

E2E Service Performance Enhancement for Tile-based Adaptive 360° VR Video Streaming

by

Yannan Wei

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Yannan Wei 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Xianbin Wang
Professor, Dept. of Electrical and Computer Engineering
Western University

Supervisor(s): Weihua Zhuang
Professor, Dept. of Electrical and Computer Engineering
University of Waterloo

Internal Member: Patrick Mitran
Professor, Dept. of Electrical and Computer Engineering
University of Waterloo

Internal Member: Zhou Wang
Professor, Dept. of Electrical and Computer Engineering
University of Waterloo

Internal-External Member: Jonathan Li
Professor, Dept. of Geography and Environmental Management
University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Network slicing technique enables a flexible and programmable network architecture. Multiple virtual networks (also known as network slices) are created over a shared physical network for customized and fine-grained service provisioning. Due to the immersive viewing experience and enormous vertical markets, 360° virtual reality (VR) video streaming service has attracted significant attention. Users wear a head-mounted display (HMD) to enjoy spherical video content and freely change their viewing orientations during the streaming. Delivering smooth and high-quality VR video content requires high end-to-end (E2E) transmission rate and low latency, posing significant challenges to current mobile networks. To reduce the requirement for a high transmission rate, a 360° video can be partitioned into multiple non-overlapping (temporal-spatial) video tiles, and only the video tiles covered by the predicted field-of-view (FoV) are transmitted with high bitrate levels. Nevertheless, tile-based 360° video streaming suffers from transmission rate fluctuations and inevitable FoV prediction errors driven by viewing behavior dynamics (i.e., head movements). Besides, encoded video tiles exhibit various properties in terms of transmission priority, deadline, and reliability requirement. To support enhanced service provisioning for tile-based 360° video streaming, requested video content needs to be efficiently delivered by considering various video tile properties under network and viewing behavior dynamics. Meanwhile, users must adaptively request video tiles and select bitrate levels based on FoV prediction results and available transmission rate. In this thesis, we focus on E2E enhancement for tile-based 360° video streaming service with core network slicing, where VR video streaming slices are deployed in the core network to facilitate enhanced VR video packet delivery.

First, we propose a customized transmission protocol based on Quick UDP Internet Connections (QUIC) which operates over a VR video slice in the core network. The QUIC protocol is tailored to accommodate the characteristics of tile-based VR video streaming where explicit mapping relations between requested video tiles and QUIC streams are established. Two customized in-network protocol functionalities including packet filtering and caching-based packet retransmission are proposed, to filter out outdated video data due to FoV prediction errors and to achieve efficient packet retransmissions with disparate transmission reliability requirements. A slice-level packet header is designed to support enhanced slice-based VR video packet transmission with the proposed protocol functionalities.

Second, we develop a performance analytical model for layer-encoded video packet services (i.e., processing or transmission) over a video streaming slice in the core network. The disparate service reliability requirements of base layer (BL) and enhancement layer

(EL) packets are considered in the analytical model for E2E packet delays, deadline violation probabilities, and throughputs of BL and EL packets. Specifically, a network function virtualization (NFV) node along the routing path of the video streaming slice is split into two consecutive logical nodes, one for packet processing and the other for transmission, based on which a segment-based analysis framework is proposed for E2E service performance modeling. A two-stage queuing model is established to obtain the approximate steady-state probability distribution of queue length at the first node in the first segment, upon which the BL/EL packet delay, deadline violation probability, and throughput at the segment are derived. In addition, the inter-departure time of successive packets departing from the first segment is analyzed based on an approximate M/D/1 system, and the packet departure process at the first segment is approximated as a Poisson process under the assumption of a large packet service rate of the first node. The independence between two consecutive segments is then achieved for analysis tractability, based on which the E2E performance measures are derived.

Lastly, we propose a two-step adaptive scheme for tile-based 360° video streaming to support multi-user enhanced viewing quality-of-experience (QoE) in a time-slotted system. Each video chunk is first prefetched based on the predictive FoV, and the FoV quality of the chunk is then enhanced at a closer-to-playback time instant based on the updated FoV prediction with improved accuracy. Both transmission-driven and device video super-resolution (VSR)-driven methods are adaptively selected to enable device energy-efficient chunk enhancement. At each time slot, the incremental QoE gain of each user is characterized via a time-difference approach, based on which the best candidate chunk is determined. Then, a single-slot problem is formulated for maximizing the total incremental QoE gain while minimizing the total device energy consumption. A particle swarm optimization (PSO)-based iterative solution is proposed to obtain optimal bandwidth allocation, bitrate level selection, and enhancement method selection for multiple users.

This research provides some insights in enhancing tile-based 360° VR video streaming service provisioning, from both network-side and end device-side perspectives.

Acknowledgements

Collaborating with Professor Weihua Zhuang and pursuing my PhD degree under her supervision is a truly rewarding experience for me. I would like to express my most sincere and deepest gratitude to Professor Zhuang for her professional guidance and generous help along this journey. Professor Zhuang believes in me more than I believe in myself, and she has been encouraging and supporting me since I became one of her students. The weekly research meetings between Professor Zhuang and me not only shape my previous understanding of what good research work is and how to conduct research rigorously, but also unconsciously influence me in dealing with challenges in work and life.

My sincere appreciation also goes to Professor Xuemin (Sherman) Shen. Professor Shen opens an additional window for me to see the cutting-edge research activities outside of the team. The beneficial talks offered by Professor Shen in terms of emerging research topics and high-quality academic writing and the weekly group research meetings he organizes have greatly broaden my vision and enriched my knowledge.

