaming using frame trimming to efficiently utilize the available network resources. However, the predictions were carried out for short intervals, i.e., 0.1s-1s, in comparison to naive and LR methods.

Hou et al. [255] proposed a deep learning-based view generation method to extract and stream only the predicted viewport tiles in advance for 360° videos and 3-Degree of Freedom (3DoF) VR applications. The authors trained their model using a real large-scale dataset[3], i.e., about 36,000 head movement traces for 19 videos. The viewpoint prediction model consists of 256 LSTM units, a fully connected layer, and a softmax layer to predict one or multiple tiles for the next 2s prediction time.

[3]Samsung VR. https://samsungvr.com

Then the FoV generation module creates the FoV tiles based on the predicted tiles. Only the FoV tiles are streamed in higher quality levels while leaving the rest, which significantly reduces the quality for wrong prediction results. In another work [258], the same researchers introduced a new predictive approach involving Multi-Layer Perceptron (MLP) and LSTM models to predict the head (i.e., viewing direction) as well as body (i.e., standing position) movements in a 3m3m 6-Degrees of Freedom (6DoF) VR environment. The predictive view is pre-rendered to enable a low-latency VR experience. However, the dataset used for this work contains only two videos, and the practical implementation of this system needs further proof.

LSTM networks have a time-consuming sequential training nature. Encoder-decoder LSTM models parallelize the training process resulting in improved prediction accuracy compared to LR and LSTM networks. **Yu and Liu [55]** used the attention-based LSTM encoder-decoder network architecture to avoid expensive recursion and to capture the viewport changes better. The attention-based encoder-decoder model was evaluated using three publicly available datasets containing head movement traces recorded on both the computer screen and VR headsets from 154 users for 35 videos. The encoder-decoder network architecture could be implemented using cross-user orientation information to further improve prediction accuracy.

Content-agnostic solutions are mainly low-complex solutions that provide reasonable prediction accuracy for different motion content. However, the perfect prediction results are not possible by processing positional information only. A viable and easy to implement solution could be utilizing multiple content-agnostic prediction mechanisms in each single prediction interval. This could hide the limitations of position-based prediction mechanisms and can generate the best tiles selection. Another solution is to utilize the content features with/without positional information to increase the prediction performance, which is more complex in general.

3.3.4.2 Content-Aware Solutions

Content-aware viewport prediction is considered to be a vital enabler for 360° video because it can improve prediction efficiency. **Aladagli et al. [260]** proposed a saliency-driven model to improve the prediction accuracy. However, this work did

Table 3.5: Summary of Viewport Prediction Approaches for 360° videos

Works		Prediction Model	Prediction Horizon	Dataset	Display
Content-Agnostic Approaches	[250]	Angular velocity, Angular acceleration	1 segment	10 videos, 17 users	OculusRift CV1
	[105]	Average, LR , WLR	2s	4 videos, 5 users	Google Cardboard
	[251]	Online machine learning	3s	10 videos, 130 users	Samsung Gear VR
	[252]	Users clustering	6s	[225]	HTC Vive
	[253]	Trajectory-based clustering	10s	[238]	Oculus DK2
	[254]	Clique clustering	3s	[229]	Razer OSVR HDK2
	[232]	LSTM model	3s	[229]	Razer OSVR HDK2
	[238]	Neural network model	1s	16 videos, 150 users	Oculus DK2
	[255]	LSTM model	1 segment	19 videos, 36000 users	Samsung Gear VR
	[55]	Attention-based encoder-decoder	3s	[229, 259, 225]	Razer OSVR HDK2, Monitor, HTC Vive
Content-Aware Approaches	[260]	Saliency-driven	2s	[238]	Oculus DK2
	[53]	Sensor and content features	2.5s	[229, 225]	Razer OSVR HDK2 HTC Vive
	[214]	CNN-LSTM model	1s	208 videos, 45 users	HTC VIVE
	[261]	DRL models	30ms	76 videos, 58 users	HTC Vive
	[5]	Sensor and content features	1s	10 videos, 25 users	Oculus Rift
	[262]	RNN+CFVT	1s	[238]	Oculus DK2
	[52]	LSTM encoder-decoder ConvLSTM	10s	[214, 225]	HTC Vive

not consider the user's viewing behavior for 360° videos. Viewport prediction errors could be minimized by understanding the user's unique visual attention for 360° videos. Most existing methods [53, 261, 214] focus on considering saliency patterns as well as positional information in 360° display to achieve better prediction results. Instead of using traditional saliency models, **Nguyen et al. [53]** proposed the panoramic saliency detection mode named PanoSalNet to capture the unique visual attention of the users in 360° frames to improve the saliency detection and head movement prediction performance. The authors employed two datasets to collect panoramic saliency. The fixation maps were created using the DBSCAN algorithm, and then a Gaussian filter was applied to smooth the fixation points. The salient regions were detected using a deep convolutional neural network (DCNN). The proposed head movement prediction network architecture then takes the saliency maps and head orientations of past points to output the predicted head orientation map for future timestamps. The fixation prediction solution with both HMD sensor features and saliency maps results in a measurable gain.

Xu et al. [214] collected a large-scale dynamic dataset[4] of 208 360° videos viewed by 45 subjects with an eye-tracking-capable HTC VIVE headset. The authors proposed predicting the gaze displacement based on the history scan path and image features. For each video clip, the trajectory encoder module encodes the histor-

[4]https://github.com/xuyanyu-shh/VR-EyeTracking

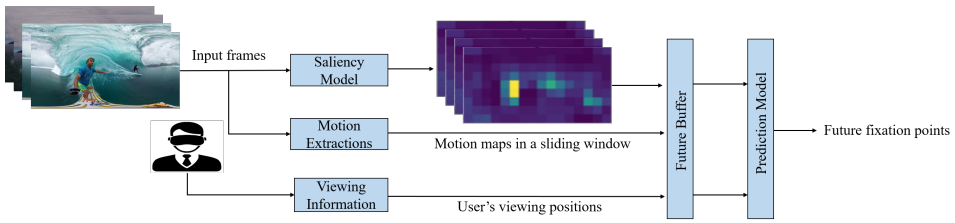


Figure 3.6: Saliency, motion maps, and viewing position based fixation prediction network in [5].

ical watching points by a stacked LSTM network. The saliency encoder module computes the saliency using the CNN network at three spatial scales based on the content popularity, i.e., local saliency, FoV saliency, and global saliency. Then both the LSTM and CNN peculiarities are coupled for gaze prediction over the next second. However, the proposed network does not deal with prediction errors.

Xu et al. [261] proposed two versions of DRL-based head movement prediction named offline-DHP and online-DHP by considering both motion trajectories and visual features. The authors introduced a VR traces dataset of 76 high-resolution (3K-8K) panoramic clips watched by 58 subjects. The offline-DHP model generates the potential heatmaps that represent the viewpoint distribution based on content popularity by running multiple DRL workflows. The online-DHL model then predicts the viewport direction and magnitude based on the obtained heatmaps from the offline-DHL model as well as the previous scan path information. However, the network aims to predict the next viewport position for only 30ms, i.e., one future frame.

Since the users are more attracted to moving objects, therefore, in addition to saliency maps, **Fan et al. [5]** investigated high-computation models by leveraging motion maps using pre-trained CNN to deal with the irregular abrupt viewport changes. The general architecture of the fixation prediction model is shown in Fig. 3.6. The feature buffer contains saliency maps, motion maps, and viewer orientation features for a sliding window. The fixation prediction network then outputs the tiles viewing probability for future frames. However, most of the existing HMDs cannot track the gaze position [263]; and the motion maps exploitation needs further investigations since there could be multiple motions, which makes the prediction not

reliable enough. Besides, the extraction of image saliency and motion maps is highly time-consuming.

Yang et al. [262] utilized both the head motions and content-related features to predict the future potential viewports using a large-scale open-source dataset [214]. First of all, the viewing trajectory is predicted using a RNN. The Correlation Filter-based Viewport Tracker (CFVT), containing more convolutional layers instead of pooling layers, performs content-aware viewport prediction by learning the relationship between sequential viewports. Then a fusion layer combines the results of the content-agnostic (i.e., RNN) and content-aware (i.e., CFVT) prediction models. Incorporating the fusion model enables better prediction and improves accuracy by 40%. However, the fusion model should be compared for longer prediction horizons, i.e., >1s.

Current prediction models are limited regarding how much they can predict in the future. **Li et al. [52]** proposed two viewport prediction models (trajectory-based and heatmap-based) for viewport-dependent tile-based streaming systems. The authors considered two versions, i.e., with/without cross-user behaviors, for each model. The first model employs an LSTM encoder-decoder network architecture to encode the watching history of the current and/or other users to predict the future viewing points similar to [55]. The second model employs a convolutional LSTM encoder-decoder architecture to predict the future heatmap sequence for the current user based on the previous heatmaps of the current user and/or the average heatmaps of other users. The experimental evaluations demonstrate that exploiting the watching history of other users can substantially improve the prediction performance compared to considering the current user's watching history only.

Accurate orientation prediction enables the 360° client to download the most relevant tiles at high resolution. Table 3.5 summarizes the content-agnostic and content-aware head movement prediction approaches. Current neural network models [5, 260, 53] that employ both the saliency and position information perform poorer than a simple no-motion baseline which directly utilizes the current viewing position for future viewport position estimation [264]. The noise level in the estimated saliency limits the prediction accuracy of these models [264]. Besides, these

models involve additional computational complexity. For the reliable prediction of attention points in 360° video and understanding of the relationship between the user's viewing likelihood and saliency maps, the saliency models must be improved and well fitted by training on large-scale datasets, especially captured with different camera rotations [265]. On the other hand, all the presented works perform prediction based on a single prediction mechanism. Utilizing multiple low complex prediction mechanisms can generate long-term prediction dependency and can bring considerable QoE improvement, especially for fast motion content.

3.4 Chapter Summary

This chapter cited existing works from the area of adaptive streaming for traditional, multi-view, and 360° videos. Several related works about the design of innovative streaming technologies and applications and possible limitations were discussed in detail. The chapter introduced the current server-side and client-side adaptive streaming solutions, which were categorized into three main types: throughput-based, buffer-based, and hybrid adaptive solutions. The existing multipath and multi-view video streaming solutions over the heterogeneous wireless environment were also listed. This chapter covered a useful classification of the tile-based streaming approaches used in the literature to model the tiles selection problem. The classification is done based on the dynamic tile size and viewport size selection problems. The four main tiles selection categories presented in this chapter are: (i) fixed viewport-based solutions; (ii) marginal region-based solutions; (iii) extended viewport-based solutions; (iv) dynamic tiling-based solutions. The significant findings from these classes and the experimental evaluations surrounding the tiles selection problem are summarized in Table 3.2, Table 3.3, and Table 3.4. Diverse efforts have been made to optimize bitrate awareness, quality balancing, and fair and stable transmission of regular, multi-view, and 360° videos. The existing 360° solutions were found to be limited in terms of accurate viewport prediction and tiles selection. This leads to new horizons that motivate smart designs and mechanisms presented in the following chapters.

Chapter 4

Throughput and Buffer Occupancy-based Adaptation

This chapter presents the Throughput and Buffer Occupancy-based Adaptation (TBOA) solution which improves the streaming quality of traditional video content. TBOA chooses most appropriate bitrates during video delivery while avoiding video quality switches and empty playback buffer-caused interruptions in both single- and multiple-client streaming environments under different network conditions. TBOA is compared against alternative throughput-based (SFTM), buffer-based (SVAA), and hybrid (AAASH, FDASH, OSMF) adaptive solutions. Testing results show that TBOA offers improved QoE streaming while efficiently utilizing the available network resources in comparison with existing solutions.

4.1 Adaptive Video Streaming: Current Experience

The current HAS framework, empowered by an adaptation algorithm helps to improve video streaming by dealing with different objectives, which include quality, efficiency, stability, and load balancing, etc. It sets the video bitrate closer to the variable network link available bandwidth to achieve both the transmission and playback objectives [266]. DASH framework as a step forward in HAS standardization pathway employs an application layer adaptation logic to minimize playback startup delay, stalling/rebuffering, and quality fluctuations for a broad range of devices. DASH achieves these goals by correctly estimating the available network capacity. In a relatively stable network environment, the throughput estimation of upcoming video segments based on the previous observations is relatively reliable and accurate. However, some works still continue to request lower bitrates [38, 150, 167] or higher bitrates [37] than the available bandwidth. In a highly variable network or multi-client streaming situation, a non-aggressive quality increment approach adopted by some works [36, 150] could lead to lower bandwidth efficiency or sub-optimal quality. The success of the adaptive streaming algorithms mainly lies in reducing the difference between the requested bitrate and available bandwidth budget. A high difference could lead to either higher loss or lower bitrates, both being associated with lower quality levels. This is done by varying the bitrate of the streamed video content during streaming. During streaming, the buffer may run dry when the throughput drops suddenly, and the buffer level increases when the segments are downloaded faster than they are consumed for the playback. Particularly, the level of the buffer increases τ seconds after the download of (i)th segment and decreases as the viewer plays the video. Let $B(i-1)$ be the buffer occupancy for $(i-1)$th segment. The current buffer level for the (i)th segment can then be expressed as:

$$B(i) = B(i-1) + \tau - D(i) \tag{4.1}$$

Eq. 4.1 reflects that if the download time $D(i)$ is higher than the buffer level $B(i-1)$, the buffer gets drained. The buffer drain and playback interruption concepts are illustrated in Fig. 4.1. To avoid playback interruptions, it is required that $B(i)$

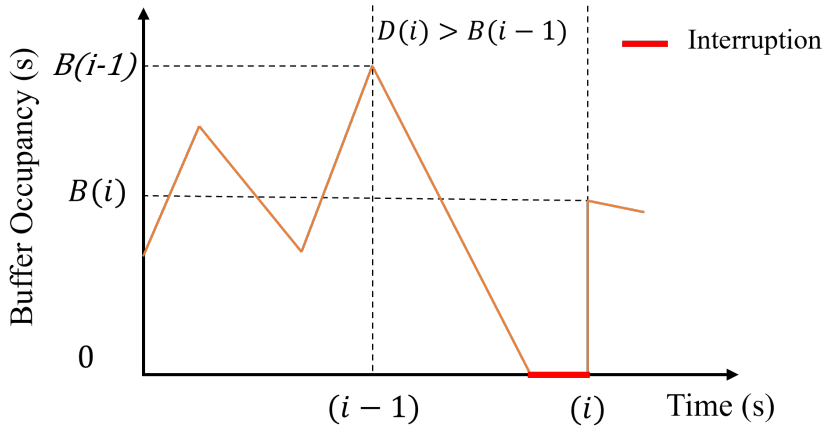


Figure 4.1: Concept of buffer dynamics in adaptive video streaming.

never reaches zero at any time during the playback. The difference in the buffer level during the download of the (i)th segment is given in Eq. 4.2.

$$\Delta B = B(i) - B(i-1) = \tau - D(i) \tag{4.2}$$

The adaptive algorithm should always try to keep this difference non-negative. At the beginning of the playback session, the HAS client quickly fills the buffer by requesting the segments as quickly as the TCP connection allows. However, when the buffer level increases to the predefined level (i.e.,B_{max}), the adaptive client enters a periodic download or ON-OFF phase. The client is active and downloads the next data segment during the ON state, but waits for a sufficient amount of time to allow for buffer data consumption before sending the next download request during the OFF state. In DASH-based adaptive streaming, the ON-OFF period is a natural phenomenon [267]. However, the increased frequency of the ON-OFF events leads to several performance issues such as sub-optimal quality selection, transmission inefficiency, quality fluctuations, unfairness, etc. Therefore, the adaptive client should be designed to sustain an optimum buffer level during the streaming session which is defined as in [167]:

$$B_{opt} = (B_l + B_h)/2 \tag{4.3}$$

where B_l and B_h represent the low and high buffer levels in a multi-level payout buffer occupancy. With this, the adaptive client should keep the requested bitrate to

the maximum possible levels without undergoing unnecessary quality fluctuations. This chapter presents a DASH-based client-side bitrate adaptation algorithm TBOA which tries to achieve enhanced QoE levels by keeping the buffer levels closer to the optimum levels.

TBOA incorporates buffer levels and is designed in such a way to keep the same bitrate levels with minimal quality switches even when the throughput changes rapidly or slower.TBOA makes efficient use of the available network measurements and improves the bandwidth utilization by having the adaptive bitrates close to the bandwidth levels. TBOA is evaluated in single and multiple-client environments. In a single-client environment, we consider 1s, 2s, 4s, and 6s segment durations to analyze the impact of segment duration on DASH-based on-demand video streaming in wired and wireless network environments. TBOA is shown to outperform other benchmark algorithms in terms of achieving higher video bitrate, reducing the video quality fluctuations, and avoiding playback interruptions. In a multi-client scenario, TBOA clients make efficient use of network bandwidth to play high-quality segments while achieving good stability and fairness. The next sections describe the proposed system architecture, algorithm, and comparative performance evaluations.

4.2 TBOA Streaming System Model

Fig. 4.2 illustrates a generic DASH-based adaptive streaming process. The DASH server is the source of video content, and where the regular video is temporally split into a number of segments, each with a duration of τ seconds. The equal duration segments are encoded into different encoding levels, i.e. $\mathcal{L}$=$\{\mathcal{L}_1(i)$,$\mathcal{L}_2(i)$,...,$\mathcal{L}_q(i)$, $\mathcal{L}_{q+1}(i)$...,$\mathcal{L}_{max}(i)\}$. For each segment, the client dynamically selects a bitrate level $\mathcal{L}_q(i)$ from the representations set $\mathcal{L}$ according to the measured buffer level and current network throughput. $\mathcal{L}_1$ and $\mathcal{L}_{max}$ are the representations with the minimum and maximum bitrates in the set $\mathcal{L}$.

The HTTP DASH client consists of *Playout Buffer*, *Bandwidth Estimation*, *Bitrate Adaptation*, and *Segment Scheduler* modules. The *Playback Buffer* stores the downloaded segments. The *Bandwidth Estimation* unit computes the available network

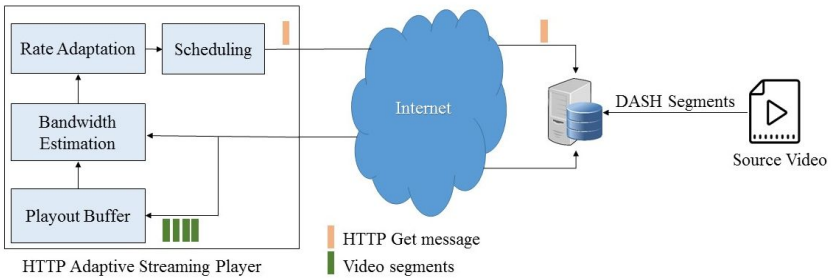


Figure 4.2: HTTP DASH-based client side bitrate adaptive streaming model.

throughput $Th^{est}(i)$ and provides input to the bitrate adaptation unit with throughput and buffer states $B(i)$ allowing it to select the bitrate for (i)th segment. The *Bitrate Adaptation* unit deploys a HAS-based adaptation algorithm by considering the output from *Bandwidth Estimation* and *Playout Buffer* modules and is responsible for dynamically performing the bitrate selection for each segment. The *Segment Scheduler* module controls the request interval for each segment by requesting the next segment with or without introducing a delay, which is calculated in Eq. 4.7. The HTTP DASH server sends the data through a bottleneck link in response to the HTTP GET requests. The downloaded segments are then decoded and displayed to the user. It is not necessary to completely download a segment before decoding and presenting it to the user. However, a specific amount of time is required by the decoder to trigger the decoding process.

4.3 Proposed TBOA Adaptation Algorithm

Estimating the throughput and computing the bitrates under variable network conditions are critically important in adaptive streaming framework. In HAS the client estimates the network throughput from previous observations, usually based on the size and download time of the previous segment, which can be represented as follows [149].

$$Th(i) = \frac{\mathcal{L}_q(i-1) \times \tau}{D(i-1)} \tag{4.4}$$

where $D(i-1)$ indicates the download time of $(i-1)$th segment and $\mathcal{L}_q(i-1)$ is the quality level of the $(i-1)$th segment. Completely relying on the last known

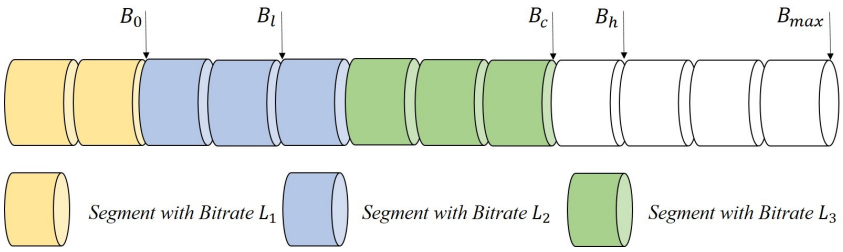


Figure 4.3: Playback buffer levels in the proposed client.

estimation results may not result in accurate results since the network throughput is highly variable. Therefore, computing average throughput alleviates the influence of unexpected variations. Following a series of throughput measurements, the average throughput is given in Eq. (4.5) [38]:

$$Th^{est}(i) = \frac{1}{k} * \sum_{r=i-k+1}^{i} Th(r) \tag{4.5}$$

where k represents the throughput measuring window in the number of segments. Under a constant bandwidth connection consideration of the average of any throughput samples can be feasible. However, in an unstable dynamic network environment, the value of k should be carefully decided to get accurate measurements.

The TBOA adaptation mechanism is provided in Algorithm 4.1. The bitrate adaptation for the (i)th segment is performed after the download of the $(i-1)$th segment. The client-side bitrate adaptation algorithm outputs the quality level for the (i)th segment, which is denoted as $\mathcal{L}_{next}$. We denote the quality level higher than the current level by $\mathcal{L}_{q+1}(i)$. Every downloaded segment is kept in a buffer with a maximum capacity B_{max}.

The TBOA algorithm considers three buffer thresholds: B_0, B_l, and B_h as shown in Fig. 4.3. These thresholds are introduced to provide safe quality increments or decrements along with keeping the buffer at stable levels. Algorithm 4.1 selects the minimum available video bitrate $\mathcal{L}_1$ for the first segment due to two reasons. First, the client does not know the network conditions at the start. Thus, we employ the conservative quality value at the start to fill the buffer quickly. Secondly, the startup delay T_s is reduced by downloading the segment with the lowest available bitrate, which is of substantial interest in HAS systems [268]. When the buffer level

goes above B_0, the maximum video bitrate is chosen provided that the video bitrate does not exceed the estimated throughput and the buffer level has crossed B_0 by $\nu * D(i-1)$, where ν is the safety margin constant for download time with a value in the range [0,1]. This safe bitrate switch-up will help to grow the buffer levels. When the buffer level increases above B_l but it is lower than B_h, i.e., $B_l < B(i) \leq B_h$, the client requests a higher video bitrate by considering the following constraints: (i) $\mathcal{L}_q(i-1) \neq \mathcal{L}_{max}$, (ii) $\mathcal{L}_{q+1}(i) \leq Th^{est}(i)$, and (iii) $\Delta B \geq 0$. Otherwise, the client will maintain the previous bitrate.

If the buffer level drops, the algorithm does not instantly respond to this drop and retains the current quality level until the buffer level drops below B_l. Unlike the immediate switch-down which negatively influences the perceived quality [269], the proposed approach helps reduce the number of quality switches that is favorable to viewers. Moreover, if the buffer level increases further and becomes higher than B_h due to an increase in throughput, the video request is delayed to bring the buffer closer to the optimum level. This is achieved by the client which implements a waiting time before sending the next GET request to avoid the buffer overflow. The waiting time before sending the next download request is calculated in Eq. 4.6.

$$w = B(i) - B_{opt} - \frac{\mathcal{L}_q(i) * \tau}{\mathcal{L}_1(i) * 2} \tag{4.6}$$

The overall waiting time t^w in Algorithm 4.1 is given in Eq. 4.7.

$$t^w = \begin{cases} w, & if B(i) > B_h \\ 0, & Otherwise \end{cases} \tag{4.7}$$

The waiting time is initialized to 0 seconds in Algorithm 4.1. If $B(i) > B_h$ then the client will wait for w seconds before sending the next segment request.

Otherwise, the adaptive client requests the maximum allowable video bitrate with an immediate request. In this way, the proposed bitrate adaptation algorithm responds well to different buffer and throughput measurements to increase the overall perceived quality levels.

Note that the bitrate is decreased when the buffer level is below B_l or above B_h;

Algorithm 4.1 Bitrate Adaptation Algorithm

Input :

$\mathcal{L} \leftarrow \{\mathcal{L}_1(i), \mathcal{L}_2(i), \mathcal{L}_3(i), \ldots, \mathcal{L}_{max}(i)\}$

$Th^{est}(i) \leftarrow$ *Estimated throughput for downloading (i)th segment*

$B(i) \leftarrow$ *Current Buffer level*

$\mathcal{L}_q(i-1) \leftarrow$ *Video quality level selected for the previous segment*

$D(i-1) \leftarrow$ *Download time of* $(i-1)$*th segment*

$t^w = 0 \leftarrow$ Waiting time to send the next download request

Result :

$\mathcal{L}_{next} \leftarrow$ Video quality level selected for the next segment

if $(B(i) \leq B_0)$ **then**
  $\mathcal{L}_{next} = \mathcal{L}_1(i)$
else if $(B(i) > B_0 \;\&\; \&B(i) \leq B_l)$ **then**
  $\mathcal{L}_{next} = max\{\mathcal{L}_q(i) | \mathcal{L}_q(i) \varepsilon \mathcal{L}, \mathcal{L}_q(i) \leq Th^{est}(i), B(i) - \nu * D(i-1) \geq B_0\}$
else if $(B(i) > B_l \;\&\; \&B(i) \leq B_h)$ **then**
  if $(\mathcal{L}_q(i-1) \neq \mathcal{L}_{max} \;\; \& \& \;\; \mathcal{L}_{q+1} \leq Th^{est}(i) \;\; \& \& \;\; \Delta B \geq 0)$ **then**
    $\mathcal{L}_{next} = \mathcal{L}_{q+1}(i)$
  else
    $\mathcal{L}_{next} = \mathcal{L}_q(i-1)$
else if $(B(i) > B_h)$ **then**
  if $(w > 0)$ **then**
    $t^w = w$
  else
    $\mathcal{L}_{next} = max\{\mathcal{L}_q(i) | \mathcal{L}_q(i) \varepsilon \mathcal{L}, \mathcal{L}_q(i) \leq Th^{est}(i), B(i) - \nu * D(i-1) \geq B_l\}$

Table 4.1: Simulation Parameters

Parameter	Value
Video Sequence	Big Buck Bunny (BBB[1])
Simulation Length	Single client (300s), Multi-clients (300s).
Segment Duration	1s, 2s, 4s, 6s
Remote Station Manager	AARF WiFi Manager
MAC Type	WiFi Station MAC, WiFi AP MAC
Mobility Model	Constant Position Mobility Model

because under the $max()$ function, the bitrate could be either decreased or increased for the next segment, if the throughput drops or increases, respectively.

4.4 Performance Evaluation

This section presents the experimental testing setup, testing results, and analysis for single and multi-client streaming environments.

4.4.1 Experimental Setup

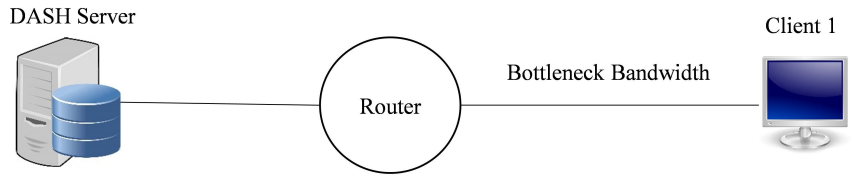
4.4.1.1 Modeling and Simulation

Network Simulator 3 (NS-3) was used for modeling and simulation in order to evaluate the proposed solution. We modified the adaptive video provided by [38] and implemented single-client and multi-client scenarios. The single client performance evaluations were performed using wired and wireless network topology shown in Fig. 4.4a and Fig. 4.4b, respectively. In Fig. 4.4a, a single client requests video segments from the DASH server through a bottleneck link. The bottleneck link is varied as follows: 6 Mbps for the first 50s, afterwards, the link rate fluctuates between 8 Mbps and 4 Mbps every 10s until 300s. This scenario is adopted to represent the bitrate selection under fast bandwidth changes. In Fig. 4.4b, the point-to-point connection between the DASH server and Access Point (AP) is implemented as follows: 5 Mbps for the first 50s and then is set to vary between 10 Mbps and 5Mbps after every 30s until the end of the simulation. These bandwidth settings reflect a stable fluctuation scenario to verify the performance of different adaptation algorithms. For the multi-clients scenario, a constant bandwidth link of 12 Mbps was shared between three and five competing clients. The values of buffer levels B_0, B_l, B_{opt}, and B_h are set to 4s, 10s, 35s, and 60s, respectively. These buffer levels are selected such as to keep the target/optimum buffer level to 35s as used in [38, 36]. The throughput measuring window k was set to 10s in the original implementations of the DASH client [38]. However, the adaptive client will rely on the last known throughput measurement instead of mean values for about 10s. We set the value of k to 5s, as this will help to accommodate both short-term and long-term bandwidth fluctuations. The description of the simulation environment is presented in Table 4.1.

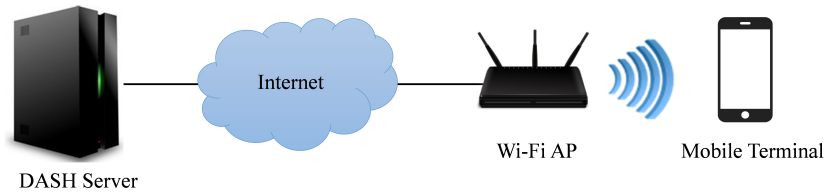
4.4.1.2 Testing Video and Bitrates

The DASH server offers various video bitrate levels and the DASH client controls the request decisions to facilitate the adaptive streaming. We choose Big Buck Bunny[2]

[2] `https://peach.blender.org/`



(a) Wired network topology for a single client scenario



(b) Wireless network topology for a single client scenario

Figure 4.4: The wired and wireless network topology for simulation experiments.

Table 4.2: Bitrate Ladder [Mbps] of Big Buck Bunny Video Sequence [1]

q	$\mathcal{Q}(\mathcal{L}_q)$	$\mathcal{L}1$	**Resolution**	$\mathcal{L}2$	**Resolution**	$\mathcal{L}3$	**Resolution**
1	0.16	0.10	256x144	0.20	320x180	0.24	384x216
2	0.33	0.37	384x216	0.55	512x288	0.75	640x360
3	0.50	1.00	768x432	1.50	1024x576	2.30	1280x720
4	0.66	3.00	1280x720	4.30	1920x1080	5.80	1920x1080
5	0.83	6.50	2560x1440	7.00	2560x1440	7.50	2560x1440
6	1.00	8.00	3840x2160	12.00	3840x2160	17.00	3840x2160

a computer-animated video sequence to perform the experiments. We choose three representation sets from [1] for this video sequence.

The chosen bitrate values in Mbps are as follows: $\mathcal{L}1 = \{0.10, 0.37, 1.00, 3.00, 6.50, 8.00\}$; $\mathcal{L}2 = \{0.20, 0.55, 1.50, 4.30, 7.00, 12.00\}$; and $\mathcal{L}3 = \{0.24, 0.75, 2.30, 5.80, 7.50, 17.00\}$. The bitrates ranging from 0.10 Mbps to 0.24 Mbps belong to low-definition (LD) video, from 0.37 Mbps to 0.75 Mbps belong to standard-definition (SD) video, from 1 Mbps to 2.3 Mbps are associated with high-definition (HD) video and from 3 Mbps to 5.8 Mbps belong to full-HD (FHD). Finally, the bitrates between 6.5 Mbps and 7.5 Mbps are mapped to quad-HD (QHD) video and from 8 Mbps to 17 Mbps represent 4K ultra-HD (UHD) video resolutions. Table 4.2 shows the relevant resolutions for these bitrate values. We simulated the video playback by choosing τ as 1s, 2s, 4s, and 6s to better analyze the streaming performance of proposed and comparative solutions. The value of ν is set to 1, 0.5, 0.25, and 0.16 for 1s, 2s, 4s, and 6s segment duration, respectively.

4.4.1.3 Baseline Algorithms

The performance of **TBOA** is evaluated in comparison with the following throughput-based, buffer-based, and hybrid adaptive streaming solutions. We set the parameter values of the various adaptation algorithms employed for comparison purposes, as suggested in their original works.

1. **AAASH** [167]: It is a hybrid algorithm for adaptive streaming over HTTP which incorporates three different buffer levels and refined throughput measurements to select the bitrate for each segment. The values of buffer levels B_0,B_l, and B_h are set to 10s, 20s, and 50s, respectively. The throughput safety margins are set to 0.75, 0.33, 0.5, 0.79, and 0.9.
2. **FDASH** [38]: It is also a hybrid adaptive streaming algorithm that uses fuzzy logic to manage adaptive decisions. The target buffering time is set to 35s and the factors of the output member functions are set to 0.25, 0.5, 1, 1.5, and 2, as used in the original work.
3. **OSMF**[3]: This is the bitrate adaptation algorithm used in Adobes Open Source Media Framework and employs an aggressive quality selection approach at the startup by requesting the highest available video bitrate for the first segment. The bitrate selection procedure for the upcoming segments mainly relies on the network throughput measurements. The buffer levels support the current quality level to be maintained for a short duration of time under sudden bandwidth drops.
4. **SFTM** [150]: It is a throughput-based adaptive streaming solution that implements bitrate stepwise switch up and multistep switch-down policies based on an adaptation metric that considers both the optimum and the measured segment fetch time.
5. **SVAA** [36]: It is a DASH-based smooth video adaptation algorithm which considers the playback buffer levels to smoothly adjust the video bitrate levels. It avoids aggressive switch-down by incorporating intermediate quality adjustments. The target buffer level is set to 35s.

[3]https://sourceforge.net/adobe/osmf/home/Home/

4.4.1.4 Performance Evaluation Metrics

We consider the following evaluation metrics to assess the performance of proposed and comparative bitrate adaptation algorithms.

1. **Video Bitrate**: We measured the per-segment video bitrate using wired and wireless network environments to reflect how the clients behave in response to network variations. In addition, we measured the average video bitrate for both single and multi-client scenarios.
2. **Quality Switches**: This metric indicates the number of quality switching events between two consecutive segments and is given as:

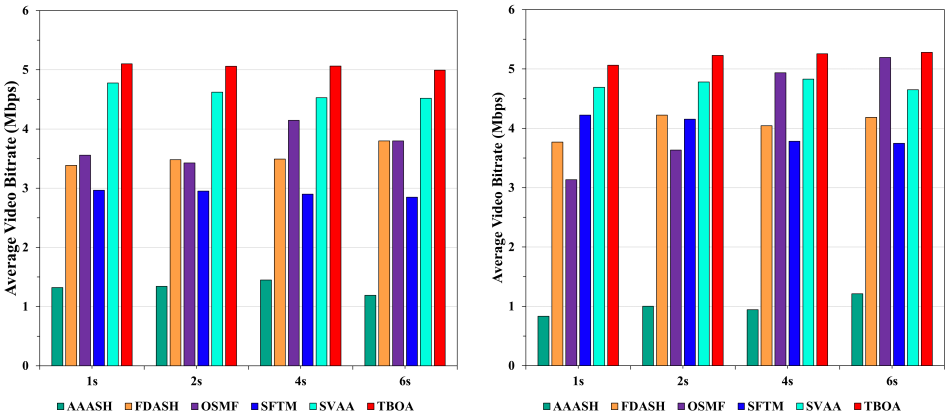
$$\mathcal{F}(i) = \begin{cases} 1, & if \mathcal{L}_q(i) \neq \mathcal{L}_q(i-1) \\ 0, & Otherwise \end{cases} \tag{4.8}$$

   We measure the average number of quality fluctuations per streaming session in single and multiple client scenarios.
3. **QoE**: Following the general QoE function defined by Yin et al. [166] and Mao et al. [40], we measure the QoE as a function of the video bitrate, rebuffering time, and quality fluctuations per segment. The QoE of the (i)th segment is measured as follows.

$$QoE(i) = \alpha * \mathcal{Q}(\mathcal{L}_q(i)) - \beta * T(i) - \gamma * \mathcal{F}(i) \tag{4.9}$$

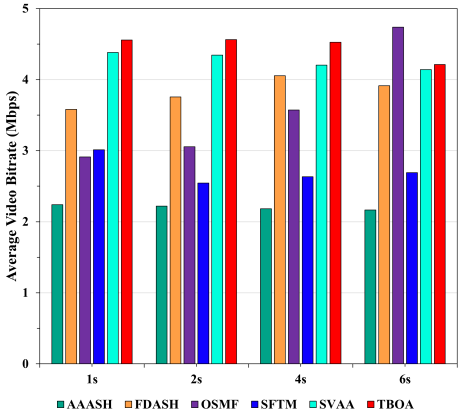
   where α, β, and γ are parameters indicating how much importance a user gives to video bitrate, rebuffering time, and quality fluctuations, respectively. $\mathcal{Q}(\mathcal{L}_q(i))$ maps the qth bitrate to the perceived quality levels, as indicated in Table 4.2. $T(i)$ represents the per segment rebuffering time measured as a ratio of buffer drainage time and segment duration. The value for the first function ranges between 0.16 and 1[4], while for the second function it is in the range of 0 and 1.
4. **Inefficiency**: The inefficiency metric represents the video bitrate requested

[4] A non-zero value was adopted for the lowest quality content, as it has a non-zero contribution to the user perceived quality.



(a) Bitrate Set $\mathcal{L}1$

(b) Bitrate Set $\mathcal{L}2$



(c) Bitrate Set $\mathcal{L}3$

Figure 4.5: Average video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link.

by multiple competing clients in response to the available bandwidth budget and is computed using Eq. 3.1.

5. **Unfairness**: The unfairness metric indicates the unfair bandwidth utilization by the competing clients and is computed using Eq. 3.2.
6. **Instability**: This metric reflects the quality fluctuations experienced by the competing clients and is measured as the ratio between the average number of quality switches of all the clients and the simulation time.

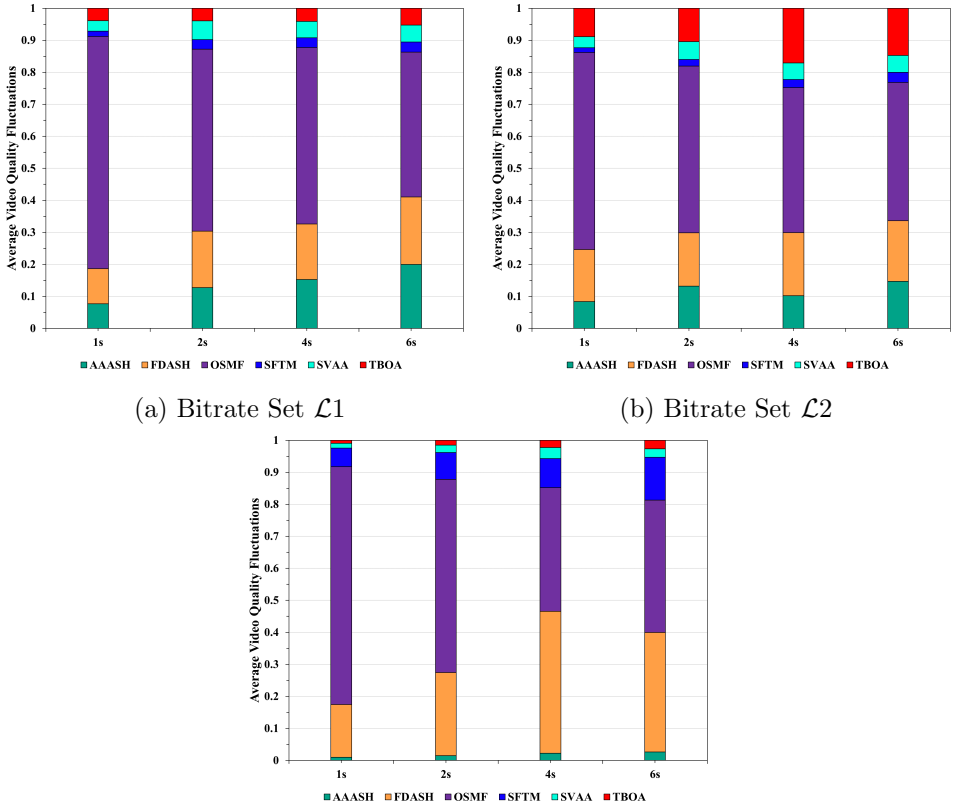
4.4.2 Experimental Results

This subsection describes how the proposed and other existing adaptation algorithms perform in single and multi-client environments in terms of the multiple defined evaluation metrics.

4.4.2.1 Single-Client Scenario - Wired Network

Fig. 4.5 presents the average video bitrates for AAASH, FDASH, OSMF, SFTM, SVAA, and **TBOA** clients over the wired network when the three bitrate sets $\mathcal{L}1$, $\mathcal{L}2$ and $\mathcal{L}3$ are used in turn and for different segment durations. It can be seen that **TBOA** achieves the highest bitrate in comparison to other adaptation algorithms for all three bitrate sets. The bitrates achieved by **TBOA** are the highest with bitrate set $\mathcal{L}2$ (Fig. 4.5b) followed by bitrate set $\mathcal{L}1$ (Fig. 4.5a). For bitrate set $\mathcal{L}3$ the average values are the lowest in Fig. 4.5c. For all three bitrate sets and for all segment durations **TBOA** achieves around 4.9 Mbps, compared to 3.8 Mbps of FDASH, 1.5 Mbps of AAASH, 4.53 Mbps of SVAA, 3.84 Mbps of OSMF, and 3.20 Mbps of SFTM clients. SVAA client achieves the second-best video bitrate after TBOA. The higher bitrates values achieved by **TBOA** indicate the highest accuracy in terms of bandwidth estimation and network link utilization. Among the compared solutions AAASH has the lowest bitrate and performs the worst for almost all segment durations. This is because it fetches minimum bitrate until the buffer goes above 10s and for the following segments it goes on with the lower bitrate values. It is interesting to note that with the increase in segment duration the average bitrate values are almost similar for **TBOA** and SVAA clients in Fig. 4.5a and Fig. 4.5b. However, the OSMF client observes higher bitrate values with the increase in segment duration. OSMF achieves the highest bitrate compared to all other solutions in Fig. 4.5c for 6s segment duration. This is because it abruptly switches to the highest bitrate of 17 Mbps and experiences 5.82 seconds of interruptions.

Fig. 4.6 represents the average number of quality fluctuations achieved by the various adaptation algorithms when the experiments are performed using different bitrate values. The average number of quality fluctuations for all bitrates and all segment durations in Fig. 4.6 are as follows: 4.06% for SVAA, 4.72% for SFTM, 6.3% for TBOA, 9.14% for AAASH, 21.88% for FDASH, and 53.86% for OSMF clients. OSMF solution has the highest number of quality changes compared to other solutions. This is because OSMF implements aggressive quality decrements without involving intermediate bitrates. SVAA, SFTM, and **TBOA** clients, on the



(a) Bitrate Set $\mathcal{L}1$

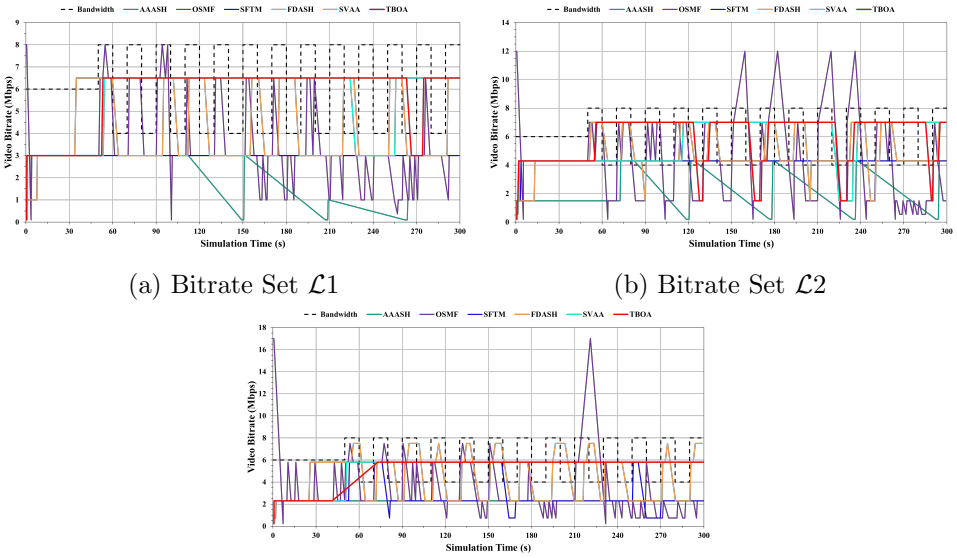
(b) Bitrate Set $\mathcal{L}2$

(c) Bitrate Set $\mathcal{L}3$

Figure 4.6: Average number of video quality switches achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link.

other hand, observe the lowest quality oscillations and offer a stable playback experience for all segment durations. The quality switches are the lowest with bitrate set $\mathcal{L}3$ (Fig. 4.6a) where **TBOA** and AAASH clients observe quality fluctuations during 1.85% of the video playback with all segment durationss. With bitrate set $\mathcal{L}3$, **TBOA** mostly requests segments with 5.8 Mbps (Table 4.2) as it is much less than the estimated average bandwidth values. However, for bitrate set $\mathcal{L}2$ in Fig. 4.6b, the requested bitrate is 7 Mbps which causes the buffer to fill up slowly when the bandwidth drops to 4 Mbps, and buffer drops results in quality variations. However, the achieved numbers of quality changes are acceptable with bitrate set $\mathcal{L}2$ as **TBOA** observes only 12.77% quality fluctuations compared to 17.88% of FDASH and 50.52 of OSMF clients. The SFTM client was stuck to the 4.3 Mbps most of the time and achieves only 3 quality switches at the cost of lower bitrate values.