Specially, I would like to thank Professor Patrick Mitran, Professor Zhou Wang, Professor Jonathan Li, and the external examiner, Professor Xianbin Wang from Western University, for serving on my thesis defense committee. Their suggestions, comments, and valuable questions have greatly helped me improve the quality of my thesis.

In the past five years, I have gotten much help from the BBCR group. Particularly, I would like to express my cordial gratitude to Professor Qiang (John) Ye and Professor Kaige Qu for their selfless help and care. I believe we have established precious and strong friendships, and I feel lucky to be one of their academic brothers. Moreover, I would like to extend my earnest thanks to Prof. Dongxiao Liu, Prof. Wen Wu, Prof. Cheng Huang, and Ruoxu Wang for their insightful suggestions and supporting hands in both research and life. I would like to express my sincere thanks to other current/former BBCR members: Prof. Conghao Zhou, Shisheng Hu, Dr. Mingcheng He, Prof. Mushu Li, Prof. Qihao Li, Dr. Xinyu Huang, Yingying Pei, Zhixuan Tang, Yixiao Zhang, Dr. Xiangyu Ren, and many others. We have spent wonderful time and made unforgettable memories.

Finally, I would like to greatly appreciate my parents for their unwavering love and support. We share joys and tears, though thousands of miles away.

Yannan Wei
July, 2025
Waterloo, Ontario, Canada

Dedication

This PhD thesis is dedicated to my beloved parents, Zhonghua Wei and Juan Guo.

The dedication is also in memory of my grandparents.

Table of Contents

Examining Committee Membership	ii
Author's Declaration	iii
Abstract	iv
Acknowledgements	vi
Dedication	vii
List of Figures	xii
List of Tables	xv
List of Abbreviations	xvi
List of Symbols	xviii
1 Introduction	1
1.1 Network Slicing-based Core Networks	1
1.2 Adaptive Tile-based 360° Video Streaming	3
1.3 Research Objectives	4
1.3.1 Customized Transmission Protocol for Tile-based 360° Video Streaming over Core Network Slices	5

1.3.2	E2E Performance Modeling for Slice-Based Video Streaming with Layered Encoding	6
1.3.3	Energy-Efficient Multi-User Adaptive 360° Video Streaming with Device Video Super-Resolution	7
1.4	Research Contributions	8
1.4.1	Customized Transmission Protocol for Tile-based 360° Video Streaming over Core Network Slices	8
1.4.2	E2E Performance Modeling for Slice-Based Video Streaming with Layered Encoding	9
1.4.3	Energy-Efficient Multi-User Adaptive 360° Video Streaming with Device Video Super-Resolution	9
1.5	Thesis Outline	10
2	Customized Transmission Protocol for Tile-based 360° VR Video Streaming over Core Network Slices	12
2.1	System Model	14
2.1.1	Network Model	14
2.1.2	Video Traffic Model	16
2.1.3	Main Protocol Functionalities	17
2.2	Protocol Customization for VR Video Transmission	18
2.2.1	Tailored QUIC	18
2.2.2	Connection Establishment	20
2.2.3	VR Video Data Transmission	20
2.2.4	Packet Loss Recovery	24
2.3	Transport Parameter Calculation	27
2.3.1	E2E Packet Transmission Analysis	29
2.3.2	Minimum Required Caching Buffer Size	31
2.4	Performance Evaluation	32
2.4.1	Data Preprocessing	32

2.4.2	Simulation Settings	33
2.4.3	Performance Metrics	34
2.4.4	Simulation Results	35
2.5	Summary	41
3	E2E Performance Modeling for Slice-Based Video Streaming with Layered Encoding	42
3.1	System Model	44
3.1.1	Network Model	44
3.1.2	Video Traffic Model	44
3.1.3	Caching-based Packet Retransmission Functionality	47
3.2	E2E Service Performance Modeling	50
3.2.1	Segment-Based Analysis Framework	50
3.2.2	Service Performance Modeling at The First Segment	51
3.2.3	Service Performance Modeling Between Segments	57
3.2.4	Average E2E Performance	58
3.2.5	Discussion	59
3.3	Simulation Results	60
3.3.1	Validation of The Proposed Performance Analytical Model	61
3.3.2	Effectiveness of The Proposed Analytical Model	66
3.4	Summary	70
4	Energy-Efficient Multi-User Adaptive 360° Video Streaming: A Two-Step Approach with Device Super-Resolution	71
4.1	System Model	73
4.1.1	Network Scenario	73
4.1.2	Video Traffic Model	75
4.1.3	Two-step Adaptive Streaming Framework	75
4.1.4	Communication Model	78

4.1.5	Delay Model	78
4.2	Problem Formulation	80
4.2.1	User Viewing QoE	80
4.2.2	Best Candidate Chunk	82
4.2.3	Problem Formulation	84
4.3	PSO-based Iterative Solution	85
4.3.1	Problem Decomposition	85
4.3.2	Algorithm Design	86
4.4	Performance Evaluation	89
4.4.1	Data Trace Preprocessing	90
4.4.2	Simulation Settings	91
4.4.3	Simulation Results	92
4.5	Summary	99
5	Conclusions and Future Research Directions	100
5.1	Conclusions	100
5.2	Future Research Directions	102
	References	104
	APPENDICES	115
A	Proof of Proposition 1	116

List of Figures

1.1	An SDN/NFV-enabled core network.	2
1.2	A general tile-based E2E 360° video streaming system.	4
2.1	The SDN/NFV-based E2E transmission network scenario.	15
2.2	Tile-based layered VR video encoding.	16
2.3	Tailored QUIC with tile-to-stream mappings.	18
2.4	SDN/NFV-based E2E QUIC connection establishment.	21
2.5	The slice-level packet header format.	22
2.6	Video packet transmission over a VR video slice.	22
2.7	The STOP_SENDING frame format.	23
2.8	The workflow of video packet filtering.	23
2.9	Caching-based BL/EL packet loss recovery	26
2.10	The workflow of event-driven and time-based caching buffer releases.	28
2.11	The considered video tiling layout in the (projected) 2D video plane.	33
2.12	The considered network scenario in the simulation.	34
2.13	Average segment downloading time and quality vs. link capacity.	36
2.14	Average packet retransmission delay, segment downloading time, and segment quality vs. random packet loss rate.	37
2.15	Average segment downloading time and quality vs. FoV prediction error.	38
2.16	Average segment downloading time, segment quality, and goodput ratio vs. congestion duration.	40

3.1	An SDN/NFV-based core network with embedded network slices.	45
3.2	The virtual routing path of a video streaming slice with aggregated BL and EL traffic.	46
3.3	The flowchart of caching-based packet retransmission.	47
3.4	The workflow of time-based caching buffer release.	48
3.5	A segment-based analysis framework.	51
3.6	The state transition diagram of queue length at the first node in the first segment.	52
3.7	Two-stage queuing at the first node of the first segment.	53
3.8	Deadline violation probability (in log-scale) of BL and EL packets at the example segment with varying packet service rates of the first node (before approximations).	56
3.9	The considered virtual network topology of a video streaming slice in the simulation.	60
3.10	Packet delay at the first segment with varying packet service rates of the first node μ^1	61
3.11	Deadline violation probability at the first segment with varying packet service rates of the first node μ^1	62
3.12	Throughput at the first segment with varying packet service rates of the first node μ^1	62
3.13	Throughput at the first segment with varying BL packet arrival rates λ_b^1	63
3.14	Packet delay at the second segment with varying packet service rates of the first node μ^2	64
3.15	Deadline violation probability at the second segment with varying packet service rates of the first node μ^2	65
3.16	Throughput at the second segment with varying packet service rates of the first node μ^2	65
3.17	Throughput at the second segment with varying BL packet arrival rates λ_b^1	66
3.18	E2E packet delay with varying packet service rates of the first node in the second segment μ^2	67

3.19	E2E deadline violation probability with varying packet service rates of the first node in the second segment μ^2	67
3.20	E2E throughput with varying packet service rates of the first node in the second segment μ^2	68
3.21	Average reserved per-node packet service rate and the minimum required caching buffer size determined using different analytical models.	69
4.1	The considered network scenario.	73
4.2	Video traffic model.	75
4.3	The proposed two-step adaptive streaming framework.	76
4.4	Enhancement methods.	77
4.5	An example playback buffer of a user $u \in \mathcal{U}_t$ at slot t	82
4.6	The flowchart of the proposed PSO-based solution.	87
4.7	The convergence performance of the proposed PSO-based solution at sample slots.	88
4.8	The considered video tiling layout with sphere-to-plane coordinate transformation.	91
4.9	The considered benchmark schemes for performance comparison.	92
4.10	Performance of different schemes with $(\alpha, \beta, \gamma) = (2/3, 1/6, 1/6)$	93
4.11	Performance of different schemes with $(\alpha, \beta, \gamma) = (1/6, 2/3, 1/6)$	93
4.12	Performance of different schemes with $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$	94
4.13	Performance of different schemes with varying total bandwidth of the BS.	95
4.14	Performance of different schemes with varying number of users.	97
4.15	Cropped snapshots of the generated demo videos (Coastal pier, slow-paced) under the proposed scheme and Flare.	98
4.16	Cropped snapshots of the generated demo videos (Mega coaster, fast-paced) under the proposed scheme and Flare.	98

List of Tables

2.1	Main simulation parameters	32
3.1	The values of N_e^1 and j^e for different packet service rates of the first node μ^1 in the example segment	55
3.2	The performance of the caching-based packet retransmission scheme	70
4.1	Main simulation parameters	90

List of Abbreviations

ACK	Acknowledgement
BL	Base Layer
BS	Base Station
CBR	Constant Bitrate
DDL	Deadline
DNN	Deep Neural Network
EL	Enhancement Layer
ERP	Equirectangular Projection
E2E	End-to-end
FIFO	First-in-first-out
FoV	Field-of-view
HMD	Head-mounted Display
ICR	Ingress Caching Release
LR	Linear Regression
MINLP	Mixed-integer Non-linear Programming
MPD	Media Presentation Description
MTP	Motion-to-photon
NFV	Network Function Virtualization
OFO	Out-of-order
PASTA	Poisson Arrivals See Time Averages
PSO	Particle Swarm Optimization
QoE	Quality of Experience
QoS	Quality of Service

QUIC	Quick UDP Internet Connections
RAN	Radio Access Network
RR	Retransmission Request
SDN	Software-defined Networking
SFC	Service Function Chain
SHVC	Scalable High-efficiency Video Encoding
SNC	Stochastic Network Calculus
SNR	Signal-to-noise Ratio
TCP	Transmission Control Protocol
TS	Timestamp
VNF	Virtual Network Function
VR	Virtual Reality
VSR	Video Super-resolution