Fig. 4.7 illustrates how the proposed and other solutions behave in response to band-

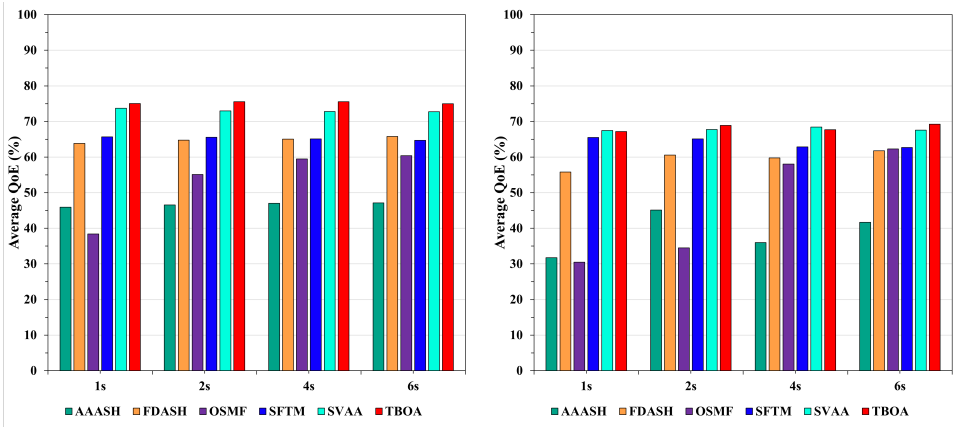


(a) Bitrate Set $\mathcal{L}1$ (b) Bitrate Set $\mathcal{L}2$

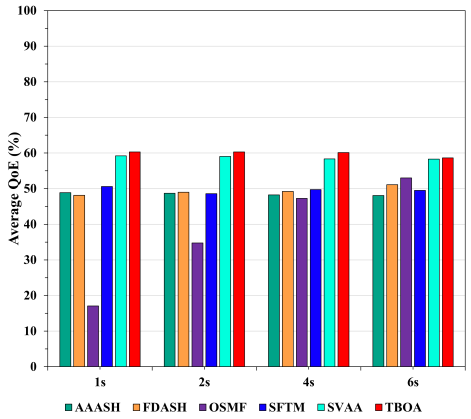
(c) Bitrate Set $\mathcal{L}3$

Figure 4.7: Per-segment video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link.

width variations *n terms of* per-segment video bitrate values. The figure presents the results with a 2s segment duration. As can be seen from Fig. 4.7a, the **TBOA** client quickly switches from 0.1 Mbps to 3 Mbps after 0.6s and then switches to 6.5 Mbps at 53.2s. **TBOA** retains this bitrate until 263s when the drop in buffer level causes it to switch back to 3 Mbps. However, at 274.9s it again achieves the highest bitrate. AAASH client switches mostly between 3 Mbps and 0.1 Mbps and is unable to respond to bandwidth fluctuations. In addition, the AAASH client experiences playback rebuffering or interruptions of 9.8s during the streaming session. With every change in bandwidth values, FDASH also switches rapidly between 6.5 and 3 Mbps. SVAA client switches from 6.5 Mbps to 3 Mbps at 225s and then increase to 6.5 Mbps after almost 50s. For bitrate set $\mathcal{L}2$ in Fig. 4.7b, **TBOA** switches down from 7 Mbps to 1.5 Mbps and then increases its bitrate to 4.3 Mbps and 7 Mbps. **TBOA** switches to 7 Mbps after 55.78s while the SVAA client achieves this bitrate after 116.52s. Similar to TBOA, SVAA also switches down from 7 Mbps to 1.5 Mbps and then requests 4.3 Mbps and eventually 7 Mbps. SFTM never reaches 7 Mbps and always stays on 4.3 Mbps after 3.5s of playback. Fig. 4.7c demonstrates how the most stable and high-quality behavior is achieved by **TBOA** client while OSMF and FDASH observe the least stable streaming experience.



(a) Bitrate Set $\mathcal{L}1$ (b) Bitrate Set $\mathcal{L}2$

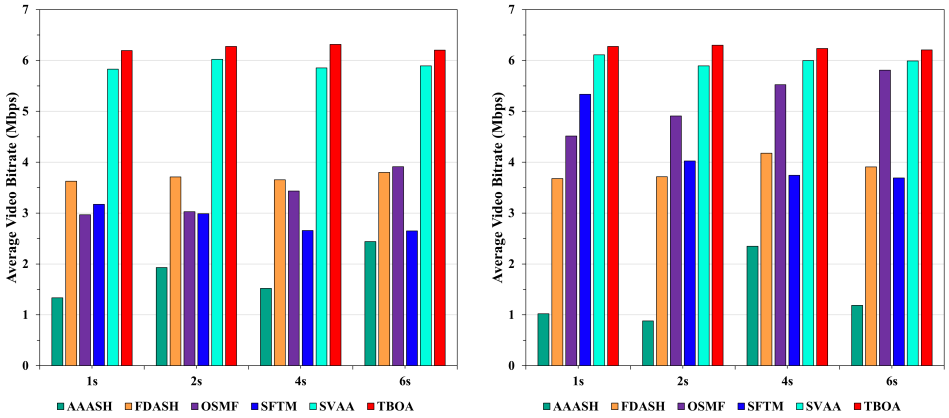


(c) Bitrate Set $\mathcal{L}3$

Figure 4.8: Average QoE achieved by **TBOA** and five streaming methods for three bitrate sets over a fluctuating wired bandwidth link.

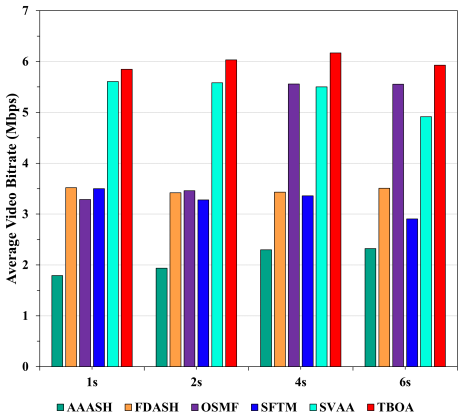
We measure the average QoE values in terms of percentages for the adaptation algorithms using Eq. 4.9. The weights for the average video quality, interruption time, and quality fluctuations are set to 41.66%, 29.16%, and 29.16%, respectively. This setting reflects a user having equal concerns about playback interruption time and quality fluctuations, however, a little more emphasis is given to the estimated video quality.

Fig. 4.8 shows the average QoE results for 1s, 2s, 4s, and 6s segment durations when a wired fluctuating link is employed. Fig. 4.8a reflect that **TBOA** outperforms AAASH, FDASH, OSMF, SFTM, and SVAA clients by about 28.62%, 10.43%, 21.92%, 10.02%, and 2.23%, respectively, for bitrate set $\mathcal{L}1$. The average QoE increments for bitrate set $\mathcal{L}1$ and set $\mathcal{L}2$ in Fig 4.8b and Fig. 4.8c are 21.81%, 9.62%, 20.54%, 7.22%, and 0.78% in comparison to AAASH, FDASH, OSMF, SFTM, and



(a) Bitrate Set $\mathcal{L}1$

(b) Bitrate Set $\mathcal{L}2$



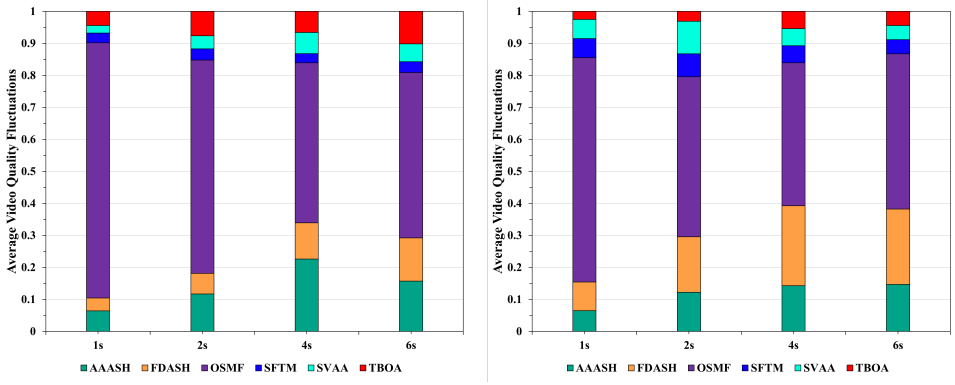
(c) Bitrate Set $\mathcal{L}3$

Figure 4.9: Average video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link.

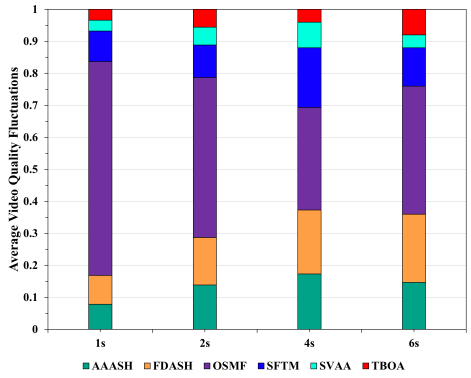
SVAA clients, respectively. The average QoE values in Fig 4.8c are lower than those depicted in Fig. 4.8b which are further lower than in Fig 4.8a. This is because the mapped quality levels are lower with achieved bitrates for QoE values in Fig. 4.8b and Fig. 4.8c. The average QoE value for all streaming solutions in Fig. 4.8a is 63.08%, compared to 57.42% in Fig. 4.8b, and 50.67% in Fig. 4.8c. From results achieved with wired bandwidth fluctuation scenario, it has been observed that **TBOA** obtains the highest average video bitrate and QoE without experiencing unnecessary quality switches.

4.4.2.2 Single-Client Scenario - Wireless Network

The adaptive behavior of the client in case of stable and constant bandwidth change in a wireless environment is shown in Fig. 4.9. The achieved bitrate values in Fig.

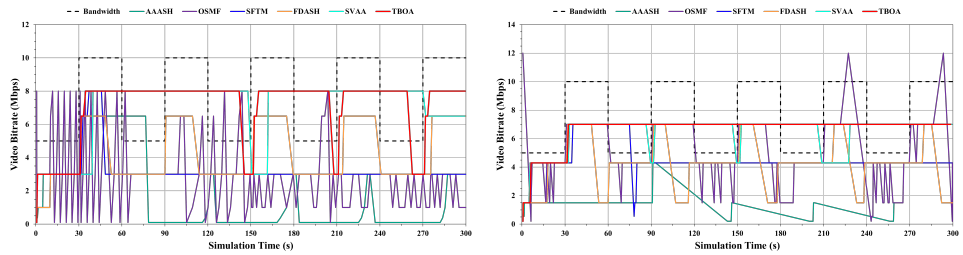


(a) Bitrate Set $\mathcal{L}1$ (b) Bitrate Set $\mathcal{L}2$

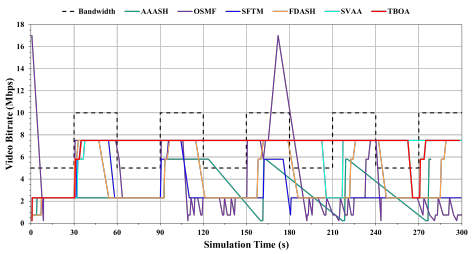


(c) Bitrate Set $\mathcal{L}3$

Figure 4.10: Average number of video quality switches achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link.



(a) Bitrate Set $\mathcal{L}1$ (b) Bitrate Set $\mathcal{L}2$

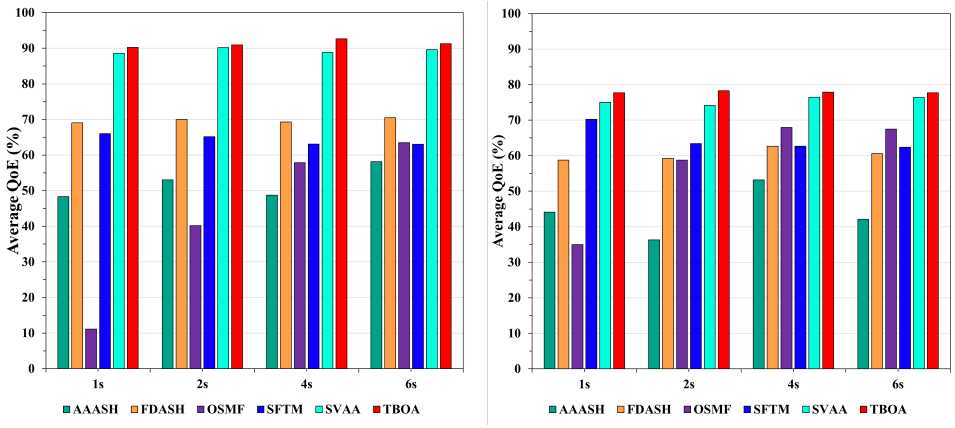


(c) Bitrate Set $\mathcal{L}3$

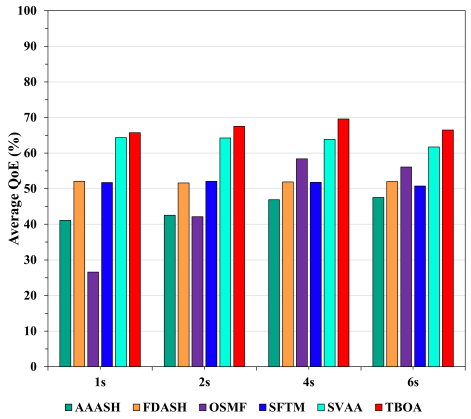
Figure 4.11: Per-segment video bitrate achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link.

4.9 are higher than in Fig. 4.5 due to the higher average bandwidth values. **TBOA** scheme achieves a higher average video bitrate for all segment lengths Fig. 4.9. In particular, **TBOA** outperforms FDASH, AAASH, SVAA, OSMF, and SFTM clients by achieving 2.48 Mbps, 4.41 Mbps, 0.39 Mbps, 1.83 Mbps, and 2.72 Mbps higher average video bitrate, respectively. Similar to Fig. 4.6, the average quality fluctuations in Fig. 4.10 are acceptable and do not lead to lower QoE values as reported in Fig. 4.12 for wireless environment. The average quality changes in **TBOA** across all segment lengths and bitrate sets are 6.42 compared to 14.58 for FDASH, 14.42 for AAASH, 6.25 for SVAA, 8.08 for SFTM, and 77 for OSMF. Following this, we show the per-segment requested bitrate values over a wireless bandwidth link in Fig. 4.11. The dominant behavior of **TBOA** client is clear from Fig. 4.11a-Fig. 4.11b. In Fig. 4.11a **TBOA** switches to the highest available bitrate (8 Mbps) in 34.68s compared to 36.48s of SFTM, and 39.81s of SVAA. FDASH and SFTM never request 8 Mbps bitrate and only touch the 6.5 Mbps bitrate value. OSMF aggressively switches between 8 Mbps and 0.1 Mbps during the first 60s. In contrast to Fig. 4.7b, **TBOA** achieves a stable playback experience with 3 quality switches from 0.1 Mbps to 1.5 Mbps, 1.5 Mbps to 4.3 Mbps, and 4.3 Mbps to 7 Mbps and continues until the end of the simulation. In Fig. 4.11c **TBOA** continues to play segments with 7.5 Mbps bitrate and expects the buffer to go below b_l until 262.37s before decreasing the bitrate to obtain a steady performance. OSMF client observes a buffer depletion of 0.81s and 3.18s for bitrate set $\mathcal{L}2$ and bitrate set $\mathcal{L}3$, respectively. However, none of the other solutions observes interruptions for bitrate set $\mathcal{L}2$.

By computing the average QoE scores with the same weights as for the wired network link in Fig. 4.8, the results for the wireless scenario are depicted in Fig. 4.12. Interestingly, the QoE values with **TBOA** solution are the highest for all performed experiments. In Fig. 4.9b and Fig. 4.9c, the highest bitrate values of OSMF client after TBOA and SVAA solutions for 4s and 6s segment lengths lead to the third-highest QoE values for 4s and 6s segment durations, as depicted in Fig. 4.12b and Fig. 4.12c. However, for the 2s segment duration, the higher quality fluctuations penalty does not improve the quality score. Moreover, with 1s segment duration,



(a) Bitrate Set $\mathcal{L}1$ (b) Bitrate Set $\mathcal{L}2$



(c) Bitrate Set $\mathcal{L}3$

Figure 4.12: Average QoE achieved by **TBOA** and five streaming methods for three bitrate sets over a stable and constantly changing wireless link.

the lowest QoE values are observed for the OSMF client, since the higher number of quality fluctuations drops the resulting QoE levels. The rebuffering events are experienced only by the AAASH client for about 2.34s during the streaming with 6s segment for bitrate set $\mathcal{L}1$. OSMF client undergoes interruptions for about 1.92s and 6.40s with a 4s segment duration and about 3.45s and 10.21s with a 6s segment duration for bitrate set $\mathcal{L}2$ and $\mathcal{L}3$, respectively.

4.4.2.3 Multi-Client Scenario

In a multi-client streaming environment, multiple DASH video segments are transmitted over the network. We simulated three and five DASH clients in a wired network environment over 300s. When multiple clients share a full bandwidth link, the clients must be able to select the highest possible video quality representations.

Table 4.3: Comparison of **TBOA** and Five Adaptive Solutions for Three Streaming Clients Using Bitrate Set $\mathcal{L}1$

Algorithms	Clients	Average Video Bitrate (Mbps)	Quality Fluctuations per-Session	Average QoE (%)	Inefficiency	Unfairness	Instability
AAASH	Client 1	2.73	3.00	63.72	0.50	0.48	0.03
	Client 2	0.46	24.00	30.11			
	Client 3	2.75	3.00	63.87			
FDASH	Client 1	2.67	3.00	63.21	0.33	0.00	0.01
	Client 2	2.67	3.00	63.21			
	Client 3	2.65	2.00	63.32			
OSMF	Client 1	2.98	4.00	65.57	0.25	0.00	0.01
	Client 2	3.00	5.00	65.50			
	Client 3	2.98	4.00	65.57			
SFTM	Client 1	2.63	3.00	62.88	0.32	0.05	0.01
	Client 2	2.62	3.00	62.77			
	Client 3	2.91	3.00	65.21			
SVAA	Client 1	2.87	3.00	64.88	0.27	0.01	0.01
	Client 2	2.95	3.00	65.55			
	Client 3	2.94	3.00	65.43			
TBOA	Client 1	2.88	2.00	65.16	0.27	0.01	0.00
	Client 2	2.94	1.00	65.95			
	Client 3	2.94	1.00	65.95			

Table 4.4: Comparison of **TBOA** and Five Adaptive Solutions for Three Streaming Clients Using Bitrate Set $\mathcal{L}2$

Algorithms	Clients	Average Video Bitrate (Mbps)	Quality Fluctuations per-Session	Average QoE (%)	Inefficiency	Unfairness	Instability
AAASH	Client 1	1.46	3.00	48.55	0.67	0.16	0.03
	Client 2	1.48	6.00	48.22			
	Client 3	1.03	16.00	37.97			
FDASH	Client 1	3.11	9.00	57.45	0.24	0.02	0.03
	Client 2	3.01	11.00	56.39			
	Client 3	2.98	7.00	57.18			
OSMF	Client 1	1.61	72.00	33.79	0.60	0.01	0.23
	Client 2	1.58	66.00	35.01			
	Client 3	1.58	68.00	34.57			
SFTM	Client 1	3.20	3.00	59.42	0.29	0.16	0.01
	Client 2	3.16	3.00	59.20			
	Client 3	2.21	3.00	53.52			
SVAA	Client 1	1.48	4.00	48.80	0.60	0.10	0.01
	Client 2	1.82	2.00	51.44			
	Client 3	1.48	6.00	48.33			
TBOA	Client 1	3.04	2.00	58.70	0.27	0.05	0.01
	Client 2	2.72	4.00	56.35			
	Client 3	3.04	2.00	58.70			

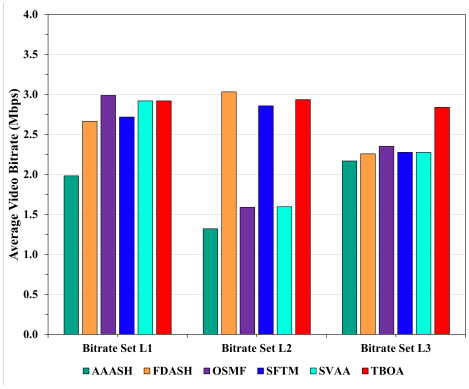
Ideally, a lower value of inefficiency is desired, which reflects that multiple clients have the highest possible bitrate for a high streaming experience. Similarly, closer to zero value of instability reveals that competing clients obtain a smooth playback experience.

Three Streaming Clients:

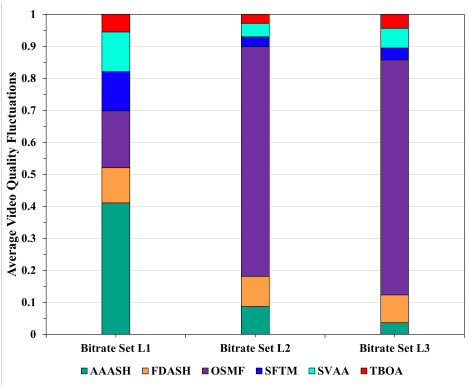
The streaming experiences of three competing clients with three bitrate sets are shown in Table 4.3-Table 4.5 and Fig. 4.13. Table 4.3 and Fig. 4.13 show that **TBOA** provides the highest average QoE and the lowest number of quality changes for each streaming client when tested for bitrate set $\mathcal{L}1$. OSMF performs quite well for 3 users and achieves an average video bitrate of 2.99 Mbps compared to 2.92 Mbps

Table 4.5: Comparison of **TBOA** and Five Adaptive Solutions for Three Streaming Clients Using Bitrate Set $\mathcal{L}3$

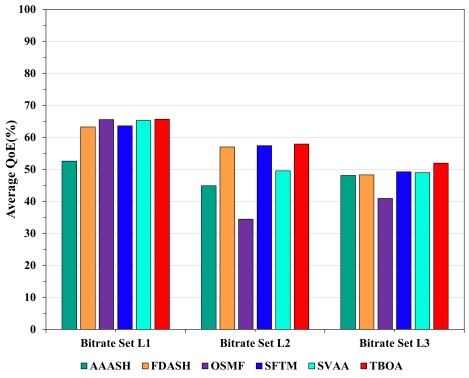
Algorithms	Clients	Average Video Bitrate (Mbps)	Quality Fluctuations per-Session	Average QoE (%)	Inefficiency	Unfairness	Instability
AAASH	Client 1	2.17	2.00	48.12	0.46	0.00	0.01
	Client 2	2.17	2.00	48.12			
	Client 3	2.17	2.00	48.12			
FDASH	Client 1	2.15	3.00	47.70	0.44	0.04	0.02
	Client 2	2.25	3.00	48.72			
	Client 3	2.37	8.00	48.48			
OSMF	Client 1	2.34	40.00	40.82	0.41	0.00	0.13
	Client 2	2.36	36.00	41.85			
	Client 3	2.35	43.00	40.16			
SFTM	Client 1	2.28	2.00	49.27	0.43	0.00	0.01
	Client 2	2.28	2.00	49.27			
	Client 3	2.28	2.00	49.27			
SVAA	Client 1	2.28	2.00	49.27	0.43	0.00	0.01
	Client 2	2.28	4.00	48.80			
	Client 3	2.28	4.00	48.80			
TBOA	Client 1	2.90	2.00	52.38	0.29	0.16	0.01
	Client 2	2.26	1.00	49.32			
	Client 3	3.36	4.00	54.12			



(a) Average video bitrate for three clients



(b) Average number of quality fluctuations for three clients



(c) Average QoE for three clients

Figure 4.13: Average video bitrate, quality fluctuations, and QoE achieved by three streaming clients of **TBOA** and comparative streaming solutions for three bitrate sets.

of **TBOA** client (as shown in Fig. 4.13a). Along with this, the inefficiency value for OSMF clients is the lowest. AAASH clients observe the highest number of quality

switches and the lowest bitrate and therefore have the lowest average QoE values for bitrate set $\mathcal{L}1$. SVAA client observes a similar video bitrate as of TBOA; however, it experiences more than double the number of quality switches. For the bitrate set $\mathcal{L}2$, FDASH, TBOA, and SFTM clients have the highest average video bitrate (Fig. 4.13a). However, due to the lower quality fluctuations, TBOA holds the slightly higher QoE values for all streaming clients, as shown in Fig. 4.13a. OSMF records the worst performance experience and achieves an average QoE of around 34.45% due to the highest instability of the competing clients Table 4.4. **TBOA** and SVAA clients share similar instability values; however, in SVAA-based streaming, the QoE is about 8.39% lower than that of TBOA. The TBOA unfairness is the third-best among the competing solutions. For the bitrate set $\mathcal{L}3$, the average video bitrate of **TBOA** clients is 2.84 Mbps (Fig. 4.13a) better than those of the OSMF and FDASH clients, at 2.35 and 2.26 Mbps, respectively. **TBOA** achieves the lowest instability and inefficiency values of 0.29 and 0.01, respectively (Table 4.5). The total number of quality switches is the lowest for the SFTM and TBOA solutions (Fig. 4.13b). The highest unfairness of **TBOA** is due to the fact that high buffer levels for a client encourage it to switch from the lowest to the maximum possible levels even if the remaining clients are at low buffer levels. However, this happens only for a short duration of time when increasing from B_0 to B_l. From results in Table 4.3-Table 4.5 and Fig. 4.13, it can be concluded that **TBOA** records the highest streaming performance for all three clients across all employed bitrate sets.

Five Streaming Clients:

Ideally, the competing clients should be able to play high-resolution segments when sharing the bottleneck bandwidth link. However, the non-cooperative clients inaccurately estimate the bandwidth, resulting in lower video quality, higher number of video quality switches, inefficient bandwidth utilization, and unfair QoE values. Table 4.6 shows the behavior of the adaptive algorithms when five competing clients request the segments with a 2s duration. The experimental results in Fig. 4.14 show that **TBOA** attains the most eminent adaptive behavior by achieving the highest average bitrate, the lowest number of video quality switches, and the highest QoE with bitrate set $\mathcal{L}1$ for all the clients in comparison to other solutions. Similarly,

Table 4.6: Comparison of **TBOA** and Five Adaptive Solutions for Five Streaming Clients Using Bitrate Set $\mathcal{L}1$

Algorithms	Clients	Average Video Bitrate (Mbps)	Quality Fluctuations per-Session	Average QoE (%)	Inefficiency	Unfairness	Instability
AAASH	Client 1	0.22	18.00	19.82	0.59	0.53	0.06
	Client 2	0.48	18.00	31.80			
	Client 3	1.68	13.00	52.62			
	Client 4	1.69	13.00	52.73			
	Client 5	0.80	21.00	39.86			
FDASH	Client 1	2.15	8.00	57.75	0.30	0.28	0.02
	Client 2	2.29	8.00	58.88			
	Client 3	1.01	7.00	48.44			
	Client 4	1.34	7.00	51.16			
	Client 5	1.55	5.00	53.41			
OSMF	Client 1	1.04	41.00	40.76	0.53	0.06	0.16
	Client 2	1.20	49.00	40.16			
	Client 3	1.18	63.00	36.73			
	Client 4	1.05	34.00	42.44			
	Client 5	1.16	51.00	39.36			
SFTM	Client 1	0.99	7.00	48.09	0.51	0.19	0.03
	Client 2	1.39	8.00	51.36			
	Client 3	1.49	12.00	51.30			
	Client 4	0.99	12.00	46.92			
	Client 5	0.99	9.00	47.62			
SVAA	Client 1	0.99	4.00	48.79	0.51	0.23	0.02
	Client 2	1.71	9.00	53.81			
	Client 3	1.20	4.00	50.76			
	Client 4	0.99	3.00	49.03			
	Client 5	0.99	3.00	49.03			
TBOA	Client 1	2.29	2.00	60.27	0.21	0.44	0.01
	Client 2	2.76	2.00	64.16			
	Client 3	0.63	4.00	39.13			
	Client 4	2.84	2.00	64.83			
	Client 5	0.98	1.00	49.29			

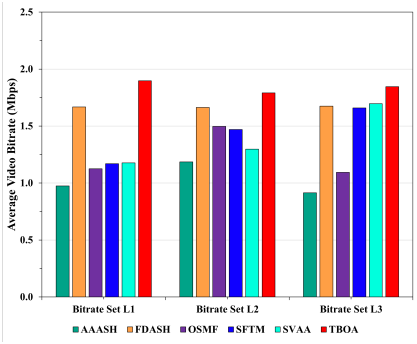
Table 4.7: Comparison of **TBOA** and Five Adaptive Solutions for Five Streaming Clients Using Bitrate Set $\mathcal{L}2$

Algorithms	Clients	Average Video Bitrate (Mbps)	Quality Fluctuations per-Session	Average QoE (%)	Inefficiency	Unfairness	Instability
AAASH	Client 1	0.59	19.00	29.53	0.51	0.26	0.02
	Client 2	1.41	2.00	48.00			
	Client 3	1.41	2.00	48.00			
	Client 4	1.11	10.00	40.74			
	Client 5	1.41	2.00	47.92			
FDASH	Client 1	1.04	6.00	40.50	0.31	0.19	0.02
	Client 2	1.84	5.00	50.83			
	Client 3	1.92	6.00	51.09			
	Client 4	1.79	6.00	50.30			
	Client 5	1.74	6.00	50.03			
OSMF	Client 1	1.55	37.00	41.65	0.38	0.07	0.06
	Client 2	1.55	14.00	46.99			
	Client 3	1.55	8.00	48.43			
	Client 4	1.55	13.00	47.26			
	Client 5	1.28	16.00	42.28			
SFTM	Client 1	1.48	3.00	49.03	0.39	0.01	0.01
	Client 2	1.48	3.00	49.01			
	Client 3	1.46	2.00	48.82			
	Client 4	1.46	2.00	48.82			
	Client 5	1.46	3.00	48.59			
SVAA	Client 1	1.48	2.00	49.27	0.46	0.27	0.01
	Client 2	0.57	5.00	32.46			
	Client 3	1.48	2.00	49.16			
	Client 4	1.48	2.00	49.16			
	Client 5	1.48	2.00	49.16			
TBOA	Client 1	1.47	1.00	49.31	0.25	0.28	0.01
	Client 2	1.96	3.00	52.03			
	Client 3	1.32	5.00	45.74			
	Client 4	2.74	4.00	56.43			
	Client 5	1.46	2.00	48.85			

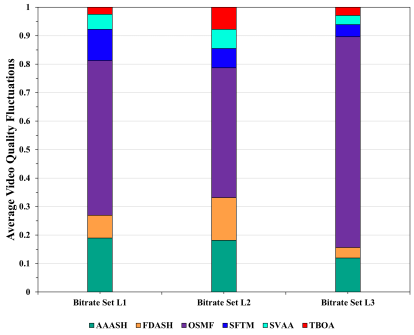
Table 4.8: Comparison of **TBOA** and Five Adaptive Solutions for Five Streaming Clients Using Bitrate Set $\mathcal{L}3$

Algorithms	Clients	Average Video Bitrate (Mbps)	Quality Fluctuations per-Session	Average QoE (%)	Inefficiency	Unfairness	Instability
AAASH	Client 1	1.08	5.00	35.65	0.62	0.20	0.03
	Client 2	0.76	16.00	29.64			
	Client 3	0.63	13.00	26.47			
	Client 4	1.04	6.00	34.98			
	Client 5	1.07	5.00	35.52			
FDASH	Client 1	1.18	4.00	37.02	0.30	0.24	0.01
	Client 2	1.16	4.00	36.79			
	Client 3	2.01	2.00	46.44			
	Client 4	2.01	2.00	46.45			
	Client 5	2.01	2.00	46.45			
OSMF	Client 1	1.09	66.00	21.50	0.54	0.09	0.19
	Client 2	1.14	54.00	24.89			
	Client 3	1.24	42.00	28.70			
	Client 4	0.94	53.00	22.97			
	Client 5	1.06	64.00	21.65			
SFTM	Client 1	1.26	3.00	38.09	0.31	0.23	0.01
	Client 2	2.03	2.00	46.59			
	Client 3	2.03	2.00	46.59			
	Client 4	1.13	4.00	36.53			
	Client 5	1.85	5.00	43.96			
SVAA	Client 1	0.55	3.00	26.04	0.29	0.37	0.01
	Client 2	2.23	2.00	48.83			
	Client 3	2.20	2.00	48.49			
	Client 4	2.20	2.00	48.49			
	Client 5	1.30	3.00	38.51			
TBOA	Client 1	1.25	4.00	37.79	0.23	0.25	0.01
	Client 2	2.26	1.00	49.32			
	Client 3	2.22	2.00	48.64			
	Client 4	2.22	2.00	48.64			
	Client 5	1.29	2.00	38.63			

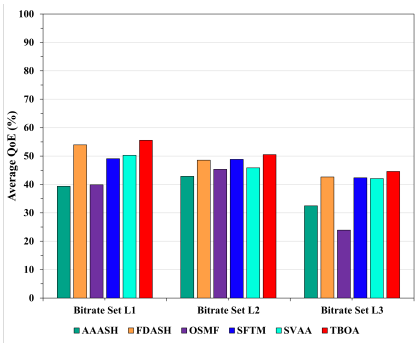
TBOA makes efficient use of the available network bandwidth to select the bitrate for future video segments (Table 4.6). Furthermore, **TBOA** has the lowest instability value, which represents that all five clients continue to play segments without switching much between different quality levels. Thus, **TBOA** is the most efficient algorithm for all five clients with the bitrate set $\mathcal{L}1$. The AAASH and OSMF client observe the lowest streaming experience with bitrate set $\mathcal{L}1$. Table 4.7 depicts the adaptive behavior of algorithms for bitrate set $\mathcal{L}2$. FDASH clients achieve nearly the same video bitrate, however, the instability and inefficiency values are higher with this method. Interestingly, SVAA client that performed well in a single-client environment faces a lower bitrate experience, especially for bitrate set $\mathcal{L}1$ and $\mathcal{L}2$ (Fig. 4.14a). OSMF performing worst in single-client streaming experience has somehow improved in the multi-client environment for bitrate set $\mathcal{L}2$ due to the relatively lower number of quality fluctuations. OSMF experiences lower inefficiency and unfairness compared to SVAA clients. Next, for bitrate set $\mathcal{L}3$ in Fig. 4.14a, **TBOA** is the most efficient algorithm and results in 0.17 Mbps, 0.93 Mbps, 0.15 Mbps, 0.75 Mbps, and 0.19 Mbps higher average video bitrate in comparison to



(a) Average video bitrate for five clients



(b) Average number of quality fluctuations for five clients



(c) Average QoE for five clients

Figure 4.14: Average video bitrate, quality fluctuations, and QoE achieved by five streaming clients of **TBOA** and comparative streaming solutions for three bitrate sets.

FDASH, AAASH, SVAA, OSMF, and SFTM solutions, respectively. OSMF and AAASH achieve the lowest unfairness values in comparison to other solutions (Table 4.8). However, the achieved inefficiency and instability scores are the highest among other solutions. This means that lower unfairness values do not guarantee the lowest inefficiency and instability values. This is because AAASH and OSMF maintain the lowest equal bitrates for all clients. **TBOA** on the other hand, provides a useful trade-off between inefficiency, instability, and unfairness while supporting higher QoE levels, as shown in Fig. 4.14c. Therefore, streaming of traditional videos to a single or multiple clients with **TBOA** solution under various bitrate levels ensures higher perceived quality levels and extended playback experience.

4.5 Conclusions

This chapter presented the proposed novel **TBOA** solution, an efficient throughput and buffer-occupancy-based adaptation algorithm for traditional videos. **TBOA** utilizes network throughput and playback buffer levels to perform appropriate video bitrate selection to improve end-user QoE assessed in terms of delivering best quality segments and continuous playback with the reduced number of video quality switches, in both single and multiple client scenarios. We assess the performance of the proposed hybrid adaptation algorithm **TBOA** in comparison to other throughput-based (SFTM), buffer-based (SVAA), and hybrid (AAASH, FDASH, and OSMF) adaptation algorithms. We considered three different bitrate sets ($\mathcal{L}1$, $\mathcal{L}2$, $\mathcal{L}3$) and four segment durations (1s, 2s, 4s, and 6s) for the Big Buck Bunny video sequences to analyze the streaming performance under both fluctuating wired and stable wireless network environments for a single client. The testing results for single-client scenarios show that **TBOA** outperforms the traditional streaming approaches by attaining higher video bitrates, reduced number of video quality switches, and improved QoE levels. Moreover, we evaluated the streaming performance in a multi-client environment involving three and five clients watching 2s segments with three bitrate levels for a duration of 300s. For this scenario, we measured video bitrates, quality fluctuations, rebuffering time, QoE, inefficiency, unfairness, and instability for all the clients. **TBOA** clients efficiently and stably utilize the available bandwidth budget to achieve higher video bitrates and a lower number of quality fluctuations in comparison with other solutions. Slightly higher unfairness values were observed for three clients with bitrate set $\mathcal{L}3$ and for five clients with bitrate set $\mathcal{L}1$. However, this slightly higher unfairness is well compensated by the lowest inefficiency and instability values. Overall, the traditional video streaming with **TBOA** solution is a promising way to achieve high quality and performance of video content streaming.

Chapter 5

Priority-aware DASH-based Multi-view Adaptation

This chapter presents the proposed MVP-DASH, a DASH-based multi-view video streaming solution to improve resources utilization for multiple concurrent video streams. The MVP-DASH streaming architecture and algorithms and its performance evaluation using multiple wireless channels are presented. The major components of the proposed solution are: (i) a throughput estimation method to estimate the available network capacity for each stream based on their priority-related weights; (ii) a unified segment scheduling mechanism to schedule the segments of selected streams in the next transmission cycle; (iii) a bitrate computation and allocation approach for multiple stream segments; and (iv) a path selection mechanism to enable high-quality transmission of high-priority streams. The proposed solution enables efficient resource utilization and enhanced viewing experience, as demonstrated by simulation-based results with parallel streaming of three video clips.

5.1 Introduction

Multi-view video streaming enables concurrent transmission of multiple streams to the remote user not only for exploration of immersive settings, but also for creating novel and exciting business and entertainment opportunities. The latest interest in multi-view video streaming solutions, including those for telepresence, commercial conferencing, remote collaboration, etc. requires support for the high-quality delivery of large amounts of content data. This requirement adds pressure to the overall existing network capacity of serving higher-resolution appropriate solutions for adaptive streaming of such multi-video streaming applications over the limited network infrastructure. Along with the traditional content capturing, segmentation, encoding, and transmission phases, enabling priority of streaming is crucial for an immersive viewing experience. The streams associated with priority views can be streamed at high-resolutions, while the streams corresponding to other views can be streamed at relatively lower resolutions for instance to mitigate the effects of limited available bandwidth. At the same time, the extensive proliferation of wireless network access technologies and availability of multiple network interfaces on modern devices require support for increased network transmission performance over various access networks. Diverse multipath-based adaptive multi-view video streaming solutions have been proposed, but they do not focus on how to implement priority-based throughput budget allocation [11] or use multiple network interfaces for prioritized bitrate streaming [185]. These aspects are fundamental when trying to secure a high-level of user satisfaction for adaptive multi-view video streaming.

This chapter presents the MVP-DASH solution which enables efficient resource utilization for enhanced viewing of multi-view videos. To achieve prioritized bitrate adaptation in a multipath streaming environment, first, a throughput estimation mechanism is proposed that is responsible for throughput estimation for each stream. This will help the multi-view client to select higher video bitrates for high-priority streams based on higher predicted throughput. Second, a unified segment scheduling mechanism in MVP-DASH is introduced to implement continuous and coordinated playback when multiple video streams are requested by a single client. Next, the

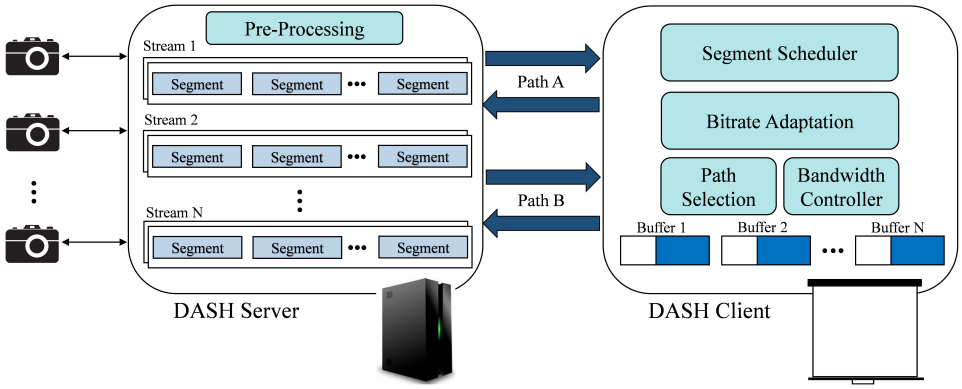


Figure 5.1: The client-server model for multi-view video streaming.

MVP-DASH bitrate adaptation algorithm is proposed to perform differentiated quality allocations by selecting the most appropriate bitrates for selected segments of multiple streams to improve the overall streaming experience. Last but not the least important, the MVP-DASH path selection mechanism is introduced to ensure that higher link rate channels are being utilized for the delivery of important segments.

5.2 Proposed MVP-DASH Framework

5.2.1 System Architecture

Fig. 5.1 illustrates the streaming architecture for an adaptive multi-view system. The DASH server stores different streams captured from multiple cameras encoded at different quality levels and divided into fixed duration segments. The DASH client utilizes multiple available paths to download multiple streams simultaneously. The client includes multiple components for supporting the streaming process, such as *Bandwidth Controller*, *Path Controller*, *Bitrate Adaptation*, and *Segment Scheduler*. The components are used in common by the multiple streams, but each video stream has an independent *Playback Buffer*.

The *Bandwidth Controller* module estimates the available bandwidth budget for each stream based on priority weights. The *Path Controller* module monitors the status of each path and organizes the paths, such as the high-priority segments will be requested over the best quality paths. The *Bitrate Adaptation* controller takes input from *Playback Buffer* and *Bandwidth Controller* units to determine the suitable

video quality for each segment of multiple streams. Finally, the *Segment Scheduler* controls the request interval for each segment of multiple streams to address synchronization. This is essential in a multi-view streaming environment.

5.2.2 Problem Formulation

In a dynamic network environment, the goal is to deliver multiple video streams at bitrates according to streams' associated priority levels, so that the delivered aggregate quality of the respective multiple streams is maximized. Let $V = \{V_1, V_2, ..., V_n, V_{n+1}, ..., V_N\}$ represent the set of N videos which are divided into two levels: high-priority streams and low-priority streams, arranged in descending priority order. Each video is segmented into a number of equal duration segments, i.e., for video V_n, the segments are: $S_n = \{S_n(1), S_n(2), ..., S_n(i), S_n(i+1), ..., S_n(I)\}$, and encoded into various representations, i.e., for video V_n, the representations are: $\mathcal{L} = \{\mathcal{L}_{1,\mathrm{n}}, \mathcal{L}_{2,\mathrm{n}}, ..., \mathcal{L}_{\mathrm{q,n}}, \mathcal{L}_{\mathrm{q+1,n}}, ..., \mathcal{L}_{\mathrm{Q,n}}\}$. The server transmits the N streams simultaneously to a user over M available paths, i.e., $P = \{P^1, P^2, ..., P^j, P^{j+1}, ..., P^M\}$. Let $\mathcal{L}^j_{q,l}(i)$ and $\mathcal{L}^j_{q,h}(i)$ represent the (q)th bitrate of (i)th segment over (j)th path for low- and high-priority streams, respectively. The main objective is to optimize the quality of the selected streams according to their priority levels.

The problem can be expressed as follows:

$$\underset{\mathcal{L}_{q,n}}{max} \sum_{n=1}^{N} \sum_{j=1}^{M} \sum_{i=1}^{I} \sum_{q=1}^{Q} w^j_{q,n}(i) * \mathcal{L}^j_{q,n}(i) \tag{5.1}$$

s.t.

$$\mathcal{L}^j_{q,n}(i) \leq \widehat{B}^j(i) * (\eta_n(i)) \tag{5.2}$$

$$\sum_{n=1}^{R^j(i)} \mathcal{L}^j_{q,n}(i) \leq \widehat{B}^j(i) \tag{5.3}$$

$$w^j_{q,n}(i) * \mathcal{L}^j_{q,n}(i) > 0, \quad \forall\ w^j_{q,n}(i) > 0, \quad w^j_{q,n}(i) \epsilon \{0, 1\} \tag{5.4}$$

$$\mathcal{L}^{j}_{q,l}(i) \leq \mathcal{L}^{j}_{q,h}(i) \tag{5.5}$$

where $R^j(i)$ represents the number of streams selected over (j)th path and $\widehat{B}^j(i)$ is the total estimated throughput of (j)th path. The constraints in Eq. 5.2 and Eq. 5.3 in the optimization problem restrict the selected bitrate for the (i)th segment of (n)th stream and all the selected streams (i.e., $R^j(i)$) over (j)th path, respectively. The priority weight $\eta_n(i)$ in constraint 5.2 plays an important role in bitrate selection for each stream. Particularly, the high-priority streams should be downloaded at high resolutions and the remaining streams at relatively low resolutions. The throughput and priority weight calculation for streams is described in section 5.3.1. Constraint 5.4 defines stream selection: $w^j_{q,n}(i) = 1$ denotes the segment (i) of (n)th stream at (q)th bitrate is selected for streaming over (j)th path, and $w^j_{q,n}(i) = 0$, otherwise. Constraint 5.5 ensures that the bitrates of low-priority streams are not higher than those of high-priority streams.

In order to solve the above problem, the proposed solution includes the following phases:

1. Throughput estimation for multiple streams based on their priority-related weight.
2. Unified segment scheduling to schedule the segments of selected streams in the next transmission cycle.
3. Computation of the video quality for multiple streams.
4. Path selection to improve the transmission of high-priority streams.

The next section elaborates on these aspects by proposing solutions for each of them.

5.3 Proposed MVP-DASH Approach

5.3.1 Throughput Estimation Scheme

The requests for video segments of multiple streams are treated within the same transmission cycle. The throughput estimation for (i)th segment of (n)th video

stream over (j)th path is calculated in Eq. (5.6).

$$Th_n^j(i) = \frac{\mathcal{L}_{q,n}^j(i-1) * \tau}{D_n(i-1)} \tag{5.6}$$

where $\mathcal{L}_{q,n}^j(i-1)$ represents the (q)th bitrate for (i-1)th segment of (n)th video over (j)th path, τ is the segment duration, and $D_n(i-1)$ indicates the download time of the $(i-1)$th segment for (n)th video stream.

As network throughput is highly variable, computing average throughput enables to reduce the influence of this variation on any decision making. Following a series of throughput measurements, the average throughput is given in Eq. (5.7) [38]:

$$\overline{Th}_n^j(i) = \frac{1}{k} * \sum_{r=i-k+1}^{i} Th_n^j(r) \tag{5.7}$$

where k represents the number of downloaded segments. The total bandwidth for N streams over path j for each segment i is calculated as in Eq. (5.8):

$$\widehat{B}^j(i) = \sum_{n=1}^{N} \overline{Th}_n^j(i) \tag{5.8}$$

In the proposed throughput estimation scheme, when multiple streams are requested over the bottleneck link, the high- and low-priority streams will get a higher and lower percentage of the estimated network throughput, respectively. The throughput estimation algorithm is presented in Algorithm 5.1. First, the maximum number of streams over the (j)th path is calculated. In a single path scenario, initially, $R^j(i)$ is equal to the total number of streams, and later on, may vary according to the network conditions. In a multipath scenario, the value of $R^j(i)$ is according to the number of streams over the selected path based on the priority level (elaborated in section 5.3.4). The number of low- and high-priority streams is recalculated if the bandwidth does not support the minimum bitrate for all streams over path j.

Next, the priority weights for each stream over the selected path are computed, such as the high-priority streams will have higher weights compared to lower priority streams since the user is mostly interested to view the high-priority streams in higher quality compared to other streams. The path estimated bandwidth will be equally

Algorithm 5.1 Throughput Estimation Algorithm

Input :
$R^j(i)\leftarrow$ *Number of streams to be requested over* (j)*th path*
$H^j(i)\leftarrow$*Number of high-priority streams over* j*th path*
$L^j(i)\leftarrow$*Number of low-priority streams over* j*th path*
$\widehat{B}^j(i)\leftarrow$ *Total estimated bandwidth of the* (j)*th path*
Output:
$R(i)\leftarrow$ *Number of streams to be requested over all the paths*
$\eta_h(i)\leftarrow$*Priority weight for high-priority stream over* (j)*th path*
$\eta_l(i)\leftarrow$ *Priority weight for low-priority stream over* (j)*th path*
$\widehat{Th}_n^j(i)\leftarrow$ *Throughput estimated for the* (n)*th stream*

```
for j ← 1 to M do
    if (∑_{n=1}^{R^j(i)} L^j_{1,n}(i) > B̂^j(i)) then
        R^j(i) = floor (B̂^j(i)/L^j_1(i))
        Recalculate H^j and L^j
    else
        R^j(i)= R^j(i)
    end
    η_l(i) = 1/(H^j(i) + R^j(i)) forL^j(i) > 0
    η_h(i) = (1 − ∑_l η_l(i))/H^j(i) forH^j(i) > 0
    if (R^j(i) == 1) then
        T̂h^j_n(i) = B̂^j(i)
    else
        for n ← 1 to R^j(i) do
            T̂h^j_n(i) = B̂^j(i) * η_n(i)
        end
    end
end
R(i) = ∑_{j=1}^{M} R^j(i)
```

shared between streams in case of only low- or high-priority streams over path j. If a single stream is requested over path j then the entire estimated path bandwidth will be used by that stream. Otherwise, the bandwidth share is calculated for each low- and high-priority streams based on their priority weights and the estimated path bandwidth. Finally, the total number of streams over all the paths is calculated.

5.3.2 Unified Segment Scheduling Strategy

In a multi-view video streaming environment, there is a need for an efficient scheduling strategy to maintain the playback synchronization between multiple streams. The periodic scheduling regularly downloads the HTTP segments to sustain a target buffer level [270]. However, this could result in frequent ON-OFF intervals which

could lead to sub-optimal throughput estimations. Following [41], the proposed segment scheduler avoids the unnecessary ON-OFF events by requesting segments of all streams in a uniform fashion.

Let t^{end}(i) and $D(i)$ be the transmission end time and download time of the segment (i), respectively. Let $B(i-1)$ and $B(h)$ represents the previous and high buffer levels, respectively. Thereby, the time, t^{start}(i), to request segment (i) is given in (5.9):

$$t^{start}(i) = \begin{cases} t^{end}(i-1), & \textbf{if } (B(i-1) \leq B(h)) \\ t^{end}(i-1) + (B(i-1) - B(h) + D(i-1)), & \text{otherwise} \end{cases} \tag{5.9}$$

Eq. 5.9 shows how for a buffer level lower or equal to $B(h)$, the client launches a request for (i)th segment immediately, provided that the $(i-1)$th segment has been completely downloaded. When the playback buffer goes above $B(h)$, the client suspends the next segment request by $(B(i-1) - B(h) + D(i-1))$ seconds.

This strategy can be applied also in a multi-view scenario. Let t_n^{start} (i) be the request time of $(i)th$ segment of (n)th stream, $t_{max}(i-1)$, and $D_{max}(i-1)$ represent the maximum transmission end time, and download time of $(i-1)$th segment in the requested streaming set, respectively, and $t_n^g(i)$ be the time the segment (i) of nth stream has become available to download from the server. The request time for segment (i) can be represented as:

$$t_n^{start}(i) = \begin{cases} max(t_{max}(i-1), t_n^g(i)), & \textbf{if } (B_n(i-1) \leq B_n(h)) \\ max(t_{max}(i-1) + BT_n, t_n^g(i)), & \text{otherwise} \end{cases} \tag{5.10}$$

$$t_{max}(i-1) = \max(t_1^{end}(i-1),\ t_2^{end}(i-1),\ \ldots,\ t_{R(i-1)}^{end}(i-1))$$

$$BT_n = \ (B_n(\mathrm{i}-1) - B_n(\mathrm{h}) + D_{max}(i-1))$$

$$D_{max}(i-1) = max(D_1(i-1), D_2(i-1), \ \ldots, D_{R(i-1)}(i-1))$$

Eq. (5.10) describes the request time of the next segment with the aim to ensure playback synchronization among segments of multiple streams. The client ensures that all the previously requested segments have been downloaded in full before sending the next GET request. When the buffer level is below or equal to $B_n(h)$, the client immediately requests the next segment of (n)th steam provided that the previous segments have been completely downloaded or the new ones have been entirely produced on the server-side. For the buffer level higher than $B_n(h)$, the client implements a waiting time before requesting segment (i)th to avoid buffer overflow. In this way, the proposed strategy synchronizes multiple streams in a way to request all the streams in a unified manner.

5.3.3 Bitrate Adaptation Approach

Algorithm 5.2 provides the bitrate adaptation algorithm for a multi-view streaming system. The bitrate for the (i)th segment is decided after the download of the $(i-1)$th segment. For each video stream, the playback buffer is associated with three thresholds: $B_n(min)$, $B_n(l)$, and $B_n(h)$. In the proposed solution, the bitrate of each stream is decided considering all the selected streams on a path. Naturally, the path estimated bandwidth should not be lower than the total bitrate of the streams selected by the client, i.e., $\sum_{n=1}^{R^j(i)} \mathcal{L}^j_{q,n}(i)$.

Algorithm 5.2 requests the lowest available video bitrate $\mathcal{L}_1$ for the first segment of all the streams because the client does not know the network conditions at the start. For $B_n(i) \leq B_n(h)$, if the total estimated throughput is higher than the total next higher bitrates of the streams over (j)th path, client changes to a higher video bitrate by considering the following conditions: (i)$\mathcal{L}^j_{q,n}(i-1) \neq \mathcal{L}_{max,n}$, (ii) $\mathcal{L}^j_{q+1,n}(i) \leq \widehat{Th}^j_n(i)$, and (iii) $B_n\,(i) - B_n(i-1) \geq 0$. The first condition verifies that the previously selected bitrate for nth stream is not the maximum bitrate. The second condition prevents the next higher video bitrate of nth stream from being greater than the estimated throughput for (i)th segment, calculated in Algorithm 5.1

Algorithm 5.2 Bitrate Adaptation Algorithm

Input :

$B_n(i) \leftarrow$ *Current buffer space of (n)th stream*

$\sum_{n=1}^{R^j(i)} \mathcal{L}^j_{q,n}(i) \leftarrow$ *Total bitrate of the requested streams over (j)th path*

$D_n(i-1) \leftarrow$ *Download time of (i-1)th segment of (n)th stream*

$\mathcal{L}^j_{q+1,n}(i) \leftarrow$ *Next higher bitrate of (n)th stream over (j)th path*

$\mathcal{L}^j_{q,n}(i-1) \leftarrow$ *Video bitrate selected for the previous segment over (j)th path*

Output:

$\mathcal{L}_{q,n}(i) \leftarrow$ *Video bitrate selected for the next segment*

if $(B_n(i) \leq B_n(min))$ **then**
  $\mathcal{L}_{q,n}(i) = \mathcal{L}_{1,n}$
else if $(B_n(min) < B_n(i) \leq B_n(l))$ **then**
  $\mathcal{L}_{q,n}(i) = max\{\mathcal{L}^j_{q,n}(i) | \mathcal{L}_{q,n} \varepsilon \mathcal{L}, \sum_{n=1}^{R^j(i)} \mathcal{L}^j_{q,n}(i) \leq \widehat{B}^j(i), \mathcal{L}^j_{q,n}(i) \leq \widehat{Th}^j_n(i),\ B_n(i) - \nu * D_n(i-1) \geq B_n(min)\}$
else if $(B_n(l) < B_n(i) \leq B_n(h))$ **then**
  if $(\sum_{n=1}^{R^j(i)} \mathcal{L}^j_{q+1,n}(i) \leq \widehat{B}^j(i))$ **then**
    if $(\mathcal{L}^j_{q+1,n}(i) \neq \mathcal{L}_{max,n}$ & & $\mathcal{L}^j_{q+1,n}(i) \leq \widehat{Th}^j_n(i)$ & &$\Delta B \geq 0)$ **then**
      $\mathcal{L}_{q,n}(i) = \mathcal{L}^j_{q+1,n}(i)$
    else
      $\mathcal{L}_{q,n}(i) = \mathcal{L}^j_{q,n}(i-1)$
else if $(B_n(i) > B_n(h))$ **then**
  $\mathcal{L}_{q,n}(i) = max\{\mathcal{L}^j_{q,n}(i) | \mathcal{L}_{q,n} \varepsilon \mathcal{L}, \sum_{n=1}^{R^j(i)} \mathcal{L}^j_{q,n}(i) \leq \widehat{B}^j(i), \mathcal{L}^j_{q,n}(i) \leq \widehat{Th}^j_n(i),\ B_n(i) - \nu * D_n(i-1) \geq B_n(l)\}$

The final condition ensures that the difference between (i)th and $(i-1)$th buffer levels is greater than or equal to zero seconds. These safety constraints ensure confident switch-up decisions. Otherwise, the client maintains the previous bitrate selected for $(i-1)$th segment.

When the buffer level of nth stream goes above $B_n(min)$ or $B_n(h)$, the client actively selects a higher bitrate to improve the streaming experience based on the following conditions: (i) total bitrate of the streams being requested on the path (j) is less than the available bandwidth of the path, (ii) selected bitrate for the (n)th stream over (j)th path should not be higher than the estimated throughput (allocated to this stream by the Algorithm 5.1), and (iii) buffer level increases by $\nu * D_n(i-1)$ seconds above $B_n(min)$ or $B_n(l)$, respectively.

In this way, the adaptive client is able to maintain high-quality playback under variable network conditions.

5.3.4 Path Selection Method

The path selection approach aims to utilize the best-quality paths for the high-priority segments, improving the overall streaming experience. The overall path selection mechanism for multi-view streaming is described as follows. For a single path, all the streams are requested over this same path. The HTTP request for each stream is received at the DASH server, and segments are transmitted in response to the requests over a single path. In the case multiple paths exists, the 1^{st} segment is requested randomly with the lowest available quality for each of the streams. For the following segments of all the streams, the path is selected such as the high-priority streams will always be requested over the best available paths in terms of available network throughput. If path (j) has the highest path estimated bandwidth, the high-priority segments will be requested over that path. If the estimated bandwidth of path (j) decreases and becomes lower over time, the high-priority segments will be requested over the path with the next highest available bandwidth. For the segments of low-priority streams, the other network links are leveraged to improve the overall streaming efficiency. After path selection, the network throughput is estimated for multiple streams. The client then runs the bitrate adaptation algorithm and performs the scheduling mechanism to retrieve the complete segments instead of dividing each segment into sub-segments. Furthermore, the independent playback buffer of each stream ensures an in-order delivery of the segments from different paths. In real-time implementations, suppose a device is connected to the network with two adapters (i.e., Ethernet and Wi-Fi), where each interface is associated with a particular priority based on the available channel conditions. The adaptive multi-view client can adjust the interface metric to specify in which order the available network adapters can be used to request and receive high- and low-priority segments for each adaptation interval, respectively.

5.4 Performance Evaluation

The modeling and evaluation of the proposed MVP-DASH solution is performed using Network Simulator NS-3.26. The DASH server contains three video streams

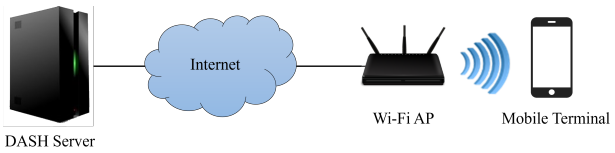


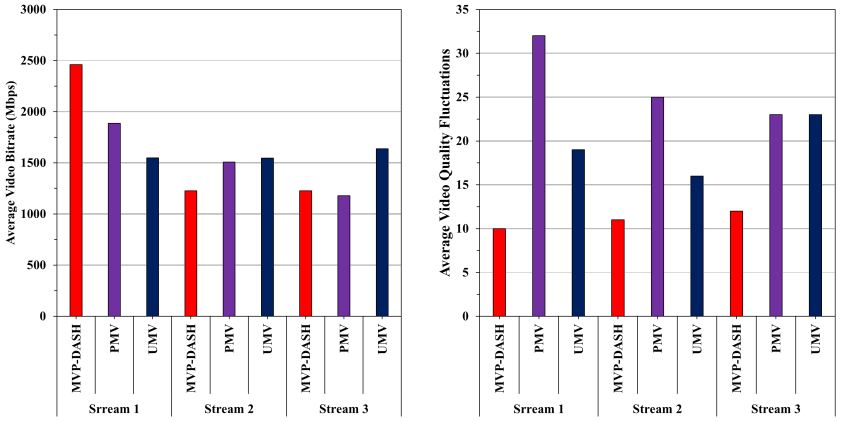
Figure 5.2: Network topology for multi-view video streaming over the wireless network.

each prepared in different representations, i.e., {45, 365, 791, 1033, 1245, 1647, 2050, 2584, 3479, 4220, 5220, and 6220} Kbps. These representations belong to different resolutions, mapped to LD, SD, HD, FHD, and QHD. The first stream is marked as high-priority while the second and third streams belong to the low-priority category. The values of buffer levels $B(min)$, $B(l)$, and $B(h)$ are set to 10s, 20s, and 40s, respectively. Each video segment has a duration τ of 2s. The value of ν is set to 25% of the segment duration, i.e., 0.5. Two benchmark algorithms referred to as priority multi-view (PMV) [185] and unified multi-view (UMV) [11] are adopted for comparative performance evaluation in single and multipath environments. The *PMV* considers increasing the bitrate starting from the high-priority stream to the last stream after allocating minimum bitrate to all the streams, while *UMV* implements a unified bandwidth approach to achieve almost the same video bitrates for all the streams over the same path. The priority levels of different streams are adopted as discussed in the original works. The *PMV* assigns different priority levels to each stream, i.e., stream 1 has higher priority compared to stream 2, and stream 2 has higher priority compared to stream 3. In *UMV* all the streams share equal priority levels. The *PMV* solution was originally designed for live streaming, but in this work, we employ it for on-demand sessions. We have deployed the proposed algorithm in the DASH classic client[1].

5.4.1 Single-Path Scenario

Fig. 5.2 shows the wireless network topology including a DASH server, a wireless AP, and a DASH client. The bottleneck link between server and AP is varied to examine the adaptation behaviors of clients under variable network conditions. The available bandwidth changes from 5 Mbps to 8 Mbps after the first 50 seconds, then

[1]https://github.com/djvergad/dash



(a) Average video bitrates (b) Average quality fluctuations

Figure 5.3: Average video bitrates and number of quality fluctuations in a single-path scenario.

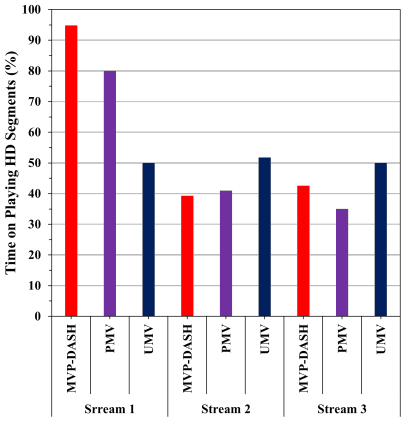


Figure 5.4: Time spent on playing HD segments in a single-path scenario.

it switches repeatedly between 8 Mbps and 4 Mbps after every 50 seconds until the end of the simulation at 250 seconds. The performance is evaluated in terms of three relevant evaluation metrics: average video bitrate, number of video bitrate switches, and time of playing high-quality segments. The time on high-quality is the percentage of time the client plays high-quality segments [271]. A video bitrate equal to or greater than 1647 Kbps is considered a high-quality level since this bitrate level represents the HD content, as represented in Table 4.2.

Fig 5.3 presents the simulation results in terms of average video bitrate and the average number of video quality fluctuations for the three solutions compared.

Fig. 5.3a shows that the average video bitrate achieved by **MVP-DASH** for stream 1 is 2460 Kbps, while for stream 2 and stream 3, it achieves nearly 1225 Kbps. The

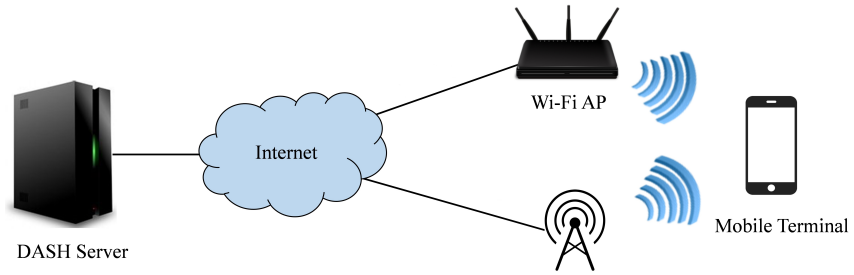


Figure 5.5: Multipath wireless network topology.

PMV client achieves 1887 Kbps, 1508 Kbps, and 1178 Kbps average video bitrates for three streams, respectively. The *UMV* client achieves about 1577.2 Kbps average video bitrate for all three streams. In particular, for all three streams, **MVP-DASH** achieves 339 Kbps and 180 Kbps higher video bitrate in comparison to the *PMV* and *UMV* clients, respectively. Besides, the **MVP-DASH** shows a more stable behavior and keeps the download bitrate to an optimum level to avoid unnecessary bitrate switches. Fig. 5.3b reflects that the *PMV* client experiences the highest number of video bitrate switches 32 for stream 1, while **MVP-DASH** experiences the least number of video bitrate fluctuations for this stream. **MVP-DASH** is the most stable solution and exhibits 47 and 25 fewer video bitrate switches for all three streams in comparison to the *PMV* and *UMV*, respectively.

Fig. 5.4 shows the average time spent by the three compared solutions playing high-quality/HD segments. **MVP-DASH** spends nearly 95% of the time playing high-quality segments for stream 1 compared to the 80% and 50% of the *PMV* and *UMV* clients, respectively. Moreover, for low-priority streams, i.e., stream 2 and stream 3, **MVP-DASH** spends nearly 40% on average of the streaming session playing high-quality segments. Consequently, **MVP-DASH** behaves as designed and supports high-quality streaming for the high-priority stream and good quality for the low-priority streams.

Table 5.1: WiFi and LTE Channels Configurations

WiFi Configuration		**LTE Configuration**	
WiFi model	IEEE 802.11g	Components	1 PGW, 1 eNodeB
PHY model type	YansWifiPhy	Antenna model	Isotropic antenna model
Channel model type	YansWifiChannel	Path loss mode	Friis propagation model
Channel width	20 MHz	Scheduler	RrFfMacScheduler

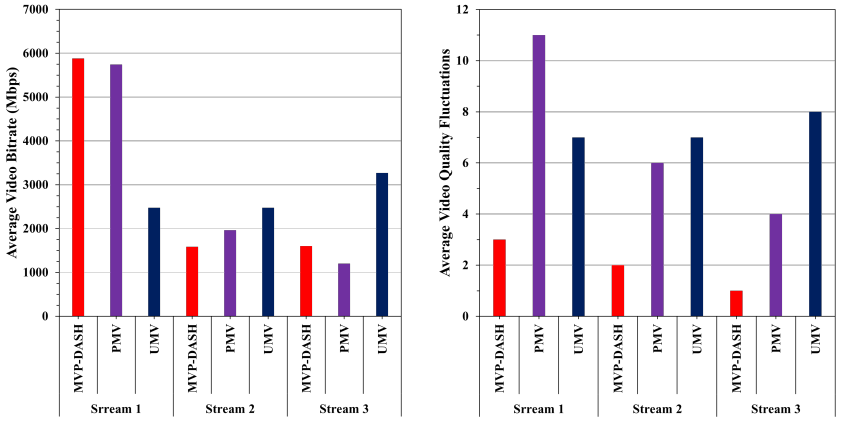
5.4.2 Multipath Scenario

Fig. 5.5 presents the network topology for a multipath testing scenario. The clients request the segments of multiple streams by utilizing both the LTE (7Mbps) and WiFi (4Mbps) interfaces simultaneously over a simulation period of 300 seconds. The WiFi and LTE characteristics used in the experiment are presented in Table 5.1. The proposed multipath approach is implemented for both *PMV* and proposed **MVP-DASH** bitrate adaptation algorithms. The *UMV* algorithm considers the network throughput and stream bitrates for the path selection.

MVP-DASH and *PMV* schemes download the high-priority stream through the LTE interface and the other two streams through the WiFi interface. In the case of the *UMV* scheme, the first two streams utilize the LTE interface, while stream 3 is requested over the WiFi interface.

Fig. 5.6 shows the adaptation behavior of the three compared solutions in a multipath environment. **MVP-DASH** performs the lowest number of resolution changes (Fig. 5.6b) and, at the same time, also provides the high-quality video bitrates for the longest (Fig. 5.6a). The *PMV* client implements a step-wise switch up and undergoes more bitrate switches. For stream 1, **MVP-DASH** has an average download bitrate of 5.87 Mbps with 3 changes, while the *PMV* client achieves nearly the same average video bitrate, but at the expense of 11 bitrate switches. The *UMV* client does not consider the buffer occupancy when making efficient switch-up decisions. In particular, **MVP-DASH** achieves about 53 Kbps and 280 Kbps higher average video bitrate and almost 15 fewer video quality switches for all three streams in comparison to the *PMV* and *UMV* clients, respectively. This is because **MVP-DASH** increases the video bitrate according to a more aggressive adjustment policy and performs fewer video quality switches.

Fig. 5.7 represents the percentage of time spent by all three compared schemes playing different video bitrates for all three streams. MVP-DASH spends 90.9% of the time playing the highest bitrate segments, i.e., 6220 Kbps, for stream 1 and experiences only 3 quality changes to reach this level, while *PMV* experiences this bitrate level for 87.88% of the session with 11 bitrate switches. For all three streams,



(a) Average video bitrates (b) Average quality fluctuations

Figure 5.6: Average video bitrates and number of quality fluctuations in a multi-path scenario.

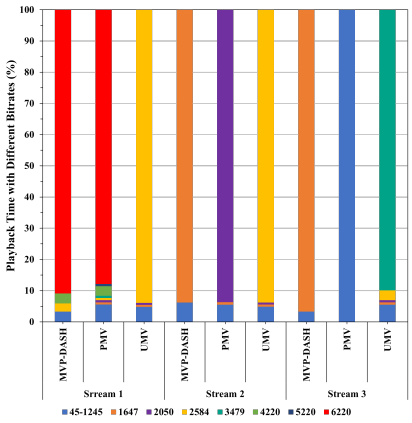


Figure 5.7: Time spent on playing different quality segments in a multi-path scenario.

MVP-DASH spends more than 95% of the time playing ≥1647 Kbps video bitrates while for stream 3, the *PMV* client spends all of the streaming session playing ≤45-1245 kbps video bitrates. Thus, **MVP-DASH** makes an efficient use of the available network resources to play high-quality segments.

5.5 Chapter Summary

In this chapter the general block-level architecture, algorithm design, and performance evaluation of the proposed **MVP-DASH** was presented. For a multi-view video streaming system, **MVP-DASH** calculates the priority-related weights and distributes the aggregated throughput with relevant importance to each low and

high-priority stream. It implements a unified segment scheduling mechanism to ensure playback synchronization and avoid buffer overflows for multi-view videos. It then leverages the path diversity in multipath environments by requesting high-priority streams over the best available paths. **MVP-DASH** was compared with two relevant multi-view streaming solutions referred to as *PMV* and *UMV* solutions in a DASH-based simulation environment. We used the same encoding levels ranging from 45 Kbps to 6.2 Mbps for three video streams to verify how different streaming solutions react to the network availability for different streams. We implemented proposed and comparative solutions and deployed them in two scenarios consisting of a single fluctuating wireless network and two stable wireless channels, respectively. In the single path scenario, **MVP-DASH** outperforms the other approaches in achieving at least 180-339 Kbps higher average video bitrate for all three streams along with exhibiting 25-47 fewer video quality switches in comparison to *PMV* and *UMV* streaming clients. In addition to this, with the proposed solution the high-priority stream was shown to spend nearly 95% of the time on playing high-quality segments as well as 40% for the low-priority streams. In a multipath streaming environment, **MVP-DASH** is the most stable streaming solution and encourages the stable and high-quality transmission of quality variable views to satisfy the streaming requirements. Based on simulation findings, it can be concluded that **MVP-DASH**-based streaming of multi-view video supports well-prioritized multi-stream transmission.

Chapter 6

Fixed Tiling-based Adaptive Solutions for 360° Video Streaming

This chapter presents the system architecture and innovative fixed tiling-based streaming solutions, CFOV and SFOV, which improve user QoE for omnidirectional 360° videos. These solutions employ FoV/viewport prediction, tiles selection, and tiles bitrate adaptation algorithms to overcome the limitations of existing tile-based streaming solutions. Simulations-oriented experimental results show how the proposed solutions improve estimated user QoE measured in terms of visual quality levels, non-visual quality levels, and spatial and temporal quality oscillations in comparison with state-of-the-art tile-based delivery solutions.

6.1 Introduction

Recently, affordable VR technologies have experienced increased attention as they support innovative and immersive 360° media applications. However, the expected high-quality immersive rich multimedia streaming experience is hindered by excessive bandwidth requirements of the high-resolution 360° content. State-of-the-art tile-based solutions lower the bandwidth requirements by performing selective streaming based on FoV prediction mechanisms. However, sometimes user movement is highly unpredictable during some parts of the video, and applying these solutions adversely affects the overall QoE [272].

The space (tile-level) and time (segment-level) separation of 360° videos make achieving high QoE levels very challenging when compared to regular videos. Therefore, proper tiles arrangement and relevant bitrate adaptation are fundamental to improving the end-users streaming experience. Fixed tiling-based adaptive streaming solutions support differentiated streaming of quality-variable tiles depending on their position within the viewport. The success of these solutions is highly dependent on dynamically selecting the range of tiles in different streaming regions. Some fixed viewport-based works [106, 49] perform non-uniform viewport quality streaming, which can make an experienced VR user feel more uncomfortable. Similarly, fixed viewport-based streaming [48, 106, 49] without marginal tiles can degrade the quality of fast motion outdoor content (e.g., sports, action movies, etc.). Therefore, there is a need for efficient solutions to improve the selection of different streaming regions during 360° video streaming.

This chapter classifies proposed tiles selection solutions into two categories: (i) prediction angles-based tiles selection; (ii) prediction angles and prediction differences-based tile selection. Some straightforward techniques [45, 106, 48, 164, 49] belonging to the first category perform viewport selection based on a single prediction mechanism. Such solutions work reasonably well for $\leq$1s segment duration. However, unfair visual quality levels and frequent quality fluctuations can be observed when tested for >1s segment duration due to the increase in viewport mismatch ratio. The proposed CFOV1 and CFOV2 solutions belonging to the first tiles selection

category employ two low-complex prediction mechanisms to address the accuracy gap and perform advanced streaming tiles selection in three scenarios: (i) *Fixed FoV*, (ii) *Marginal Region*, and (iii) *Extended FoV*, in order to improve the 360° video visualizations. Some solutions [56, 44, 273] consider prediction angles as well as prediction errors to estimate the viewport. However, they do not involve any marginal region [44] or viewport extension [56, 273] to truly reflect the proactive streaming. Moreover, including marginal tiles surrounding viewport in all directions [107] or sub-areas-based streaming [56] can lead to significant bandwidth wastage. The proposed SFOV1 and SFOV2 solutions jointly employ prediction angles from a single prediction mechanism and observed prediction errors to select the best-fit streaming regions. Different from [4, 230, 107], CFOV and SFOV solutions do not select marginal regions all the time to absorb the discrepancy between actual and predicted viewport. The extended viewport in CFOV and SFOV solutions is more profound than [108, 48] and can accommodate highly variable head rotations. Different from CFOV1 and SFOV1 solutions which only consider the viewing information to perform relevant tiles selection, CFOV2 and SFOV2 solutions include content information in addition to viewing behaviors to proactively select the streaming regions. The proposed tiles selection algorithms input different streaming regions tiles to the prioritized bitrate adaptation algorithm. We propose a bitrate adaptation algorithm in the context of fixed tiling-based streaming, which dynamically implements *Uniform* and *Weighted* bitrate adaptation scenarios. Unlike [108, 48, 107, 56] only the viewport tiles are streamed to the client if the available bandwidth is limited. The proposed bitrate adaptation algorithms further refine the tiles selection based on the available bandwidth budget and then arrange the streaming into viewport only, viewport and background, or viewport, marginal, and background regions streaming. Following this, the viewport tiles are always streamed in higher possible quality levels than the background tiles, which are allocated with the lowest quality levels. The considerable performance analysis and comparisons indicate the success of the proposed solutions against the highly-cited fixed tiling-based solutions.

- CFOV solutions employ two viewpoint prediction mechanisms for each adaptation interval. CFOV1 is a content-independent solution and arranges the

tiled videos into viewport, marginal, and background tiles sets based on the relationship between two predicted viewport tiles sets. On the other hand, CFOV2 is a content-dependent solution that performs tiles selection based on content complexity variations and predicted viewpoints to arrange the tiles into different streaming regions. CFOV solutions consider the following three tiles selection scenarios: (i) *Fixed FoV*; (ii) *Marginal Region*; (iii) *Extended FoV* to lower the distance between predicted and true viewport positions.

- SFOV solutions consider single predicted viewpoint and viewport differences observed during previous segment playback to decide the tiles for the next segment request. SFOV1 mainly decides to keep the original viewport size or extend it or add marginal tiles based on the distance between previously observed original and predicted viewpoint tiles. SFOV2 considers different thresholds to measure the closeness between original and predicted viewpoints and the content complexity information. Similar to CFOV, SFOV1 and SFOV2 solutions arrange tiles by considering *Fixed VP*, *Marginal Region*, and *Extended VP* scenarios.
- The fixed tiling-based bitrate adaptation algorithms implement different priority-related policies to effectively utilize the available network resources to output the maximized viewing qualities for selected tiles of CFOV and SFOV Solutions. In this context, one of the following streaming regions: (i) viewport; (ii) viewport and background; (iii) viewport, marginal, and background; are selected based on available network bandwidth and tiles distributions. Only the viewport tiles are encouraged to be downloaded for CFOV and SFOV solutions when the available bandwidth is limited. For viewport and background regions streaming, the bitrate of the viewport region is incremented *Uniformly* after allocating the minimum bitrate to the entire tiles sets. A *Weighted* bitrate adaptation is performed for viewport, marginal, and background tiles streaming to keep a safe quality difference between the highest viewport and immediate lowest background tiles. These effective bitrate adjustment policies are designed to address the different quality objectives defined in section 6.3.1.

The solutions are evaluated in Section 6.4.2 which demonstrates how they improve

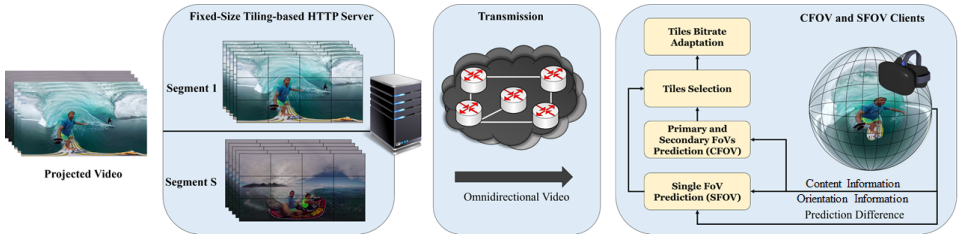


Figure 6.1: The proposed end-to-end 360° video streaming framework.

streaming performance when compared with existing adaptive tile-based solutions. The evaluation was performed with four videos from performance, sport, and film categories, prepared in 4x3, 6x4, and 8x6 tiling patterns and in 1s, 2s, and 3s segment durations.

6.2 Proposed Fixed Tiling-based Architecture

6.2.1 Fixed Tiling-based System Architecture

The proposed end-to-end on-demand 360° video streaming architecture is presented in Fig. 6.1. The server is responsible for storing and pre-processing of 360° video content. The 360° sphere representation is transformed into an equirectangular projection format [81] following capturing and stitching steps. The equirectangular projected video is temporally split into S equal duration segments, and each segment is prepared in M spatial tiles, and each tile encoded into N bitrate levels. Let $\mathcal{L}_j^k(i)$ represents the quality level $j \in [1, N]$ of tile $k \in [1, M]$ in segment $i \in [1, S]$. Let $x^k(i)$ be a decision variable representing that the kth tile for (i)th segment is selected for streaming (i.e., $x^k(i) = 1$) or not (i.e., $x^k(i) = 0$).

On the client-side, the *Single FoV Prediction* module predicts the future FoV coordinates based on the orientation information and prediction differences between the last predicted and actual FoVs. SFOV2 adjusts this prediction difference by including the content complexity variations. The *Combined FoV* prediction module predicted primary and secondary FoVs based on orientation information. CFOV2 measures the arc distance between two predicted viewpoints and adjusts it by incorporating content information. Accordingly, the *Tiles Selection* module in CFOV and SFOV clients selects the viewport, marginal, and background tiles sets for the (i)th

Table 6.1: Notations Used in this Chapter

Symbol	Meaning
S, N, M	Total number of segments, bitrates, and tiles
i, j, k	Index of segment, bitrate, and tiles
τ	Segment duration
$\hat{\mathcal{T}^v}(i)$	Set containing the actual viewport tiles
$\hat{\mathcal{T}^b}(i)$	Set containing the actual background tiles
$\mathcal{T}(i)$	Set containing all the tiles in a streaming session
$\mathcal{T}^v(i)$	Set containing the predicted viewport tiles
$\mathcal{T}^m(i)$	Set containing the predicted marginal tiles
$\mathcal{T}^b(i)$	Set containing the predicted background tiles
$\mathcal{L}_j^k(i)$	Video bitrate selected for kth tile of (i)th segment
$\widehat{\mathcal{B}}(i)$	Estimated bandwidth for the (i)th segment
$\mathcal{B}^v(i), \mathcal{B}^m(i)$	Estimated bandwidth for the viewport and marginal tiles
α, β, γ, δ	Weight Coefficients
$TO(i)$	Tiles overlap for (i)th segment

segment, i.e., $\mathcal{T}^v(i)$, $\mathcal{T}^m(i)$, $\mathcal{T}^b(i)$, from the tiles set, $\mathcal{T}(i)$. Based on the output of the *Tiles Selection* module, the *Bitrate Adaptation* unit selects suitable bitrates for each tile according to the associated region and the estimated network bandwidth. Once the segments are received, the client performs decoding and stitching of the requested tiles to reconstruct the 360° video. It then performs the rendering and starts playing the requested content. Table 6.1 includes the mathematical symbols and their meanings used in the following discussion.

6.2.2 Problem Statement

The high-quality expectations of the user mainly depend on the quality of the visible area. The lower quality visible tiles may not satisfy the user even if the background tiles are played in good quality. Some key challenges to help support the high QoE levels include real-time scene update, accurate FoV prediction, tiles selection, adaptive quality adjustments, and employing efficient delivery protocols, among others [274]. The client seeks optimal bitrates for each segment, intending to optimize the user's long-term QoE reward, subject to the constraints (6.2-6.7). Mathematically, the optimization problem can be formulated as the following problem:

Problem:

$$arg \max_{i \in [1,S]} QoE(i) \tag{6.1}$$

Constraints:

$$\sum_{k \in \mathcal{T}(i)} \sum_{j \in [1,N]} \mathcal{L}_j^k(i) * x^k(i) \leq \widehat{\mathcal{B}}(i) \tag{6.2}$$

$$\mathcal{L}_j^k(i) * x^k(i) = \mathcal{L}_j^{k'}(i) * x^{k'}(i), \forall k, k' \in \mathcal{T}^v(i), \forall j \in [1, N] \tag{6.3}$$

$$\mathcal{L}_j^k(i) = \mathcal{L}_1^k(i) * x^k(i), \forall k \in \mathcal{T}^b(i) \tag{6.4}$$

$$\sum_{k \in \mathcal{T}^v(i)} \sum_{j \in [1,N]} \mathcal{L}_j^k(i) * x^k(i) \leq \mathcal{B}^v(i) \tag{6.5}$$

$$\sum_{k \in \mathcal{T}^m(i)} \sum_{j \in [1,N]} \mathcal{L}_j^k(i) * x^k(i) \leq \mathcal{B}^m(i) \tag{6.6}$$

$$\sum_{k \in \mathcal{T}^v(i)} \sum_{j \in [1,N]} \mathcal{L}_j^k(i) * x^k(i) \geq \sum_{k \in \mathcal{T}^m(i)} \sum_{j \in [1,N]} \mathcal{L}_j^k(i) * x^k(i) \tag{6.7}$$

The constraints in Eq. (6.2) limit the selected bitrate of all the tiles in *(i)th* segment. Constraints in Eq. (6.3) state that all the viewport tiles should have the same selected video bitrate. Eq. (6.4) restricts the bitrate of all the background tiles to the lowest quality level (i.e., $\mathcal{L}_1^k(i)$). The constraints in Eq. (6.5) and Eq. (6.6) make sure that the bitrate of the viewport and marginal tiles is not higher than the bandwidth of the viewport and marginal tiles, respectively. Particularly, the viewport tiles are downloaded in higher bitrates than the marginal tiles. The bandwidth allocation based on the importance of each region is described in section 6.3.3. Finally, the constraints in Eq. (6.7) ensure that the bitrates of marginal tiles would not be higher than the viewport tiles. The following three steps solve the above problem:

1. defines a user QoE metric that assesses the perceived quality based on the visual quality and the quality allocated to background regions and visual smoothness.
2. employs content-agnostic prediction angles or prediction angles and prediction errors-based tiles selection approaches that dynamically perform the viewing area selection to improve the overlap between real and predicted viewport tiles.
3. selects optimal quality levels by performing different priority-related bitrate

allocation policies for each tile of different regions.

The following subsection elaborates on these aspects.

6.3 Fixed Tiling-based Algorithms

6.3.1 QoE Metric

With the adaptive transmission of omnidirectional video, it is imperative to recognize the unique quality aspects of the consumer due to its highly prevalent nature. The QoE assessment is a key factor in optimizing the adaptive video streaming [275]. However, calculating visual quality alone is not adequate for a complete VR QoE framework. In evaluating the user's QoE, it is also essential to define the effect of other parameters, e.g., bandwidth savings, spatial and temporal quality fluctuations, etc.

- **Viewport Quality:** By averaging the quality of the viewport tiles based on the real viewport traces, we get the viewport quality in segment (i) as follows [276, 216]:

$$f_1(i) = \frac{\sum_{k \in \mathcal{T}^{\hat{v}}(i)} \sum_{j \in [1,N]} \mathcal{Q}(\mathcal{L}_j^k(i))}{|\mathcal{T}^{\hat{v}}(i)|} \tag{6.8}$$

  where $\mathcal{T}^{\hat{v}}(i)$ represents the actual viewport tiles set and $|\mathcal{T}^{\hat{v}}(i)|$ indicates the number of tiles in viewport tiles set. $\mathcal{Q}(\mathcal{L}_j^k(i))$ maps the video bitrate to the relevant quality level for the (i)th segment.

- **Background Quality:** Ideally, the 360° client should only stream the viewport tiles at the best possible quality with no background tiles. But several solutions stream the background tiles to lower the impact of viewport anomalies due to the limited precision of prediction mechanisms. This metric explicitly indicates the average quality of the background tile in the (i)th segment and is given as follows:

$$f_2(i) = \frac{\sum_{k \in \mathcal{T}^{\hat{b}}(i)} \sum_{j \in [1,N]} \mathcal{Q}(\mathcal{L}_j^k(i))}{|\mathcal{T}^{\hat{b}}(i)|} \tag{6.9}$$

  where $\mathcal{T}^{\hat{b}}(i)$ represents the background tiles set which contains tiles not vis-

ible to the user based on the ground truth viewport traces during the (i)th segment. The term in the denominator $|\mathcal{T}^{\hat{b}}(i)|$ represents the number of tiles in background tiles set.

- **Temporal Quality Oscillations:** The efficiency of tile-based streaming schemes can be impaired by the disparity in quality levels between two viewports of consecutive segments. Therefore, the temporal quality fluctuations need not be drastic and can be calculated by [276]:

$$f_3(i) = \mid f_1(i) - f_1(i-1)| \tag{6.10}$$

- **Spatial Quality Oscillations:** Cybersickness, viewing irritation, and other physiological effects, such as nausea, fatigue, and aversion [131], can be driven by variable quality levels within the viewport. That leads, therefore, to lower QoE levels. Following [232], we measure this value according to the coefficient of variation (CV) of the viewport quality levels.

$$f_4(i) = \frac{\sigma(\mathcal{Q}(\mathcal{L}_j^k(i)))}{\mu(\mathcal{Q}(\mathcal{L}_j^k(i)))}, \quad \forall k \in \mathcal{T}^{\hat{v}}(i), \forall j \in [1, N] \tag{6.11}$$

The term in the numerator represents the standard deviation of the viewport quality samples, while the denominator represents the mean of the samples.

Following the principle behind the QoE metric for traditional video [40], some works [251, 216, 277] consider video quality, quality variations, rebuffering events, etc. to model a QoE metric for 360° videos. In contrast to these works, the proposed metric considers the impact of non-visual quality allocation specifically in a scenario where the bandwidth budget is actively allocated to different predicted frame areas. This is particularly important to cover the quality assessment gap for 360° videos. The proposed QoE metric based on the above-mentioned quality functions is given as follows:

$$QoE(i) = \alpha * f_1(i) - \beta * f_2(i) - \gamma * f_3(i) - \delta * f_4(i) \tag{6.12}$$

where α, β, γ, and δ are the non-negative weight coefficients corresponding to the background quality and temporal and spatial quality oscillations, respectively. As

the background quality, and spatial and temporal quality oscillation negatively impact the perceived quality. Therefore, We want to minimize the $f_2(i)$, $f_3(i)$, and $f_4(i)$.

6.3.2 Fixed Tiling-based Streaming Tiles Selection Algorithms

One of the essential requirements of VR devices is to update the scene according to the viewer's head movement. When a user changes his viewpoint, the end terminal processes the feedback signals and can render the relevant FoV so that a view is accessible from a normal visual angle. Accurate viewport prediction plays a significant role in tiles selection, leading to improved streaming performance. The tiles are usually selected based on the predicted points. However, the accuracy of prediction mechanisms is limited when tested for different motion content, i.e., outdoor sports, action films, etc. Therefore, it is necessary to consider the prediction errors in addition to prediction angles to improve the viewport hit ratio. This section presents the proposed tiles selection solutions, CFOV and SFOV. Next, these innovative tile selection approaches are illustrated in detail.

6.3.2.1 Prediction Angles-based Tiles Selection Solutions

360° video has become an integral part of popular multimedia applications, as the consumer is interested in an increasingly interactive and immersive streaming experience. Tile-based streaming solutions should be able to select the best-fit tiles in response to the variable head motions of the consumer. The existing prediction models based on a single prediction mechanism, for instance, position-based (i.e., LR, speed-based, machine learning-based, etc.) and complex content-based (i.e., saliency maps, motion maps, etc.), could undergo lower long-term prediction accuracy. Interestingly, the *last known* prediction model that simply outputs the current viewing position for the future segments performs better than the state-of-the-art viewport prediction models [264].

Due to the limited prediction accuracy of existing solutions and the extremely unpredictable viewing nature of the user, the basic idea is to stream more tiles than necessary to cover the actual viewing area. The proposed prediction angles-based

Algorithm 6.1 Tiles Selection Algorithm-CFOV1

Input :

$\mathcal{T}(i) \leftarrow$ *Sorted tile set in the streaming session*; $\mathcal{T}^{vp}(i) \leftarrow$ *Tiles set for the primary predicted FoV*; $\mathcal{T}^{vs}(i) \leftarrow$ *Tiles set for the secondary predicted FoV*

Result :

$\mathcal{T}^{v}(i), \mathcal{T}^{m}(i), \mathcal{T}^{b}(i) \leftarrow$ *Predicted viewport, marginal, and background tiles sets for the* (i)*th segment*

Calculate the tiles sets $\mathcal{T}^{vp}(i)$ and $\mathcal{T}^{vs}(i)$ with $\Phi^{p}(i), \Theta^{p}(i), \Phi^{s}(i), \Theta^{s}(i)$ and $\mathcal{VS}$
if $(\mathcal{T}^{vp}(i) \cap \mathcal{T}^{vp}(i) \neq \emptyset)$ **then**
    if $(\mathcal{T}^{vs}(i) - \mathcal{T}^{vp}(i)) == \emptyset)$ **then**
        $\mathcal{T}^{v}(i) = \mathcal{T}^{vp}(i)$
    else
        $\mathcal{T}^{v}(i) = \mathcal{T}^{vp}(i) \cup (\mathcal{T}^{vs}(i) - \mathcal{T}^{vp}(i))$
    $\mathcal{T}^{m}(i) = \emptyset$
    $\mathcal{T}^{b}(i) = \mathcal{T}(i) - \mathcal{T}^{v}(i)$
else
    $\mathcal{T}^{v}(i) = \mathcal{T}^{vp}(i)$
    $\mathcal{T}^{m}(i) = \mathcal{T}^{vs}(i)$
    $\mathcal{T}^{b}(i) = \mathcal{T}(i) - (\mathcal{T}^{v}(i) \cup \mathcal{T}^{m}(i))$

tiles selection solutions (CFOV1 and CFOV2) perform tiles selection by considering two viewpoint/viewport prediction mechanisms without involving any prediction errors. The *last known* prediction mechanism, which outputs the previously actual viewpoint as the future predicted viewpoint, is employed as the primary predicted mechanism. A *spherical walk* approach proposed in [48] is used as the secondary prediction mechanism that considers the user's motion as a walk on a sphere and predicts the future position based on the spherical movement from one point to another point. The prediction angles for primary and secondary prediction mechanisms for (i)th segment are represented as $\Phi^{p}(i), \Theta^{p}(i)$ and $\Phi^{s}(i), \Theta^{s}(i)$, respectively.

For each video segment, the CFOV client classifies a 360° video frame into the viewport, marginal, and background regions. Following this, we consider *Fixed FoV*, *Marginal Region*, and *Extended FoV* cases for tiles selections in CFOV1 and CFOV2 solutions.

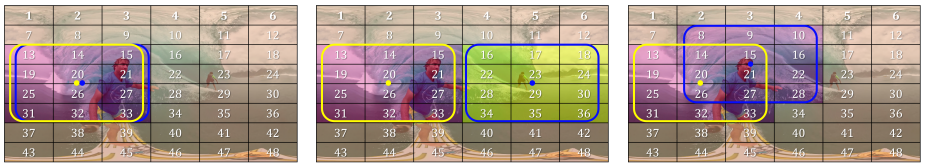
1. *Fixed FoV: The Fixed FoV tiles set contains only the tiles corresponding to the original size/coverage of the FoV.*
2. *Marginal Region: A dynamic range of tiles is allocated in the marginal region tiles set. The marginal tiles could be surrounding the fixed FoV region in any*

direction.