List of Symbols

k_c	Cycle of the periodic timer
$C_{\min}^b/C_{\min}^e$	Minimum required caching buffer size (in packet) for BL/EL packets
μ^g	Packet service rate (in packet/s) of the first node in segment g
λ^g	Total packet arrival rate at segment g . $\lambda^g = \lambda_b^g + \lambda_e^g$ where λ_b^g and λ_e^g are the arrival rates of BL and EL packets arriving at segment g , respectively
$\bar{d}^{b,g}/\bar{d}^{e,g}$	Average remaining service deadline of BL/EL packets arriving at segment g
N_e^1	Queue length bound for EL packets that can successfully pass through the first node in the first segment without being discarded due to service deadline violations
$\hat{q}_i, i \geq 0$	Approximated steady-state probability distribution of queue length at the first node in a segment
P_j	Steady-state probability distribution of queue length for stage one queuing
π_n	Steady-state probability distribution of queue length for stage two queuing
ρ_1^1/ρ_2^1	Traffic intensity for stage one/two queuing at the first node in the first segment
$\tilde{T}^1$	Total service time for a packet passing through the rest of nodes except the first one in the first segment
T_i^1	Time consumption for the i th packet in the queue of the first node traversing the first segment
$T_{\theta}^1, \theta \in \{b, e\}$	Average BL/EL packet delay for traversing the first segment

$j^\theta - 1, \theta \in \{b, e\}$	Queue length bound for BL/EL packets at the first node that can successfully traverse a segment without violating their service deadlines
$V_\theta^1, \theta \in \{b, e\}$	Deadline violation probability of BL/EL packets when traversing the first segment
λ_t^1	Total throughput at the first segment. $\lambda_t^1 = \lambda_{t,b}^1 + \lambda_{t,e}^1$ where $\lambda_{t,b}^1$ and $\lambda_{t,e}^1$ are the throughputs of BL and EL traffic after traversing the first segment, respectively
Y^1	Inter-departure time of successive packets departing from the first segment
$T_\theta/V_\theta/\lambda_{t,\theta}, \theta \in \{b, e\}$	E2E packet delay, deadline violation probability, and throughput of BL/EL packets traversing the video streaming slice
τ, T_c	The slot and video chunk lengths
$\mathcal{I}_u$	The set of video chunks of user u
$\mathcal{J}$	The set of video tiles composing the panoramic video scene
$\mathcal{R}$	The set of encoding bitrate levels for a video tile
$u \in \mathcal{U}_t$	The set of users at slot t
$\mathcal{B}_{t,u}, O_{t,u}$	The playback buffer and the current playback buffer occupancy of user u at slot t
Ω	The total radio spectrum bandwidth (in MHz) of the BS
$A_{t,u,i}, e_{t,u,i}, f_{t,u,i}$	The FoV quality, enhancement status, and effective factor for video chunk i of user u at slot t
$r_{u,i}^P, \mathcal{J}_{u,i}^P$	The bitrate level and the set of predictive FoV tiles for video chunk i of user u
$\mathcal{J}_{t,u,i}^P, \mathcal{J}_{t,u,i}^M, \mathcal{J}_{t,u,i}^H$	The sets of predictive FoV tiles, miss-predicted tiles, and hit tiles when a new video chunk of user u is prefetched or when prefetched chunk i of user u is enhanced at slot t
$r_{t,u,i}^P, r_{t,u,i}^M, r_{t,u,i}^H$	The bitrate levels of predictive FoV tiles, miss-predicted tiles, and hit tiles when a new video chunk of user u is prefetched or when prefetched chunk i of user u is enhanced at slot t
$\rho_{t,u,i}^M, \rho_{t,u,i}^H$	The enhancement methods for miss-predicted tiles and hit tiles when prefetched chunk i of user u is enhanced at slot t
$w_{t,u}, \kappa_{t,u}$	The fraction of bandwidth allocated to user u at slot t and the corresponding transmission rate

$d_{t,u}^P, d_{t,u}^E$	The total experienced delay when a new chunk of user u is prefetched or when a prefetched chunk of user u is enhanced at slot t
$\phi_u(r_b, r'_b)$	The time for reconstructing a video tile from bitrate level r_b to a higher bitrate level $r'_b \geq r_b$ by user device u
$Q_{t,u}^1, Q_{t,u}^2, Q_{t,u}^3$	The average perceived FoV quality, video stall time, and temporal quality smoothness of user u at slot t
$G_{t,u}^1, G_{t,u}^2, G_{t,u}^3$	The incremental gains in terms of different QoE components for user u at slot t
(α, β, γ)	The importance coefficients regarding different QoE components
$E_{t,u}$	The incremental average (normalized) device energy consumption of user u at slot t
N, K	The swarm size and the total number of iterations in the PSO algorithm

Chapter 1

Introduction

1.1 Network Slicing-based Core Networks

Next-generation mobile networks are envisioned to support diverse emerging services and applications with heterogeneous quality of service (QoS) requirements, such as immersive communication and autonomous driving. Supporting the coexistence of these new applications over a common physical substrate infrastructure with QoS satisfaction requires a flexible and programmable network architecture. The concept of *network slicing* then comes into play, where software-defined networking (SDN) and network function virtualization (NFV) are two key enabling technologies [1]. They achieve agile and scalable service deployment and network management.