3. *Extended FoV: The original area of the FoV is virtually enlarged to deal with the potential viewport prediction errors with no marginal region.*

CFOV1: Algorithm 6.1 describes the *Tiles Selection* mechanism in CFOV1. For each segment, the tiles belonging to the different regions are chosen dynamically based on the performance of the prediction mechanisms. Algorithm 6.1 starts by calculating the primary and secondary FoV tiles sets by considering the predicted viewpoints (primary and secondary) and viewport size ($\mathcal{VS}$), usually in the range of 90°-110°. The tiles for both viewports are selected by calculating the spherical distance between the predicted viewpoints and the center of each of the tiles. The tiles whose centers are less than half of the FoV size apart from the viewpoint will belong to the viewport region. In this way, both the primary (*last known*) and secondary (*spherical walk*) viewport sets represented by $\mathcal{T}^{vp}(i)$, $\mathcal{T}^{vs}(i)$, respectively, are computed for (i)th segment (line 1). Algorithm 6.1 then finds the intersection between two predicted FoVs to select the tiles for each of the three regions. The non-empty intersection between two predicted sets indicates that the prediction accuracy of the two mechanisms is closer. Following this, if the tiles in both FoVs are identical, this refers to *Fixed FoV* case, where the similar tiles of both viewport ($\mathcal{T}^{vp}(i)$) are selected as viewport tiles set ($\mathcal{T}^{v}(i)$) for (i)th segment (line 4). If the intersection of two predicted FoVs is non-empty, they contain non-identical tile/tiles. This refers to the *Extended FoV* case, where the unique tiles of both FoVs are labeled as viewport tiles (line 6). Next, the background tiles set is computed by adding all the remaining tiles (line 8). In *Fixed FoV* and *Extended FoV* cases, the marginal tiles set ($\mathcal{T}^{m}(i)$) does not contain any tiles. For the *Marginal Region* scenario, where the two predicted FoVs contain different tiles, the tiles belonging to the primary and secondary FoVs are labeled as viewport and marginal tiles, respectively (lines 10-11). The range of background tiles set is then computed for the (i)th segment (line 12).

Fig. 6.2 illustrates the tiles selection cases in CFOV1 solution. The yellow and blue rectangles represent the primary and secondary predicted FoVs, respectively. The viewport, marginal, and background tiles are represented in light red, green, and



(a) Fixed FoV (b) Marginal Region (c) Extended FoV

Figure 6.2: Tiles selection cases in CFOV1 for the (i)th segment. The yellow rectangle represents the primary predicted FoV, while the blue rectangle represents the secondary predicted FoV. The potential viewport tiles are represented in light red, while the marginal and background tiles are marked in light green and brown, respectively.

brown. Fig. 6.2a represents the *Fixed FoV* case in CFOV1, where both the primary and secondary FoVs contain identical tiles. The *Marginal Region* scenario is shown in Fig. 6.2b, where the predicted secondary FoV is represented as the marginal region. Fig. 6.2c represents the case where both the predicted FoVs partially overlap during the (i)th segment. The goal here is to extend the FoV coverage by adding non-overlapping tiles of the secondary FoV tiles set to the primary FoV tiles set to deal with possible head movement prediction errors. With no marginal tiles in *Extended FoV* case, the rest of the tiles belong to the background region.

CFOV2: Different prediction mechanisms can generate different prediction results. Using a combination of these prediction mechanisms reduces the viewport mismatch ratio. Similar to Algorithm 6.1, Algorithm 6.2 is also based on the relationship between two prediction mechanisms to adapt better to the highly variable head movements for different content characteristics. However, CFOV2 involves content complexity variations to improvise the tiles selection.

Spatial Information (SI) and Temporal Information (TI), defined by ITU-T P.910 [126], have been extensively used to approximate video scene complexity. SI is based on the Sobel filter, while TI is based upon the difference between successive frames. The research community employs these spatial and temporal complexity measurements to select content for subjective experiments, quality approximation as well as estimating compression levels and bandwidth requirements for a diverse range of streaming applications. More spatially and temporally rich 360° videos result in more drastic viewing patterns. Following this, the proposed solution combines viewing patterns and content complexity measurements to dynamically select the

Algorithm 6.2 Tiles Selection Algorithm-CFOV2

Input :

$\mathcal{T}(i)\leftarrow$ *Sorted tile set for the streaming session*; $\Phi^p(i), \Theta^p(i), \Phi^s(i), \Theta^s(i)\leftarrow$ *Predicted angles*; $\mathcal{D}(i)\leftarrow$ *Arc distance between two predicted viewpoints*; $\mathcal{VS}\leftarrow$ *Size of the viewport in radians*; $\widehat{\mathcal{VS}} = 0\leftarrow$ *Updated size of the viewport initialized to 0*; $\mathcal{CC}(i)\leftarrow$ *Content complexity variations*; $\mathcal{AD}(i)\leftarrow$ *Adjusted distance*; $\Omega_1, \Omega_2\leftarrow$ *Distance thresholds*

Result :

$\mathcal{T}^v(i), \mathcal{T}^m(i), \mathcal{T}^b(i)\leftarrow$ *Estimated viewport, marginal, and background tiles sets for the* (i)*th segment*

Calculate $\mathcal{T}^p(i)$ and $\mathcal{T}^s(i)$ with tiles having distance to viewpoints $((\Phi^p(i), \Theta^p(i)))$ $(\Phi^s(i), \Theta^s(i))$ in range $[0{:}\mathcal{VS}/2]$
$\mathcal{D}(i) = ArcDistance(\Phi^p(i), \Theta^p(i), \Phi^s(i), \Theta^s(i))$
$\mathcal{AD}(i) = \mathcal{D}(i) + \mathcal{D}(i) * \mathcal{CC}(i)$
if $\mathcal{AD}(i) \geq 0$ && $\mathcal{AD}(i) <= \Omega_1$) **then**
    $\mathcal{T}^v(i) = \mathcal{T}^p(i)$
    $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$
end
else if ($\mathcal{AD}(i) > \Omega_1$ && $\mathcal{AD}(i) <= \Omega_2$) **then**
    $\mathcal{T}^v(i) = \mathcal{T}^p(i)$
    $\mathcal{T}^m(i) = \mathcal{T}^s(i) - \mathcal{T}^p(i)$
    $\mathcal{T}^b(i) = \mathcal{T}(i) - (\mathcal{T}^v(i) \cup \mathcal{T}^m(i))$
end
else
    $\widehat{\mathcal{VS}} = \mathcal{VS} + \mathcal{D}(i)$
    Recalculate $\mathcal{T}^p(i)$ with tiles having distance to viewpoint $(\Phi^p(i), \Theta^p(i))$ in range $[0{:}\widehat{\mathcal{VS}}/2]$
    $\mathcal{T}^v(i) = \mathcal{T}^p(i)$
    $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$
end

viewing region for 360° video. DVS measures SI and TI values on a frame-by-frame basis for each 360° video. These values are then averaged across each segment. To establish the best spatial and temporal correlation, DVS employs the product of the averaged SI and TI values for each segment. Let $S(i)$ and $T(i)$ be the spatial and temporal scores, respectively, for the (i)th segment. The standard deviation reflects how much dispersion occurs over a while. The variation for the (i)th segment is given as:

$$ST(i) = std(S(i) * T(i) : S(i-1) * T(i-1)) \tag{6.13}$$

The content complexity for (i)th segment is approximated as follows:

$$\mathcal{CC}(i) = \frac{ST(i)}{max\{ST(i)|\forall i \in [1:S]\}} \tag{6.14}$$

Algorithm 6.2 is based on the difference between two prediction mechanisms and content complexity variations to adapt better to the high dynamics of head movements. It starts by calculating the primary ($\mathcal{T}^p(i)$) and secondary viewport ($\mathcal{T}^s(i)$) tiles sets considering the predicted coordinates and viewport size ($\mathcal{VS}$). Each viewport contains tiles whose centers are not located more than half of the viewport size from the center tile. The prediction difference between two prediction mechanisms is adjusted according to the content complexity variations computed in Eq. 6.14. Let $\mathcal{AD}(i)$ represent the adjusted distance of two prediction mechanisms with respect to the content information. We choose different thresholds to calculate the tiles for different streaming regions. Let Ω_1 and Ω_2 represent the thresholds of the great circle distance between two predicted viewpoints, such as $\Omega_1 < \Omega_2$ and Ω_1, $\Omega_2 \epsilon [0,\mathcal{VS}]$. If the adjusted distance between two predicted viewpoints is less than Ω_1; this refers to the *Fixed FoV* scenario, where the viewport tiles set containing all the tiles of the anticipated primary viewport and the background tiles are selected for streaming to the client. The marginal tiles set is empty and does not contain any tile. If the adjusted prediction difference is in between Ω_1 and Ω_2, we refer to this case as the *Neighbor region* scenario, where the goal is to include the marginal tiles to deal with possible head movement prediction errors. The primary viewport tiles set is considered as the potential viewport set for the (i)th segment. The non-identical tiles of the secondary viewport are considered the marginal tiles. The rest of the tiles are added to the background tiles set. Suppose the primary and secondary mechanisms' predicted viewpoints are farther than Ω_2. This represents the *Extended viewport* scenario, where the viewport size is extended by adding the actual arc distance between primary and secondary predicted viewpoints to the default viewport size. The primary viewport tiles set is recalculated with the updated viewport size. In this scenario, the primary viewport tiles set is labeled as viewport and streamed with the background tiles sets for the (i)th segment.

6.3.2.2 Prediction Error and Prediction Angles-based Tiles Selection

360° videos require high-resolution for the achievement of better-perceived quality. Tiles selection based on prediction angles as well as prediction errors can bound the discrepancy between actual and predicted tiles sets. The proposed tiles selection solutions (SFOV1 and SFOV2) select the tiles for different streaming regions based on a single viewport prediction mechanism. SFOV1 solutions consider the current and history angle information to output the future viewing positions. Let $(\Theta^{\hat{v}}(t\text{-}g), ..., \Theta^{\hat{v}}(t\text{-}1), \Theta^{\hat{v}}(t))$ represents the original yaw angles from time $(t-g)$ to time (t). The predicted yaw angle for a future horizon $(t+h)$ is given as:

$$\Phi^{v}(t{+}h) = \zeta * \text{Avg}(\Phi^{\hat{v}}(t\text{-}g) : \Phi^{\hat{v}}(t)) + (1-\zeta) * \Phi^{\hat{v}}(t) \tag{6.15}$$

where Avg() function returns the average of g samples and ζ is the smoothness factor. Similarly, the future pitch angle can be computed from the above equation.

$$\Theta^{v}(t{+}h) = \zeta * \text{Avg}(\Theta^{\hat{v}}(t\text{-}g) : \Theta^{\hat{v}}(t)) + (1-\zeta) * \Theta^{\hat{v}}(t) \tag{6.16}$$

The viewport prediction results ($\Phi(i)$ and $\Theta(i)$) are then used to compute the tiles to be requested for the next segment. Based on the predicted angles and the FoV of headset, the viewport tiles set is calculated similar to the CFOV tiles selection algorithms. In addition, SFOV solutions classify a 360° video frame into viewport, marginal, and background tiled videos. For each segment, the tiles are arranged in the ascending order of their distance from the viewport coordinates. Let $\mathcal{T}^{v}(i-1)$ be the predicted viewport tiles set with a center viewpoint tile represented as $t_c^{v}(i{-}1)$ for (i)th segment. Similarly, the original viewport tiles set for the $(i-1)th$ segment denoted as $\mathcal{T}^{\hat{v}}(i{-}1)$ contains a center viewpoint tile, $t_c^{\hat{v}}(i{-}1)$, along with surrounding tiles.

SFOV1: Algorithm 6.3 describes the overall tiles selection algorithm in SFOV1. To avoid the viewport deviations, SFOV1 dynamically adapts the coverage of the FoV by considering the closeness between actual and predicted viewports. Firstly, the SFOV1 client calculates the viewport tiles set by considering the predicted points

Algorithm 6.3 Tiles Selection Algorithm-SFOV1

Input :

$\mathcal{T}(i)\leftarrow$ *Sorted tile set for the streaming session*; $\Phi^v(i), \Theta^v(i)\leftarrow$ *Predicted viewpoint angles from Eq.6.16 and Eq. 6.15*; $\Phi^{\hat{v}}(i), \Theta^{\hat{v}}(i)\leftarrow$ *Actual viewpoint angles for the* (i)*th segment*; $\mathcal{D}(i-1)\leftarrow$ *Arc distance between actual and predicted viewpoints during* $(i-1)$*th segment*; $\mathcal{VS}\leftarrow$ *Size of the viewport in radians*

Result :

$\mathcal{T}^v(i), \mathcal{T}^m(i), \mathcal{T}^b(i)\leftarrow$ *Estimated viewport, marginal, and background tiles sets for the* (i)*th segment*

1 Calculate the tiles sets $\mathcal{T}^v(i)$ with predicted angles $(\Phi^v(i), \Theta^v(i))$ and $\mathcal{VS}$
2 **if** $(t_c^{\hat{v}}(i-1)\epsilon\mathcal{T}^v(i-1))$ **then**
3 $\quad\mathcal{T}^m(i-1) = \overline{(\mathcal{T}^v(i-1)\cap\mathcal{T}^{\hat{v}}(i-1))}\cap\mathcal{T}^{\hat{v}}(i-1)$
4 $\quad\mathcal{T}^m(i)= t_k(i), \forall k\epsilon[|\mathcal{T}^v(i)|+1; |\mathcal{T}^v(i)|+|\mathcal{T}^m(i-1)|]$
5 $\quad\mathcal{T}^o(i) = \overline{(T^v(i)\cup\mathcal{T}^m(i))}$
6 **end**
7 **else**
8 $\quad\widehat{\mathcal{VS}} = \mathcal{VS}+\mathcal{D}(i-1)$
9 $\quad$Calculate the tiles sets $\mathcal{T}^v(i)$ and $\mathcal{T}^b(i)$ with predicted angles $(\Phi^v(i), \Theta^v(i))$ and $\widehat{\mathcal{VS}}$
10 **end**
11 $\mathcal{D}(i) = ArcDistance(\Phi^{\hat{v}}(i), \Theta^{\hat{v}}(i), \Phi^v(i), \Theta^v(i))$

and the viewport size. If the predicted viewport tile set $\mathcal{T}^v(i-1)$ contains the viewpoint tile $t_c^{\hat{v}}(i-1)$ of the original viewport tiles set $\mathcal{T}^{\hat{v}}(i-1)$, the marginal tiles are calculated for $(i)th$ segment (lines 2-4). The tiles not belonging to the viewport and marginal regions are considered background tiles (line 5). In *Marginal Region* case, the size of the viewport is fixed similar to the *Fixed FoV* case. For *Extended FoV* case, if the viewpoint tile, $t_c^{\hat{v}}(i-1)$, does not belong to the predicted viewport tiles set $\mathcal{T}^v(i-1)$, the viewport size is updated by adding the great circular distance between original and predicted points to the actual or by default viewport size (line 8). The viewport tiles and background tiles sets are recalculated according to the predicted points and updated viewport size (line 11).

Fig. 6.3 shows the tiles selection cases in SFOV1 solution for the (i)th segment. Fig. 6.3a represents the *Fixed FoV* case in the SFOV1 solution where only the viewport and background tiles are selected under original viewport coverage because the original and predicted FoV tiles sets are identical. The marginal region selection in SFOV1 is different than what is depicted in Fig. 6.2b. Instead of adding a full viewport as a marginal region, SFOV1 adds a specific number of tiles based on the

(a) Fixed FoV (b) Marginal Region (c) Extended FoV

Figure 6.3: Tiles selection cases in SFOV1 for the (i)th segment. The black rectangle represents the predicted FoV, while the red rectangle represents the actual FoV for the $(i-1)$th segment. The potential viewport tiles are represented in red, whereas the marginal and background tiles are marked in green and light brown, respectively.

difference between original and predicted FoV tiles sets during $(i-1)$th segment, as shown in Fig. 6.3b. Similarly, the extended coverage of the FoV is also different from CFOV solutions, where the FoV is extended when the original viewpoint tile does not belong to the predicted viewport tile set during the previous segment. Fig. 6.3c represents *Extended FoV* scenario for the SFOV1 solution during the (i)th segment.

SFOV2: Similar to SFOV1, SFOV2 solution selects the tiles for different streaming regions based on a single viewport prediction mechanism. However different from SFOV1 and similar to CFOV2, SFOV2 involves content complexity variations for dynamic tiles selection. Algorithm 6.2 considers the adjusted prediction distance between two prediction mechanisms, while Algorithm 6.4 employs adjusted prediction difference between actual and predicted viewpoints for dynamic viewport selection. Algorithm 6.4 adjusts the prediction difference measured during the previous segment according to the content complexity variations in order to best select the tiles for the next segment. Let Ω'_1 and Ω'_2 represent the thresholds of great circular distance between original and predicted viewpoints, such as $\Omega'_1 < \Omega'_2$ and Ω'_1, Ω'_2 ϵ $[0,\mathcal{VS}]$. Suppose the adjusted distance between predicted and actual viewpoints during $(i-1)$th segment is not greater than the Ω'_1. This refers to the *Fixed FoV* scenario, where only the viewport and background tiles sets are calculated by considering the predicted angles and the actual viewport size. For the *Marginal Region* scenario, the viewport and marginal tiles are added with tiles having distance to viewpoint centre in range of $[0 : \mathcal{VS}/2]$ and $(\mathcal{VS}/2 : \mathcal{VS}/2 + \mathcal{D}(i-1)]$, respectively. The prediction performance could be worse under quick head turns for different motion scenes, and the adjusted distance could be even larger than Ω'_2. In this case, an

Algorithm 6.4 Tiles Selection Algorithm-SFOV2

Input :

$\mathcal{T}(i)\leftarrow$ *Sorted tile set for the streaming session*; $\Phi^{p}(i), \Theta^{p}(i), \Phi^{s}(i), \Theta^{s}(i)\leftarrow$ *Predicted angles*; $\Phi^{\hat{v}}(i), \Theta^{\hat{v}}(i), \Phi^{v}(i), \Theta^{v}(i)\leftarrow$ *Actual and Predicted angles*; $\mathcal{D}(i-1)\leftarrow$ *Arc distance between actual and predicted viewpoints*; $\mathcal{VS}\leftarrow$ *Size of the viewport in radians*; $\widehat{\mathcal{VS}} = 0\leftarrow$ *Updated size of the viewport initialized to 0*; $\mathcal{CC}(i)\leftarrow$ *Content complexity variations*; $\mathcal{AD}(i)\leftarrow$ *Adjusted distance*; $\Omega'_1, \Omega'_2\leftarrow$ *Distance thresholds*

Result :

$\mathcal{T}^{v}(i), \mathcal{T}^{m}(i), \mathcal{T}^{b}(i)\leftarrow$ *Estimated viewport, marginal, and background tiles sets for the* (i)*th segment*

$\Phi^{v}(i), \Theta^{v}(i) = (\Phi^{p}(i) + \Phi^{s}(i))/2, (\Theta^{p}(i) + \Theta^{s}(i))/2$

$\mathcal{AD}(i) = \mathcal{D}(i-1) + \mathcal{D}(i-1) * \mathcal{CC}(i)$

if $(\mathcal{AD}(i) \geq 0$ && $\mathcal{AD}(i) <= \Omega'_1)$ **then**

  Calculate $\mathcal{T}^{v}(i)$ with tiles having distance to viewpoint $(\Phi^{v}(i), \Theta^{v}(i))$ in range $[0{:}\mathcal{VS}/2]$

  $\mathcal{T}^{b}(i) = \mathcal{T}(i) - \mathcal{T}^{v}(i)$

end

else if $(\mathcal{AD}(i) > \Omega'_1$ && $\mathcal{AD}(i) <= \Omega'_2)$ **then**

  Calculate $\mathcal{T}^{v}(i)$ with tiles having distance to viewpoint $(\Phi^{v}(i), \Theta^{v}(i))$ in range $[0{:}\mathcal{VS}/2]$

  Calculate $\mathcal{T}^{m}(i)$ with tiles having distance to viewpoint $(\Phi^{v}(i), \Theta^{v}(i))$ in range $(\mathcal{VS}/2{:}\mathcal{VS}/2{+}\mathcal{D}(i-1)]$

  $\mathcal{T}^{b}(i) = \mathcal{T}(i) - (\mathcal{T}^{v}(i) \cup \mathcal{T}^{m}(i))$

end

else

  $\widehat{\mathcal{VS}} = \mathcal{VS} + \mathcal{D}(i-1)$

  Calculate $\mathcal{T}^{v}(i)$ with tiles having distance to viewpoint $(\Phi^{v}(i), \Theta^{v}(i))$ in range $[0{:}\widehat{\mathcal{VS}}/2]$

  $\mathcal{T}^{b}(i) = \mathcal{T}(i) - \mathcal{T}^{v}(i)$

end

$\mathcal{D}(i) = ArcDistance(\Phi^{\hat{v}}(i), \Theta^{\hat{v}}(i), \Phi^{v}(i), \Theta^{v}(i))$

Extended viewport streaming approach is adopted by increasing the viewport size with added background tiles to lower the impact of viewport mismatch.

6.3.3 Fixed Tiling-based Prioritized Bitrate Adaptation

In an HTTP adaptive streaming environment, the client mainly focuses on using the network resources efficiently to promote the desirable end-user streaming experience. The streaming client usually infers the network bandwidth from prior measurements for effective bitrate allocations.

Algorithm 6.5 allocates the bitrate to the outputted tiles of the *Tiles Selection* module to achieve the optimization aims described in Eq. (6.1) and Eq. (6.12). The

Algorithm 6.5 Tile Selection-specific Prioritized Bitrate Adaptation

Input :
$\{\mathcal{L}_1(i), ..., \mathcal{L}_j(i), ..., \mathcal{L}_N(i)\} \leftarrow$ *Video bitrate levels for tiles in* (i)*th segment*
$\mathcal{T}(i) \leftarrow$ *Tiles set containing* M *tiles for the* (i)*th segment*
$\mathcal{T}^v(i), \mathcal{T}^m(i) \leftarrow$ *Tiles sets for the viewport and marginal regions*
$|\mathcal{T}^v(i)|, |\mathcal{T}^m(i)| \leftarrow$ *Number of tiles in viewport and marginal regions*
$\widehat{\mathcal{B}}(i) \leftarrow$ *Estimated bandwidth for* (i)*th segment*
Result :
$w^{\mathcal{T}^v}(i), w^{\mathcal{T}^m}(i) \leftarrow$ *Priority related weights for viewport and marginal tiles*
$\mathcal{B}^v(i), \mathcal{B}^m(i) \leftarrow$ *Estimated bandwidth for the viewport and marginal tiles*
$\mathcal{L}^{\mathcal{T}}(i), \mathcal{L}^{\mathcal{T}^v}(i), \mathcal{L}^{\mathcal{T}^m}(i) \leftarrow$ *Video bitrates selected for the tiles of* (i)*th segment*

1 **if** $((1+\Delta) * \widehat{\mathcal{B}}(i) \leq \sum_{k \in \mathcal{T}(i)} \mathcal{L}_1^k)$ **then**
2 $\quad \mathcal{L}^{\mathcal{T}^v}(i) = \max_{j \in [1,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq \widehat{\mathcal{B}}(i)\}$
3 **else**
4 $\quad \mathcal{L}^{\mathcal{T}}(i) = \mathcal{L}_1^k(i), \forall k \in \mathcal{T}(i)$
5 $\quad \mathcal{B}(i) = \widehat{\mathcal{B}}(i) - \sum_{k \in \mathcal{T}(i)} \mathcal{L}_1^k(i)$
6 $\quad$ **if** $(\mathcal{T}^m(i) = \emptyset)$ **then**
7 $\quad\quad \mathcal{B}^v(i) = \mathcal{B}(i)$
8 $\quad\quad \mathcal{L}^{\mathcal{T}^v}(i) = \max_{j \in [2,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq \mathcal{B}^v(i)\}$
9 $\quad$ **else**
10 $\quad\quad w^{\mathcal{T}^m}(i) = (|\mathcal{T}^m(i)| / (2 * |\mathcal{T}^v(i)| + |\mathcal{T}^m(i)|))$
11 $\quad\quad w^{\mathcal{T}^v}(i) = 1 - w^{\mathcal{T}^m}(i)$
12 $\quad\quad \mathcal{B}^v(i) = \mathcal{B}(i) * w^{\mathcal{T}^v}(i)$
13 $\quad\quad \mathcal{B}^m(i) = \mathcal{B}(i) * w^{\mathcal{T}^m}(i)$
14 $\quad\quad \mathcal{L}^{\mathcal{T}^v}(i) = \max_{j \in [2,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq \mathcal{B}^v(i)\}$
15 $\quad\quad \mathcal{L}^{\mathcal{T}^m}(i) = \max_{j \in [2,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^m(i)} \mathcal{L}_j^k(i) \leq \mathcal{B}^m(i)\}$

adaptation for the (i)th segment is performed after completely fetching the $(i-1)$th segment. Algorithm 6.5 ensures that the segment size does not exceed the available bandwidth budget in fulfilling the constraints from Eq. (6.2). For the selected tiles of CFOV and SFOV solutions, Algorithm 6.5 outputs bitrates for one of the following three streaming regions: (i) viewport; (ii) viewport and background; (iii) viewport, marginal, and background regions. Suppose the lowest available video bitrate for the entire 360° segment is $(1 + \boldsymbol{\Delta})$ times greater than the estimated network bandwidth. In that case, only the viewport tiles with the highest permitted video bitrate are streamed to ensure a seamless video playback corresponding to Eq. (6.3) for the actual spatial smoothness defined in Eq. (6.11) (lines 1-2). This refers to the viewport-only streaming option for all the scenarios of CFOV and SFOV tiles

selection methods.

Next, the bitrate allocation is carried out for the entire frame by firstly assigning the lowest bitrate to all the tiles (Eq. (6.4)) to achieve a lower background penalty for the actual background tiles, defined in Eq. (6.9) (line 4). The bandwidth budget is revised then (line 5). Following this, if the marginal region does not contain any tiles, this refers to the streaming of tiles from two regions only, i.e., viewport and background tiles. The viewport bandwidth is determined to select the best possible video bitrate for the viewport tiles (lines 6-7). All the viewport tiles are streamed with the same selected uniform rate to improve the perceived visual quality levels mentioned in Eq. (6.8) (line 8). Next, the Algorithm 6.5 allocates the bitrates for three streaming regions, i.e., viewport, marginal, and background. If the marginal tiles set is non-empty, the proposed bitrate adaptation algorithm ensures that similar to the constraints in Eq. (6.5) and Eq. (6.6), the bitrates of the viewport and marginal tiles are not higher than the bandwidth of the viewport and marginal tiles, respectively. The bitrate allocation is performed for viewport and marginal tiles after calculating their priority-related weights. The weights are determined depending on the number of tiles in the viewport and marginal regions (lines 10-11). As the viewer is more interested in watching the viewport content at higher quality levels; therefore, viewport tiles are assigned with double weights compared to the marginal tiles to fulfill the constraints in Eq. (6.7). After that, the bandwidth for the viewport and marginal tiles are computed based on computed weights (lines 12-13). Finally, the video bitrate levels for the viewport and the marginal tiles are calculated. The maximum available video bitrates not exceeding each region's corresponding bandwidth budget are allocated to each viewport and marginal tiles (lines 14-15). Noteworthy is to state that for different tiles selections of CFOV and SFOV solutions, Algorithm 6.5 provides a solution to achieve an important balance between different quality objectives defined in Eq. (6.8)-Eq. (6.11) under constraints in Eq. (6.2)-Eq. (6.7) and maximize the optimization goal defined in Eq. (6.12).

Table 6.2: Average and Standard Deviations of Video Bitrates for the *Boxing*, *Conan*, *Football*, and *Spotlight* videos [Mbps]

Video	**QP**	**1s**			**2s**			**3s**		
		4x3	**6x4**	**8x6**	**4x3**	**6x4**	**8x6**	**4x3**	**6x4**	**8x6**
Boxing	22	24.81±8.28	25.39±8.47	26.13±8.63	49.63±16.16	50.8±16.53	52.29±16.84	74.44±23.56	76.18±24.1	78.41±24.55
	27	13.38±4.77	13.76±4.88	14.29±4.96	26.78±9.28	27.53±9.49	28.6±9.66	40.16±13.52	41.27±13.83	42.87±14.08
	32	7.21±2.72	7.46±2.77	7.83±2.82	14.44±5.27	14.92±5.38	15.69±5.46	21.65±7.68	22.37±7.84	23.5±7.96
	37	3.96±1.54	4.12±1.57	4.39±1.59	7.92±2.98	8.26±3.03	8.81±3.07	11.88±4.35	12.37±4.41	13.19±4.47
	42	2.19±0.85	2.31±0.86	2.52±0.87	4.39±1.64	4.63±1.65	5.07±1.67	6.58±2.4	6.93±2.41	7.57±2.44
Conan	22	10.6±3.6	10.68±3.6	10.88±3.62	21.22±6.79	21.37±6.78	21.77±6.82	31.63±9.62	31.85±9.6	32.44±9.66
	27	5.05±1.81	5.12±1.8	5.3±1.82	10.1±3.34	10.25±3.32	10.63±3.35	15.05±4.66	15.26±4.64	15.83±4.69
	32	2.43±0.91	2.51±0.9	2.68±0.91	4.87±1.62	5.02±1.61	5.39±1.62	7.26±2.22	7.48±2.2	8.01±2.22
	37	1.24±0.47	1.32±0.46	1.5±0.47	2.49±0.81	2.66±0.79	3.03±0.8	3.71±1.08	3.95±1.07	4.49±1.09
	42	0.72±0.24	0.8±0.24	0.98±0.24	1.44±0.4	1.62±0.39	1.99±0.4	2.14±0.53	2.4±0.53	2.94±0.55
Football	22	6.9±2.35	7.02±2.36	7.19±2.36	13.88±4.62	14.12±4.64	14.48±4.64	20.69±6.86	21.04±6.9	21.56±6.9
	27	3.55±1.38	3.64±1.39	3.81±1.39	7.14±2.71	7.34±2.73	7.68±2.73	10.64±4.03	10.92±4.06	11.41±4.06
	32	1.97±0.8	2.07±0.8	2.23±0.8	3.98±1.57	4.16±1.58	4.5±1.59	5.92±2.34	6.19±2.35	6.68±2.36
	37	1.15±0.47	1.24±0.47	1.4±0.47	2.31±0.93	2.49±0.94	2.82±0.94	3.44±1.38	3.7±1.39	4.18±1.39
	42	0.69±0.28	0.77±0.28	0.93±0.28	1.39±0.55	1.56±0.56	1.88±0.56	2.06±0.81	2.31±0.82	2.78±0.83
Spotlight	22	13.63±8.81	13.93±8.86	14.31±8.92	27.18±17.09	27.77±17.19	28.55±17.31	40.76±25.25	41.64±25.4	42.8±25.58
	27	7.21±5.33	7.44±5.36	7.77±5.4	14.37±10.35	14.84±10.41	15.5±10.48	21.54±15.3	22.25±15.39	23.23±15.49
	32	4.06±3.15	4.24±3.17	4.5±3.2	8.11±6.12	8.47±6.16	9.01±6.21	12.15±9.05	12.69±9.11	13.48±9.18
	37	2.36±1.79	2.49±1.81	2.72±1.82	4.7±3.48	4.98±3.51	5.44±3.54	7.04±5.14	7.46±5.18	8.13±5.24
	42	1.35±0.91	1.46±0.92	1.66±0.94	2.69±1.77	2.93±1.79	3.35±1.82	4.03±2.61	4.38±2.64	4.99±2.68

Table 6.3: Experimental Settings

Parameter	Configuration
Video, Duration, and Category	Boxing -2′52″ (Sport)
	Conan-2′44″ (Performance)
	Football- 2′44″ (Sport)
	Spotlight- 4′53″ (Film)
Resolution and FPS	Boxing (3840x1920)-29FPS
	Conan (3840x2160)-29FPS
	Football (3840x2160)-25FPS
	Spotlight (3840x2160)-30FPS
Tiling patterns	4x3, 6x4, 8x6
Encoding Versions	QPs (22, 27, 32, 37, 42)
Segment Duration (τ)	1s, 2s, 3s
Buffer Length	2s, 4s, 6s
Viewport Size	110°
Simulation Length	Video duration
Ω_1, Ω_1'	$\mathcal{VS}/3$
Ω_2, Ω_2'	$\mathcal{VS}/2$
QoE Coefficients	$(\alpha = 1, \beta = 0.5, \gamma = 0.1, \delta = 0.1)$
	$(\alpha = 1, \beta = 0.6, \gamma = 0.2, \delta = 0.2)$
	$(\alpha = 1, \beta = 0.8, \gamma = 0.3, \delta = 0.1)$

6.4 Experimental Evaluation

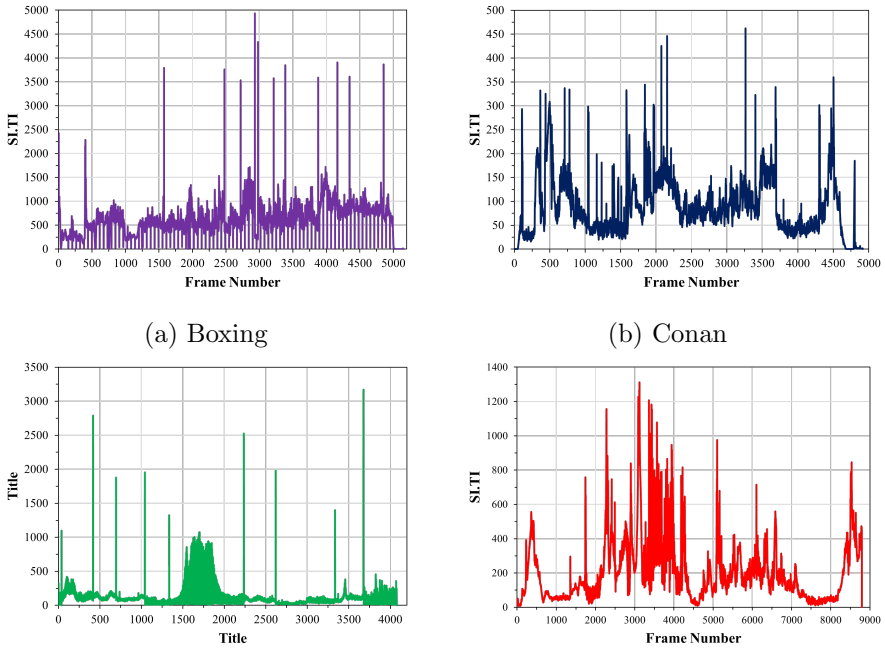
This section illustrates the trace-driven experimental evaluations of the proposed and closest state-of-the-art tile-based streaming solutions. We tested the solutions under a wide variety of content and network characteristics. Next, we present the experimental testing setup and the comparison schemes. Then, we show the experimental results and their analysis for each of the streaming solutions.

6.4.1 Experimental Setup

6.4.1.1 360° Video Player

The modeling and evaluation of the proposed system were conducted by employing an enhanced version of a VR player[1] proposed by Hooft et al. [48] running on an Ubuntu 16.04 machine with a 64-bit Intel Core i7-7500U CPU 2.7 GHz quad-core and 16 GB memory. The 360° player requested video segments from the HTTP server based on the available bandwidth and estimated viewpoint coordinates.

[1] https://github.com/jvdrhoof/VRClient



(a) Boxing (b) Conan

(c) Football (d) Spotlight

Figure 6.4: The per-frame SI.TI values for Boxing, Conan, Football, and Spotlight videos.

6.4.1.2 360° Videos and Head Movement Traces

The experimental evaluation was performed using a trace-driven simulation involving real viewport traces of 48 VR users from an open-source dataset [225]. This dataset is widely used, including in [53, 252, 216, 278]. From this dataset, we chose four videos which include a wide range of motion content: *SHOWTIME Boxing*[2], *Conan360-Sandwich*[3], *LOSC Football*[4], and *Google Spotlight-HELP*[5]. The content category also differed across the selected clips; the first and third streams belong to the sports category, while the second and fourth clips belong to artistic performance and action film, respectively. These videos are referred to as *Boxing*, *Conan*, *Football*, and *Spotlight* throughout this chapter. The videos were encoded at 4K resolution using FFmpeg[6] software. Fig. 6.4 illustrates the SI.TI values for each video, computed using code from [279]. The content complexity variations are then computed for each video using Eq. 6.13 and Eq. 6.14.

[2]https://www.youtube.com/watch?v=raKh0OIERew
[3]https://www.youtube.com/watch?v=FiClYLgxJ5s
[4]https://www.youtube.com/watch?v=lvH89OkkKQ8
[5]https://www.youtube.com/watch?v=G-XZhKqQAHU
[6]https://ffmpeg.org/

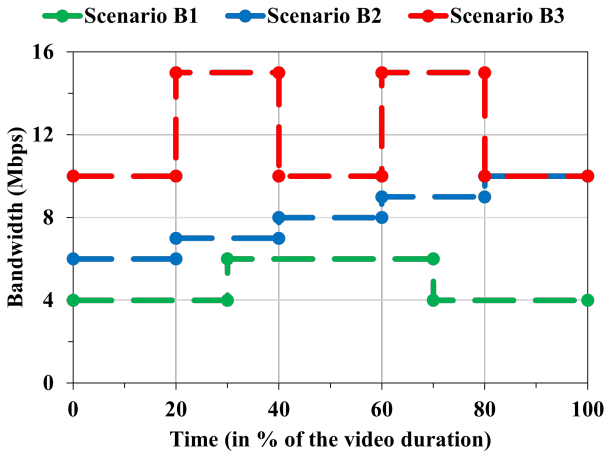


Figure 6.5: Bandwidth scenarios used in experiments.

All the videos were spatially split into 4x3, 6x4, and 8x6 tiling patterns. The tiles were processed using an open-source Kvazaar encoder [280] considering five different quantization parameters (QPs) (i.e., 22, 27, 32, 37, and 42). Subsequently, DASH video segments were generated using GPAC MP4Box[7] with a duration of 1s, 2s, and 3s, respectively. The average segment sizes for each video and encoding rate are illustrated in Table 6.2. The values of ζ, Δ, and g were set to 0.2, 0.5, and 0.5s. The viewport coverage was set to 110°, as used in the head movement collection by Wu et al. [225]. The length of each experiment was adjusted following the duration of the employed video. In practice, for 360° video streaming, the buffer should be as small as possible to accommodate the new video chunks in response to the user movements within the immersive video. For each experiment, the buffer level is set to two video segments.

We performed simulations using different values of Ω_1, Ω_2, Ω'_1, and Ω'_2. The reported experimental results are obtained by setting Ω_1 , $\Omega'_1 = \mathcal{VS}/3$ and Ω_2, $\Omega'_2 = \mathcal{VS}/2$. Table 6.3 presents the content characteristics and experimental settings in terms of tiling pattern, encoding versions, segment length, viewport size, and buffer levels.

6.4.1.3 Bandwidth Scenarios

The experiments were performed using the following dynamic bandwidth scenarios, also shown in Fig. 6.5:

[7]https://gpac.wp.imt.fr/mp4box/

1. Scenario B1: The bandwidth of the link between the HTTP client and server was varied for each video as follows: 4 Mbps for the first 30% of the segments, 6 Mbps for the following 40% segments, and then back to 4 Mbps until the end of the video playback. Scenario B1 was used for the experiments performed for 1s segment duration. This scenario considers relatively stable low bandwidth connections.
2. Scenario B2: A stepwise switch-up connection was considered for 2s video segment duration where the bandwidth was set to 6 Mbps for the first 20% of the segments, and then increased with 1 Mbps after every 20% of the playback. This stepwise switch up scenario with an average bandwidth of 8 Mbps is employed to test how the proposed and comparative clients utilize this bandwidth to increase the viewport quality.
3. Scenario B3: The bandwidth of the link between the client and HTTP server varied in an on-off fashion between 10 Mbps and 15 Mbps after each 20% of the playback for all the videos. Scenario B3 was employed for 3s video segment duration. This scenario helps to observe the spatial and temporal visual quality variations trends with the increase in bandwidth fluctuations.

6.4.1.4 QoE Weight Coefficients

The QoE weight coefficients (α, β, γ, δ) are such selected in order to emphasize different QoE objectives or a combination of such objectives, i.e., maximize viewport quality, minimize background content quality, and reduce spatial and temporal quality variations. A large α indicates that the user cares more for the viewport quality, whereas large β, γ, and δ values indicate that the user prefers low background quality, low temporal and low spatial quality variance, respectively. To verify the effectiveness of the proposed solution, the following sets of QoE coefficients are chosen:

1. Coefficients C1: $(\alpha = 1, \beta = 0.5, \gamma = 0.1, \delta = 0.1)$: This case reflects situations when users are more interested in visual quality compared to the visual smoothness.
2. Coefficients C2: $(\alpha = 1, \beta = 0.6, \gamma = 0.2, \delta = 0.2)$: This case refers to situa-

Table 6.4: Comparison Between Proposed and Existing Tile-based Adaptive 360° Video Streaming Solutions in terms of Prediction Mechanisms, Tiles Selection Decisions, Tiles Selection Scenarios, and Streaming Regions

Works	Prediction Mechanism		Tiles Selection Decisions		Tiles Selection Scenarios				Streaming Regions		
	Single	Combined	Prediction Angles	Prediction Errors	*FFoV*	*MR*	*EFoV*	*FF*	Viewport	Viewport Background	Viewport Marginal Background
UVP [48]	✓	×	✓	×	✓	×	×	×	×	✓	×
CTF [48]	✓	×	✓	×	×	×	×	✓	✓	×	×
PBA [49]	✓	×	✓	×	✓	×	×	×	×	✓	×
AVR [107]	✓	×	✓	×	✓	✓	×	×	×	×	✓
VIS	×	×	×	×	×	×	×	✓	✓	×	×
CFOV1	×	✓	✓	×	✓	✓	✓	×	✓	✓	✓
CFOV2	×	✓	✓	×	✓	✓	✓	×	✓	✓	✓
SFOV1	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓
SFOV2	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓

tions in which there is more sensitivity towards bandwidth wastage for outside tiles and visual smoothness compared to C1.

3. Coefficients C3: $(\alpha = 1, \beta = 0.8, \gamma = 0.3, \delta = 0.1)$: In the third case, we consider increasing the weights of background quality penalty and the temporal quality oscillations for a scenario where a user is more concerned about bandwidth allocation to background tiles and temporal visual smoothness.

In practice, the QoE weight coefficients can be selected in order to emphasize different QoE objectives such as maximizing the viewport quality, minimizing the background content quality, and reducing the spatial and temporal quality variations or their combination.

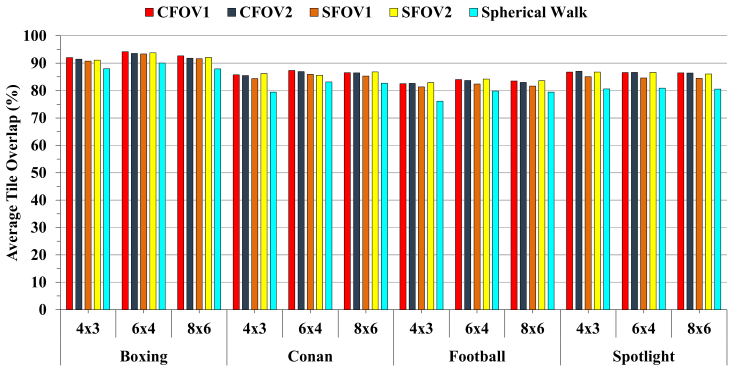
6.4.1.5 Baseline Algorithms

We compare the performance of **CFOV** and **SFOV** solutions with five tile-based streaming solutions. The first approach, denoted as **UVP** [48], classifies the tiles into the viewport and non-viewport regions; no marginal region is considered here. Initially, it allocates the minimum representation to all the tiles. Later, the quality representation of viewport tiles is uniformly increased to the maximum level. If the bandwidth is still available, the quality of the non-viewport tiles is also improved. The second method, referred to as **CTF** [48], increases the quality, starting from the viewport center to the last tile. This scheme is an extended version of **UVP** where the tiles are sorted based on the distance from the predicted points. Instead of increasing the viewport quality uniformly, this approach improves the tiles'

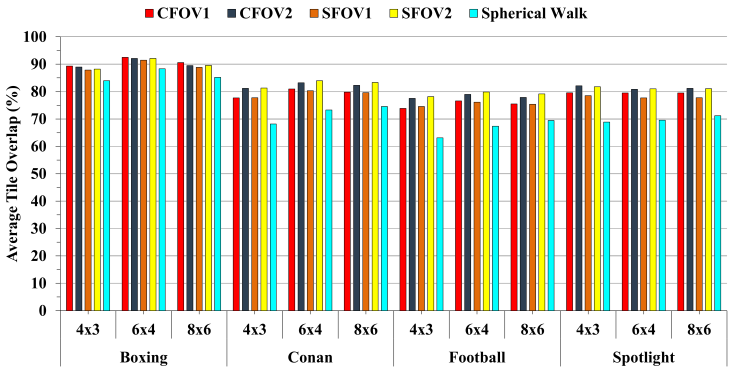
quality from the viewpoint to the last tile. The third approach denoted as **PBA** [49], performs divided tiles into three zones, i.e., Z_1 (viewpoint tile), Z_2 (viewpoint surrounding tiles), and Z_3 (background tiles). It performs priority-based bitrate adaptation for tiles belonging to these regions while respecting the available bandwidth budget. The fourth approach, denoted as **AVR** [107], divides the tiles of the equirectangular frames into three regions (i.e., viewport, adjacent, and outside) and assigns variable bitrates to the visible and non-visible regions by incorporating the viewer head movements. All of the reference tile-based delivery solutions incorporate viewer head movements for adaptive bitrate selection. The fifth method used for comparison is Viewport-Independent Streaming (**VIS**). It's a widely deployed 360° streaming method over the popular streaming platforms, e.g., YouTube, that comprises streaming the whole 360° video in the same quality subject to the bandwidth constraint. Unlike the previous works, the marginal area in our method is a special case. It could be adjacent to the viewport or can reside at a distance from the viewport, depending on the difference in the prediction mechanisms. Table 6.4 illustrates the significant differences between the proposed and the comparative schemes in terms of prediction mechanism, FoV selection, and streaming strategy. The *Full Frame* scenario is employed in **CTF** and **VIS** solutions. **CTF** and **VIS** extracts the full-frame as potential viewport set for each segment during the entire streaming sessions.

6.4.2 Experimental Results

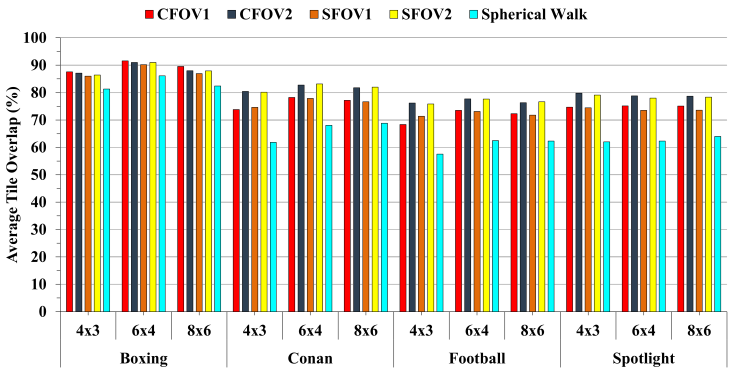
This section presents the experimental results and analysis of how the proposed and existing tile delivery solutions react under different testing profiles. First, the performance of tiles selection in terms of tiles overlap is assessed. This metric indicates the closeness of actual and predicted viewport. Secondly, the effectiveness of proposed bitrate adaptation algorithms involving tiles selection solutions is discussed in terms of the QoE metric elaborated in section 6.3.1. Next, the bandwidth utilization results for the viewport region achieved by the streaming solution are discussed. Finally, the streaming performance of the proposed solution in terms of tiles selection and bitrate adaptation scenarios is described.



(a) Segment duration: 1s



(b) Segment duration: 2s



(c) Segment duration: 3s

Figure 6.6: Average tiles overlap achieved by **CFOV** and Spherical Walk methods for 1s, 2s, and 3s prediction horizons; results from four videos watched by 48 users.

6.4.2.1 Viewport Prediction and Tile Selection Assessment

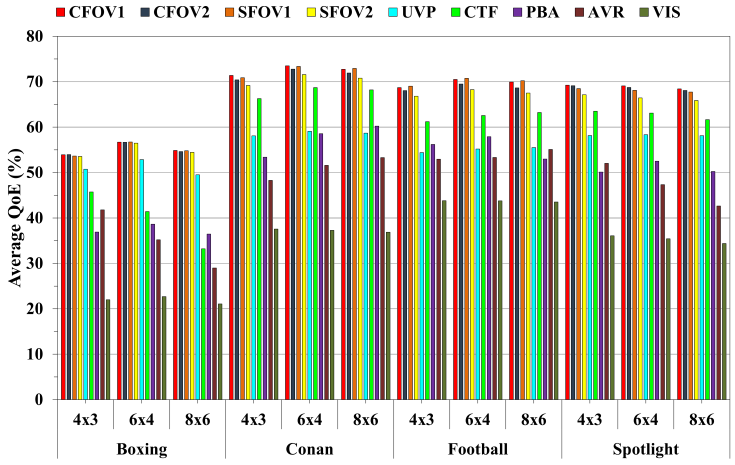
This metric directly calculates the fraction of the actual viewport tiles $\mathcal{T}^{\hat{v}}(i)$ covered by the predicted viewport tiles $\mathcal{T}^{v}(i)$. We compare **CFOV** and **SFOV** solutions with a *spherical walk* approach proposed in [48]. Fig. 6.6 illustrates the average tiles overlap for four videos prepared with three segment durations and three tiling patterns played using 48 head movement traces. The *spherical walk* method is adopted

in **UVP**, **CTF**, **PBA**, and **AVR** streaming algorithms. A slightly lower tile overlap is observed for the 4x3 tiling pattern since all the methods arrange the tiles considering arc distance between the viewpoint and each tile's center. The viewers have relatively fast head movements when watching the *Football* video since it is an outdoor sports video and contains several fast-moving objects. A higher average tile overlap is observed for the *Boxing* video since the users have more regular and stable head movements for this video, which are easy to forecast. It is notable that proactive tile selection in **CFOV** and **SFOV** solutions mainly yields high matching performance and outperforms the *spherical walk* approach for different user behaviors. In particular, for all four videos, **CFOV** and **SFOV** solutions experience an average tile overlap of 86.9% (Fig. 6.6a). The lower average tiles overlap results for *spherical walk* approach are mainly due to the higher mismatch between actual and predicted viewport positions even when the head movements are stable. As seen, for the *Conan* video, **CFOV1** and **CFOV2** outperform the *spherical walk* method by 4.79% and 4.56% for 1s (Fig. 6.6a), by 7.48% and 10.23% for 2s (Fig. 6.6b), and by 10.18%, 15.41% for 3s prediction horizon (Fig. 6.6c). Similarly, **SFOV1** and **SFOV2** solutions show their superior capability in increasing viewport match for the *Spotlight* video and outperforms the *spherical walk* method across all tiling patterns by 4.04% and 5.82% for 1s (Fig. 6.6a), by 8.12% and 11.41% for 2s (Fig. 6.6b), and by 11.10% and 15.73% for 3s prediction horizon (Fig. 6.6c). Fig. 6.6a shows that for the 1s prediction window, **CFOV** and **SFOV** solutions gets very close to a perfect viewport match (e.g., 92.63% and 92.15% for the *Boxing* video, 86.4% and 86.73% for the *Conan* video, 83.25% and 82.71% for the *Football* video, and 86.68% and 85.61% for the *Spotlight* video) across all tiling patterns. At the same time, a very small percentage of viewport mismatch is observed when increasing the prediction horizon from 1s to 3s (e.g., 3.5% and 4.07% for the *Boxing* video, 7.42% and 6.65% for the *Conan* video, 9.17% and 8.31% for the *Football* video, and 9.64% and 9.43% for the *Spotlight* video) (Fig. 6.6c). The average tiles overlap drops for *spherical walk* approach is 5.39% for the *Boxing* video, 15.54% for the *Conan* video, 17.69% for the *Football* video, and 17.90% for the *Spotlight* video. It is also notable from Fig. 6.6c that **SFOV2** outperforms the *spherical walk* approach by up to 5.18% for

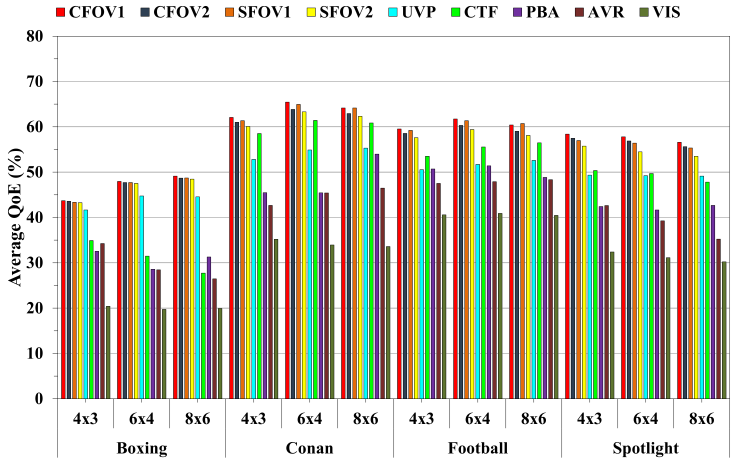
the *Boxing* video, up to 15.54% for the *Conan* video, up to 15.94% for the *Football* and up to 15.73% for the *Spotlight* videos. There is a higher viewport matching performance for the proposed solutions because the attractive tiles selection cases adapts better to the varying user behaviors for different video characteristics. The evaluation results show that our dynamic tiles selection methods ensures stable visual angles to provide users with a favorable QoE. As a result, it can be concluded that **CFOV** and **SFOV** solutions exploit user-watching information better than the *spherical walk* method and reduce the mismatch between the actual and predicted FoV tiles.

6.4.2.2 Bitrate Adaptation Assessment: Average QoE

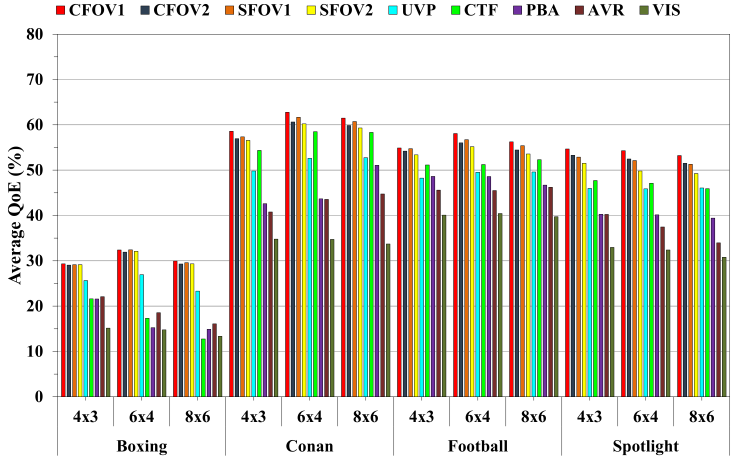
Average QoE with Coefficient Set C1: To assess the bitrate adaptation algorithm performance, we computed the average quality score based on the QoE metric defined in Section 6.3.1. Fig. 6.7 presents the average QoE scores achieved by each streaming algorithm under three dynamic bandwidth scenarios for the *Boxing*, *Conan*, *Football*, and *Spotlight* videos, which are spatially and temporally split into three tiling patterns and three segment durations, respectively. The performance results are depicted for the QoE weight coefficient set C1 in Fig. 6.7. The experimental findings reveal that **CFOV** and **SFOV** attain an optimal trade-off when selecting the streaming tiles quality and the highest QoE among the approaches compared. The algorithms' performance decreases accordingly with the increase of segment length. The *Boxing* video requires higher bitrates for achieving a particular quality score compared to those of the other videos (as can be seen in Table 6.2). Hence, achieving higher QoE with limited dynamic bandwidth is more challenging for this kind of video. The higher quality scores for *Conan* video are due to the higher average tiles overlap compared to the *Football* video. Contrarily, *Football* video achieve slighter higher QoE levels compared to the *Spotlight* video since it has the lowest average segment sizes. This indicates that content features and user interaction strongly influence the streaming performance of 360° videos. It can be noticed that all of the streaming methods achieve slightly higher performance for the 6x4 tiling pattern followed by the 8x6 tiling pattern for all four videos. This



(a) Segment duration: 1s, Bandwidth: Scenario B1



(b) Segment duration: 2s, Bandwidth: Scenario B2



(c) Segment duration: 3s, Bandwidth: Scenario B3

Figure 6.7: Average QoE results achieved by nine streaming clients for 48 VR users with QoE weight coefficients set to C1; results from four videos prepared in three tiling patterns and three segment durations.

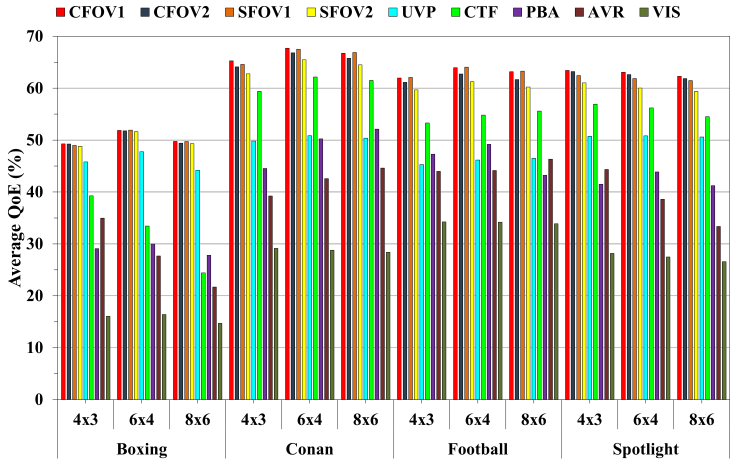
is because the higher viewport overlap and feasible segment sizes yield higher QoE levels with this tiling pattern.

The streaming results for 1s segment duration in dynamic bandwidth Scenario B1 illustrated in Fig. 6.7a show that **CFOV** solutions achieve about 55.12% and 72.12% average QoE scores for the *Boxing* and *Conan* videos, and **SFOV** solutions achieve on average 68.77% and 67.29% for the *Football* and *Spotlight* videos. Comparatively, **UVP** and **CTF** achieves about 51.03% and 40.12% for the *Boxing* video, **CTF** and **PBA** achieves about 67.72% and 57.39% for the *Conan* video, **PBA** and **AVR** achieves about 55.69% and 53.79% for *Football* video, and **AVR** and **VIS** achieves about 47.33% and 35.29% for the *Spotlight* video. The lowest average quality scores are observed for the **VIS**, followed by the **AVR** since these solutions observe the highest background quality penalty. Fig. 6.7b displays the average QoE values of the proposed solutions and five reference methods for the 2s segment duration when a stepwise switch-up bandwidth scenario B2 is employed. Interestingly, **CFOV** and **SFOV** solutions always have better performance than the reference methods. This is because our approaches favor high viewport quality. Compared to **UVP**, **CTF**, and **VIS** methods, **SFOV** solutions improve the average QoE by up to 4.05%, 20.87%, 17.29%, 28.65%, respectively, for the 8x6 *Boxing* video. The average improvements in QoE for **CFOV1** and **CFOV2** over **UVP**, **PBA** and **AVR** methods are 9.78%, 19.21%, and 19.25% for the 6x4 *Conan* video. For the *Spotlight* video, the proposed solutions achieves about 7.05%, 6.98%, 14.03%, 17.22%, and 25.03% higher QoE values compared to the **UVP**, **CTF**, **PBA**, **AVR**, and **VIS** algorithms, respectively (Fig. 6.7b). The **CTF** method performs better than **UVP** for *Conan* and *Football* videos. This is because **CTF** undergoes lower spatial and temporal quality variations for lower data rates of these videos. On the other hand, **UVP** performs better than **CTF** for the higher segment sizes video (i.e., *Boxing*) by uniformly increasing the quality of the visible and non-visible regions.

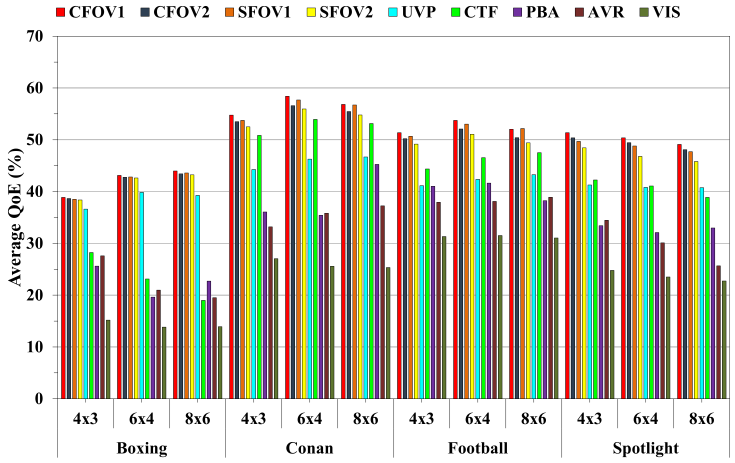
Similar to the 1s and 2s cases, the proposed solutions mostly achieve the highest average QoE value for all three tiling patterns when the segment duration is set to 3s (Fig. 6.7c). Interestingly, the average performance of all proposed and reference bitrate adaptation algorithms drops to 7.82% and 13.52% when segment duration is

increased from 1s to 2s and 1s to 3s, respectively. The *Football* video has the most petite, average segment sizes; however, the head movement traces for this video contain significant variations in viewing directions. Therefore, the viewport-based methods achieve slightly lower QoE values than the *Conan* video for the bandwidth fluctuation scenario B3. The **CFOV1** solution achieves 62.76% on average QoE for the 6x4 *Conan* video compared to the scores of 60.64%, 61.67%, 60.22%, 52.62%, 58.49%, 43.68%, 43.54%, and 34.71% achieved by **CFOV2**, **SFOV1**, **SFOV2**, **UVP**, **CTF**, **PBA**, **AVR**, and **VIS** algorithms, respectively. The proposed solutions surpass the baseline algorithms by obtaining 5.27%, 14.89%, 16.96%, 13.66%, and 17.43% higher QoE scores for the 6x4 *Boxing* video (Fig. 6.7c). The performance drop for the **PBA** and **AVR** methods, even under stable head movements, is mainly because they needlessly increase the quality of the adjacent tiles.

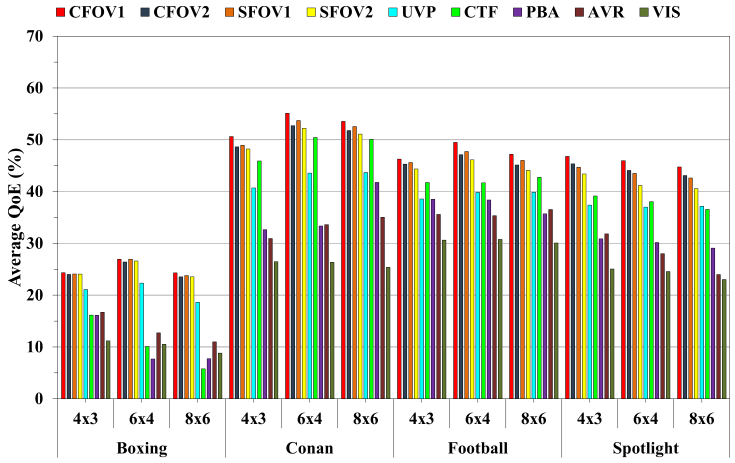
Average QoE with Coefficient Set C2: To better understand bitrate adaptation approaches' performance, we increase the weights of the background quality and spatial and temporal quality oscillations penalties. Fig. 6.8 compares the average gains on QoE for different content types with the QoE weight coefficient set C2. Fig. 6.8a indicates that **CFOV2** outperforms **UVP**, **CTF**, **PBA**, **AVR**, and **VIS** methods by 4.22%, 17.77%, 21.17%, 22.03%, and 34.42%, respectively, for *Boxing* video with a 1s segment duration. Similarly, **SFOV2** achieves about 8.69%, 1.78%, 15.50%, 18.99%, and 28.44% higher QoE values for *Conan* video with a 2s segment duration (Fig. 6.8b). **CFOV1** outperforms **UVP** and **CTF** methods by 8.23% and 5.59% for *Football* video with a 3s segment duration (Fig. 6.8c). Similarly, for *Spotlight* video with a 3s segment duration, **SFOV1** attains 13.57%, 15.66%, and 19.39% higher quality scores than **PBA**, **AVR**, and **VIS** methods, respectively. The proposed solutions are highly vulnerable to imperfect viewport predictions because only the viewport tiles can be streamed to the client under low network bandwidth. However, streaming a lower amount of data provides significant benefits for **CFOV**. For all videos in Fig. 6.8a, **CFOV1** and **CFOV2** solutions achieve about 12.12%, 9.41%, 18.69%, 21.91%, and 33.86% higher average QoE compared to **UVP**, **CTF**, **PBA**, **AVR**, and **VIS** methods. Fig. 6.8b show that the average gain on QoE achieved by **SFOV** solutions for all four videos with 4x3, 6x4, and 8x6 tiling patterns is



(a) Segment duration: 1s, Bandwidth: Scenario B1



(b) Segment duration: 2s, Bandwidth: Scenario B2

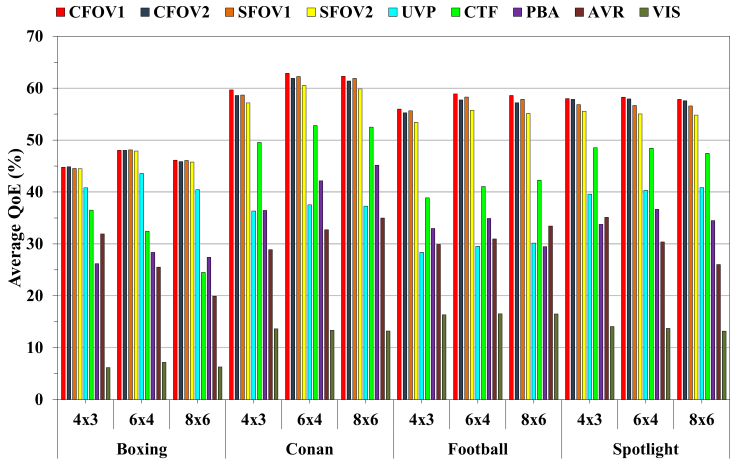


(c) Segment duration: 3s, Bandwidth: Scenario B3

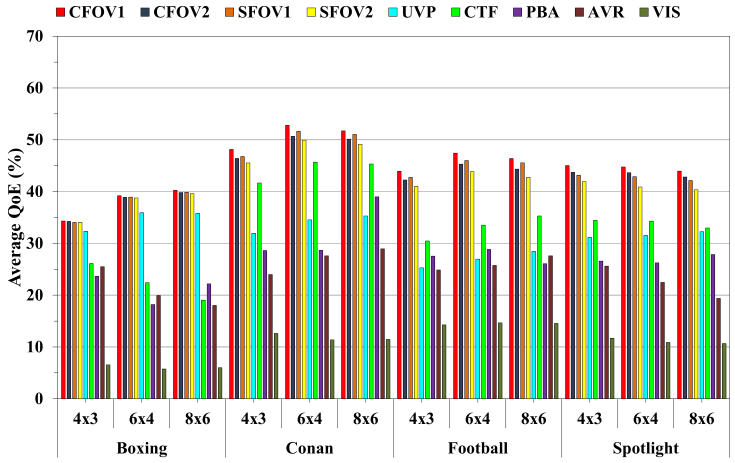
Figure 6.8: Average QoE results achieved by nine streaming clients for 48 VR users with QoE weight coefficients set to C2; results from four videos prepared in three tiling patterns and three segment durations.

about 7.01%, 8.15%, 15.22%, 17.26%, and 25.08% compared to other solutions. For all the videos with a 3s segment duration in Fig. 6.8c, all the proposed solutions outperforms **UVP**, **CTF**, **PBA**, **AVR**, and **VIS** methods by about 6.65%, 6.76%, 13.12%, 14.02%, and 18.89%, respectively. It can be noticed that with the more considerable difference between the actual and predicted viewports, the average QoE in Fig. 6.8c is lower than what is depicted in Fig. 6.8a and Fig. 6.8b. Interestingly, all the comparative methods tend to download the highest quality levels when bandwidth is higher than the available quality levels. This favors the QoE metric by lowering spatial and temporal quality variations. **CFOV** and **SFOV** solutions' average QoE is the highest for *Conan* and *Football* videos, followed by the **CTF** and **UVP** methods. This is due to the fact that the proposed solutions preserve the highest visual quality by dealing effectively with the abrupt view switching during each adaptation interval. **CTF** leads to better results than **UVP** for the *Conan* and *Football* videos since it directly assigns the highest quality to the viewpoint tile based on the available bandwidth budget. Instead of completely relying on bandwidth, **CFOV** adjusts the viewport quality by dynamically deciding the coverage of the FoV. Moreover, **CFOV** incurs a lower bandwidth consumption without noticeable quality degradation by streaming background tiles at the lowest quality.

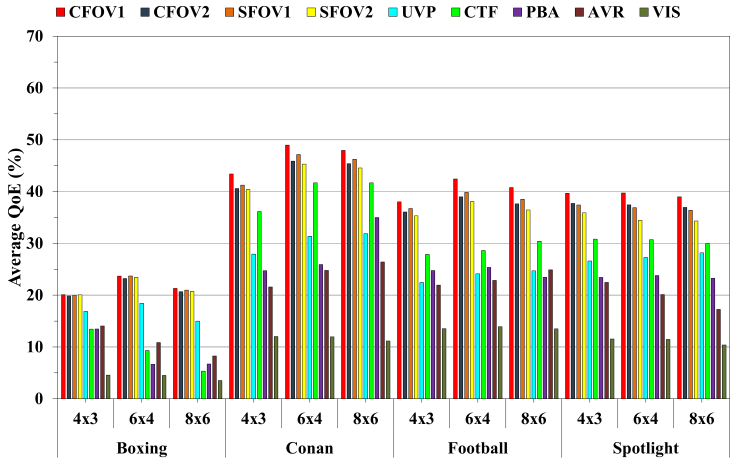
Average QoE with Coefficient Set C3: Next, the performance of fixed tiling-based bitrate adaptation algorithm incorporating **CFOV** and **SFOV** tiles selection solutions and those of the other reference approaches are tested by setting the QoE weight coefficients to set C3. Fig. 6.9 illustrates the video quality experienced, averaged across the 48 users for the different videos and tiling patterns. It can be noted that with the increase of background quality and spatial and temporal quality penalties, the average quality scores in Fig. 6.9 is lower than what are depicted in Fig. 6.7 and Fig. 6.8. Fig. 6.9a shows that **CFOV1** outperforms the baseline streaming approaches by achieving 19.83%, 26.50%, 27.73%, and 24.53% on average QoE improvements for the *Boxing*, *Conan*, *Football*, and *Spotlight*l videos, respectively. In particular, **CFOV1** improves the average QoE by up to 18.88% compared to the **UVP**, up to 13.03% compared to the **CTF**, up to 21.94% compared to the **PBA**, up to 25.96% compared to the **AVR**, and up to 42.42% compared to



(a) Segment duration: 1s, Bandwidth: Scenario B1



(b) Segment duration: 2s, Bandwidth: Scenario B2



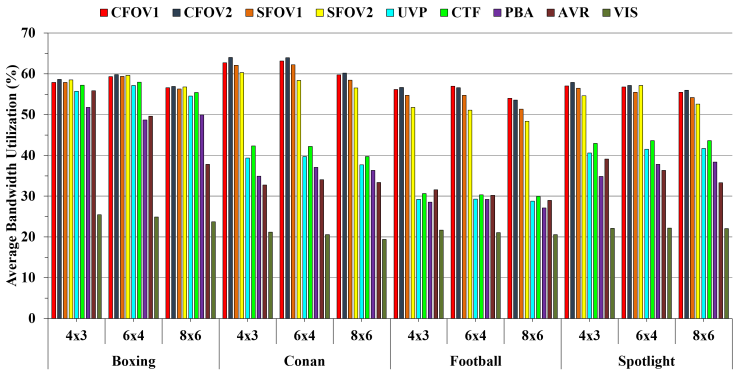
(c) Segment duration: 3s, Bandwidth: Scenario B3

Figure 6.9: Average QoE results achieved by nine streaming clients for 48 VR users with QoE weight coefficients set to C3; results from four videos prepared in three tiling patterns and three segment durations.

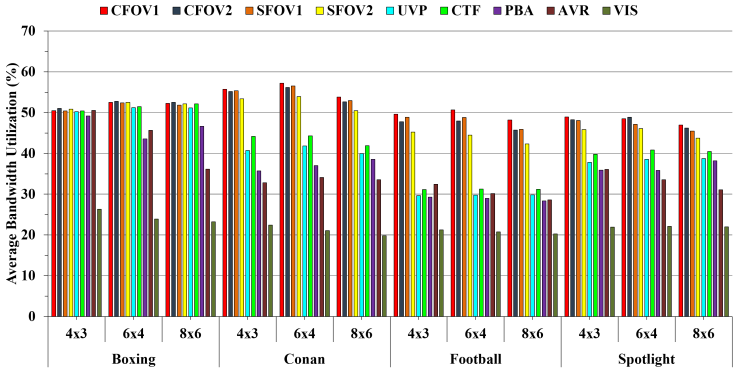
VIS methods for the entire test dataset. By eliminating **VIS** from comparison as it results in too low QoE scores, **SFOV1** outperforms the four comparative methods by resulting in 14.77% (for *Boxing* video), 20.39% (for *Conan* video), 23.76% (for *Football* video), and 18.23% (for *Spotlight* video) higher QoE values. Fig. 6.9b shows the average QoE comparison under Scenario B2 when the segment duration is set to 2s. **SFOV2** experiences only 7.9% and 16% viewport deviations for the *Boxing* and *Conan* videos with a 6x4 tiling pattern (Fig. 6.6b); therefore, it efficiently utilizes the available bandwidth budget and achieves average QoE improvements of up to 14.26%, 3.97%, 16.10%, 21.35%, and 36.38% for the *Conan* video, and up to 15.65%, 9.44%, 15.05%, 16.48%, and 28.04% for the *Football* video, in comparison to the other five methods. Fig. 6.9c shows that for the *Football* video with a segment duration of 3s, **CFOV2** and **CFOV1** achieve about 13.81%-16.65%, 8.60%-11.44%, 13.04%-15.88%, 14.34%-17.18%, and 23.91%-26.76% higher QoE in comparison to the **UVP**, **CTF**, **PBA**, and **AVR** methods, respectively. Similarly, for the *Spotlight* video, **SFOV2** and **SFOV1** has an effective QoE improvement between 5.39% and 24.77% in comparison with the other approaches.

6.4.2.3 Bitrate Adaptation Assessment: Bandwidth Utilization

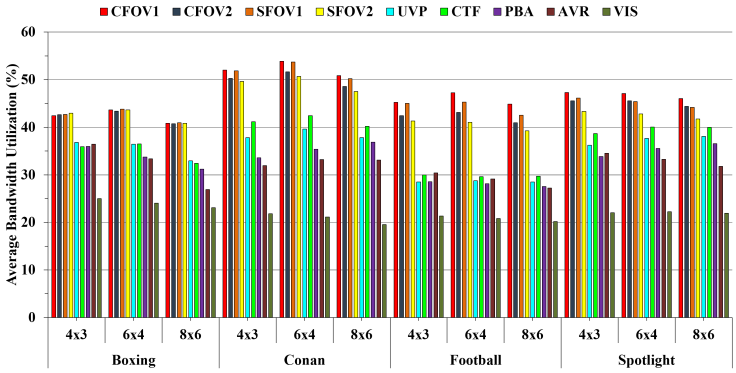
The results for bandwidth utilization for the viewport region are discussed in this section. The bandwidth utilization for the (i)th segment is measured as the ratio of the video bitrates of the viewport and the bitrates of the whole 360° segment. Fig. 6.10 shows the average bandwidth utilization for the viewport region. The depicted results are achieved by nine streaming clients for four videos watched by 48 VR users in three different segment durations. Fig. 6.10a shows the average bandwidth utilization results for 1s segment duration. It can be noted that **CFOV** and **SFOV** solutions achieve nearly similar higher viewport bandwidth utilization in comparison to other streaming solutions. At the same time, the lowest bandwidth utilization is observed for the **VIS** streaming client when playing all the four videos. Interestingly, **VIS** solution occupies about 22.08% bandwidth for the visible region and 77.91% for the non-visible region. This is truly unacceptable for 360° video streaming. Comparatively, the proposed solutions consumes on average 57.21% bandwidth



(a) Segment duration: 1s



(b) Segment duration: 2s



(c) Segment duration: 3s

Figure 6.10: Average bandwidth Utilization for viewport tiles achieved by 9 streaming clients for 48 VR users; results from four videos prepared in three tiling patterns and three segment durations.

for the viewport region. **CFOV** and **SFOV** solutions achieve on average 20.98% higher bandwidth utilization for all videos watched in three tiling patterns compared to other streaming solutions. To be more specific, **SFOV** and **CFOV** solution outperforms **UVP** by 14.94%-16.95%, **CTF** by 13.21%-15.23%, **PBA** by 18.32%-20.33%, **AVR** by 19.3%-21.31%, and **VIS** by 34.13%-36.14% across the entire test dataset. Among proposed solutions, **CFOV2** slightly achieves higher bandwidth

utilization for *Conan*, *Spotlight*, and *Surfing* videos compared to **CFOV1**, **SFOV1**, and **SFOV2** solutions. The comparative solutions achieves higher bandwidth utilization for the **Boxing** video compared to other videos. For instance, the average bandwidth utilization for the **Boxing** video is 47.05%, while it is 34.05%, 27.81%, and 36.01% for the **Conan**, **Football**, and **Spotlight** videos. This is because each streaming method strives for higher bandwidth utilization for the higher segment sizes of the *Boxing* video. In proposed solutions, the higher bandwidth utilizations for *Conan* (60.98%), *Football* (53.84%), and *Spotlight* (55.9%) videos are due to the fact that the proposed solutions do not only utilize the lowest background bitrate but also the highest possible viewport bitrate. For *Conan* video, **CFOV1** outperforms **UVP** by 22.94%, **CFOV2** outperforms **CTF** by 21.31%, **SFOV1** outperforms **PBA** by 24.82%, and **SFOV2** achieves 25% higher bandwidth utilization in comparison to **AVR** method across all three tiling patterns. In a similar way, **CFOV1** and **CFOV2** solutions attain up to 28.11% more bandwidth utilization for 4x3 *Football* video, and **SFOV1** and **SFOV2** solutions achieves nearly 20.03% higher bandwidth utilization for 6x4 *Spotlight* video.

Fig. 6.10b and 6.10c show the average bandwidth utilization when the segment duration is set to 2s and 3s, respectively. The average viewport bandwidth utilization results in Fig. 6.10b are 4,13% higher than the results shown in Fig. 6.10c. For *Boxing* video in Fig. 6.10b, **CFOV2** solution exceeds in performance compared to **CFOV1** by 0.23%, **SFOV1** by 0.35%, **SFOV2** by 0.25%, **UVP** by 1.55%, **CTF** by 1.30%, **PBA** by 9.21%, **AVR** by 7.13%, and **VIS** by 28.84%. **CFOV1** and **CFOV2** solutions achieves upto 14.28% (**UVP**), 11.65% (**CTF**), 18.01% (**PBA**), 21.62% (**AVR**), and 34.02% (**VIS**) for *Conan* video with a 2s segment duration (Fig. 6.10b). Similarly, both solutions obtain 14.40%, 13.19%, 15.35%, 15.68%, and 22.73% higher bandwidth utilization compared to other solutions in the order listed in Fig. 6.10c for 8x6 *Football* video. For the action film *Spotlight* video, the bandwidth utilization for the viewport region is 48.13% for **CFOV1**, 47.8% for **CFOV2**, 46.89% for **SFOV1**, 45.24% for **SFOV2**, 38.38% for **UVP**, 40.38% for **CTF**, 36.62% for **PBA**, 33.59% for **AVR**, and 22% for **VIS** solution for 2s segment duration (Fig. 6.10b). For the same video with a 3s segment duration (Fig. 6.10c)

Table 6.5: Streaming Behaviors of **CFOV** and **SFOV** clients in terms of Selecting Different Tiles Selection Scenarios for Four 360° videos Prepared in 4x3, 6x4, and 8x6 Tiling Patterns and 1s, 2s, and 3s Segment Lengths

Video	Segment Duration	Tiling Pattern	CFOV1			CFOV2			SFOV1			SFOV2		
			FFOV	MR	EFOV	FFOV	MR	EFOV	FFOV	MR	EFOV	FFOV	MR	EFOV
Boxing	1s	4x3	85.74	0.86	13.40	96.52	1.93	1.55	82.35	13.99	3.66	95.51	2.43	2.06
		6x4	82.04	0.52	17.44	96.80	1.66	1.54	82.29	15.88	1.83	95.57	2.39	2.05
		8x6	77.13	0.21	22.66	96.90	1.57	1.53	67.65	31.03	1.32	95.58	2.36	2.06
	2s	4x3	79.26	2.06	18.68	91.50	4.46	4.05	78.20	16.47	5.33	92.56	4.36	3.08
		6x4	80.26	0.92	18.82	91.84	4.14	4.02	79.24	17.95	2.81	92.73	4.36	2.91
		8x6	68.48	0.65	30.86	92.56	3.97	3.46	60.34	37.40	2.25	92.76	4.26	2.98
	3s	4x3	72.20	3.48	24.32	87.50	5.85	6.65	75.90	17.89	6.21	90.30	4.89	4.81
		6x4	76.04	1.36	22.59	87.68	5.78	6.54	76.94	20.08	2.98	90.16	4.99	4.85
		8x6	59.09	0.72	40.19	88.22	5.50	6.29	56.36	41.27	2.37	90.19	4.99	4.81
Conan	1s	4x3	76.50	2.19	21.31	90.37	5.31	4.33	70.26	20.22	9.52	82.93	8.05	9.02
		6x4	71.94	0.77	27.29	91.59	4.45	3.96	63.65	29.99	6.36	83.13	7.98	8.90
		8x6	67.00	0.42	32.59	91.74	4.41	3.85	53.21	41.08	5.71	83.31	7.90	8.79
	2s	4x3	63.24	8.65	28.11	76.45	7.92	15.64	63.09	19.98	16.94	72.92	8.95	18.14
		6x4	59.58	3.50	36.91	77.75	7.28	14.98	56.99	30.17	12.84	72.92	8.85	18.24
		8x6	55.39	2.43	42.18	78.73	7.11	14.17	47.57	40.54	11.89	73.06	8.77	18.16
	3s	4x3	55.70	14.07	30.23	68.02	8.99	22.99	59.39	20.39	20.21	68.53	8.66	22.81
		6x4	50.37	7.42	42.21	68.71	8.77	22.51	53.87	30.37	15.75	68.71	8.74	22.55
		8x6	45.47	5.15	49.38	69.08	9.03	21.89	44.88	40.35	14.77	68.82	8.41	22.77
Football	1s	4x3	75.40	2.31	22.29	89.66	5.44	4.90	65.72	23.16	11.12	79.94	9.45	10.61
		6x4	69.85	0.77	29.38	90.27	5.11	4.62	58.21	34.27	7.53	79.95	9.49	10.56
		8x6	64.77	0.44	34.79	91.36	4.47	4.17	45.88	47.37	6.74	80.06	9.43	10.52
	2s	4x3	59.98	10.09	29.93	73.50	9.78	16.72	59.48	22.33	18.19	68.78	11.33	19.89
		6x4	55.97	4.40	39.63	74.39	9.60	16.01	53.73	31.78	14.48	68.70	11.26	20.05
		8x6	52.29	2.64	45.07	76.65	8.71	14.63	41.23	45.45	13.31	68.93	11.13	19.94
	3s	4x3	51.44	17.80	30.76	64.81	10.19	25.00	57.27	20.64	22.08	63.30	12.23	24.47
		6x4	47.92	9.39	42.69	66.40	9.92	23.67	51.10	30.38	18.52	63.52	12.05	24.43
		8x6	41.67	5.68	52.65	66.70	10.04	23.26	39.70	43.41	16.89	63.37	12.50	24.13
Spotlight	1s	4x3	77.45	2.21	20.34	90.44	4.93	4.64	70.04	21.64	8.33	81.79	8.62	9.58
		6x4	69.49	1.24	29.27	91.27	4.39	4.34	58.48	35.05	6.47	91.27	4.39	4.34
		8x6	59.84	0.88	39.28	91.16	4.47	4.37	47.41	46.55	6.04	81.99	8.58	9.43
	2s	4x3	64.80	7.78	27.42	77.95	8.42	13.63	61.79	23.34	14.87	71.30	10.81	17.89
		6x4	55.65	4.37	39.98	80.37	7.45	12.17	49.19	38.04	12.77	71.16	11.01	17.83
		8x6	48.33	3.03	48.64	78.91	7.99	13.10	39.26	48.29	12.46	71.43	10.84	17.73
	3s	4x3	56.34	13.56	30.10	68.45	9.99	21.56	57.29	23.45	19.26	62.84	13.50	23.66
		6x4	45.64	8.38	45.98	68.90	9.95	21.15	45.60	36.80	17.60	62.63	13.71	23.66
		8x6	39.20	6.51	54.29	69.90	9.59	20.51	35.54	47.60	16.86	62.97	13.27	23.77

the bandwidth utilization for non-viewport region is 53.21% for **CFOV1**, 54.85% for **CFOV2**, 54.78% for **SFOV1**, 57.37% for **SFOV2**, 62.67% for **UVP**, 60.45% for **CTF**, 64.70% for **PBA**, 66.81% for **AVR**, and 77.95% for **VIS** solutions. On a concluding note, **CFOV** and **SFOV** solutions outperform other methods resulting in noticeably lower bandwidth consumption for non-viewport tiles. This is because the proposed frameworks streams non-viewport tiles in the lowest quality or not at all to improve the bandwidth utilization for viewport streaming. While, all other methods unnecessarily increase the bitrate for the non-viewing region and consume upto 65% of the available bandwidth budget for the non-visible region. Thus, **CFOV** and **SFOV** solutions empowers differentiated streaming of quality-variable tiles based on the user's interest by consuming more than 65% of the bandwidth for the viewport region averaged across entire data.

6.4.2.4 Streaming Behavior of CFOV and SFOV Clients

Table 6.5 represents the streaming behavior of **CFOV** and **SFOV** clients. The percentage results of tiles selection cases for each video in these solutions are presented. **CFOV** and **SFOV** solutions share similar tiles selection cases (e.g., *Fixed FOV*, *Marginal Region*, and *Extended FOV*); however, the decision variables are different in these approaches. It is interesting to note from Table 6.5 that with an increase in segment duration the percentage of *Fixed FoV* scenarios drops with an increase in *Marginal Region* or *Extended FoV* scenarios. For instance, increasing the segment duration from 1s to 3s for *Conan* video results in 21.3%, 22.63%, 9.66%, and 14.43% decrease in *Fixed FoV* scenarios for **CFOV1**, **CFOV2**, **SFOV1**, and **SFOV2** solutions, respectively. Similarly, for *Football* video, the decrease in *Fixed FoV* scenario is 23% (for **CFOV1**), 24.66% (for **CFOV2**), 7.25% (for **SFOV1**), and 16.59% (for **SFOV2**). As a reaction of this, the *Extended FoV* scenario is increased by 13.54%, 18.42%, 9.71%, and 13.81% for **CFOV** and **SFOV** solutions when switching from 1s to 3s *Conan* video. Similarly, a 13.22%, 19.41%, 10.7%, and 13.78% increase in *Extended FoV* is observed for all proposed solutions when increasing the segment duration from 1s to 2s for *Football* video.

The highest *Fixed FOV* case is observed in **CFOV2** (82.03%) followed by **SFOV2** (77.88%), **CFOV1** (62.82%), and **SFOV1** (58.59%) solutions. This shows that the adjusted distance values remains mostly in the first threshold in **CFOV2** solution. **SFOV1** observes the highest average (i.e., 30.68%) *Marginal Region* scenario because the original viewpoint tile appears in the predicted viewport tiles set and leads to the higher percentage of marginal tiles. At the same time, **CFOV1** and **CFOV2** observes 4.36% and 6.63% of *Marginal Region* scenario, respectively. **SFOV1** observes the highest *Marginal Region* percentage (48.29%) for 8x6 *Spotlight* video with a 2s segment duration. The highest *Extended FoV* case is observed in the **CFOV1** solution, which requests two combined FOVs almost during 32.82% of the streaming sessions for entire videos. The least *Extended FOV* case appears in **SFOV1**, which performs FOV extension almost during 10.73% of the streaming sessions. **CFOV1** extends the FoV during 54.29% playback of 8x6 *Spotlight* video with a 3s segment

duration. The previous sections described how different tiles selection scenarios in proposed solutions result in improved streaming performance measured in terms of tiles overlap, QoE, and bandwidth utilization in comparison to the baseline algorithms.

6.4.3 Discussions

The key observations and findings following the simulation-based experiments performed can be summarized as follows:

- The **VIS** method performs worse than all the tiles-based solutions. The average QoE values achieved are the lowest and the highest bandwidth wastage is observed for this solution.
- The prediction angles-based tiles selection solution **CFOV1** achieves slightly higher streaming performance compared to **CFOV2**. Similarly **SFOV1** solution outperforms **SFOV2** in terms of achieving higher QoE and bandwidth utilization. The reason is that under defined thresholds, **CFOV2** and **SFOV2** solutions are limited to the *Fixed FoV* tiles selection scenario since we choose a wider area as a safety threshold to avoid the unnecessary FoV extension.
- The uniform bitrate adaptation combined with prediction angles-based (**CFOV1** and **CFOV2**) and prediction angles and prediction differences-based (**SFOV1** and **SFOV2**) tiles selection scenarios can facilitate achieving improved perceived QoE compared to the methods focusing on increasing the quality uniformly (**UVP** and **AVR**) and methods improving the quality from the center tile (**CTF** and **PBA**). The quality improvements were observed for all tiling patterns and across all segment durations.
- Employing marginal tiles in all directions as in **AVR** can lead to inefficient bandwidth utilization and thus to lower perceived quality levels. Instead, dynamically defining the marginal region based on prediction results significantly improves streaming performance for content with different motion characteristics.
- The weighted quality increment for the marginal region scenario provides a cushion and absorbs the quick quality transitions from viewport (highest) to

background (lowest) image region and thereby lower spatial oscillations under both the high or medium-to-high prediction performance due to the equal-quality differentiated streaming regions.

- With the increase in the QoE weight coefficients and segment duration, the average QoE values decreases since the user head movements become more difficult to predict over longer prediction horizons due to which spatial and temporal quality oscillations and background quality penalty drastically increase even for the stable motion content (i.e., Boxing).
- Streaming the viewport with the same size or virtual extension of the viewport size along with aggressive or weighted bitrate selections for **CFOV** and **SFOV** solutions improve the viewport quality levels under longer prediction horizons.

In summary, the proposed interactive tiles selection solutions, along with the priority-oriented bitrate adaptation method, achieve higher viewport overlap, QoE and bandwidth utilization in relation to the 360° video streaming: stream high-quality visible tiles along with low-quality background tiles while maintaining spatial and temporal viewport smoothness.

6.5 Conclusions

This chapter presented the system architecture, proposed algorithms, and experimental analysis of the proposed solutions. We divided the tiles selection solutions into two categories: tiles selection based on prediction angles and tiles selection based on angles and prediction errors. **CFOV1** and **CFOV2** solutions belonging to the first category, and **SFOV1** and **SFOV2** solutions from the second category were proposed. **CFOV1** mainly checks the overlap between two predicted FoVs, while **CFOV2** uses different thresholds to calculate the tiles in three streaming regions. **CFOV1** and **CFOV2** solutions reduce the complexity of tile selection by adopting two FoV prediction mechanisms to better accommodate the user's viewing region in response to the different head movements. In contrast to **CFOV1** and **CFOV2**, **SFOV1** and **SFOV2** solutions employ a single FoV prediction mechanism and consider closeness between the original and predicted viewpoints during

previous segment to calculate the tiles set for the next segment. **CFOV2** and **SFOV2** involve content complexity information to dynamically output the streaming tiles for each adaptation interval. The tiles selection results are employed by a priority-based bitrate adaptation algorithm, which further refines the tiles selection results based on the available bandwidth budget in order to achieve improved user QoE. In the context of quality-efficient 360° remote video services, the proposed bitrate adaptation algorithm results in one of the following three streaming regions for selected groups of tiles: viewport-only tiles, viewport and background tiles, or viewport, marginal, and background tiles. An extensive experimental assessment was performed using four video streams prepared in three tiling patterns and three segment durations under three dynamic bandwidth scenarios. The experimental results show that **CFOV** and **SFOV** solutions result in 5.61%, 9.78%, and 12.54% higher average viewport overlap for 1s, 2s, and 3s prediction horizons. Following higher prediction trends, the proposed solutions result in 19.67%, 29.65%, and 43.70% higher average QoE for QoE weight coefficients C1, C2, and C3, respectively. In addition, compared to alternative viewport-dependent (i.e. **UVP**, **CTF**, **PBA**, and **AVR**) and viewport-independent (**VIS**) solutions, **CFOV** and **SFOV** solutions output 21.41%, 15.40%, and 13.68% higher viewport bandwidth utilization for 1s, 2s, and 3s segment durations, respectively.