SDN decouples control functions from network nodes (e.g., switches), which are integrated into a logical centralized controller (in the control plane) with the global view over an underlying physical network. The data plane is simplified with programmable switches connected to the SDN controller. With the softwarized global control, the SDN controller can establish optimal E2E routing paths for different service traffic flows to efficiently utilize link transmission resources and alleviate network congestion. Besides, the SDN controller collects network state information (e.g., traffic loads) from the underlying SDN switches and instructs data forwarding and processing operations with in-network awareness, via southbound protocols such as the OpenFlow [2]. On the other hand, NFV is a complementary technology to enable elastic and cost-effective network function deployment. Functions from dedicated hardware are decoupled as virtual network functions (VNFs) hosted in virtual machines on commodity servers (or NFV nodes). With SDN and NFV, multiple virtual networks, also known as *network slices*, can be created over a shared

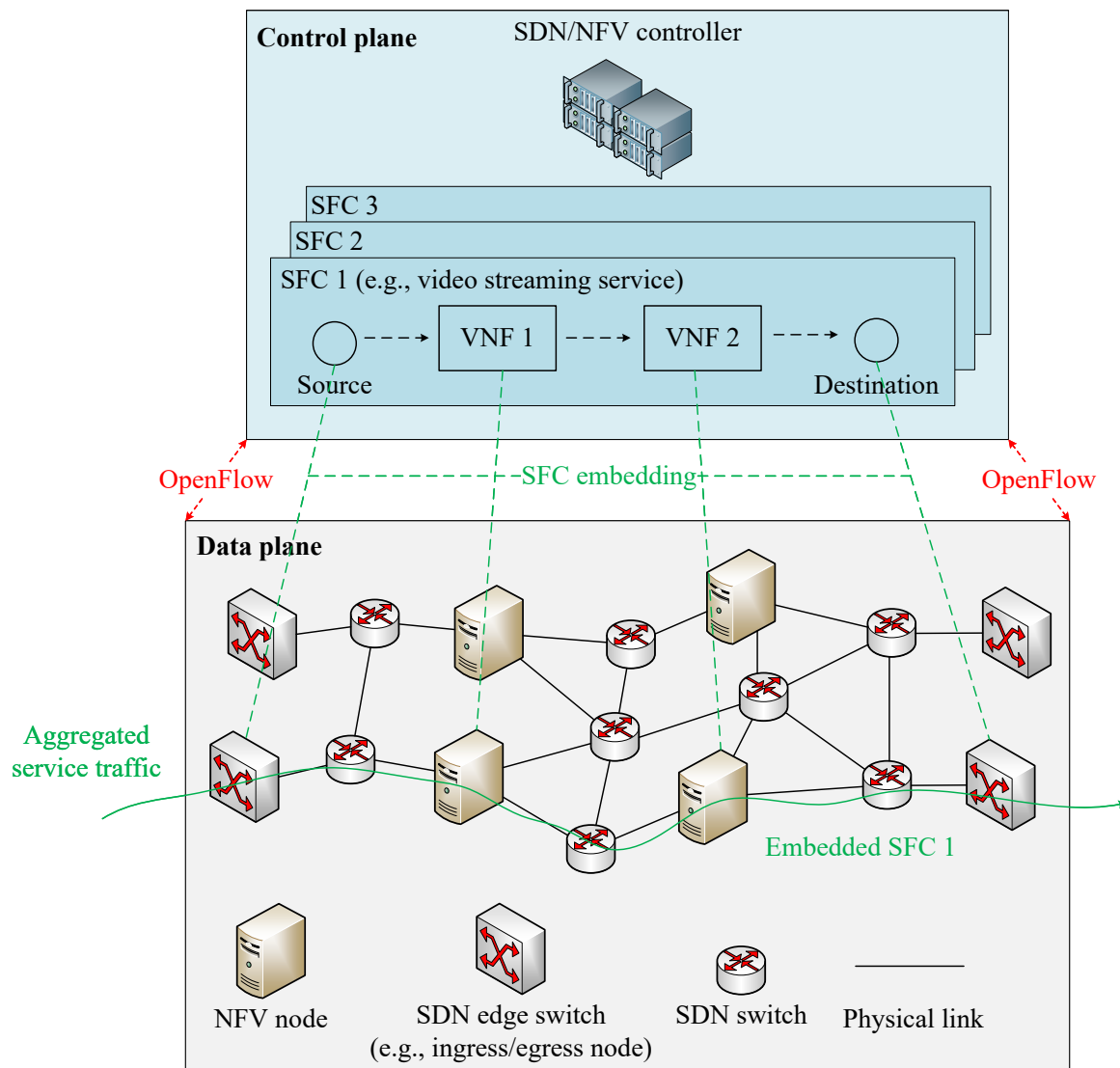


Figure 1.1: An SDN/NFV-enabled core network.

physical network with dedicated resources (e.g., processing and transmission resources) for supporting different services [3]. Fig. 1.1 shows an SDN/NFV-enabled core network. Each service is described by a service function chain (SFC) which is a specific sequence of VNFs interconnected in a predefined order. Each SFC is embedded in the core network by the SDN/NFV controller to create a core network slice for supporting a particular service. Specifically, the virtual network topology of the slice is configured. Dedicated processing and link transmission resources are reserved on NFV nodes and physical links along the routing path of the slice for data packet processing and transmission, respectively [4, 5]. In addition, within the network slice for a particular service, customized protocol functionalities can be deployed on specific network nodes, such as an NFV node or an SDN edge switch. These customized protocol functionalities enable in-network fine-grained transmission control and support enhanced service provisioning without affecting other services.

1.2 Adaptive Tile-based 360° Video Streaming

In traditional (on-demand) video streaming, the viewing area is predetermined, and the viewing orientation is thus fixed. In contrast, in 360° virtual reality (VR) video streaming, users (or video clients) wear a head-mounted display (HMD) to watch a panoramic (spherical) video and can freely switch their viewing orientations during the streaming. Due to immersive viewing experience and enormous vertical markets, such as entertainment and remote education [6, 7, 8], 360° video streaming has attracted significant attention from both academia and industry. It is forecasted that the number of HMDs and other types of mobile VR devices will reach over 110 million by 2026 [9]. Due to the large area of omni-directional video content, a 360° video has the data size typically 4-6 times larger than a conventional video at an equivalent quality. Supporting smooth and high-quality streaming of a 360° video requires extremely high end-to-end (E2E) transmission rate (e.g., 400 Mbps) and poses significant challenges to current mobile networks [10, 11]. As a user watches only part of a 360° video at any time instant due to the limited span of an HMD (e.g., $120^\circ \times 120^\circ$), referred to as field-of-view (FoV), the tile-based 360°/VR video streaming has been proposed to reduce the high demand on transmission rate [6].

Fig. 1.2 shows a general tile-based E2E 360° video streaming system with emphasis on the end sides. At the server side, raw 360° videos are projected onto a 2D video plane using techniques such as the equirectangular projection (ERP). Each 360° video is first temporally divided into a sequence of video chunks (or segments), each with a playback time of typically 2-10 seconds. Then, each video chunk is spatially partitioned into multiple non-overlapping video tiles. Each tile is further independently encoded into

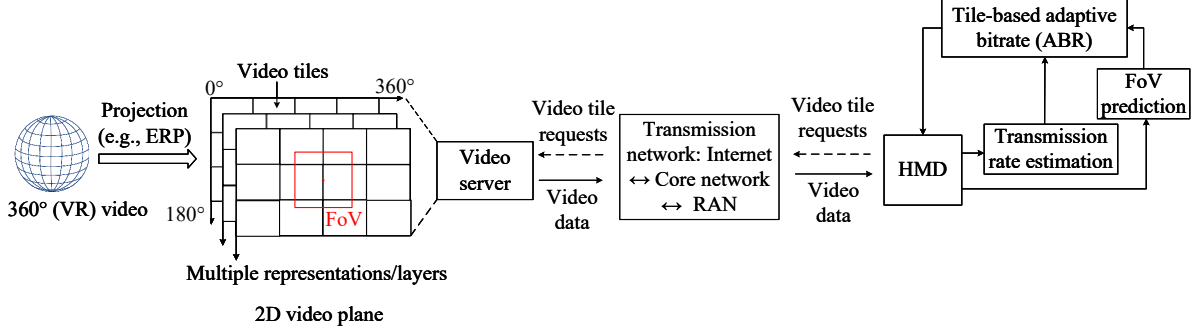


Figure 1.2: A general tile-based E2E 360° video streaming system.

multiple representations or interdependent layers with different bitrate (or quality) levels to support adaptive tile-based streaming.

At the user side, head movements are tracked by the HMD for FoV prediction, which is a critical module in tile-based adaptive 360° video streaming [6, 12, 13, 14, 15]. When a user requests a video chunk, the FoV of the chunk is first predicted. Then, based on the predicted FoV and the estimated transmission rate, the user selectively requests a set of video tiles with different bitrate levels. Generally, the priority of a video tile being requested depends on its likelihood of being watched by the user. More transmission resources are allocated to deliver the video tiles covered by the predicted FoV than the peripheral tiles outside the predicted FoV. Nevertheless, FoV prediction and downloading of a video chunk are conducted before it is played. The predicted FoV may diverge from the real one due to the inevitable prediction errors caused by user viewing behavior dynamics, especially when the prediction time gap is large.

Throughout this thesis, we consider slice- and tile-based E2E adaptive 360° video streaming, where video streaming slices are pre-deployed in a core network to support enhanced service provisioning.

1.3 Research Objectives

Tile-based adaptive 360° video streaming suffers from transmission rate fluctuations and inevitable FoV prediction errors. To support enhanced provisioning of 360° video streaming service, two fundamental questions need to be answered: (1) How requested video tiles should be efficiently delivered from remote video servers to users under viewing behavior

and network dynamics (e.g., sudden head movements and random packet loss); (2) How to request video tiles and select bitrate levels based on the predicted FoV and the available transmission rate to enhance user viewing quality-of-experience (QoE). Specifically, encoded video tiles exhibit various properties in terms of transmission priority, deadline, and reliability requirement. A supporting transmission protocol is imperative to efficiently deliver requested video tiles to a user, which incorporates various video tile properties while adapting to network and viewing behavior dynamics. In addition, an analytical model is essential to evaluate the performance of slice-based video packet transmission. The analytical model can be used to optimize the configuration of a video streaming slice for service-level QoS guarantee in the long run. Finally, under transmission rate variations, a user needs to adaptively select bitrate levels for requested video tiles to balance between video playback smoothness and video quality. Moreover, transmission resources are not efficiently utilized when the FoV prediction accuracy is low, as the predictive FoV tiles normally assigned with high bitrate levels may not be actually watched by the user. Hence, an adaptive streaming scheme is needed. The streaming scheme adaptively selects bitrate levels while considering the impact of FoV prediction accuracy on user perceived FoV quality for user viewing QoE optimization.