Chapter 7

Dynamic Tiling-based Adaptive Solutions for 360° Video Streaming

This chapter presents the system architecture and the performance evaluation of one of the major contributions of this thesis, the Dynamic video Frames Tiling (DFT) solution for 360° video. ***DFT*** *solutions aim to select suitable tiling patterns for each adaptation interval during 360° video streaming. Compared to fixed tiling-based solutions discussed in previous chapter,* ***DFT*** *introduces two dynamically tiling pattern selection solutions in addition to streaming regions selection and tiles bitrate adaptation algorithms.* ***DFT*** *was compared against alternative solutions in simulation-based testing. Testing results show that* ***DFT*** *efficiently and fairly utilizes the available network resources and achieves higher QoE levels in comparison with the alternative solutions.*

7.1 Introduction

360° omnidirectional videos have improved the traditional content streaming by offering an entire wide-angle spherical spatial FoV. Users can have an incredibly immersive experience, especially when watching 360° videos on high-resolution Head-Mounted Display (HMD) devices [281]. These videos enable an interactive look-around viewing experience for its audience. The user's head motion patterns recorded with a VR headset are essential information for predicting future viewport positions [282], which is limited in many cases [283]. Viewport prediction encourages downloading the predicted tiles in advance before the user moves. According to the prediction results, the client can assign more bits to the visible tiles. An accurate viewport prediction is only possible for short video segments. However, short video segments lead to higher request-response overhead [284]. Delivering viewport tiles in case of inaccurate viewport prediction significantly deteriorates the degree of satisfaction. Therefore, the spatial partitioning structure of tiles plays an important role in concealing head motion dynamics. For instance, on the one hand, the larger the rendered viewport area with a constant resolution can compensate the impact of quick head turns. On the other hand, using smaller resolution tiles may reduce redundant bits during good prediction accuracy.

The existing fixed tiling-based algorithms [48, 49, 107, 56, 44, 231, 236] have set a stable background by considering user-specific viewing preferences to boost user QoE. However, the tiles dynamics based on space and time separation of 360° videos make it challenging to develop a successful VR streaming framework. Similarly, accurate prediction and selection of the viewport and marginal regions are important, although challenging and need to be investigated.

This chapter introduces *Dynamic video Frames Tiling-based (DFT)* adaptive solutions for 360° video, which incorporate an additional adaptive layer compared to fixed tiling-based solutions which dynamically perform streaming regions selection and bitrate adaptation algorithms. Compared to the existing dynamic tiling-based solutions [58, 57, 246], this chapter introduces two novel solutions to dynamically decide appropriate tiling patterns based on the visual area-based weighted quality

Table 7.1: Tile-based Streaming Solutions for Adaptive 360° Video

Works	Tiling Mechanism		Viewport Prediction		Tiles Selection		
	Fixed	Dynamic	Single	Combined	Fixed VP	Marginal	Extended VP
[48]	✓	×	✓	×	✓	×	✓
[49]	✓	×	✓	×	✓	×	×
[107]	✓	×	✓	×	✓	✓	×
[56]	✓	×	✓	×	✓	✓	×
[44]	✓	×	✓	×	✓	×	✓
[231]	✓	×	✓	×	✓	✓	×
[236]	✓	×	✓	×	✓	✓	✓
[58]	×	✓	✓	×	✓	×	×
[57]	×	✓	✓	×	✓	×	✓
DFT1	×	✓	×	✓	✓	✓	✓
DFT2	×	✓	×	✓	✓	✓	✓

metric and viewport prediction differences and explores their advantages over using a fixed tiling pattern. The first solution named DFT1 considers a visual-area-based weighted quality metric to decide the optimal tiling pattern for each segment. The second solution, referred to as DFT2, adapts the tiling versions based on viewport prediction differences to improve the viewport availability for different motion content. The proposed solutions employ two viewpoint/viewport prediction mechanisms to decide the best-fit viewport tiles according to the relationship between the prediction mechanisms. DFT solutions consider the following three cases for innovative tiles selection according to the user watching behavior: (i) *Case 1: Fixed viewport with no marginal tiles*; (ii) *Case 2: Fixed viewport with marginal tiles*; (iii) *Case 3: Extended viewport with no marginal tiles.* Last but not least important, an effective bitrate adaptation algorithm is designed to improve bitrate budget allocation between different tiles sets by learning tile distribution and real-time transmission capacity. Table 7.1 illustrates the significant differences between the proposed solutions and existing schemes in terms of tiling mechanism, viewport prediction, and tiles selection.

The main contributions of this chapter are as follows:

1. introduces innovative tiling pattern selection solutions that perform interactive tiling pattern selection based on the newly proposed visual area-based weighted quality metric and head movement prediction errors observed during the previous segment playback. The proposed solutions dynamically select one of the three tiling patterns, i.e., small, medium, or large resolution tiles,

for each segment to overcome the viewing uncertainty. We present the performance of our proposed method using an open-source dataset of real head movement traces recorded from 48 VR users.

2. proposes a new tile selection method for 360° video to lower the impact of drastic head rotations based on the combination of two viewport prediction mechanisms, where the user viewport is defined in terms of 110° angles in both horizontal and vertical directions.
3. designs a DFT adaptation algorithm for actively allocating the video bitrate budget to different frame areas to improve VR user perception levels in a dynamic tiling-based streaming context. The DFT bitrate adaptation heuristic renders viewport, marginal, and background tiles in uniform quality levels.
4. presents an extensive experimental evaluation of the proposed solution in comparison with dynamic tiling-based (ATS) and fixed tiling-based (i.e. UVP, CTF, PBA, and AVR) solutions considering different motion 4K videos. The videos were spatially split into 4x3, 6x4, 8x6 tiling patterns and temporally split into 1s, 1.5s, 2s segment durations. DFT1 and DFT2 solutions improve the streaming performance under dynamic bandwidth conditions in comparison to the state-of-the-art solutions, as reported in section 7.4.

7.2 Proposed Dynamic Tiling-based Architecture

7.2.1 Dynamic Tiling-based System Architecture

The system architecture for the dynamic tiling-based solutions, DFT1 and DFT2 is shown in Fig. 7.1. Different from fixed tiling-based architecture, the DFT server stores the multiple pre-processed tiling versions of 360° video. The equirectangular projected video is temporally split into numerous equal duration segments, i.e., $\mathcal{S} = \{\mathcal{S}(1), \mathcal{S}(2), ..., \mathcal{S}(i), ..., \mathcal{S}(I)\}$. For each segment, the equirectangular frames are spatially divided into p tiling patterns, $\mathcal{T}_p$, $\forall\ p \in \{x, y, z\}$, containing small, medium, and large number of tiles, respectively, such that $|\mathcal{T}_x| < |\mathcal{T}_y| < |\mathcal{T}_z|$. Each tiling pattern contain a number of tiles, i.e., $\mathcal{T}_p = \{\mathcal{T}_p^1(i), \mathcal{T}_p^2(i), ..., \mathcal{T}_p^k(i), ..., \mathcal{T}_p^K(i)\}$. For each segment, the tiled videos are then encoded into J bitrate representations,

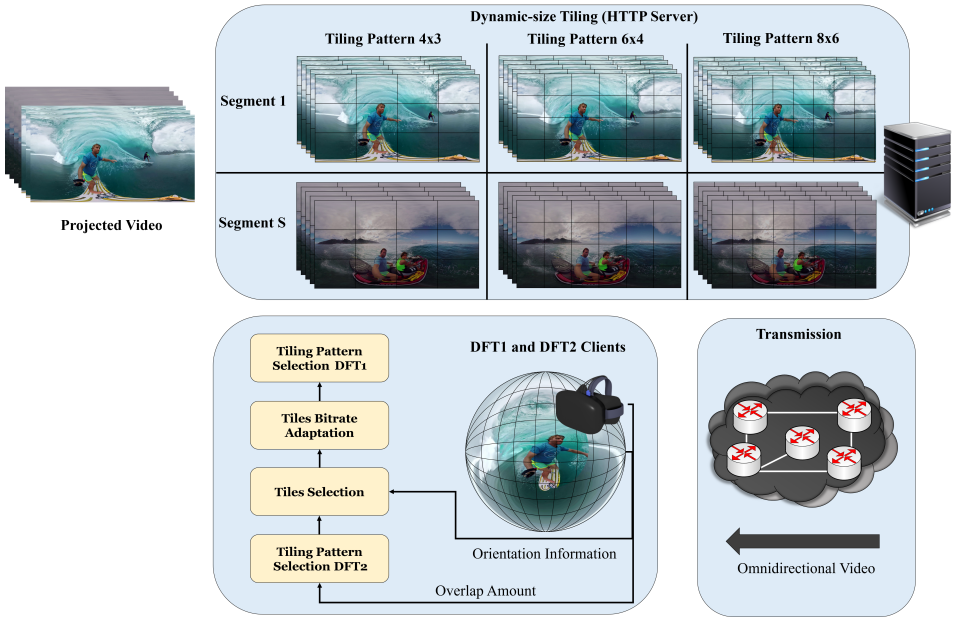


Figure 7.1: The proposed 360° client-server streaming architecture.

Table 7.2: Notations Used in This Chapter

Symbol	Meaning
$\mathcal{S}$, $\mathcal{L}$, $\mathcal{T}_p$	Sets of segment, bitrates, and tiles
I, J, K	Number of segments, bitrates, and tiles
i, j, k	Index of segment, bitrate, and tile
$\mathcal{T}_p(i)$	Tiling pattern selected for (i)th segment
$\mathcal{T}_x(i)$, $\mathcal{T}_y(i)$, $\mathcal{T}_z(i)$	Different resolutions tiles sets
$\mathcal{T}_p^{vn}(i)$, $\mathcal{T}_p^{vs}(i)$	Viewport tiles sets of two prediction mechanisms
$\mathcal{T}_p^{\hat{v}}(i)$	Actual viewport tiles set for pth tiling pattern
$\mathcal{T}_p^{v}(i), \mathcal{T}_p^{m}(i), \mathcal{T}_p^{b}(i)$	Tiles sets for pth tiling pattern
$\widehat{B}(i)$	Available bandwidth for the (i)th segment
$B^v(i), B^m(i), B^b(i)$	Measured bandwidth for different tiles sets
$\mathcal{L}_{p,j}^k(i)$	Video bitrate selected for tile of pth tiling pattern
α, β, γ, δ	Weight coefficients

i.e., $\mathcal{L} = \{\mathcal{L}_{p,1}^k(i), \mathcal{L}_{p,2}^k(i), ..., \mathcal{L}_{p,j}^k(i), ..., \mathcal{L}_{p,J}^k(i)\}$. Let $\mathcal{L}_{p,j}^k(i)$ represents the jth bitrate for kth tile of pth tiling pattern in (i)th segment.

The DFT clients responsible for controlling the adaptive streaming operations need information in advance about the available tiling patterns at the server-side. DFT2 performs tiling pattern selection before computing the streaming tiles and bitrate allocations for each adaptation interval. The *tiling pattern selection* module in DFT2 checks the overlap between actual and predicted viewport areas during the previous segment. It then dynamically decides the particular tiling pattern for the next seg-

ment. For each selected tiling pattern, the *tiles selection* module selects tiles sets for different priority regions, i.e., viewport ($\mathcal{T}_p^v(i)$), marginal ($\mathcal{T}_p^m(i)$), and background ($\mathcal{T}_p^b(i)$), based on the predicted viewport coordinates for each segment. Accordingly, the *bitrate adaptation* unit selects suitable bitrates for each tile according to the associated region and the available network capacity. On the other hand, DFT1 first computes the streaming tiles using orientation information for each available tiling pattern. Then, it performs the bitrate adaptation for the computed tiles using the measured bandwidth. Finally, the tiling pattern resulting in a higher weighted-area-based visual quality score is selected for streaming during each adaptation interval. Once the segments are received, the clients decode and reconstruct the requested bitstreams similar to the fixed tiling information for each selected tiling pattern without any additional decoding overhead. Then, they perform the rendering and start playing the requested content. Table 7.2 includes mathematical symbols and their meanings used in the following discussion.

7.2.2 Problem Definition

In 360° adaptive video streaming, it is highly imperative to recognize the consumer's unique quality aspects. The high-quality expectations of the user mainly depend on the quality of the visible area. The lower rate visible tiles may not satisfy the user even if the background tiles are played in good quality. The QoE metric defined in this section includes viewport quality and spatial and temporal smoothness factors similar to the Chapter 6 along with playback buffer risk.

- **Viewport Quality:** By averaging the quality of the actual viewport tiles in segment (i), for pth tiling pattern, the viewport quality is given as follow:

$$f_1(i) = \frac{\sum_{k \in \mathcal{T}_p^{\hat{v}}(i)} \sum_{j \in \mathcal{L}} \mathcal{Q}(\mathcal{L}_{p,j}^k(i))}{|\mathcal{T}_p^{\hat{v}}(i)|} \tag{7.1}$$

  where $\mathcal{T}_p^{\hat{v}}(i)$ represents the actual viewport tiles set in the (i)th segment and $|\mathcal{T}_p^{\hat{v}}(i)|$ indicates the cardinality of the set. $\mathcal{Q}(\mathcal{L}_{p,j}^k(i))$ maps the video bitrate to the particular quality level.

- **Temporal Quality Oscillations:** The efficiency of tile-based streaming

schemes can be impaired by the disparity in quality levels between two viewports of consecutive segments. Therefore, the inter segments quality fluctuations need not be drastic and can be calculated by:

$$f_2(i) = \mid f_1(i) - f_1(i-1)\mid \tag{7.2}$$

- **Spatial Quality Oscillations:** Cybersickness, viewing irritation, and other physiological effects, such as nausea, fatigue, and aversion [131], can be driven by inconsistent quality levels within the viewport, which leads towards the dizzier perception levels. Spatial quality variations are measured as follows:

$$f_3(i) = \frac{\sigma(\mathcal{Q}(\mathcal{L}^k_{p,j}(i)))}{\mu(\mathcal{Q}(\mathcal{L}^k_{p,j}(i)))}, \quad \forall k \in \mathcal{T}^{\hat{v}}_p(i), \forall j \in \mathcal{L} \tag{7.3}$$

  The term in the numerator represents the standard deviation of the viewport quality samples, while the denominator represents the mean of the samples.

- **Playback Buffer Risk:** This metric directly indicates the risky buffer events when the available connection bandwidth is lower than the sum of selected video bitrates and is given as:

$$f_4(i) = \begin{cases} 1, & \textbf{if } (\widehat{B}(i) < \sum_{k \in \mathcal{T}_p(i)} \mathcal{L}^k_{p,j}(i)) \\ 0, & Otherwise \end{cases} \tag{7.4}$$

  where $\widehat{B}(i)$ represents the available bandwidth budget for (i)th segment.

The user-perceived QoE for each 360° segment is defined by a weighted summation formulation [251]:

$$QoE(i) = \alpha * f_1(i) - \beta * f_2(i) - \gamma * f_3(i) - \delta * f_4(i) \tag{7.5}$$

where α, β, γ, and δ are the non-negative constants corresponding to the viewport quality, temporal quality oscillations, spatial quality oscillations, and buffer risk events, respectively. As users do not want to experience spatial and temporal quality fluctuations and rebuffering events, the functions $f_2(i)$, $f_3(i)$, and $f_4(i)$ are set to

negative.

Accurate QoE evaluation plays a critical role in optimizing the performance of traditional as well as immersive video content. How long a user feels the VR immersion in a video scene dictates the user's satisfaction level. The client seeks optimal bitrates for each segment in a dynamic tiling-based 360° video streaming, intending to optimize the user's long-term QoE reward. The mathematical problem formulation is given as follows:

Problem:

$$arg \quad \max_{i \in \mathcal{S}} QoE(i) \tag{7.6}$$

The proposed solution solves this problem by selecting a relevant tiling pattern for each segment based on the viewport prediction errors for the previous segment. Next, it dynamically performs the viewing area selection based on the two viewport prediction mechanisms to predict the most likely watched tiles. Finally, the tiles bitrate adaptation mechanism improves the bitrate budget distribution between different viewing regions. These solutions are explained in the next section.

7.3 Proposed Dynamic Tiling-based Adaptation Algorithms

This section presents the adaptation algorithms for DFT1 and DFT2 streaming clients. First, the tiling pattern selection algorithms in DFT1 and DFT2 solutions are presented. Next, the streaming tiles selection mechanism in DFT solutions is described. Finally, the DFT bitrate adaptation algorithm aiming to achieve higher visual quality levels in dynamic tiling-based streaming context is presented.

7.3.1 DFT Tiling Pattern Selection Algorithms

Tile-based encoding brings several opportunities such as efficient video coding [33, 32], improved quality distribution, parallel [285, 286], and partial decoding [287, 288], etc., for VR video applications. The choice of the appropriate tiling pattern, which reflects the spatial partitioning of frame areas [288] impacts the overall video

compression performance. Employing a smaller number of tiles (i.e., larger resolution tiles) can improve the compression performance in some cases. Yet, at the same time, it may include additional pixels that contribute to the waste of bandwidth resources [58]. Contrary, using a large number of tiles (i.e., smaller resolution tiles) reduces the number of redundant pixels; however, it may introduce several visual distortions, e.g., flickering, floating, and blurring at the edges of the tiles [59]. Therefore, the proposed solution employs dynamic tiling patterns that aim to lower the amount of redundant data and provide a finer boundary to the user FoV according to the watching behaviors of the users.

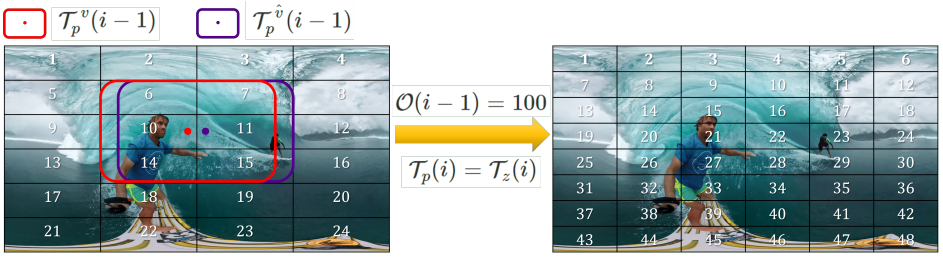
DFT1: The proposed DFT1 solution decides an optimal tiling pattern during each adaptation interval based on the visual quality score. Considering the fact that the user gaze-point is mostly located around the center of the viewport [289], the quality of the viewpoint tile, should have a higher priority compared to other tiles. Therefore, we design a new visual area-based weighted quality metric to select the suitable tiles for different streaming regions during 360° video streaming. The viewport tiles are sorted based on the distance from the viewpoint. The priority weights are assigned to tiles based on their distance from the viewport center, such as the highest and lowest weights are allocated to the viewpoint and the last tile, respectively, in the viewport tiles set. The weighted quality metric is given in Eq. (7.7):

$$\mathcal{WQ}_p^v(i) = \frac{\sum_{k=1}^{|\mathcal{T}_p^v(i)|} \sum_{j=1}^{J} (2)^{|\mathcal{T}_p^v(i)|-k} * \mathcal{Q}(\mathcal{L}_{p,j}^k(i))}{(2)^{|\mathcal{T}_p^v(i)|} - 1} \tag{7.7}$$

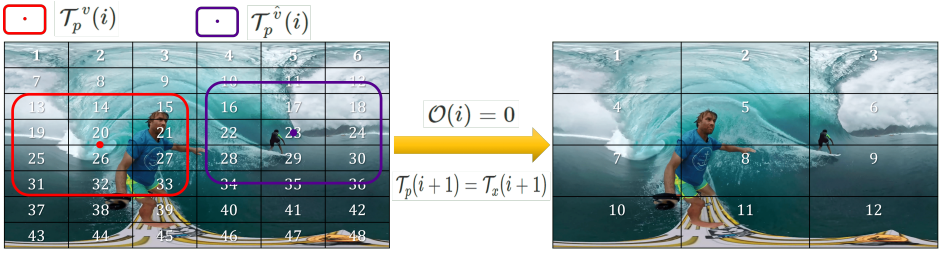
where $|\mathcal{T}_p^v(i)|$ represents the number of tiles in predicted viewport tiles set and $\mathcal{Q}(\mathcal{L}_{p,j}^k(i))$ maps the video bitrate to the particular quality level. As for each tiling pattern, the visual area can be different based on different tiles selection cases elaborated in section 7.3.2. Therefore, we define visual area-based weighted video quality metric which tries to balance the visual area and the weighted quality and is given in Eq. (7.8):

$$\mathcal{VQ}_p^v(i) = \frac{|\mathcal{T}_p^v(i)|}{|\mathcal{T}_p(i)|} * \mathcal{WQ}_p^v(i) \tag{7.8}$$

where $|\mathcal{T}_p(i)|$ represents the number of tiles in the tiling pattern p. The tiling pattern



(a) Tiling pattern selection in DFT2 for the (i)th segments.



(b) Tiling pattern selection in DFT2 for the $(i+1)$th segments.

Figure 7.2: Tiling pattern selection in an equirectangular space for two consecutive segments.

selection procedure for DFT1 is given as follows:

1. For each tiling pattern:
   - Calculate the viewport, marginal, and background tiles using Algorithm 7.2.
   - Assign appropriate bitrates to the viewport, marginal, and background tiles using Algorithm 7.3.
   - Calculate the visual quality scores using Eq. 7.7 and Eq. 7.8.
2. Select the tiling pattern that achieves the highest visual quality score.

DFT1 is scalable and can work with any number of tiling patterns. The proposed solution is implementable in both simulation and real-time environments.

DFT2: The DFT2 solution decides an optimal tiling pattern based on the viewport prediction performance. Unlike DFT1 which is based on visual quality metrics, DFT2 measures the overlap between actual and predicted viewport tiles to select the appropriate tiling pattern. Let $\mathcal{O}(i-1)$ denotes the overlap percentage of the actual and predicted viewport tiles for the $(i-1)$th segment, and is given as [290]:

$$\mathcal{O}(i-1) = \frac{|\mathcal{T}_p^{\hat{v}}(i-1) \cap \mathcal{T}_p^{v}(i-1)|}{|\mathcal{T}_p^{\hat{v}}(i-1)|} * 100 \tag{7.9}$$

Algorithm 7.1 Tiling Pattern Selection Algorithm in DFT2

Input : $\mathcal{O}(i-1) \leftarrow$ *Tiles overlap percentage during the* $(i-1)$*th segment.*
Result : $\mathcal{T}_p(i)$ *Tiling pattern selected for the* (i)*th segment.*

```
1 if (i = 1) then
2     T_p(i) = T_z(i)
3 else if (O(i − 1) == 0) then
4     T_p(i) = T_x(i) ;                                  // No overlap
5 else if (O(i − 1) == 100) then
6     T_p(i) = T_z(i) ;                                  // Perfect overlap
7 else
8     T_p(i) = T_y(i) ;                                  // Partial overlap
```

Algorithm 7.1 details the tiling pattern selection procedure in DFT2. It checks the closeness of actual and predicted viewport tiles to decide on a suitable tiling pattern for the next segment. As no information is available at the start, the tiling pattern with a larger number of tiles ($\mathcal{T}_z(i)$) is selected for the first segment (lines 1-2). For the $(i-1)$th segment, if there is no overlap between actual and predicted viewing tiles, then the tiling pattern with a smaller number of tiles ($\mathcal{T}_x(i)$) is selected for the (i)th segment to deal with fast head rotations (lines 3-4). If actual and predicted viewports perfectly overlap during the previous segment, the largest tiling pattern is selected to lessen the abundance of redundant pixels outside the viewport region (lines 5-6). If the actual and predicted viewports partially overlap during the adaptation interval, medium resolution tiles represented as $\mathcal{T}_y(i)$ are streamed for the next segment (lines 7-8).

Suppose the server includes three tiling patterns with $K = 12$, $K = 24$, and $K = 48$, number of tiles. Fig. 7.2a and Fig. 7.2b illustrate the main idea of tiling pattern selection in DFT2 for the (i)th and $(i+1)$th segments, respectively. The red and purple rectangles represent the actual and predicted viewports. With good prediction accuracy, the actual and predicted group of tiles are similar for the $(i-1)$th tiling pattern; therefore, the larger number of tiles are picked for the (i)th segment (Fig. 7.2a). Due to the abrupt user's head movements, if the actual and predicted viewports do not overlap during the (i)th segment, the smaller number of tiles are selected for the $(i+1)$th segment to include the redundant pixels (Fig. 7.2b).

Algorithm 7.2 Tiles Selection Algorithm in DFT

Input : $\mathcal{T}_p(i) \leftarrow$ *Selected tiles set with tiling pattern p for the* (i)*th segment*; $\mathcal{T}_p^{vn}(i) \leftarrow$ *Tiles set of the primary predicted viewport.*; $\mathcal{T}_p^{vs}(i) \leftarrow$ *Tiles set of the secondary predicted viewport.*

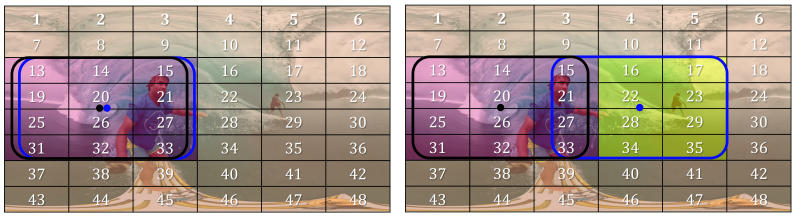
Result : $\mathcal{T}_p^{v}(i), \mathcal{T}_p^{m}(i), \mathcal{T}_p^{b}(i) \leftarrow$ *Estimated viewport, marginal, and background tiles sets for the* (i)*th segment.*

if $(\mathcal{T}_p^{vn}(i) \cap \mathcal{T}_p^{vs}(i) \neq \emptyset)$ **then**
  $\mathcal{T}_p^{v}(i) = \mathcal{T}_p^{vn}(i)$
  if $(\mathcal{T}_p^{vs}(i) - \mathcal{T}_p^{vn}(i)) == \emptyset)$ **then**
    $\mathcal{T}_p^{m}(i) = \emptyset$; `// Case 1`
  else
    $\mathcal{T}_p^{m}(i) = \mathcal{T}_p^{vs}(i) - \mathcal{T}_p^{vn}(i))$; `// Case 2`
  $\mathcal{T}_p^{b}(i) = \mathcal{T}_p(i) - (\mathcal{T}_p^{v}(i) \cup \mathcal{T}_p^{m}(i))$
else
  $\mathcal{T}_p^{v}(i) = \mathcal{T}_p^{vn}(i) \cup \mathcal{T}_p^{vs}(i)$; `// Case 3`
  $\mathcal{T}_p^{m}(i) = \emptyset$
  $\mathcal{T}_p^{b}(i) = \mathcal{T}_p(i) - \mathcal{T}_p^{v}(i)$

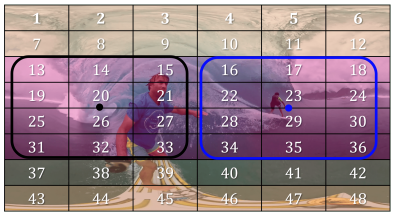
7.3.2 DFT Streaming Tiles Selection Algorithm

DFT solutions consider two viewpoint/viewport prediction mechanisms to perform the tile selection during each adaptation interval similar to CFOV1 and CFOV2 solutions presented in Chapter 6. The current viewing point will be used as the predicted viewing position for the primary viewpoint prediction. The spherical walk approach proposed in [48] is adopted for the secondary viewpoint prediction that considers the user's motion as a walk on a sphere and predicts the future position based on the spherical movement from one point to another point. Based on the predicted viewpoints and headset FoV (usually in the range of 90°-110°), the tiles for both viewports are selected by calculating the spherical distance between the predicted viewpoints and the center of each of the tiles. The tiles whose centers are less than half of the FoV size apart from the viewpoint will belong to the viewport region. In this way, both the primary (last known) and secondary (spherical walk) viewport sets represented by $\mathcal{T}_p^{vn}(i)$, $\mathcal{T}_p^{vs}(i)$, respectively, are computed for each segment.

Algorithm 7.2 aims to find appropriate tiles for the viewport, marginal, and background regions, respectively. The tiles identification and selection are dynamically performed for each adaptation interval. Algorithm 7.2 starts by determining the



(a) Case 1 (b) Case 2

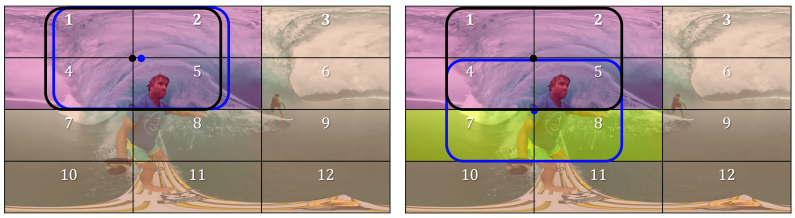


(c) Case 3

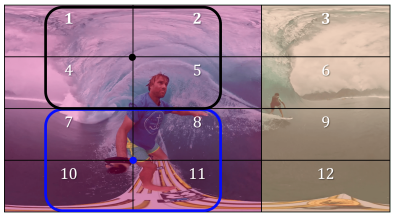
Figure 7.3: Tiles selection cases in DFT for $\mathcal{T}_z(i)$ tiling pattern of (i)th segment.

intersection of two predicted tiles sets and then selects the potential viewport and other tiles. The viewport tiles set ($\mathcal{T}_p^v(i)$) would include all the tiles of the primary viewport if the intersection of two predicted sets is non-empty for the (i)th segment (lines 1-2). Suppose all the tiles in both primary and secondary viewport sets are identical. In that case, it reflects *Case 1: Fixed viewport with no marginal tiles*, where only the viewport and background tiles sets are selected for streaming for the next segment (lines 3-4). Suppose the primary and secondary viewports contain some non-identical tiles, then for *Case 2: Fixed viewport with marginal tiles*, the marginal tiles set ($\mathcal{T}_p^m(i)$) contains the non-overlapping tiles of the secondary viewport (line 6). All the remaining tiles are classified as background tiles ($\mathcal{T}_p^b(i)$) for *Case 1* or *Case 2* (line 7). When both predicted viewports contain unique tiles, then for *Case 3: Extended viewport with no marginal tiles*, all the tiles belonging to the primary and secondary viewports are labeled as the viewport tiles for the next segment (line 9). For *Case 3*, the marginal region is empty and does not contain any tile (line 10). Finally, the range of background tiles is computed for the (i)th segment (line 11).

Fig. 7.3 and Fig. 7.4 illustrate the tiles selection cases based on the output of Algorithm 7.1 for two consecutive segments in an equirectangular space of 360° video. The black rectangle represents the primary predicted viewport, while the blue rectangle represents the secondary predicted viewport. The potential viewport tiles



(a) Case 1 (b) Case 2



(c) Case 3

Figure 7.4: Tiles selection cases in DFT for $\mathcal{T}_x(i+1)$ tiling pattern of $(i+1)$th segment.

are represented by a purple window, whereas the marginal tiles are marked in light green. Fig. 7.3 reflects the three cases based on the closeness of the two predicted viewports for the (i)th segment. The selected tiling pattern for (i)th segment is $\mathcal{T}_z(i)$. If the primary and secondary predicted viewport tiles are identical, then the tiles belonging to only two regions, i.e., viewport and background, are selected for streaming (Fig. 7.3a). The marginal tiles are included in Fig. 7.3b following the uncertainty of the two prediction mechanisms. For the extended viewport case in Fig. 7.3c, the goal is to extend the viewport area to deal with possible head movement prediction errors. Similarly, Fig. 7.4 represents the tiles selection cases for the selected tiling pattern $(\mathcal{T}_x(i+1))$ of $(i+1)$th segment. If both viewport sets contain identical tiles, we can positively stream the same viewport tiles with no marginal region (Fig. 7.4a). The rest of the tiles are added to the background tiles set. Under quick head turns for different motion scenes, the primary and secondary mechanisms' predicted viewpoints could be far away from each other. In this case, we can stream tiles of both viewports as the potential predicted viewport tiles to allow equal-quality streaming (Fig. 7.4c). With no marginal tiles in *Case 3*, the rest of the tiles belong to the background region.

7.3.3 DFT Tiles Bitrate Adaptation Algorithm

This algorithm is specifically designed for dynamic tiling-based 360° video streaming. Both DFT1 and DFT2 clients employ this bitrate adaptation algorithm to decide the suitable bitrates for tiles. In an adaptive multimedia streaming environment, accurate bandwidth utilization is crucial to achieving higher playback performance [145]. The proposed bitrate allocation algorithm considers aggressive, weighted, and conservative quality adjustments for different tiles selection cases to improve the corresponding bitrate choice for each tile that the network can support. For tiles selection *Case 1*, an aggressive quality adjustment is performed for viewport tiles. The algorithm performs a weighted quality adjustment if the marginal region is non-empty (*Case 2* of Algorithm 7.2). A relatively conservative bitrate selection is performed for *Case 3*, where the viewport region is extended to lower the viewport mismatch while sacrificing the quality.

Algorithm 7.3 determines the bitrate selection for the tiles belonging to different priority regions calculated in Section 7.3.2. After completely downloading the previous segment, the playback adaptation is performed for each segment. The algorithm simply requests the whole segment in the lowest quality if the available bandwidth is insufficient (lines 1-2). This can prevent viewport deviation but can lead to playback buffer risk. If the available bandwidth is higher than the largest available quality level, the highest representation for the entire 360° segment is requested to enable the higher visual levels (lines 3-4). In other cases, firstly, the minimum quality level is allocated to all the tiles, and next, the available bandwidth budget is updated (lines 6-7).

Suppose the marginal region is non-empty (*Case 2*). In that case, a priority weight calculation is performed for viewport and marginal tiles, where the weights are determined depending on the number of tiles in these regions (lines 9-10). The viewer is more interested in watching the viewport content at higher quality levels; therefore, these tiles are assigned double weights compared to the marginal tiles. Next, the viewport and marginal tiles' bandwidth are computed based on priority-related weights (lines 11-12). Finally, the video bitrate levels for the viewport and the

Algorithm 7.3 Tiles Bitrate Adaptation Algorithm in DFT

Input : $\mathcal{L}\leftarrow$ *Video bitrates set of 360° segments*; $\mathcal{T}_p^v(i), \mathcal{T}_p^m(i), \mathcal{T}_p^b(i)\leftarrow$ *Tiles sets of viewport, marginal, and background regions*; $|\mathcal{T}_p^v(i)|, |\mathcal{T}_p^m(i)|\leftarrow$ *Number of tiles in viewport and marginal regions*; $\widehat{B}(i)\leftarrow$ *Available bandwidth for the* (i)*th segment.*

Result : $w^{\mathcal{T}_p^v}(i), w^{\mathcal{T}_p^m}(i)\leftarrow$ *Weights of viewport and marginal tiles sets*; $B^v(i), B^m(i), B^b(i)\leftarrow$ *Bandwidth for the viewport, marginal, and background tiles*; $\mathcal{L}^{\mathcal{T}_p^v}(i), \mathcal{L}^{\mathcal{T}_p^m}(i), \mathcal{L}^{\mathcal{T}_p^b}(i)\leftarrow$ *Video bitrates selected for the tiles of* (i)*th segment.*

1 **if** $(\widehat{B}(i) \leq \sum_{k\in\mathcal{T}_p(i)} \mathcal{L}_{p,1}^k(i))$ **then**
2 $\quad \mathcal{L}^{\mathcal{T}_p}(i) = \mathcal{L}_{p,1}^k(i),\ \forall\ k \in \mathcal{T}_p(i)$
3 **else if** $(\widehat{B}(i) \geq \sum_{k\in\mathcal{T}_p(i)} \mathcal{L}_{p,J}^k(i))$ **then**
4 $\quad \mathcal{L}^{\mathcal{T}_p}(i) = \mathcal{L}_{p,J}^k(i),\ \forall\ k \in \mathcal{T}_p(i)$
5 **else**
6 $\quad \mathcal{L}^{\mathcal{T}_p}(i) = \mathcal{L}_{p,1}^k(i),\ \forall\ k \in \mathcal{T}_p(i)$
7 $\quad B(i) = \widehat{B}(i) - \sum_{k\in\mathcal{T}_p(i)} \mathcal{L}_{p,1}^k(i)$
8 $\quad$ **if** $(\mathcal{T}_p^m(i) \neq \emptyset)$ **then**
9 $\quad\quad w^{\mathcal{T}_p^m}(i) = (|\mathcal{T}_p^m(i)|/(2 * |\mathcal{T}_p^v(i)| + |\mathcal{T}_p^m(i)|))$
10 $\quad\quad w^{\mathcal{T}_p^v}(i) = 1 - w^{\mathcal{T}_p^m}(i)$
11 $\quad\quad B^v(i) = B(i) * w^{\mathcal{T}_p^v}(i)$
12 $\quad\quad B^m(i) = B(i) * w^{\mathcal{T}_p^m}(i)$
13 $\quad\quad \mathcal{L}^{\mathcal{T}_p^v}(i) = \max_{j\in[2:J]} \{\mathcal{L}_{p,j}^k(i) | \sum_{k\in\mathcal{T}_p^v(i)} \mathcal{L}_{p,j}^k(i) \leq B^v(i)\}$
14 $\quad\quad \mathcal{L}^{\mathcal{T}_p^m}(i) = \max_{j\in[2:J]} \{\mathcal{L}_{p,j}^k(i) | \sum_{k\in\mathcal{T}_p^m(i)} \mathcal{L}_{p,j}^k(i) \leq B^m(i)\}$
15 $\quad\quad B^b(i) = B(i) - (\sum_{k\in\{\mathcal{T}_p^v(i), \mathcal{T}_p^m(i)\}} \mathcal{L}_{p,j}^k(i))$
16 $\quad$ **else**
17 $\quad\quad B^v(i) = B(i)$
18 $\quad\quad \mathcal{L}^{\mathcal{T}_p^v}(i) = \max_{j\in[2:J]} \{\mathcal{L}_{p,j}^k(i) | \sum_{k\in\mathcal{T}_p^v(i)} \mathcal{L}_{p,j}^k(i) \leq B^v(i)\}$
19 $\quad\quad B^b(i) = B(i) - (\sum_{k\in\mathcal{T}_p^v(i)} \mathcal{L}_{p,j}^k(i))$
20 $\quad \mathcal{L}^{\mathcal{T}_p^b}(i) = \max_{j\in[2:J]} \{\mathcal{L}_{p,j}^k(i) | \sum_{k\in\mathcal{T}_p^b(i)} \mathcal{L}_{p,j}^k(i) \leq B^b(i)\}$

marginal tiles are calculated. The maximum available video bitrates not exceeding each region's corresponding bandwidth are allocated to each viewport and marginal tiles (lines 13-14). The bandwidth for the background tiles is calculated by subtracting the allocated budget to the viewport and marginal tiles from the updated bandwidth budget (line 15). If there are no marginal tiles, then for *Case 1* or *Case 3* of Algorithm 7.2, an aggressive or relatively conservative quality allocation is considered for viewport tiles, respectively, to ensure visual smoothness (line 18). Following this, the background bandwidth budget is calculated (line 19). Finally,

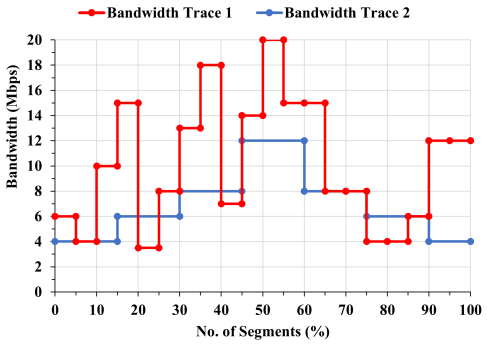


Figure 7.5: Bandwidth traces employed in experiments.

for *Case 1*, *Case 2*, or *Case 3*, the quality of the background tiles is increased as well by respecting the bandwidth budget (line 20).

7.4 Experimental Evaluation

This section reports on our proposed scheme's performance evaluation with a wide variety of content and in diverse network conditions.

7.4.1 Experimental Setup

The proposed solution evaluation is performed by modifying a VR player provided by [48], running on Ubuntu 16.04 machine with a 64-bit Intel Core i7-7500U CPU 2.70 GHz quad-core and 16 GB memory. The VR player requests the 360° video segments from the HTTP server based on the available bandwidth and estimated viewpoint coordinates. The viewport coverage is set to 110°, as used by Wu et al. [225] to collect users' head movement traces. The experiments are performed using dynamic connection speeds between the HTTP client and the HTTP server, shown in Fig. 7.5. Bandwidth trace 1 has more irregular increasing and decreasing trends compared to bandwidth trace 2. The maximum connection speed for trace 1 is 20 Mbps, while for trace 2, the maximum bandwidth value is 12 Mbps.

7.4.1.1 Video Traces

This work employs real head movement records of 48 VR users provided in [225]. This dataset contains viewing traces collected through two experiments for the con-

Table 7.3: Content Characteristics

Videos	Category	Duration	Resolution	FPS
Football	Sport	2′44″	3840x2160	25
Performance	Performance	4′38″	3840x1920	29
Spotlight	Film	4′53″	3840x2160	30
Surfing	Sport	3′25″	3840x1920	29
VR-Interview	Talkshow	3′07″	3840x1920	25

tent belonging to five categories: Documentary, Film, Sport, Performance, and Talkshow. Experiment 1 includes nine on-demand videos, while experiment 2 contains 9 live VR videos. We choose five videos from both experiments, namely, **LOSC Football**[1], **Weekly Idol-Dancing**[2], **Google Spotlight-HELP**[3], **GoPro VR-Tahiti Surf**, and **Rio Olympics VR Interview**[4] from this dataset. The five different duration immersive clips in this dataset can be classified into four categories: Sport (**LOSC Football** and **GoPro VR-Tahiti Surf**), Performance (**Weekly Idol-Dancing**), Film (**Google Spotlight-HELP**), and Talkshow (**Rio Olympics VR Interview**). These videos are referred to as **Football**, **Performance**, **Spotlight**, **Surfing**, and **VR Interview** throughout the remaining chapter. Table 7.3 summarizes the content features of five videos. All the videos are rescaled to 4K resolution using FFmpeg[5] software. The videos are spatially split into 4x3, 6x4, and 8x6 tiling patterns. The tiles are encoded using an open-source Kvazaar encoder [280] considering five different quantization parameter (QP) values (i.e., 22, 27, 32, 37, and 42). Subsequently, the MPEG-DASH video segments are generated using GPAC MP4Box[6] with a duration of 1s, 1.5s, and 2s. The average segment sizes for each video are illustrated in Table 7.4. The simulation length is adjusted following the duration of each video. The experimental settings are described in Table 7.5.

7.4.1.2 Comparative Approaches

DFT solutions are compared with dynamic tiling-based (ATS) and fixed tiling-based (UVP, CTF, PBA, AVR) solutions. UVP, CTF, PBA, and AVR comparative

[1] `https://www.youtube.com/watch?v=lvH89OkkKQ8`
[2] `https://www.youtube.com/watch?v=bFE2PN9_THI`
[3] `https://www.youtube.com/watch?v=G-XZhKqQAHU`
[4] `https://www.dropbox.com/s/kvvzox0dl3nu0qy/2-3-RioVR.mp4?dl=0`
[5] `https://ffmpeg.org/`
[6] `https://gpac.wp.imt.fr/mp4box/`

Table 7.4: Average and Standard Deviations of Segment Bitrates [Mbps] for the **Football**, **Performance**, **Spotlight**, **Surfing**, and **VR Interview** Videos

Video	QP	1s			1.5s			2s		
		4x3	6x4	8x6	4x3	6x4	8x6	4x3	6x4	8x6
Football	22	6.9±2.3	7.0±2.3	7.2±2.3	10.5±5.1	10.6±5.1	10.9±5.2	13.8±4.6	14.1±4.6	14.4±4.6
	27	3.5±1.3	3.6±1.4	3.8±1.4	5.3±2.8	5.5±2.8	5.7±2.9	7.1±2.7	7.3±2.7	7.6±2.7
	32	1.9±0.8	2±0.8	2.2±0.8	2.9±1.6	3.1±1.6	3.3±1.6	3.9±1.5	4.1±1.5	4.5±1.6
	37	1.1±0.4	1.2±0.4	1.4±0.4	1.7±0.9	1.8±0.9	2.1±1	2.3±0.9	2.4±0.9	2.8±0.9
	42	0.7±0.2	0.7±0.2	0.9±0.2	1±0.5	1.1±0.5	1.4±0.6	1.3±0.5	1.5±0.5	1.8±0.5
Performance	22	8.5±2.9	8.6±2.9	8.9±3.0	12.8±5.9	13.0±5.9	13.4±6.0	17.0±4.7	17.3±4.7	17.8±4.8
	27	4.6±1.7	4.7±1.7	5.0±1.7	6.9±3.3	7.1±3.3	7.5±3.4	9.3±2.7	9.5±2.7	10.0±2.7
	32	2.6±0.9	2.7±0.9	2.9±0.9	4.0±1.9	4.1±1.9	4.5±2.0	5.3±1.5	5.5±1.5	6.0±1.5
	37	1.6±0.5	1.7±0.5	1.9±0.5	2.4±1.1	2.5±1.1	2.8±1.2	3.2±0.9	3.4±0.8	3.8±0.9
	42	0.9±0.3	1.0±0.3	1.2±0.3	1.4±0.6	1.6±0.6	1.9±0.7	1.9±0.5	2.1±0.5	2.5±0.5
Spotlight	22	13.6±8.8	13.9±8.8	14.3±8.9	20.4±15.0	20.9±15.2	21.5±15.4	27.1±17.1	27.7±17.2	28.5±17.3
	27	7.2±5.3	7.4±5.3	7.7±5.4	10.8±8.9	11.1±9.0	11.6±9.1	14.3±10.3	14.8±10.4	15.5±10.5
	32	4.0±3.1	4.2±3.1	4.5±3.2	6.1±5.2	6.3±5.3	6.7±5.4	8.1±6.1	8.4±6.1	9.0±6.2
	37	2.3±1.8	2.4±1.8	2.7±1.8	3.5±2.9	3.7±3.0	4.1±3.1	4.7±3.5	4.9±3.5	5.4±3.5
	42	1.3±0.9	1.4±0.9	1.6±0.9	2.0±1.5	2.2±1.6	2.5±1.6	2.7±1.7	2.9±1.8	3.3±1.8
Surfing	22	22.7±11.2	23.0±11.3	23.5±11.4	34.0±21.0	34.5±21.2	35.3±21.4	45.3±22.2	45.9±22.3	46.9±22.5
	27	12.8±6.7	13.0±6.8	13.4±6.8	19.2±12.4	19.5±12.5	20.2±12.7	25.5±13.3	26.0±13.4	26.8±13.5
	32	7.2±3.9	7.4±3.9	7.7±3.9	10.8±7.1	11.1±7.2	11.6±7.3	14.4±7.7	14.7±7.8	15.4±7.8
	37	4.0±2.2	4.1±2.2	4.4±2.2	6.0±3.9	6.2±4.0	6.6±4.1	7.9±4.3	8.2±4.3	8.8±4.3
	42	2.1±1.1	2.2±1.1	2.5±1.1	3.2±2.0	3.4±2.1	3.7±2.2	4.2±2.2	4.5±2.2	5.0±2.2
VR Interview	22	7.6±1.0	7.7±1.1	7.8±1.1	11.4±4.1	11.5±4.2	11.8±4.3	15.2±2.0	15.4±2.0	15.7±2.1
	27	3.7±0.7	3.8±0.7	3.9±0.7	5.5±2.1	5.7±2.2	5.9±2.3	7.4±1.3	7.6±1.3	7.9±1.4
	32	1.7±0.3	1.8±0.4	2.0±0.4	2.6±1.0	2.8±1.1	3.0±1.2	3.5±0.7	3.7±0.7	4.0±0.8
	37	0.9±0.2	1.0±0.2	1.2±0.2	1.4±0.6	1.6±0.6	1.8±0.7	1.9±0.4	2.1±0.4	2.5±0.4
	42	0.6±0.1	0.7±0.1	0.8±0.1	0.9±0.3	1.0±0.4	1.3±0.4	1.2±0.2	1.4±0.2	1.7±0.2

Table 7.5: Simulation Settings

Parameter	Configuration
Tiling patterns	4x3, 6x4, 8x6
Encoding versions (QPs)	22, 27, 32, 37, 42
Segment length (s)	1s, 1.5s, 2s
Buffer length (s)	2s, 3s, 4s
Viewport size	110°
Simulation length	Video duration
QoE Weight coefficients	$\alpha = 1, \beta = 0.8, \gamma = 0.6, \delta = 0.2$

solutions are described in section 6.4.1.5. The ATS [246] solution performs adaptive tiles selection based on weighted viewport distortions. The tiling pattern resulting in minimum viewport distortion or maximum viewport bitrate is selected for streaming during each decision interval. The tiles are grouped into viewport and background regions, where the viewport tiles are allocated with the maximum possible bitrate, and background tiles are streamed at a minimum quality level.