In this PhD research, we investigate E2E service performance enhancement for tile-based adaptive 360° video streaming from video servers on the Internet to users. Specifically, we study three research problems. In the first problem, we focus on transmission protocol customization over a VR video streaming slice based on the Quick UDP Internet Connections (QUIC) protocol. In the second problem, we study E2E performance analytical modeling for slice-based video packet transmission. For the first two problems, we focus on downlink slice-level video packet transmission over a core network. In the third research problem, we focus on the radio access network (RAN) domain and investigate adaptive streaming scheme design. Both downlink wireless video data transmission and uplink video content requests are jointly considered.

1.3.1 Customized Transmission Protocol for Tile-based 360° Video Streaming over Core Network Slices

Encoded video tiles have various properties in terms of transmission priority, deadline, and reliability requirement, which are expected to be supported by the underlying operating transport protocol. With scalable high-efficiency video encoding (SHVC), each video tile is encoded into one base layer (BL) and multiple enhancement layers (ELs) [16]. BL ensures video playback smoothness with a basic quality, which is an essential layer of decoding and

can be decoded independently. ELs increase the encoding bitrate and provide improved quality. An EL can be decoded only if the BL and all lower-layer ELs are decoded. Thus, BL tiles have a higher transmission priority than EL tiles. Each BL/EL tile has a strict deadline to be delivered. Especially, when viewers suddenly rotate heads, additional urgent video tiles need to be requested with a small deadline to compensate for the current viewing experience, referred to as the motion-to-photon (MTP) latency requirement (usually less than 25 ms [17]). Moreover, BL packets have a higher transmission reliability requirement than EL packets. BL packets need to be reliably transmitted to avoid black spots in the FoV or long video stall time. Deadline-violated EL packets can be directly discarded without retransmissions for congestion alleviation and high transmission efficiency [18]. In addition, tile-based 360° video streaming suffers from viewing behavior dynamics constantly driven by head movements, causing FoV prediction errors. En-route VR video packets may contain outdated video data which should be dropped for transmission efficiency.

To support efficient VR video data transmission, a customized transmission protocol is imperative. In a slice-based core network, a VR video streaming slice can be deployed to support enhanced service provisioning. Customized protocol functionalities and operations can be implemented to realize prompt in-network transmission control. In Problem I, we aim to develop a customized slice-level transmission protocol based on QUIC for tile-based 360° video streaming. The proposed transmission protocol will support various VR video packet properties and promptly react to network and viewing behavior dynamics to achieve enhanced slice-level VR video packet transmission.

1.3.2 E2E Performance Modeling for Slice-Based Video Streaming with Layered Encoding

To evaluate the performance of slice-based and layer-encoded video packet transmission, an E2E performance analytical model is essential, which analyzes E2E packet delay, deadline violation probability, and throughput. Deadline violation probability is a critical performance metric for layer-encoded video streaming. This is because video packets delivered within an average delay bound may not always improve video playback smoothness or quality due to deadline violations. In addition, the analytical model is also helpful in determining key transport parameters of customized protocol functionalities implemented in the video streaming slice.

The challenges in the E2E analytical modeling mainly come from two parts. First, the packet service (i.e., processing or transmission) performance at one node is challenging to analyze due to the disparate service reliability requirements of BL and EL packets. Due to

the deadline-violated EL packet dropping, arriving BL and EL packets at a single node may undergo different service processes, resulting in disparate average performance. Second, packet service performance at each intermediate node is closely correlated to its preceding node. Specifically, the packet arrival process at the node depends on the packet departure process of its preceding node. As deadline-violated EL packets are directly dropped at each node without being forwarded to the next node, packet service processes at different nodes may be different. It is intractable to analyze the packet service performance of every node in a multi-hop packet service setting.

In Problem II, we aim to develop an E2E performance analytical model for slice-based and layer-encoded multi-hop video packet transmission. The impact of disparate service reliability requirements of BL and EL packets and the correlations between consecutive service nodes are studied in the E2E analytical modeling.

1.3.3 Energy-Efficient Multi-User Adaptive 360° Video Streaming with Device Video Super-Resolution

Tile-based adaptive 360° video streaming suffers from transmission rate fluctuations and inevitable FoV prediction errors, which affect user perceived FoV quality. Maintaining a large playback buffer occupancy ensures high video playback smoothness, whereas reducing the FoV prediction accuracy due to a large prediction time gap. In this case, predictive FoV tiles assigned with high bitrate levels may not be actually watched by the user due to high FoV prediction errors, leading to low transmission rate utilization and perceived FoV quality. Conversely, the FoV prediction accuracy becomes high when the playback buffer occupancy is small due to a short prediction time gap. Transmission resources can be efficiently utilized to deliver predictive FoV tiles with high bitrate levels, at the risk of video stalling under transmission rate fluctuations. Thus, an adaptive streaming scheme is critical, which balances video playback smoothness and perceived FoV quality by considering the impact of FoV prediction accuracy on user perceived FoV quality. On the other hand, the technique of video super-resolution (VSR) can be exploited to further enhance video tile quality, while breaking the strong dependency between user viewing QoE and available transmission rate [19]. A video tile with a low bitrate level can be transmitted by a remote server first. Then, leveraging the computing capability of an edge server and/or a user device, it can be reconstructed into a higher bitrate level. This approach reduces transmission time but incurs additional reconstruction time and device energy consumption.