7.4.1.3 Evaluation Metrics

The performance of the compared schemes is assessed in terms of the following metrics:

1. **Tiles Overlap:** This metric measures the real and predicted viewport tiles overlap as defined in eq. (7.9).

2. **Average QoE:** It reflects the average quality score of all the users for each video for the QoE metric defined in eq. (7.5).

7.4.2 Experimental Results

This subsection presents the experimental results and analysis for each solution under different settings. Table 7.6 represents the overall streaming behavior of the **DFT1** client for five different motion videos watched by 48 VR users. **DFT1** supports the larger visual area with higher quality streaming; therefore, for all the videos, larger resolution tiles (i.e., 4x3 and 6x4) are predominantly selected. However, the percentage of the 6x4 tiling pattern selection is decreased, and the percentage of 4x3 tiling pattern selection is slightly (5.38%) increased by increasing the segment duration from 1s to 2s for all the videos. Overall, a small percentage of smaller resolution tiles (i.e., 8x6) is selected for all videos. **DFT1** selects a 4x3 tiling pattern for more than 67% for **VR Interview** video. This is because a higher visual area leads to a higher visual quality score for this video. **DFT1** mostly performs aggressive bitrate selection for selected tiling patterns. **DFT1** fetches the segments of **Football**, **Performance**, **Spotlight**, **Surfing**, and **VR Interview** videos by performing aggressive bitrate selection by up to 59.14%, 75.33%, 66.27%, 58.75%, and 78.43%, respectively, averaged across three segment durations. **DFT1** performs weighted quality adjustments for segments of these videos by up to 32.37%, 21.08%, 27.64%, 32.66%, and 17.18%. Interestingly, with the increase in segment duration, the percentage of aggressive quality adjustments is decreased, leading to more weighted quality adjustments. In addition, a tiny percentage of conservative bitrate selection is observed for all the videos in the **DFT1** solution.

The streaming behavior of the **DFT2** client is presented in Table 7.7. **DFT2** achieves a perfect viewport match (by up to 65.50%), a partial viewport match (by up to 27.71%), and a complete viewport mismatch (by up to 6.77%) by selecting on average 8x6, 6x4, and 4x3 tiling patterns, respectively. In particular, **DFT2** observes a perfect viewport match (i.e., 57.35% for the **Football** video, 73.88% for the **Performance** video, 64.95% for the **Spotlight** video, 55.46% for the **Surfing** video, and 75.86% for the **VR Interview** video) averaged across three prediction

Table 7.6: Streaming Behavior [%] of **DFT1** Client in terms of Tiling Pattern Selection, Tiles Selection, and Bitrate Adaptation Scenarios

Videos	Segment Duration	Tiling Pattern [%]			Tiles Selection: Case 1 Bitrate: Aggressive			Tiles Selection: Case 2 Bitrate: Weighted			Tiles Selection: Case 3 Bitrate: Conservative		
		8x6	6x4	4x3	8x6	6x4	4x3	8x6	6x4	4x3	8x6	6x4	4x3
Football	1	17.73	52.27	30.00	8.84	37.39	20.45	8.70	14.36	6.45	0.19	0.53	3.09
	1.5	17.53	50.10	32.38	7.42	32.59	17.74	9.54	16.02	7.91	0.57	1.49	6.73
	2	17.48	47.97	34.55	6.43	30.06	16.51	9.88	15.19	9.07	1.17	2.72	8.97
Performance	1	4.96	66.74	28.30	2.68	55.40	21.41	2.16	11.08	5.42	0.12	0.25	1.47
	1.5	5.42	63.77	30.81	2.80	51.00	21.32	2.48	12.18	6.48	0.14	0.59	3.02
	2	6.71	60.49	32.79	3.75	46.66	20.97	2.67	13.08	7.69	0.30	0.75	4.14
Spotlight	1	21.00	44.49	34.51	12.99	32.75	27.48	7.71	11.13	5.09	0.30	0.61	1.93
	1.5	19.01	42.00	39.00	10.30	27.67	27.24	8.10	12.86	7.47	0.61	1.46	4.28
	2	18.06	38.90	43.04	8.57	24.14	27.66	8.65	12.50	9.41	0.84	2.27	5.97
Surfing	1	20.46	42.23	37.32	11.12	28.75	27.63	9.10	12.89	7.00	0.24	0.59	2.68
	1.5	19.30	39.81	40.89	8.67	23.62	24.68	10.04	14.32	9.87	0.59	1.87	6.34
	2	17.84	36.67	45.49	7.12	19.72	24.96	9.59	14.02	11.17	1.13	2.93	9.36
VR Interview	1	6.46	26.42	67.11	3.93	19.11	59.60	2.46	6.87	5.99	0.07	0.45	1.52
	1.5	7.29	24.14	68.57	4.33	16.15	57.48	2.67	7.17	7.64	0.29	0.82	3.44
	2	8.60	23.16	68.23	4.61	14.45	55.65	3.20	7.53	8.00	0.78	1.19	4.59

horizons. The lower values of perfect viewport match for the sports videos, i.e., **Football** and **Surfing**, reflects the fast-moving objects within these videos. Therefore, the client observes a lower percentage of aggressive bitrate adaptation, 34.03%, 31.31% for the **Football** and **Surfing** videos, with an 8x6 tiling pattern in comparison to other videos. A very small percentage of viewport mismatch is observed for the relatively static motion content, i.e., **Performance** and **VR Interview**, even when the segment duration is set to 2s. **DFT2** requests a 4x3 tiling pattern by up to 3.36% and 5.94% for the **Performance** and **VR Interview** videos, respectively when the segment duration is set to 2s. Therefore, these videos observe a limited percentage of extended viewport and conservative bitrate adaptation cases compared to the other videos. Interestingly, the percentage of the 4x3 and 6x4 tiling pattern selection increased with the increase in segment duration. Moreover, the client observed a higher percentage of weighted and conservative quality adjustments with the increase in segment duration. The percentage of fixed viewport case that comes with aggressive bitrate adjustment is reduced with the increase in segment duration. This is because when predicting longer in the future, the prediction performance is reduced due to the highly dynamic user viewing behavior.

Fig. 7.6 represents the streaming behavior of the ATS algorithm in terms of selecting the average tiling pattern for the entire video data set. ATS results in selecting a 4x3 tiling pattern mostly for **Football** video, followed by 8x6 and 6x4 tiling grids. ATS requests 6x4 and 8x6 tiling patterns for about 15.06%, and 50.94% of the streaming

Table 7.7: Streaming Behavior [%] of **DFT2** Client in terms of Tiling Pattern Selection, Tiles Selection, and Bitrate Adaptation Scenarios

Videos	Segment Duration	Tiling Pattern [%]			Tiles Selection: Case 1 Bitrate: Aggressive			Tiles Selection: Case 2 Bitrate: Weighted			Tiles Selection: Case 3 Bitrate: Conservative		
		8x6	6x4	4x3	8x6	6x4	4x3	8x6	6x4	4x3	8x6	6x4	4x3
Football	1	62.94	31.97	5.09	39.46	10.66	0.74	23.24	21.16	2.39	0.24	0.15	1.96
	1.5	56.31	33.68	10.02	33.03	9.27	0.73	22.76	24.06	4.05	0.52	0.34	5.24
	2	52.82	34.55	12.63	29.62	8.51	0.76	22.41	25.28	4.07	0.79	0.76	7.80
Performance	1	77.81	20.51	1.68	57.95	8.03	0.20	19.66	12.40	0.85	0.19	0.08	0.62
	1.5	73.85	23.28	2.87	52.48	7.38	0.17	21.07	15.64	1.19	0.30	0.26	1.51
	2	69.99	26.65	3.36	48.91	7.76	0.10	20.65	18.51	0.99	0.43	0.37	2.26
Spotlight	1	71.39	24.18	4.44	46.03	8.26	0.75	25.21	15.73	1.92	0.15	0.19	1.76
	1.5	64.12	28.40	7.48	38.76	7.69	0.56	24.96	20.42	2.69	0.41	0.29	4.23
	2	59.35	30.77	9.88	35.50	9.18	1.50	23.38	21.03	2.95	0.47	0.55	5.43
Surfing	1	62.08	32.17	5.74	37.10	10.25	0.87	24.75	21.78	2.71	0.23	0.14	2.15
	1.5	54.41	34.93	10.66	30.52	8.06	0.81	23.51	26.51	4.43	0.38	0.36	5.43
	2	49.90	36.25	13.86	26.31	7.71	0.85	22.92	27.51	4.53	0.67	1.03	8.47
VR Interview	1	79.18	17.67	3.15	58.25	6.38	0.29	20.68	11.23	1.59	0.25	0.06	1.27
	1.5	74.75	20.33	4.92	53.41	6.15	0.34	20.93	13.93	1.65	0.40	0.25	2.94
	2	73.66	20.41	5.94	50.13	5.62	0.31	22.92	14.14	1.68	0.60	0.65	3.94

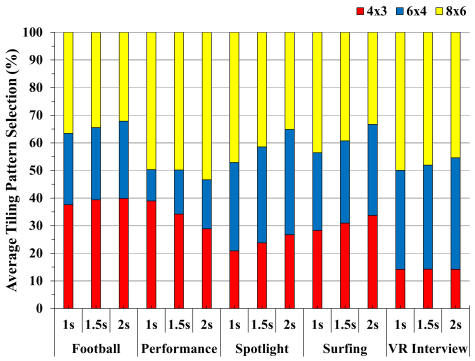
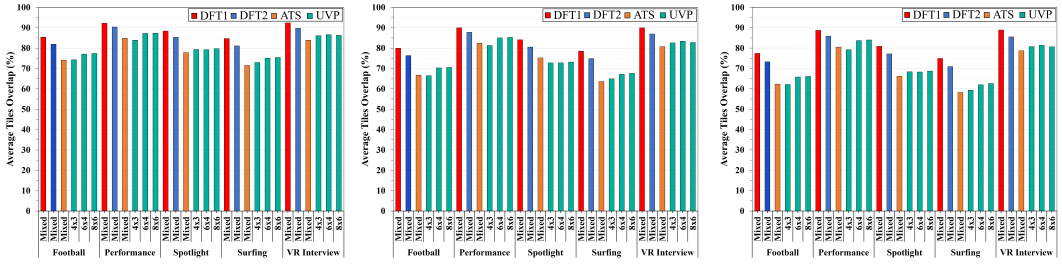


Figure 7.6: Average Tiling Pattern Selection in the ATS method.

session for **Performance** video with 1s, 1.5s, and 2s. The 6x4 tiling pattern is mostly requested for **VR Interview** video with a 2s segment duration. For **Spotlight** and **Surfing** videos, ATS mostly requests an 8x6 tiling pattern (41.23% and 38.71%) followed by the 6x4 (34.97% and 30.29%) and 4x3 (23.79% and 30.98%), respectively. For the entire test dataset, the ATS method achieves 42.61% for selecting 8x6, 29% for 6x4, and 28.39% for 4x3. This is because the larger tiling patterns result in relatively larger segment sizes, and ATS selects tiling patterns based on the minimum weighted viewport distortions measured in terms of achieving maximum viewport bitrate.

7.4.2.1 Average Tiles Overlap

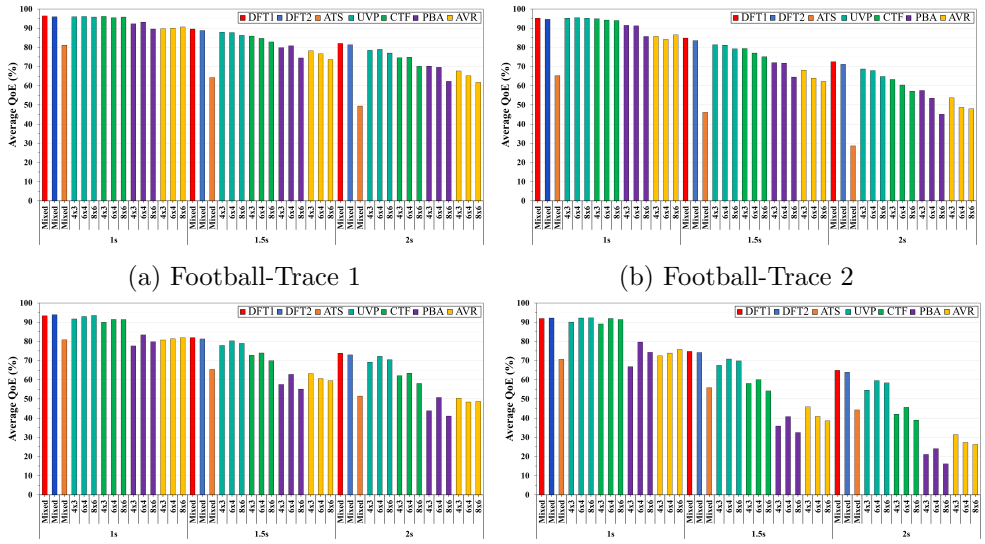
Fig. 7.7 summarizes the average tiles overlap results (per video 48 head movement traces) for the **DFT1**, **DFT2**, ATS, and UVP methods under various prediction horizons. The spherical walk prediction method is adopted for the ATS, UVP, CTF,



(a) Segment duration: 1s (b) Segment duration: 1.5s (c) Segment duration: 2s

Figure 7.7: Average tiles overlap achieved by **DFT1**, **DFT2**, and Spherical Walk methods for the **Football**, **Performance**, **Spotlight**, **Surfing**, and **VR Interview** videos. These videos were prepared in 4x3, 6x4, and 8x6 tiling patterns, which were watched by 48 VR users. The recorded results are for 1s, 1.5s, and 2s segment durations.

PBA, and AVR streaming algorithms. Fig. 7.7 shows that **DFT1** achieves higher tiles overlap for all five videos since the tiles in dynamic size tiling patterns are arranged considering the arc distance between the viewpoint and each tile's center. The users have relatively dynamic head movements for the **Football** and **Surfing** videos since both videos contain several outdoor sports-related fast-moving objects. On the contrary, a higher average tile overlap is observed for the **Performance** and **VR interview** videos since the point of attentions is only the slow-moving indoor objects. It is notable that **DFT1** and **DFT2** attain higher matching performance and outperform the ATS and UVP methods adopting a single prediction mechanism for different user behaviors. For all 48 VR users, **DFT1** and **DFT2** experience an average tiles overlap of 85.40% and 81.95% (**Football**), 92.22% and 90.43% (**Performance**) for 1s (Fig. 7.7a), 84.09% and 80.50% (**Spotlight**), 90.03% and 86.94% (**VR Interview**) for 1.5s (Fig. 7.7b), 74.93% and 70.91% (**Surfing**) and 88.92% and 85.54% (**Performance**) for 2s (Fig. 7.7c) prediction windows. The reason is that the proactive tiles selection cases adapt better to the diverse spatial and temporal information of different motion scenes. Simultaneously, the ATS method observes the lowest average tile overlap than the UVP method for content with fast and stable head movements. As seen, for the **Spotlight** videos, **DFT1** outperforms the ATS and UVP methods by up to 8.88% and 14.66% for the next 1.5s (Fig. 7.7b), and by 11.19% and 12.43% for 2s prediction horizon (Fig. 7.7c), respectively. Similarly, **DFT2** shows its superior capability in increasing viewport overlap



(a) Football-Trace 1 (b) Football-Trace 2

(c) Performance-Trace 1 (d) Performance-Trace 2

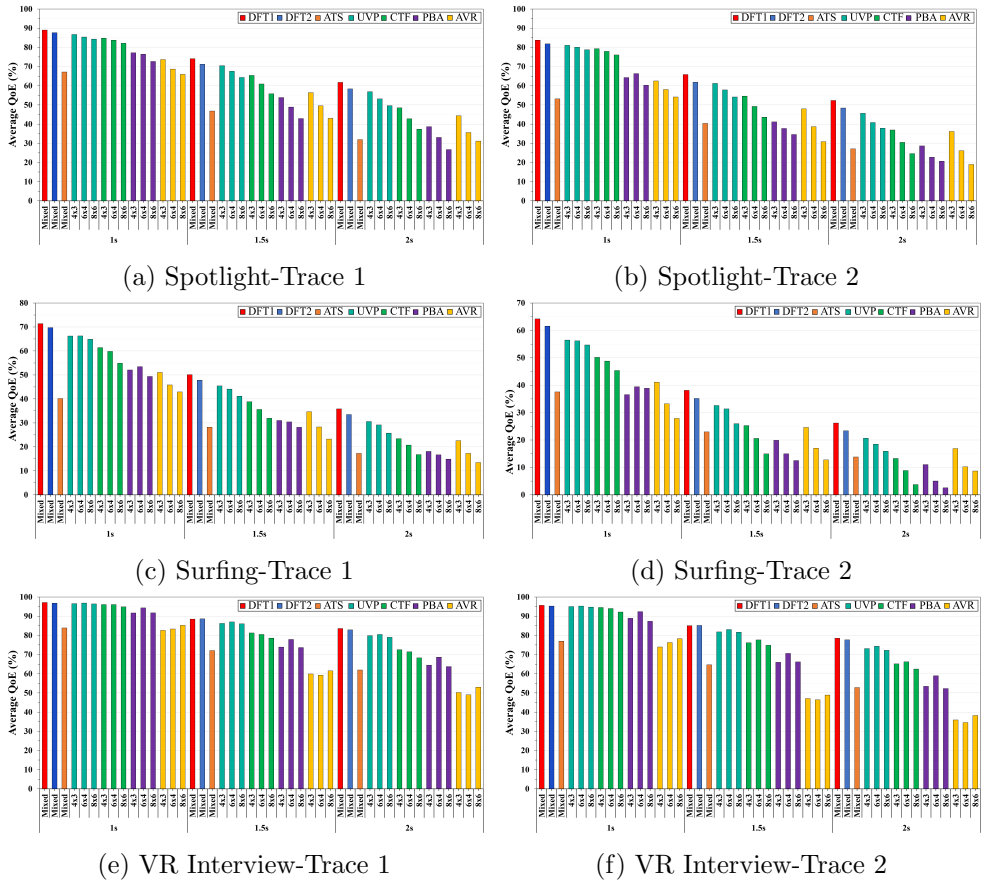
Figure 7.8: Average QoE achieved by **DFT1**, **DFT2**, UVP, CTF, PBA, and AVR streaming clients for **Football** and **Performance** videos. These videos were prepared in 4x3, 6x4, and 8x6 tiling patterns, which were watched by 48 VR users. The recorded results are for 1s, 1.5s, and 2s segment durations while employing the bandwidth trace 1 and trace 2.

for **Surfing** video and outperforms by achieving about 7.37% 9.02%, 10.35% higher viewport overlap for 1s, 1.5s, and 2s prediction time, respectively. For **Spotlight** video, the average gain of **DFT** methods ranges from 6.27%-9.32%, 7.02%-10.61%, and 9.28%-12.98% for different prediction horizons. The tiles overlap for the **DFT2** is reduced by (**Football**) and by 9.78% (**Surfing**) when segment duration is increased from 1s to 2s, while for the ATS and UVP methods, it is reduced by 11.57% and 11.83% (**Football**) and by 13.17% and 13.29% (**Surfing**) (Fig. 7.7). As a result, it can be concluded that employing two prediction mechanisms (**DFT**) leads to better viewing probability than employing a single prediction mechanism for fixed (UVP) and dynamic (ATS) tiling-based streaming.

7.4.2.2 Average QoE

Next, the performance of the proposed solutions is tested against five tile-based methods by employing bandwidth trace 1 and trace 2 for **Football** and **Performance** videos. The QoE weight coefficients are set as $\alpha = 1, \beta = 0.8, \gamma = 0.6, \delta = 0.2$. The reference tile-based delivery solutions incorporate viewers' head motion

patterns for adaptive bitrate selection. Fig. 7.8 depicts the video quality experienced and averaged across 48 users for 1s, 1.5s, and 2s segments. It can be noted that the algorithms' performance in Fig. 7.8a and Fig. 7.8c is higher than what is depicted in Fig. 7.8b and Fig. 7.8d. The average QoE values are lower accordingly with bandwidth decrease for the same QoE weight coefficients. The higher QoE scores of larger tiling patterns (i.e., 6x4 and 8x6) for the 1s **Performance** video are due to the higher average tiles overlap (Fig. 7.8c-Fig. 7.8d). Despite the lower tiles overlap, UVP, CTF, PBA, and AVR streaming methods achieve higher quality scores for the **Football** video with a 1s segment duration due to the lowest average segment sizes (Fig. 7.8a and Fig. 7.8b). Fig. 7.8a results show that **DFT1** improves the QoE compared to other methods by about 3.96%, 9.29%, and 12.90% for **Football** video with 1s, 1.5s, and 2s segment durations when employing bandwidth trace 1. For both bandwidth traces, **DFT1** outperforms ATS by about 25.31%-38.71%, UVP by about 2.25%-4.25%, CTF by about 5.08%-7.67%, PBA by about 11.16%-15.42%, and AVR by about 13.37%-20.07% for **Football** video with a 1.5s segment duration. Fig. 7.8b shows that **DFT2** achieves 5.44% (for 1s), 12.56%(for 1.5s), and 15.98% (for 2s), higher average QoE for **Football** video streaming in comparison to other solutions. The increment in quality with the increase in segment duration reflects that **DFT** solutions have better prediction accuracy with longer segment duration. Similarly, Fig. 7.8c and Fig. 7.8d show that **DFT** solutions observe the highest visual quality levels for all segment durations since they better accommodate the user's viewing directions than the other methods. In particular, **DFT1** achieves an average gain of 7.45% (1s), 14.42% (1.5s), and 17.69% (2s) for **Performance** video streaming under bandwidth trace 1, while it is increased to 10.34% (1s), 23.20% (1.5s) and 27.23% (2s) for bandwidth trace 2. Viewport mismatch leads to uncomfortable quality degradation for tile-based streaming methods for higher segment lengths. In **DFT** methods, the combination of viewport coverage selection and bitrate selection policies favor the higher quality perceptibility of the viewing area. For **Performance** video with a 2s segment duration, **DFT2** outperforms fixed tiling-based solutions by about 2.34%-6.39%, 11.76%-21.67%, 27.75%-43.46%, and 23.80%-35.58% for both bandwidth scenarios.



(a) Spotlight-Trace 1 (b) Spotlight-Trace 2

(c) Surfing-Trace 1 (d) Surfing-Trace 2

(e) VR Interview-Trace 1 (f) VR Interview-Trace 2

Figure 7.9: Average QoE achieved by **DFT1**, **DFT2**, UVP, CTF, PBA, and AVR streaming clients for the **Spotlight**, **Surfing**, and **VR Interview** videos. These videos were prepared in 4x3, 6x4, and 8x6 tiling patterns, which were watched by 48 VR users. The recorded results are for 1s, 1.5s, and 2s segment durations while employing the bandwidth trace 1 and trace 2.

The improved performance of **DFT** solutions over CTF and PBA methods is for the reason that they perform a uniform quality allocation for the predictive tiles to favor the higher visual quality levels with a reduced amount of data for the background tiles.

Fig. 7.9 illustrates the average QoE trends for the experiments performed for **Spotlight**, **Surfing**, and **VR Interview** videos. The QoE weight coefficients are set as: $\alpha = 1$, $\beta = 0.8$, $\gamma = 0.6$, and $\delta = 0.2$. It can be noted that the **Surfing** and **Spotlight** videos require higher bitrates for achieving higher quality scores than other videos (as can be seen in Table 7.4). Hence, achieving higher QoE with a limited network connection and higher QoE expectations is more difficult. On the other hand, **VR Interview** video results in higher QoE scores due to the lower average

segment sizes and higher viewport overlap. Therefore, segment size, bandwidth capacity, and viewport prediction strongly influence the streaming performance of 360° videos. For **Spotlight** video with a 2s segment duration, **DFT1** actively utilizes the available bandwidth budget and achieves average QoE improvements of up to 29.8%, 12.15%, 24.36%, 28.7%, and 30.6% compared to ATS, UVP 8x6, CTF 8x6, PBA 6x4, and AVR 8x6, respectively (Fig. 7.9a). This is because **DFT1** observes 14.65% and 12.15% higher average tiles overlap than the ATS and UVP methods for **Spotlight** video with a 2s segment duration (Fig. 7.7c). The average quality score for the **Surfing** video with a 1s segment duration under bandwidth trace 2 (Fig. 7.9d) are 64.21% for **DFT1**, 61.57% for **DFT2**, 37.53% for ATS, 56.45% for the UVP 4x3, 48.79% for the CTF 6x4, 38.9% for PBA 8x6, and 41.08% for the AVR 4x3 methods. For the **VR Interview** video with a 1.5s segment duration, **DFT2** improves the average QoE by up to 20.55% compared to ATS, 3.02% compared to UVP, 9% compared to CTF, 17.61% compared to PBA, and 37.74% compared to AVR for bandwidth trace 2 (Fig. 7.9f), while the average improvement for **DFT1** is 25.7%, 5.3%, 13.94%, 23.64%, and 42.3% across all tiling patterns of ATS, UVP, CTF, PBA, and AVR methods for 2s **VR Interview** video (Fig. 7.9e). The ATS method only outperforms the AVR method for some cases of **Performance** and **VR Interview** videos under different bandwidth profiles. The worst performance of the ATS method is because it restricts the quality of the background tiles to the minimum levels, which leads to lower quality scores under lower and medium prediction performance. Fig. 7.9 shows that **DFT1** and **DFT2** result in 16.53%, 15.56%, and 13.62% higher QoE for **Spotlight**, **Surfing**, and **VR Interview** videos, respectively, when simulated with all tiling patterns, segment durations, and bandwidth profiles. The reason for the quality improvement is that our QoE metric favors higher visible quality. The low QoE values for the PBA algorithm are because it assigns different priorities within the viewport zones (Z_1 and Z_2) while sacrificing the quality of the remaining tiles. As a result, PBA struggles with the user-perceived quality and visual smoothness objectives. UVP receives good performance on the spatial and temporal quality objectives because it uniformly increases the quality of the viewport and outer regions. However, the visible tiles' rate is reduced substan-

tially due to lower prediction performance. The degraded performance of the AVR method even under stable head movements is because it unnecessarily increases the quality of the adjacent tiles.

7.4.3 Discussion

The existing fixed tiling-based adaptive streaming solutions attempt to improve visual quality levels while reducing the spatial and temporal quality variations and the risk of playback interruptions. However, the proposed dynamic tiling-based streaming solutions result in more accurate viewport prediction and higher QoE levels since it systematizes the best resolution tiles for both static and dynamic motion scenes. The ATS and UVP solution uniformly allocates bitrate to the tiles belonging to the same classification to improve the visual smoothness objectives defined in Eq. 7.2-7.3. However, ATS limits the background quality to the minimum level with lower viewport matching performance and achieves the lowest average QoE values for the entire test dataset. UVP, on the other hand, increases the quality of the background tiles to the highest possible level and outputs better quality scores even under difficult to predict head movements. For CTF and PBA solutions, the primary focus is on improving the quality of the center tile, which significantly leads to degraded quality levels for poor viewport prediction and spatial quality variations even for the stable viewport prediction results. The underperformance of the AVR streaming method under drastic and stable viewport switches is due to the inefficient tiles' arrangement to consume an essential share of the network bandwidth. In contrast, **DFT** solutions consume a much larger bandwidth share for the most likely to be watched tiles and result in higher QoE scores than the comparative methods for all tested datasets. **DFT1** provides a useful trade-off between visual area and visual quality, and **DFT2** works to minimize the viewport mismatch ratio. Both proposed solutions work reasonably well under different testing settings and try to avoid unacceptable viewport deviations for end-users. **DFT** solutions achieve a fair share of bandwidth consumption for tiles belonging to the viewport, marginal, and background regions; therefore, it results in lower spatial and temporal quality variations for different viewport prediction results. Under stable or variable motions

of experienced or naive VR users, the dynamic selection of the tiling patterns and coverage of the visible region (Fixed/Extended) along with the aggressive, weighted, and/or conservative quality adjustment policies provide improved QoE for different bandwidth settings, segment sizes, and motion trends. Therefore, the **DFT**-based delivery of 360° videos performs better than when the other benchmark methods are employed.

7.5 Conclusions

This chapter proposed and evaluated dynamic video frame tiling-based solutions, **DFT1** and **DFT2**, for advancing predictive tiles selection during adaptive 360° video streaming. **DFT1** performs an interactive tiling pattern selection by leveraging the visual area and associated weighted quality to overcome the attention field's dynamics. **DFT2** observes the potential viewport prediction differences to best accommodate different tiling versions. **DFT** solutions extract the user attention fields by leveraging two viewport prediction mechanisms to transmit the high-quality tiled videos. Three tiles selection scenarios were adopted in this work to output stable and perfectly matched viewing tiles. The proposed region-based bitrate adaptation algorithm reduces the intricacy of bitrate selection according to the level of interest in tiles by implementing aggressive, weighted, or conservative quality adjustment policies. Consequently, **DFT** solutions achieve an appropriate balance between viewport availability and visual quality. The effectiveness of the **DFT** algorithms was evaluated through extensive trace-driven experiments. The experimental results on publicly available datasets under different segment lengths and bandwidth settings demonstrate that the proposed solutions can improve the streaming performance by achieving an important balance between different QoE objectives, i.e., increasing the perceived quality levels and reducing playback buffer emptying risk while maintaining spatial and temporal viewport smoothness, in comparison to other state-of-the-art methods.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

The very large popularity of on-demand multimedia streaming applications is very much due to the associated enriched user experience. Unfortunately, this user experience is mostly affected by the underlying network conditions and transmission performance. For instance, sub-optimal network utilization leads to inferior video quality, which influences negatively user perceived QoE levels. The ever-evolving multimedia applications have increasing demands in terms of exceptional QoE levels including in highly dynamic network conditions.

In this context, this work focuses on improving bandwidth utilization and end-user QoE in HAS for regular, multi-view, and 360° videos. The proposed solutions for various multimedia streaming types share some common components including content preparation, packaging, storage, transmission, bitrate selection, rendering, and playing. Chapter 4 introduces **TBOA**, a novel rate adaptation algorithm for DASH-based on-demand adaptive video streaming. **TBOA** monitors both the network throughput and playback buffer information to fetch the video segments with the most suitable video bitrates for each adaptation interval. The proposed *TBOA* solution has the advantage of increasing the video bitrate or sustaining the same bitrate to make efficient use of the available network bandwidth. The proposed **TBOA** algorithm is evaluated using different bandwidth and segment duration settings in

(i) single-client and (ii) multi-client environments. The functioning of the proposed method is assessed in NS-3 against popular throughput-based (i.e., SFTM) buffer-based (i.e., SVAA) and hybrid (i.e., AAASH, FDASH, OSMF) adaptive streaming solutions. The experimental results under short-term bandwidth fluctuations in a wired network show that **TBOA** achieves 1.7 Mbps, 0.36 Mbps, 3.4 Mbps, 1.10 Mbps, and 1.06 Mbps higher average video bitrate in comparison to SFTM, SVAA, AAASH, FDASH, and OSMF clients. Similarly, the average video bitrate gain under long-term bandwidth fluctuations in the wireless transmission is recorded as 2.72 Mbps, 0.39 Mbps, 4.41 Mbps, 2.48 Mbps, and 1.83 Mbps compared to other algorithms. Similarly, **TBOA** experiences 27.58%, 14.04%, 22.84%, 13.39%, and 5.13% higher average QoE compared to AAASH, FDASH, SFTM, OSMF, and SVAA algorithms in both wired and wireless network environments for all bitrates and segment durations. In multi-client scenarios, **TBOA** has the potential to achieve the highest average QoE and lowest inefficiency and instability values when the bandwidth is shared between three and five streaming clients in remote video playback. In particular, **TBOA** outperforms other algorithms by achieving higher average QoE, i.e., about 3.63% and 9.02% (for bitrate set $\mathcal{L}1$), 9.26% and 4.19% (for bitrate set $\mathcal{L}2$), and 4.82% and 7.91% (for bitrate set $\mathcal{L}3$) for three and five competing clients, respectively. The average achieved efficiency scores are about 6.63%, 21.38%, and 14.34% higher than those of the compared algorithms for three clients for each bitrate set, respectively, while for five clients this improvement is about 28.11%, 15.35%, and 18.27%. **TBOA** also receives on average 1.09%-5.32% and 1.37%-4.97% higher stability values for three and five streaming clients, respectively. No playback interruptions were observed for **TBOA** clients in both single-client and multi-client scenarios. Moreover, the slightly higher unfairness values among clients are well compensated by the highest video bitrate and lowest video quality switches in a multi-client environment.

Chapter 5 presents the system architecture, proposed design, and evaluation of the **MVP-DASH** framework under different testing scenarios. **MVP-DASH** is a priority-oriented hybrid adaptation solution that utilizes stream-associated weights to distribute the available bandwidth between different videos. **MVP-DASH** en-

sures that all video segments have been received during the previous transmission cycle before issuing the requests for the next segments. **MVP-DASH**'s prioritized bitrate adaptation algorithm enables the individual playback buffer of each stream to grow and sustain a steady-state level by allocating the best possible video bitrates to each stream. **MVP-DASH** tries to keep the requested bitrates of all the streams closer to the estimated throughput based on their associated priorities so that the buffer levels could be maintained at good levels across all streams. Moreover, it also leverages the path diversity in such a way that the transmission rates for high priority streams are always higher than those of the low-priority streams. The experimental evaluation using three video streams in (i) single-path and (ii) multi-path environments in comparison to *PMV* and *UMV* streaming solutions demonstrate that **MVP-DASH** mitigates the inefficient bandwidth utilization problem of existing solutions by streaming prioritized segments with higher bitrates and other segments at good quality levels. **MVP-DASH** achieves 339 Kbps and 180 Kbps higher average video bitrate for all three streams in comparison to *PMV* and *UMV* streaming clients in a single path scenario. **MVP-DASH** is the most stable multi-view algorithm and exhibits around 47 and 25 fewer video quality switches in a WiFi network environment. **MVP-DASH** plays high-quality segments with bitrates $\geq$ 1647 Kbps for high-priority stream over 95% of the time, while this time for low-priority streams is around 40%. During a multi-path streaming session, **MVP-DASH** outperforms *PMV* and *UMV* clients by achieving about 53 Kbps and 282 Kbps higher average video bitrate and 15 and 16 fewer video quality switches. **MVP-DASH** quickly switches to the highest bitrate segments (6220 Kbps) while utilizing a higher bandwidth path i.e., 7 Mbps, compared to the *PMV* streaming mechanism for high-priority stream. The transmission of low-priority streams (i.e., stream 2 and stream 3) with **MVP-DASH** over a lower bandwidth link (i.e., 5 Mbps) experiences a little quality difference. However the difference is the highest when employing the *PMV* streaming algorithm.

Chapter 6 presents **CFOV** and **SFOV**, prediction-oriented fixed tiling-based adaptive streaming solutions for 360° videos. **CFOV** and **SFOV** are prediction angles-based and prediction angles and differences-based innovative tiles selection solutions,

respectively. Two variants of each solution namely **CFOV1**, **CFOV2**, and **SFOV1**, **SFOV2** are introduced. **CFOV1** mainly identifies the streaming tiles based on two predicted FoVs, while **CFOV2** performs tiles selection by considering the adjusted prediction difference of two predicted viewpoints based on content complexity variations. In contrast, **SFOV1** involves a single prediction mechanism and the prediction difference between actual and predicted FoVs to perform region-based tiles selection. **SFOV2** similar to **CFOV2** involves content information, however, only a single prediction mechanism is employed. **CFOV** and **SFOV** solutions differentiate tiles into viewport, marginal and background regions by considering *Fixed FoV*, *Marginal Region*, and *Extended FoV* scenarios. It is interesting to note that the of the marginal and extended viewport regions is highly dynamic and could be different for subsequent segments. The fixed tiling-based bitrate adaptation algorithm is designed in such a way to achieve maximum possible quality levels under dynamic network and head movements. It terminates the marginal and background tiles when the available bandwidth is not sufficient. In this way, it further refines the streaming tiles selection by streaming viewport-only tiles, viewport and background tiles, or viewport, marginal, and background tiles. We compared the proposed solutions against the closest tile-based viewport-dependent (i.e., UVP, CTF, PBA, AVR) and viewport-independent (i.e., VIS) solutions by employing four 360° videos (i.e., Boxing, Conan, Football, and Spotlight) having different spatial and temporal information prepared in three tiling patterns (i.e., 4x3, 6x4, 8x6) and three segment lengths (1s, 2s, 3s). The success of the tiles selection solutions is verified by comparing **CFOV1**, **CFOV2**, **SFOV1**, and **CFOV2** solutions against a content-independent *spherical walk* prediction method. Results showed that the proposed solutions have significant improvements in terms of achieving higher tile overlap for all four videos under all segment lengths. In particular, **SFOV** and **CFOV** solutions outperform *spherical walk* prediction mechanism employed in UVP, CTF, PBA, and AVR methods by 4.15%-4.85% for 1s, by 8.30%-8.54% for 2s, and by 11.17%-11.55% for 3s prediction horizons for all videos and tiling patterns. To assess the performance of the bitrate adaptation algorithm involving tiles selection scenarios under three different bandwidth profiles, we define a new QoE metric that

considers the impact of not only visual quality and visual smoothness but also how the background quality deteriorates the perceived quality. We tested using three different weight coefficients to reflect the importance of different quality objectives to the users. Interestingly, all the proposed tiles selection solutions with the proposed bitrate adaptation algorithm outperform the five baseline algorithms by achieving higher QoE scores for all 48 VR users. For instance, the performance gain for **SFOV** and **CFOV** solutions is recorded under QoE-C1 as 16.47%-17.17% for 1s, 13.29%-14.06% for 2s, and 10.81%-11.73% for 3s, under QoE-C2 as 18.36%-19.20% for 1s, 14.54%-15.44% for 2s, and 11.33%-12.45% for 3s, under QoE-C3 as 23.24%-24.35% for 1s, 17.58%-18.73% for 2s, and 14.17%-15.48% for 3s segment lengths. The increase in QoE gain with an increase in weight coefficients reflects that the proposed solutions are capable to satisfy the requirements of users who have more concerns regarding bandwidth wastage and viewport quality variations. In addition, the proposed solutions (i.e., **CFOV** and **SFOV**) achieve significant bandwidth efficiency for all the videos and outperforms UVP by 15.94%, 10.13%, and 10.42%, CTF by 14.22%, 8.51%, and 8.98%, PBA by 19.33%, 12.83%, and 12.28%, AVR by 20.31%, 14.71%, and 13.58%, and VIS by 35.14%, 28.01%, and 23.43% for 1s, 2s, and 3s segment lengths, respectively.

The final significant contributions in this thesis were presented in Chapter 7. The dynamic tiling-based solutions (i.e., **DFT1** and **DFT2**) presented in this chapter are important as they involve a hierarchical adaptation structure. Usually, the tile-based solutions implement two-tier adaptations in terms of streaming tiles selection and bitrate adaptation. However, in **DFT** solutions a third level adaptation, i.e., tiling pattern selection, is performed to leverage the opportunity created by different resolution tiles in 360° video. In this context, a novel visual area-based weighted quality metric is defined in the **DFT1** solution, which assigns more weights to the quality levels of tiles closer to the viewpoint. The tiling pattern resulting in the highest weighted quality score is selected for streaming in the next adaptation interval. **DFT2** solution, on the other hand, checks the dispersion between original and predicted viewports and then decides among small, medium, or larger resolution tiles in each adaptation interval. Both tiling pattern selection solutions are supported by

tiles selection and bitrate adaptation algorithms. The tiles selection solution in **DFT** is a prediction angle-based tile selection solution and considers the same streaming regions as described in Chapter 6. However, the range of the streaming regions and decision factors are different here. The dynamic tiling-based bitrate adaptation algorithm implements *aggressive*, *weighted*, and *conservative* quality adaptations based on spatial distribution of tiles. We compared **DFT** solutions with one dynamic tiling-based (i.e., ATS) and four fixed tiling-based (i.e., UVP, CTF, PBA, and AVR) streaming solutions using five different motion 360° videos (i.e., Football, Performance, Spotlight, Surfing, and VR-Interview) for 48 real VR head movement traces. We showed how **DFT** and ATS solutions adapt different tiling versions for each video during 360° video streaming. We measured how much tile overlap is achieved by DFT, ATS, and UVP methods for 1s, 1.5s, and 2s segment duration. Both **DFT** solutions outperform ATS and UVP methods by achieving 9.42% and 6.62%, 10.32% and 7.83%, and 12.31% and 8.82% higher tiles overlap for 1s, 1.5s, and 2s segment durations, respectively. The proposed QoE metric in dynamic tiling-based adaptive streaming considers maximizing perceived visual quality levels and minimizing visual fluctuations and playback buffer risks objectives. The adaptive transmission under two bandwidth traces show that the bitrate adaptation algorithm for **DFT1** and **DFT2** solutions result in QOE increment of 9.7%-10.56% for **Football**, 16.33%-16.72% for **Performance**, 15.08%-17.99% for **Spotlight**, 14.33%-16.79% for **Surfing**, and 13.45%-13.79% for **VR Interview** videos in comparison to other solutions for all segment durations.

8.2 Future Works

Despite the amount of research done on interactive and immersive multimedia streaming applications, several research challenges still exist and provide opportunities for further investigations to reach the goal of having a viable and close to the real-life viewing experience. The development of new applications for diverse domains is natural; however, of concern is the need for addressing other issues related to immersive media content transmission and providing support for a large

number of users, long-term accurate viewport prediction, and QoE-oriented delivery solutions.

1. The influence of segment and buffer duration on DASH-based on-demand streaming plays a critical role. Therefore, the configuration of buffer space is vital to achieving an efficient adaptive playback experience. The testbed for **TBOA** and **MVP-DASH** could be extended to observe the adaptation performance under various segment and buffer lengths. **MVP-DASH** can be extended by employing aggregated bandwidth estimation of all paths to enhance bandwidth utilization for the larger number of streams in a multi-user multi-view streaming environment. The adaptation algorithm for fixed and dynamic tiling-based adaptation solutions can be extended by including buffer-based adaptation decisions for 360° videos.
2. Accurate bitrate selection for the more- and less-interesting regions based on exploration and exploitation of environmental variables plays a significant factor in optimizing the 360° video streaming service and is a fundamental one for adaptive delivery solutions. Therefore, machine learning-based bitrate selection models can be designed to make more advanced adaptation decisions involving several time-variant features, i.e., number of users, energy, mobility, bandwidth variance, buffer space, etc. This adaptation framework can then be extended for 360° video streaming considering the user viewing information to balance various QoE metrics.
3. Another research avenue that can be considered is a combined user's interest-based viewport prediction mechanism. Compared to complex content-dependent machine learning models, cluster-based prediction approaches can be suitable to enable more accurate long-term predictions and can bring considerable QoE improvement, especially in a cooperative streaming environment. The cross-users watching behavior for viewport prediction is an approach not worth ignoring to achieve long-term viewport prediction accuracy and improved viewport hit ratio.
4. A final direction of interest can be the edge-assisted adaptive transmission of traditional and 360° video services. The traditional network architecture with

centralized cloud-based computing and storage is not adequate for real-time high-bitrate content delivery. Leveraging the power of mobile edge computing can boost mobility, portability, location-awareness, effective data distribution, network context understanding, safety for service provisioning, etc. The hierarchical edge computing and proactive caching architecture can be employed for **MVP-DASH**, **CFOV**, **SFOV**, and **DFT** solutions to ensure the real-time availability of video segments to a large audience in dynamic network conditions.

5. Naturally, the QoE modeling of 360° videos relates to a broad range of technical research. Despite a growing interest in both subjective and objective quality assessment in recent years, research is still in its early stages. Challenges include learning how to define the test protocols needed during the subjective assessment for 360° videos, how to aggregate the quality measured by different users under different rating scales, how to assess the impacts of different network and content characteristics on the overall quality ratings, how to plan the test sessions to minimize the associated simulator sickness, how to assess the quality loss by considering the eye movements within the viewport, how to develop objective metrics that consider the heterogeneous quality of omnidirectional video and how to statistically analyze the effectiveness of objective quality metrics in correlation with MOS considering large-scale datasets.

Bibliography

[1] A. Zabrovskiy, C. Feldmann, and C. Timmerer, "Multi-Codec DASH Dataset," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 438443. [Online]. Available: https://doi.org/10.1145/3204949.3208140

[2] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-Degree Video Streaming Using Scalable Video Coding," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1689–1697.

[3] A. Mahzari, A. Taghavi Nasrabadi, A. Samiei, and R. Prakash, "FoV-Aware Edge Caching for Adaptive 360 Video Streaming," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 173–181.

[4] M. Graf, C. Timmerer, and C. Mueller, "Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP: Design, Implementation, and Evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 261–271. [Online]. Available: http://doi.acm.org/10.1145/3083187.3084016

[5] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Fixation Prediction for 360 Video Streaming in Head-Mounted Virtual Reality," in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2017, pp. 67–72.

[6] Cisco Systems Inc., "Cisco Visual Networking Index: Forecast and Trends, 20172022 White Paper," *Cisco Forecast and Methodology*, 2019.

[7] D. Li, R. Du, A. Babu, C. D. Brumar, and A. Varshney, "A Log-Rectilinear Transformation for Foveated 360-degree Video Streaming," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2638–2647, 2021.

[8] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021," *Cisco white paper*, 2017.

[9] N. De la Peña, P. Weil, J. Llobera, E. Giannopoulos, A. Pomés, B. Spanlang, D. Friedman, M. V. Sanchez-Vives, and M. Slater, "Immersive Journalism: Immersive Virtual Reality for the First-Person Experience of News," *Presence: Teleoperators and virtual environments*, vol. 19, no. 4, pp. 291–301, 2010.

[10] C. Cox, "The use of Computer Graphics and Virtual Reality for Visual Impact Assessments," Ph.D. dissertation, University of Nottingham, 2003.

[11] W. U. Rahman and K. Chung, "A Multi-Path-Based Adaptive Scheme for Multi-View Streaming Over HTTP," *IEEE Access*, vol. 6, pp. 77 869–77 879, 2018.

[12] S. Hollister, "Youtubes ready to blow your mind with 360-degree videos," *March 13th*, 2015.

[13] Netflix, "Internet Connection Speed Recommendations," 2014. [Online]. Available: https://help.netflix.com/en/node/306

[14] J. Thompson, J. Sun, R. Möller, M. Sintorn, and G. Huston, "Q1 2017 State of the Internet - Connectivity Report," *State of the Internet Report*, 2017.

[15] T. Nguyen and A. Zakhor, "Distributed video streaming with forward error correction," in *Packet Video Workshop*, vol. 2002, 2002.

[16] R. Puri, K. Ramchandran, K.-W. Lee, and V. Bharghavan, "Forward Error Correction (FEC) Codes based Multiple Description Coding for Internet Video Streaming and Multicast," *Signal Processing: Image Communication*, vol. 16, no. 8, pp. 745–762, 2001.

[17] S. Rane, A. Aaron, and B. Girod, "Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting-A Wyner-Ziv Coding Approach," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 5. IEEE, 2004, pp. 3101–3104.

[18] J.-W. Suh and Y.-S. Ho, "Error Concealment Techniques for Digital TV," *IEEE transactions on broadcasting*, vol. 48, no. 4, pp. 299–306, 2002.

[19] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error Resilient Video Coding Techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, 2000.

[20] Y. Zhang, W. Gao, H. Sun, Q. Huang, and Y. Lu, "Error Resilience Video Coding in H. 264 Encoder with Potential Distortion Tracking," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 1. IEEE, 2004, pp. 163–166.

[21] X. Wang and H. Schulzrinne, "Comparison of Adaptive Internet Multimedia Applications," *IEICE Transactions on Communications*, vol. 82, no. 6, pp. 806–818, 1999.

[22] D. Sisalem and F. Emanuel, "QoS Control Using Adaptive Layered Data Transmission," in *Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No. 98TB100241)*. IEEE, 1998, pp. 4–12.

[23] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, "Advances in Channel-Adaptive Video Streaming," *Wireless Communications and Mobile Computing*, vol. 2, no. 6, pp. 573–584, 2002.

[24] A. Hanjalic, "Adaptive Extraction of Highlights from a Sport Video based on Excitement Modeling," *IEEE transactions on Multimedia*, vol. 7, no. 6, pp. 1114–1122, 2005.

[25] G.-M. Muntean, "Efficient Delivery of Multimedia Streams Over Broadband Networks Using QOAS," *IEEE Transactions on Broadcasting*, vol. 52, no. 2, pp. 230–235, 2006.

[26] G.-M. Muntean, P. Perry, and L. Murphy, "Quality-oriented adaptation scheme (QOAS) for high bit-rate multimedia streaming," in *Proc. 6th International Conference on Technical Informatics (CONTI)*, vol. 4. Citeseer, 2004, pp. 199–204.

[27] ——, "A New Adaptive Multimedia Streaming System for All-IP Multi-Service Networks," *IEEE Transactions on Broadcasting*, vol. 50, no. 1, pp. 1–10, 2004.

[28] J. E. Hurtado Guzman, "Development and Implementation of an Adaptive HTTP Streaming Framework for H264/MVC Coded Media," *Politecnico di Torino*, vol. 108, 2010.

[29] T. Thang, J. Lee, J. Kang, S. Bae, S. Jung, and S. Park, "Signaling Metadata for Adaptive HTTP Streaming," *ISO/IEC JTC1/SC29/WG11 m17771, Geneva*, 2010.

[30] T. C. Nguyen and J.-H. Yun, "Predictive Tile Selection for 360-Degree VR Video Streaming in Bandwidth-Limited Networks," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1858–1861, 2018.

[31] C. Zhou, Z. Li, and Y. Liu, "A Measurement Study of Oculus 360 Degree Video Streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 2017, pp. 27–37.

[32] Y. Sánchez, R. Skupin, and T. Schierl, "Compressed Domain Video Processing for Tile based Panoramic Streaming using HEVC," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2244–2248.

[33] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-Compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 601–605.

[34] R. Aksu, J. Chakareski, and V. Swaminathan, "Viewport-Driven Rate-Distortion Optimized Scalable Live 360° Video Network Multicast," in *2018*

IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2018, pp. 1–6.

[35] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Viewport-Adaptive Encoding and Streaming of 360-Degree Video for Virtual Reality Applications," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 583–586.

[36] G. Tian and Y. Liu, "Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 109–120.

[37] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, *QDASH: A QoE-Aware DASH System*. New York, NY, USA: Association for Computing Machinery, 2012, p. 1122. [Online]. Available: https://doi.org/10.1145/2155555.2155558

[38] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados, and P. Chatzimisios, "FDASH: A Fuzzy-based MPEG/DASH Adaptation Algorithm," *IEEE Systems Journal*, vol. 10, no. 2, pp. 859–868, 2016.

[39] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[40] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming With PENSIEVE," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 197–210.

[41] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming With FESTIVE," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[42] 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," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.

[43] S. Q. Jabbar, D. J. Kadhim, and Y. Li, "Proposed an Adaptive Bitrate Algorithm based on Measuring Bandwidth and Video Buffer Occupancy for Providing Smoothly Video Streaming," *technology*, vol. 9, no. 2, 2018.

[44] Y. Hu, Y. Liu, and Y. Wang, "VAS360: QoE-Driven Viewport Adaptive Streaming for 360 Video," in *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2019, pp. 324–329.

[45] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-Aware Adaptive 360 Video Streaming using Tiles for Virtual Reality," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2174–2178.

[46] J. van der Hooft, S. Petrangeli, N. Bouten, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, "An HTTP/2 push-based approach for SVC adaptive streaming," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 104–111.

[47] A. Ghosh, V. Aggarwal, and F. Qian, "A Rate Adaptation Algorithm for Tile-based 360-degree Video Streaming," *arXiv e-prints*, p. arXiv:1704.08215, Apr. 2017.

[48] J. V. d. Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. D. Turck, "Tile-based Adaptive Streaming for Virtual Reality Video," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–24, 2019.

[49] M. Hosseini and V. Swaminathan, "Adaptive 360 VR Video Streaming: Divide and Conquer," in *Multimedia (ISM), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 107–110.

[50] K. M. Stanney, R. R. Mourant, and R. S. Kennedy, "Human Factors Issues in Virtual Environments: A Review of the Literature," *Presence:*

Teleoperators and Virtual Environments, vol. 7, no. 4, pp. 327–351, 1998. [Online]. Available: https://doi.org/10.1162/105474698565767

[51] S. Shi, V. Gupta, M. Hwang, and R. Jana, "Mobile VR on Edge Cloud: A Latency-Driven Design," in *Proceedings of the 10th ACM Multimedia Systems Conference*. ACM, 2019, pp. 222–231.

[52] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very Long Term Field of View Prediction for 360-degree Video Streaming," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 297–302.

[53] A. Nguyen, Z. Yan, and K. Nahrstedt, "Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 1190–1198.

[54] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-Degree Videos Using Super-Resolution," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 1977–1986.

[55] J. Yu and Y. Liu, "Field-of-View Prediction in 360-Degree Videos with Attention-based Neural Encoder-Decoder Networks," in *Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems*. ACM, 2019, pp. 37–42.

[56] D. V. Nguyen, H. T. Tran, A. T. Pham, and T. C. Thang, "An Optimal Tile-Based Approach for Viewport-Adaptive 360-Degree Video Streamings," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 29–42, 2019.

[57] C. Kattadige and K. Thilakarathna, "VAD360: Viewport Aware Dynamic 360-Degree Video Frame Tiling," *arXiv preprint arXiv:2105.11563*, 2021.

[58] C. Ozcinar, J. Cabrera, and A. Smolic, "Visual Attention-Aware Omnidirectional Video Streaming Using Optimal Tiles for Virtual Reality," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 217–230, March 2019.

[59] R. G. d. A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, "Visual Distortions in 360-Degree Videos," *arXiv preprint arXiv:1901.01848*, 2019.

[60] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. S. Davie, J. Wroclawski, and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks," *RFC*, vol. 2998, pp. 1–31, 2000.

[61] S. Blake, D. L. Black, M. A. Carlson, E. B. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC*, vol. 2475, pp. 1–36, 1998.

[62] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification," *RFC*, vol. 2205, pp. 1–112, 1996.

[63] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *RFC*, vol. 1889, pp. 1–75, 1996.

[64] J. Postel, "User Datagram Protocol," *RFC*, vol. 768, pp. 1–3, 1980.

[65] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," *RFC*, vol. 2326, pp. 1–92, 1998.

[66] T. Friedman, R. Cáceres, and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)," *RFC*, vol. 3611, pp. 1–55, 2003.

[67] J. Postel, "Transmission Control Protocol," *RFC*, vol. 793, pp. 1–91, 1981.

[68] I. Irondi, Q. Wang, and C. Grecos, "Empirical Evaluation of H.265/HEVC-based Dynamic Adaptive Video Streaming over HTTP (HEVC-DASH)," in *Real-Time Image and Video Processing 2014*, vol. 9139. International Society for Optics and Photonics, 2014, p. 91390L.

[69] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP –: Standards and Design Principles," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.

[70] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[71] Move Networks, "Move Networks," 2010. [Online]. Available: http://www.movenetworkshd.com

[72] A. Zambelli, "IIS Smooth Streaming Technical Overview," *Microsoft Corporation*, vol. 3, p. 40, 2009.

[73] Apple Inc, "HTTP Live Streaming Overview," 2013. [Online]. Available: https://developer.apple.com/library/ios/documentation/networkinginternet/conceptual/streamingmediaguide/Introduction/Introductoon.html

[74] "Adobe Systems Inc., HTTP Dynamic Streaming," 2013. [Online]. Available: http://www.adobe.com/products/hds-dynamic-streaming.html

[75] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46–59, 1991.

[76] T. Bi, A. Pichon, L. Zou, S. Chen, G. Ghinea, and G.-M. Muntean, "A DASH-based Mulsemedia Adaptive Delivery Solution," in *Proceedings of the 10th International Workshop on Immersive Mixed and Virtual Environment Systems*. ACM, 2018, pp. 1–6.

[77] T. Lohmar, T. Einarsson, P. Fröjdh, F. Gabin, and M. Kampmann, "Dynamic Adaptive HTTP Streaming of Live Content," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 2011, pp. 1–8.

[78] R. Castagno and D. Singer, "MIME Type Registrations for 3rd Generation Partnership Project (3GPP) Multimedia files," *RFC*, vol. 3839, pp. 1–7, 2004.

[79] H. Garudadri, "MIME Type Registrations for 3GPP2 Multimedia Files," *RFC*, vol. 4393, pp. 1–7, 2006.

[80] V. Swaminathan, "Are We in the Middle of a Video Streaming Revolution?" *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 9, no. 1s, pp. 1–6, 2013.

[81] D. Salomon, *Transformations and Projections in Computer Graphics*. Springer Science & Business Media, 2007.

[82] E. Kuzyakov and D. Pio, "Next-Generation Video Encoding Techniques for 360 Video and VR," 2016. [Online]. Available: https://code.facebook.com/posts/1126354007399553/nextgeneration-video-encoding-techniques-for-360-video-and-vr

[83] G. Van der Auwera, H. M. Coban, and M. Karczewicz, "Truncated Square Pyramid Projection (TSP) For 360 Video, JVET Doc," *D0071*, 2016.

[84] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the Edge: How to Deliver 360 Videos in Mobile Networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2017, pp. 30–35.

[85] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand *et al.*, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[86] V. Sze, M. Budagavi, and G. J. Sullivan, "High Efficiency Video Coding (HEVC)," *Integrated Circuit and Systems, Algorithms and Architectures. Springer*, vol. 39, p. 40, 2014.

[87] G. J. Sullivan, "Video Coding Standards Progress Report: Joint Video Experts Team Launches the Versatile Video Coding Project," *SMPTE Motion Imaging Journal*, vol. 127, no. 8, pp. 94–98, 2018.

[88] V. Avelar, "Practical Options for Deploying Small Server Rooms and Micro Data Centers," *Schneider electric, white paper 174*, 2015.

[89] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and its Role in the Internet of Things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[90] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing _ A Key Technology Towards 5G," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[91] A. Manzalini and N. Crespi, "An Edge Operating System Enabling Anything-as-a-Service," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 62–67, 2016.

[92] L. Lin, X. Liao, H. Jin, and P. Li, "Computation Offloading Toward Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019.

[93] D. Antonogiorgakis, A. Britzolakis, P. Chatziadam, A. Dimitriadis, S. Gikas, E. Michalodimitrakis, M. Oikonomakis, N. Siganos, E. Tzagkarakis, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A View on Edge caching Applications," *arXiv e-prints*, p. arXiv:1907.12359, Jul. 2019.

[94] S.-Y. Lien, S.-C. Hung, H. Hsu, and D.-J. Deng, "Energy-Optimal Edge Content Cache and Dissemination: Designs for Practical Network Deployment," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 88–93, 2018.

[95] M. Yan, C. A. Chan, W. Li, L. Lei, Q. Shuai, A. F. Gygax, and I. Chih-Lin, "Assessing the Energy Consumption of 5G Wireless Edge Caching," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[96] Q.-V. Pham, F. Fang, V. Nguyen Ha, M. Jalil Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art," *arXiv e-prints*, p. arXiv:1906.08452, Jun. 2019.

[97] Fraunhofer FOKUS, "Cloud-based 360° Video Playout," 2016, Retrieved 10. [Online]. Available: https://www.fokus.fraunhofer.de/go/360

[98] L. Sun, Y. Mao, T. Zong, Y. Liu, and Y. Wang, "Flocking-based Live Streaming of 360-Degree Video," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 26–37.

[99] O. Eltobgy, O. Arafa, and M. Hefeeda, "Mobile Streaming of Live 360-Degree Videos," *IEEE Transactions on Multimedia*, 2020.

[100] X. Feng, Y. Liu, and S. Wei, "LiveDeep: Online Viewport Prediction for Live Virtual Reality Streaming Using Lifelong Deep Learning," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2020, pp. 800–808.

[101] X. Liu, B. Han, F. Qian, and M. Varvello, "LIME: Understanding Commercial 360° Live Video Streaming Services," in *Proceedings of the 10th ACM Multimedia Systems Conference*. ACM, 2019, pp. 154–164.

[102] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "A Two-Tier System for On-Demand Streaming of 360 Degree Video over Dynamic Networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 43–57, 2019.

[103] J. Song, F. Yang, W. Zhang, W. Zou, Y. Fan, and P. Di, "A Fast FoV-Switching DASH System Based on Tiling Mechanism for Practical Omnidirectional Video Services," *IEEE Transactions on Multimedia*, 2019.

[104] Y. Mao, L. Sun, Y. Liu, and Y. Wang, "Low-latency FoV-adaptive Coding and Streaming for Interactive 360° Video Streaming," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3696–3704.

[105] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 Video Delivery Over Cellular Networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[106] X. Chen, D. Wu, and I. Ahmad, "Optimized Viewport-adaptive 360-degree Video Streaming," *CAAI Transactions on Intelligence Technology*, 2021.

[107] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An HTTP/2-based Adaptive Streaming Framework for 360 Virtual Reality Videos," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 306–314.

[108] X. Hu, W. Quan, T. Guo, Y. Liu, and L. Zhang, "Mobile Edge Assisted Live Streaming System for Omnidirectional Video," *Mobile Information Systems*, vol. 2019, 2019.

[109] C. Timmerer, "Immersive Media Delivery: Overview of Ongoing Standardization Activities," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 71–74, 2017.

[110] ISO, "ISO/IEC 13818-1:2018 Generic Coding of Moving Pictures and Associated Audio Information, Part 1: Systems," *ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information*, 2018.

[111] D. Singer, W. Belknap, and G. Franceschini, "ISO/IEC 14496-14: Information Technology Coding of Audio-Visual Objects Part 14: MP4 File Format," *Final Draft International Standard (FDIS)*, 2 2004.

[112] Y. Lim, K. Park, J. Y. Lee, S. Aoki, and G. Fernando, "MMT: An Emerging MPEG Standard for Multimedia Delivery over the Internet," *IEEE MultiMedia*, vol. 20, no. 1, pp. 80–85, 2013.

[113] B. Choi, Y. Wang, M. Hannuksela, Y. Lim, and A. Murtaza, "Study of ISO/IEC DIS 23000-20 Omnidirectional Media Format," *ISO/IEC JTCl/SC29/WG11, Torino, Italy, Tech. Rep. N16950*, 2017.

[114] S. Oh and S. Hwang, "OMAF: Generalized Signaling of Region-wise Packing for Omnidirectional Video," in *118th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG2017/m40423*, 2017.

[115] R. Skupin, Y. Sanchez, Y.-K. Wang, M. Hannuksela, J. Boyce, and M. Wien, "Standardization Status of 360 Degree Video Coding and Delivery," in *Visual*

Communications and Image Processing (VCIP), 2017 IEEE. IEEE, 2017, pp. 1–4.

[116] M. M. Hannuksela, Y. Wang, and A. Hourunranta, "An Overview of the OMAF Standard for 360° Video," in *2019 Data Compression Conference (DCC)*, March 2019, pp. 418–427.

[117] K. Hughes and D. Singer, "Information Technology – Multimedia Application Format (MPEG-A) – Part 19: Common Media Application Format (CMAF) for Segmented Media, ISO/IEC 23000-19," 2017.

[118] D. Singer, "Information Technology – Coding of Audio-Visual ObjectsPart 12: ISO Base Media File Format, ISO/IEC 14496-12:2005," 2005.

[119] R. Doré, J. Fleureau, B. Chupeau, and G. Briand, "3DoF Plus Intermediate View Synthesizer Proposal," in *122nd MPEG meeting of ISO/IEC JTC1/WG11, MPEG2018/m42486*, 2018.

[120] 3GPP, "3GPP TR 126 918: Virtual Reality (VR) Media Services Over 3GPP," *Computer Science and Communications Dictionary*, 2018.

[121] M. S. Brennesholtz, "VR Standards and Guidelines," in *SID Symposium Digest of Technical Papers*, vol. 49. Wiley Online Library, 2018, pp. 1–4.

[122] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an Efficient Representation of Visual Reality," *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, Oct 2016.

[123] T. Richter, "On the Standardization of the JPEG XT Image Compression," in *2013 Picture Coding Symposium (PCS)*. IEEE, 2013, pp. 37–40.

[124] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "JVET Common Test Conditions and Evaluation Procedures for 360 Video," *Joint Video Exploration Team of ITU-T SG*, vol. 16, 2017.

[125] ITU-T: Rec., "P. 800: Methods for Subjective Determination of Transmission Quality," *International Telecommunication Union, Geneva*, vol. 22, 1996.

[126] ITU-T Recommendation P.910, "Subjective Video Quality Assessment Methods for Multimedia Applications," *International Telecommunication Union, Geneva*, 1999.

[127] ITU-T Recommendation P.911, "Subjective Audiovisual Quality Assessment Methods for Multimedia Applications," 1998.

[128] ITU-R Recommandation BT.500-11, "Methodology for the Subjective Assessment of the Quality of Television Pictures," *International Telecommunication Union*, 2002.

[129] ITU-T P. 913, "Methods for the Subjective Assessment of Video Quality Audio Quality and Audiovisual Quality of Internet Video and Distribution Quality Television in any Environment," *SERIES P: TERMINALS AND SUBJECTIVE AND OBJECTIVE ASSESSMENT METHODS*, 2016.

[130] ITU-T RECOMMENDATION, P.920, "Interactive Test Methods for Audiovisual Communications," 2000.

[131] A. S. Fernandes and S. K. Feiner, "Combating VR Sickness Through Subtle Dynamic Field-of-View Modification," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2016, pp. 201–210.

[132] ITU-T Recommendation P. 919., "Subjective Test Methodologies for 360 Degree Video on HMD," 2020.

[133] Z. Wang and Q. Li, "Video Quality Assessment using a Statistical Model of Human Visual Speed Perception," *JOSA A*, vol. 24, no. 12, pp. B61–B69, 2007.

[134] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale Structural Similarity for Image Quality Assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[135] M. Yu, H. Lakshman, and B. Girod, "A Framework to Evaluate Omnidirectional Video Coding Schemes," in *2015 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2015, pp. 31–36.

[136] S. Chen, Y. Zhang, Y. Li, Z. Chen, and Z. Wang, "Spherical Structural Similarity Index for Objective Omnidirectional Video Quality Assessment," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, July 2018, pp. 1–6.

[137] Y. Sun, A. Lu, and L. Yu, "Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video," *IEEE signal processing letters*, vol. 24, no. 9, pp. 1408–1412, 2017.

[138] Y. Zhou, M. Yu, H. Ma, H. Shao, and G. Jiang, "Weighted-to-Spherically-Uniform SSIM Objective Quality Evaluation for Panoramic Video," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2018, pp. 54–57.

[139] V. Zakharchenko, K. P. Choi, and J. H. Park, "Quality Metric for Spherical Panoramic Video," in *Optics and Photonics for Information Processing X*, vol. 9970. International Society for Optics and Photonics, 2016, p. 99700C.

[140] J. van der Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. De Turck, "Quality Assessment for Adaptive Virtual Reality Video Streaming: A Probabilistic Approach on the User's Gaze," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2019, pp. 19–24.

[141] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea," in *2012 Fourth International Workshop on Quality of Multimedia Experience*. IEEE, 2012, pp. 1–6.

[142] Y. Qi and M. Dai, "The Effect of Frame Freezing and Frame Skipping on Video Quality," in *2006 international conference on intelligent information hiding and multimedia*. IEEE, 2006, pp. 423–426.

[143] T. N. Minhas and M. Fiedler, "Impact of Disturbance Locations on Video Quality of Experience," in *EuroITV 2011 Workshop: Quality of Experience*

for Multimedia Content Sharing. EuroITV 2011 Workshop: Quality of Experience for Multimedia Content Sharing, 2011.

[144] L. De Cicco, S. Mascolo, and V. Palmisano, *Feedback Control for Adaptive Live Video Streaming.* New York, NY, USA: Association for Computing Machinery, 2011, p. 145156. [Online]. Available: https://doi.org/10.1145/1943552.1943573

[145] Z. Yuan and G.-M. Muntean, "A Prioritized Adaptive Scheme for Multimedia Services over IEEE 802.11 WLANs," *IEEE Transactions on Network and Service Management*, vol. 10, no. 4, pp. 340–355, 2013.

[146] L. Zou, R. Trestian, and G.-M. Muntean, "DOAS: Device-Oriented Adaptive Multimedia Scheme for 3GPP LTE Systems," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC).* IEEE, 2013, pp. 2180–2184.

[147] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. M. Batalla, and E. Borcoci, "MS-Stream: A Multiple-Source Adaptive Streaming Solution Enhancing Consumer's Perceived Quality," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017, pp. 427–434.

[148] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 562–585, Firstquarter 2019.

[149] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," in *Proceedings of the second annual ACM conference on Multimedia systems.* ACM, 2011, pp. 169–174.

[150] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, "Rate Adaptation for Dynamic Adaptive Streaming Over HTTP in Content Distribution Network," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 288–311, 2012.

[151] Adobe, "OSMF," 2012. [Online]. Available: https://sourceforge.net/projects/osmf.adobe/

[152] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984.

[153] 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 Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, 2014.

[154] 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 *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 2016, pp. 272–285.

[155] J. Jiang, V. Sekar, H. Milner, D. Shepherd, I. Stoica, and H. Zhang, "CFA: A Practical Prediction System for Video QoE Optimization," in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016, pp. 137–150.

[156] J. Jiang, S. Sun, V. Sekar, and H. Zhang, "Pytheas: Enabling Data-Driven Quality of Experience Optimization Using Group-based Exploration-Exploitation," in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 393–406.

[157] M. Xiao, V. Swaminathan, S. Wei, and S. Chen, "Dash2m: Exploring HTTP/2 for Internet Streaming to Mobile Devices," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 22–31.

[158] W. ur Rahman and K. Chung, "A Novel Adaptive Logic for Dynamic Adaptive Streaming over HTTP," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 433–446, 2017.

[159] C. Mueller, S. Lederer, R. Grandl, and C. Timmerer, "Oscillation Compen-

sating Dynamic Adaptive Streaming over HTTP," in *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[160] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, "ABMA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 2.

[161] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. IEEE, 2013, pp. 1318–1323.

[162] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying QoE Optimal Adaptation of HTTP Adaptive Streaming based on Subjective Studies," *Computer Networks*, vol. 81, pp. 320–332, 2015.

[163] A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[164] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming," *Proceedings of the ACM Multimedia Conference*, pp. 315–323, 2017. [Online]. Available: 10.1145/3123266.3123291

[165] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-Adaptive Navigable 360-Degree Video Delivery," in *2017 IEEE international conference on communications (ICC)*. IEEE, 2017, pp. 1–7.

[166] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *ACM SIGCOMM Computer Communication Review*, vol. 45. ACM, 2015, pp. 325–338.

[167] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation Algorithm for Adaptive Streaming over HTTP," in *2012 19th International Packet Video Workshop (PV)*, May 2012, pp. 173–178.

[168] A. Sobhani, A. Yassine, and S. Shirmohammadi, "A Video Bitrate Adaptation and Prediction Mechanism for HTTP Adaptive Streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 2, p. 18, 2017.

[169] P. Kaufman, *Smarter Trading*. New York: McGraw-Hill, 1995, vol. 22.

[170] S. Liu and J. Y. L. Forrest, *Grey Systems: Theory and Applications*. Springer Science & Business Media, 2010.

[171] C. Wang, A. Rizk, and M. Zink, "SQUAD: A Spectrum-based Quality Adaptation for Dynamic Adaptive Streaming over HTTP," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 1.

[172] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "TFDASH: A Fairness, Stability, and Efficiency Aware Rate Control Approach for Multiple Clients over DASH," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[173] B. Friedland, "Steady-state behavior of kalman filter with discrete-and continuous-time observations," *IEEE Transactions on Automatic Control*, vol. 25, no. 5, pp. 988–992, 1980.

[174] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," *RFC*, vol. 6824, pp. 1–64, 2013.

[175] B. Hesmans, F. Duchene, C. Paasch, G. Detal, and O. Bonaventure, "Are TCP Extensions Middlebox-proof?" in *Proceedings of the 2013 workshop on Hot topics in middleboxes and network function virtualization*. ACM, 2013, pp. 37–42.

[176] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: Quality-Aware Adaptive Concurrent Multipath Data Transfer in Heterogeneous Wireless Networks," *IEEE transactions on mobile computing*, vol. 12, no. 11, pp. 2193–2205, 2012.

[177] Y.-C. Chen, D. Towsley, and R. Khalili, "MSPlayer: Multi-Source and Multi-Path Video Streaming," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2198–2206, 2016.

[178] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-Layer Fairness-Driven Concurrent Multipath Video Delivery Over Heterogeneous Wireless Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1175–1189, July 2015.

[179] Y. Kim and K. Chung, "Multipath-based Transmission Scheme for Improving the QoE of HTTP Adaptive Streaming," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 12–20, 2018.

[180] A. Elgabli, K. Liu, and V. Aggarwal, "Optimized Preference-Aware Multi-Path Video Streaming with Scalable Video Coding," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 159–172, 2020.

[181] F. Silva, D. Bogusevschi, and G. Muntean, "A MPTCP-based RTT-aware Packet Delivery Prioritisation Algorithm in AR/VR Scenarios," in *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2018, pp. 95–100.

[182] M. Kheirkhah, I. Wakeman, and G. Parisis, "Multipath-TCP in NS-3," *CoRR*, vol. abs/1510.07721, 2015. [Online]. Available: http://arxiv.org/abs/1510.07721

[183] Y. Go, O. C. Kwon, and H. Song, "An Energy-Efficient HTTP Adaptive Video Streaming with Networking Cost Constraint over Heterogeneous Wireless Networks," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1646–1657, 2015.

[184] K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. Hansen, and P. Engelstad, "Improving the Performance of Quality-Adaptive Video Streaming over Multiple Heterogeneous Access Networks," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 57–68.

[185] W. Zhang, S. Ye, B. Li, H. Zhao, and Q. Zheng, "A Priority-based Adaptive Scheme for Multi-view Live Streaming over HTTP," *Computer Communications*, vol. 85, pp. 89–97, 2016.

[186] D. Yun and K. Chung, "DASH-Based Multi-View Video Streaming System," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1974–1980, Aug 2018.

[187] X. Zhang, L. Toni, P. Frossard, Y. Zhao, and C. Lin, "Adaptive Streaming in Interactive Multiview Video Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1130–1144, 2018.

[188] K. Bilal, A. Erbad, and M. Hefeeda, "Crowdsourced Multi-View Live Video Streaming using Cloud Computing," *IEEE Access*, vol. 5, pp. 12 635–12 647, 2017.

[189] S. Afzal, J. Chen, and K. Ramakrishnan, "Characterization of 360-Degree Videos," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2017, pp. 1–6.

[190] E. Domínguez, "Going Beyond the Classic News Narrative Convention: The background to and challenges of immersion in journalism," *Frontiers in digital humanities*, vol. 4, p. 10, 2017.

[191] S. Pérez-Seijo and B. García-Orosa, "Use of 360-Degree Video in Organizational Communication: Case Study of Humanitarian Aid NGOs," in *Journalistic metamorphosis*. Springer, 2020, pp. 99–111.

[192] M. Gänsluckner, M. Ebner, and I. Kamrat, "360 Degree Videos within a Climbing MOOC." *International Association for Development of the Information Society*, 2017.

[193] A. Hebbel-Seeger, "360 Degrees Video and VR for Training and Marketing within Sports," *Athens Journal of Sports*, vol. 4, no. 4, pp. 243–261, 2017.

[194] J. Reyna Zeballos, "The Potential of 360-Degree Videos for Teaching, Learning

and Research." in *The 12th annual International Technology, Education and Development Conference.* INTED, 2018.

[195] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "Viewport-Dependent 360 Degree Video Streaming based on the Emerging Omnidirectional Media Format (OMAF) Standard," in *2017 IEEE International Conference on Image Processing (ICIP).* IEEE, 2017, pp. 4592–4592.

[196] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Optimal Set of 360-Degree Videos for Viewport-Adaptive Streaming," in *Proceedings of the 2017 ACM on Multimedia Conference.* ACM, 2017, pp. 943–951.

[197] T. Biatek, W. Hamidouche, P.-L. Cabarat, J.-F. Travers, and O. Dforges, "Scalable Video Coding for Backward-Compatible 360° Video Delivery Over Broadcast Networks," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 322–332, 2020.

[198] D. Naik, I. D. Curcio, and H. Toukomaa, "Optimized Viewport Dependent Streaming of Stereoscopic Omnidirectional Video," in *Proceedings of the 23rd Packet Video Workshop.* ACM, 2018, pp. 37–42.

[199] T. El-Ganainy and M. Hefeeda, "Streaming Virtual Reality Content," *arXiv e-prints*, p. arXiv:1612.08350, Dec. 2016.

[200] D. He, C. Westphal, and J. J. Garcia-Luna-Aceves, "Network Support for AR/VR and Immersive Video Application: A Survey," in *ICETE*, 2018.

[201] M. Xu, C. Li, S. Zhang, and P. Le Callet, "State-of-the-art in 360 Video/Image Processing: Perception, Assessment and Compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 5–26, 2020.

[202] M. Zink, R. K. Sitaraman, and K. Nahrstedt, "Scalable 360° Video Stream Delivery: Challenges, Solutions, and Opportunities," *Proceedings of the IEEE*, vol. 107, pp. 639–650, 2019.

[203] R. G. d. A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli,

and P. Frossard, "Visual Distortions in 360° Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019.

[204] X. Zhang, X. Hu, L. Zhong, S. Shirmohammadi, and L. Zhang, "Cooperative Tile-based 360-degree Panoramic Streaming in Heterogeneous Networks using Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[205] D. V. Nguyen, H. Van Trung, H. L. Dieu Huong, T. T. Huong, N. P. Ngoc, and T. C. Thang, "Scalable 360 Video Streaming Using HTTP/2," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2019, pp. 1–6.

[206] R. Schatz, A. Sackl, C. Timmerer, and B. Gardlo, "Towards Subjective Quality of Experience Assessment for Omnidirectional Video Streaming," in *Proc. 9th Int. Conf. Qual. Multimedia Exp.(QoMEX)*, 2017, pp. 1–6.

[207] T. Ogasawara and M. Bandai, "Buffer Based Adaptation Using Scalable Video Coding for 360-Degree Video Streaming over NDN," in *2020 International Conference on Information Networking (ICOIN)*. IEEE, 2020, pp. 142–145.

[208] J. Son, D. Jang, and E.-S. Ryu, "Implementing Motion-Constrained Tile and Viewport Extraction for VR Streaming," in *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2018, pp. 61–66.

[209] H. Ahmadi, O. Eltobgy, and M. Hefeeda, "Adaptive Multicast Streaming of Virtual Reality Content to Mobile Users," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. ACM, 2017, pp. 170–178.

[210] H.-W. Kim and S.-H. Yang, "Region of Interest–based Segmented Tiled Adaptive Streaming Using Head-Mounted Display Tracking Sensing Data," *International Journal of Distributed Sensor Networks*, vol. 15, no. 12, p. 1550147719894533, 2019.

[211] H. W. Kim, J. W. Yang, J. Y. Yang, J. H. Jang, and W. C. Park, "MPEG-DASH SRD based 360 VR Tiled Streaming System for Foveated Rendering," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 587–591.

[212] J. van der Hooft, S. Petrangeli, M. Claeys, J. Famaey, and F. De Turck, "A Learning-based Algorithm for Improved Bandwidth-Awareness of Adaptive Streaming Clients," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 131–138.

[213] J. Fu, X. Chen, Z. Zhang, S. Wu, and Z. Chen, "360SRL: A Sequential Reinforcement Learning Approach for ABR Tile-Based 360 Video Streaming," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 290–295.

[214] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze Prediction in Dynamic 360° Immersive Videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 5333–5342.

[215] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360° Video Viewing Dataset in Head-Mounted Virtual Reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 211–216.

[216] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "DRL360: 360-degree Video Streaming with Deep Reinforcement Learning," in *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2019, pp. 1252–1260.

[217] S. Sukhmani, M. Sadeghi, M. Erol-Kantarci, and A. El Saddik, "Edge Caching and Computing in 5G for Mobile AR/VR and Tactile Internet," *IEEE MultiMedia*, vol. 26, no. 1, pp. 21–30, 2018.

[218] K. Bilal and A. Erbad, "Edge Computing for Interactive Media and Video Streaming," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2017, pp. 68–73.

[219] M. Erol-Kantarci and S. Sukhmani, "Caching and Computing at the Edge for Mobile Augmented Reality and Virtual Reality (AR/VR) in 5G," in *Ad Hoc Networks*, Y. Zhou and T. Kunz, Eds. Cham: Springer International Publishing, 2018, pp. 169–177.

[220] M. Viitanen, J. Vanne, T. D. Hämäläinen, and A. Kulmala, "Low Latency Edge Rendering Scheme for Interactive 360 Degree Virtual Reality Gaming," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1557–1560.

[221] W.-C. Lo, C.-Y. Huang, and C.-H. Hsu, "Edge-Assisted Rendering of 360° Videos Streamed to Head-Mounted Virtual Reality," in *2018 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2018, pp. 44–51.

[222] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and Operator Cooperation Policies for Layered Video Content Delivery," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[223] J. Poderys, M. Artuso, C. M. O. Lensbøl, H. L. Christiansen, and J. Soler, "Caching at the Mobile Edge: A Practical Implementation," *Ieee Access*, vol. 6, pp. 8630–8637, 2018.

[224] G. Li, Q. Shen, Y. Liu, H. Cao, Z. Han, F. Li, and J. Li, "Data-Driven Approaches to Edge Caching," in *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*. ACM, 2018, pp. 8–14.

[225] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 193–198.

[226] G. Papaioannou and I. Koutsopoulos, "Tile-based Caching Optimization for 360° Videos," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2019, pp. 171–180.

[227] J. Zhang, Q. Gao, and G. Zhang, "Edge Cache Replacement Strategy for SVC-Encoding Tile-Based 360-degree Panoramic Streaming," in *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2020, pp. 122–128.

[228] L. Liu, R. Zhong, W. Zhang, Y. Liu, J. Zhang, L. Zhang, and M. Gruteser, "Cutting the Cord: Designing a High-Quality Untethered VR System with Low Latency Remote Rendering," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2018, pp. 68–80.

[229] X. Corbillon, F. De Simone, and G. Simon, "360-Degree Video Head Movement Dataset," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 199–204.

[230] M. Ben Yahia, Y. Le Louedec, G. Simon, and L. Nuaymi, "HTTP/2-Based Streaming Solutions for Tiled Omnidirectional Videos," in *2018 IEEE International Symposium on Multimedia (ISM)*, Dec 2018, pp. 89–96.

[231] J. Zou, C. Li, C. Liu, Q. Yang, H. Xiong, and E. Steinbach, "Probabilistic Tile Visibility-based Server-Side Rate Adaptation for Adaptive 360-Degree Video Streaming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 161–176, 2019.

[232] X. Jiang, Y.-H. Chiang, Y. Zhao, and Y. Ji, "Plato: Learning-based Adaptive Streaming of 360-Degree Videos," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. IEEE, 2018, pp. 393–400.

[233] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[234] D. V. Nguyen, H. T. Tran, and T. C. Thang, "Impact of Delays on 360-degree Video Communications," in *TRON Symposium (TRONSHOW), 2017*. IEEE, 2017, pp. 1–6.

[235] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan, "Mixing Tile Resolutions in Tiled Video: A Perceptual Quality Assessment," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. ACM, 2014, p. 25.

[236] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. Kwong, "Spatial and Temporal Consistency-Aware Dynamic Adaptive Streaming for 360-Degree Videos," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 177–193, 2019.

[237] Y. Bao, T. Zhang, A. Pande, H. Wu, and X. Liu, "Motion-Prediction-based Multicast for 360-Degree Video Transmissions," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2017, pp. 1–9.

[238] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a Moving Target: Motion-Prediction-based Transmission for 360-Degree Videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[239] N. Kan, J. Zou, K. Tang, C. Li, N. Liu, and H. Xiong, "Deep Reinforcement Learning-based Rate Adaptation for Adaptive 360-Degree Video Streaming," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4030–4034.

[240] D. He, C. Westphal, and J. Garcia-Luna-Aceves, "Joint Rate and FoV Adaptation in Immersive Video Streaming," in *ACM Sigcomm workshop on AR/VR Networks*, 2018.

[241] L. D'Acunto, J. van den Berg, E. Thomas, and O. Niamut, "Using MPEG DASH SRD for Zoomable and Navigable Video," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 34.

[242] P. Maniotis, E. Bourtsoulatze, and N. Thomos, "Tile-Based Joint Caching and Delivery of 360° Videos in Heterogeneous Networks," *2019 IEEE 21st*

International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6, 2019.

[243] Y. Li and W. Gao, "MUVR: Supporting Multi-User Mobile Virtual Reality with Resource Constrained Edge Cloud," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 1–16.

[244] C. Guo, Y. Cui, and Z. Liu, "Optimal Multicast of Tiled 360 VR Video," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 145–148, 2018.

[245] K. Long, C. Ye, Y. Cui, and Z. Liu, "Optimal Multi-Quality Multicast for 360 Virtual Reality Video," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[246] D. V. Nguyen, H. T. Tran, and T. C. Thang, "Adaptive Tiling Selection for Viewport Adaptive Streaming of 360-Degree Video," *IEICE Transactions on Information and Systems*, vol. 102, no. 1, pp. 48–51, 2019.

[247] C. Zhou, M. Xiao, and Y. Liu, "ClusTile: Toward Minimizing Bandwidth in 360-degree Video Streaming," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018, pp. 962–970.

[248] W. Quan, Y. Pan, B. Xiang, and L. Zhang, "Reinforcement Learning Driven Adaptive VR Streaming with Optical Flow Based QoE," *arXiv e-prints*, p. arXiv:2003.07583, Mar. 2020.

[249] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "HEVC Tile based Streaming to Head Mounted Displays," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 613–615.

[250] Y. Sanchez, G. S. Bhullar, R. Skupin, C. Hellge, and T. Schierl, "Delay Impact on MPEG OMAF's Tile-based Viewport-Dependent 360° Video Streaming," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2019.

[251] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices," in *Proceedings of*

the 24th Annual International Conference on Mobile Computing and Networking. ACM, 2018, pp. 99–114.

[252] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "Cub360: Exploiting Cross-users Behaviors for Viewport Prediction in 360 Video Adaptive Streaming," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[253] S. Petrangeli, G. Simon, and V. Swaminathan, "Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos," in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2018, pp. 157–160.

[254] S. Rossi, F. De Simone, P. Frossard, and L. Toni, "Spherical Clustering of Users Navigating 360° Content," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4020–4024.

[255] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive View Generation to Enable Mobile 360-degree and VR Experiences," in *Proceedings of the 2018 Morning Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2018, pp. 20–26.

[256] C. Bron and J. Kerbosch, "Algorithm 457: Finding All Cliques of an Undirected Graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[257] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[258] X. Hou, J. Zhang, M. Budagavi, and S. Dey, "Head and Body Motion Prediction to Enable Mobile VR Experiences with Low Latency," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–7.

[259] F. Duanmu, Y. Mao, S. Liu, S. Srinivasan, and Y. Wang, "A Subjective Study of Viewer Navigation Behaviors When Watching 360-Degree Videos

on Computers," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[260] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov, and A. Kondoz, "Predicting Head Trajectories in 360 Virtual Reality Videos," in *2017 International Conference on 3D Immersion (IC3D)*. IEEE, 2017, pp. 1–6.

[261] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2693–2708, Nov 2019.

[262] Q. Yang, J. Zou, K. Tang, C. Li, and H. Xiong, "Single and Sequential Viewports Prediction for 360-Degree Video Streaming," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[263] J. Fu, Z. Chen, X. Chen, and W. Li, "Sequential Reinforced 360-Degree Video Adaptive Streaming With Cross-User Attentive Network," *IEEE Transactions on Broadcasting*, 2020.

[264] M. Fabian Romero Rondon, L. Sassatelli, R. Aparicio Pardo, and F. Precioso, "Revisiting Deep Architectures for Head Motion Prediction in 360° Videos," *arXiv e-prints*, p. arXiv:1911.11702, Nov. 2019.

[265] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. McMahan, R. Prakash, M. C. Farias, and M. M. Carvalho, "A Taxonomy and Dataset for 360° Videos," in *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019, pp. 273–278.

[266] G.-M. Muntean and N. Cranley, "Resource Efficient Quality-Oriented Wireless Broadcasting of Adaptive Multimedia Content," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 362–368, 2007.

[267] P. Xuan-Tan, E. Kamioka *et al.*, "Early Detection of QoE Deterioration in HTTP Adaptive Streaming," *Journal of Computer and Communications*, vol. 5, no. 14, p. 14, 2017.

[268] S. Egger, B. Gardlo, M. Seufert, and R. Schatz, "The Impact of Adaptation Strategies on Perceived Quality of HTTP Adaptive Streaming," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, 2014, pp. 31–36.

[269] Y. Shen, Y. Liu, H. Yang, and D. Yang, "Quality of Experience Study on Dynamic Adaptive Streaming based on HTTP," *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 62–70, 2015.

[270] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth?" in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*, 2012, pp. 9–14.

[271] 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 Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, 2012.

[272] R. I. T. da Costa Filho, M. C. Luizelli, M. T. Vega, J. van der Hooft, S. Petrangeli, T. Wauters, F. D. Turck, and L. P. Gaspary, "Predicting the Performance of Virtual Reality Video Streaming in Mobile Networks," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 270–283.

[273] N. Kan, J. Zou, C. Li, W. Dai, and H. Xiong, "RAPT360: Reinforcement Learning-Based Rate Adaptation for 360-degree Video Streaming with Adaptive Prediction and Tiling," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021.

[274] A. Yaqoob, T. Bi, and G. M. Muntean, "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[275] N. Eswara, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "Perceptual QoE-Optimal Resource Allocation for Adaptive

Video Streaming," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 346–358, 2020.

[276] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Rubiks: Practical 360-Degree Streaming for Smartphones," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. New York, NY, USA: ACM, 2018, pp. 482–494.

[277] S. Park, A. Bhattacharya, Z. Yang, S. R. Das, and D. Samaras, "Mosaic: Advancing User Quality of Experience in 360-Degree Video Streaming With Machine Learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 1000–1015, 2021.

[278] L. Xie, X. Zhang, and Z. Guo, "CLS: A Cross-User Learning Based System for Improving QoE in 360-Degree Video Adaptive Streaming," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 564572. [Online]. Available: https://doi.org/10.1145/3240508.3240556

[279] N. Barman, N. Khan, and M. G. Martini, "Analysis of Spatial and Temporal Information Variation for 10-bit and 8-bit Video Sequences," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2019, pp. 1–6.

[280] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Hämäläinen, "Kvazaar: Open-Source HEVC/H.265 Encoder," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016. [Online]. Available: http://doi.acm.org/10.1145/2964284.2973796

[281] J.-B. Jeong, S. Lee, D. Jang, and E.-S. Ryu, "Towards 3DoF+ 360 Video Streaming System for Immersive Media," *IEEE Access*, vol. 7, pp. 136 399–136 408, 2019.

[282] Y. Zhu, G. Zhai, X. Min, and J. Zhou, "Learning a Deep Agent to Predict Head Movement in 360-Degree Images," *ACM Transactions on Multimedia*

Computing, Communications, and Applications (TOMM), vol. 16, no. 4, pp. 1–23, 2020.

[283] J. Li, L. Han, C. Zhang, Q. Li, and W. Li, "Adaptive Panoramic Video Multicast Streaming with Limited FoV Feedback," *Complexity*, vol. 2020, 2020.

[284] C. Moldovan, F. Loh, M. Seufert, and T. Hoßfeld, "Optimizing HAS for 360-Degree Videos," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–6.

[285] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards Foveated Rendering for Gaze-Tracked Virtual Reality," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 179, 2016.

[286] P. Lungaro, R. Sjöberg, A. J. F. Valero, A. Mittal, and K. Tollmar, "Gaze-Aware Streaming Solutions for the Next Generation of Mobile VR Experiences," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 4, pp. 1535–1544, 2018.

[287] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, "An Overview of Tiles in HEVC," *IEEE journal of selected topics in signal processing*, vol. 7, no. 6, pp. 969–977, 2013.

[288] C. Concolato, J. Le Feuvre, F. Denoual, F. Mazé, E. Nassor, N. Ouedraogo, and J. Taquet, "Adaptive Streaming of HEVC Tiled Videos Using MPEG-DASH," *IEEE transactions on circuits and systems for video technology*, vol. 28, no. 8, pp. 1981–1992, 2017.

[289] E. Upenik and T. Ebrahimi, "A Simple Method to Obtain Visual Attention Data in Head Mounted Virtual Reality," in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2017, pp. 73–78.

[290] A. T. Nasrabadi, A. Samiei, and R. Prakash, "Viewport Prediction for 360° Videos: A Clustering Approach," in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2020, pp. 34–39. [Online]. Available: https://doi.org/10.1145/3386290.3396934