In Problem III, we aim to design an adaptive scheme for tile-based 360° video streaming

to support enhanced multi-user viewing QoE. The adaptive streaming scheme will capture the impact of FoV prediction accuracy on user perceived FoV quality. It will also incorporate the merits of device-VSR-assisted FoV quality enhancement while considering device energy consumption.

1.4 Research Contributions

In this section, we present the research contributions for each research problem.

1.4.1 Customized Transmission Protocol for Tile-based 360° Video Streaming over Core Network Slices

In Chapter 2, we present a customized slice-level transmission protocol based on QUIC to achieve enhanced VR video packet delivery [20]. Specifically, the QUIC protocol is tailored to accommodate the characteristics of tile-based 360° video streaming. An explicit one-to-one mapping between requested EL tiles and QUIC streams is established. The behaviors of stream multiplexing and QUIC packet assembly at the server side are modified accordingly. Besides, two customized protocol functionalities are designed, including packet filtering and caching-based packet retransmission. The packet filtering functionality removes outdated video data due to FoV prediction errors in prompt response to viewing behavior dynamics (i.e., head movements). The caching-based packet retransmission functionality conducts efficient packet loss recovery according to disparate reliability requirements. A slice-level packet header is designed to support enhanced slice-based VR video packet transmission with the proposed protocol functionalities. The main contributions are summarized as follows.

- The QUIC protocol is tailored to better support the tile-based 360° VR video streaming service. Explicit mapping relations are established between requested video tiles with various properties and QUIC streams;
- Customized protocol functionalities including packet filtering and caching-based packet retransmission are proposed to support responsive and efficient transmission control. A slice-level packet header is designed to support enhanced VR video packet transmission with the proposed protocol functionalities;
- Key transport parameters in the proposed transmission protocol are determined via analytical modeling.

1.4.2 E2E Performance Modeling for Slice-Based Video Streaming with Layered Encoding

In Chapter 3, we present an E2E service performance modeling for layer-encoded video traffic traversing a video streaming slice in the core network. The discrepant service reliability requirements of BL and EL packets are considered in the analytical modeling [21]. Specifically, each NFV node along the routing path of the video streaming slice is split into two consecutive logical nodes, one for packet processing and the other for packet transmission. The E2E packet processing and transmission over the video streaming slice is abstracted as a multi-hop packet service process with different service reliability requirements. Based on the analysis of the correlations between two consecutive nodes, a segment-based analysis framework is proposed. Within each segment, a two-stage queuing model is established to obtain the approximate steady-state probability distribution of queue length at the first node. Then, the BL/EL packet delay, deadline violation probability, and throughput at the segment are analyzed. In addition, the departure process from a segment is discussed based on an approximated M/D/1 queuing system, upon which the E2E performance is analyzed. The main contributions are summarized as follows.

- A segment-based analysis framework is proposed for E2E service performance modeling of slice-based video streaming with layered encoding. A two-stage queuing model is established to derive the approximate steady-state probability distribution of queue length at the first node in each segment;
- The departure process of successive packets departing from a segment is analyzed as a mixed process alternating between Poisson and deterministic processes. The independence between two consecutive segments is achieved under the assumption of a large packet service rate of the first node in the preceding segment for E2E analysis tractability;
- The proposed performance analytical model analyzes the E2E BL/EL packet delay, deadline violation probability, and throughput between the pair of ingress and egress nodes of a video streaming slice.

1.4.3 Energy-Efficient Multi-User Adaptive 360° Video Streaming with Device Video Super-Resolution

In Chapter 4, we propose a two-step adaptive scheme for tile-based 360° video streaming to support enhanced multi-user viewing QoE in a time-slotted system [22]. Each video chunk

is first prefetched based on the predictive FoV and then enhanced at a closer-to-playback time instant based on the updated and more accurate FoV prediction. Transmission-driven and device-VSR-driven methods are proposed, which are adaptively enabled for device energy-efficient FoV quality enhancement. To determine when to prefetch a new video chunk or enhance a prefetched chunk, a time-difference approach is applied for user viewing QoE characterization. A single-shot problem is then formulated for maximizing the total incremental QoE gain while minimizing the total device energy consumption. A practical particle swarm optimization (PSO)-based algorithm is developed to determine the per-slot optimal decisions of bandwidth allocation, bitrate level selection, and enhancement method selection for each user. The main contributions are summarized as follows.

- A two-step adaptive streaming scheme is proposed to achieve enhanced 360° video viewing QoE, which captures the impact of FoV prediction accuracy on user perceived FoV quality. Both transmission-driven and device-VSR-driven methods are proposed and adaptively enabled to achieve device energy-efficient FoV quality enhancement;
- The incremental QoE gain of each user is characterized via a time-difference approach. The best candidate chunk of each user is then determined, which balances the tradeoff between video playback smoothness and perceived FoV quality;
- A single-slot problem is formulated for maximizing the total incremental QoE gain while minimizing the total device energy consumption. It is decomposed into multiple independent per-user subproblems given bandwidth allocation decisions. A PSO-based algorithm is developed to obtain the optimal bandwidth allocation, bitrate level selection, and enhancement method selection decisions for multiple users at each slot.

1.5 Thesis Outline

The rest of this thesis is organized as follows. The three research problems are presented in Chapters 2, 3, and 4, respectively.

In Chapter 2, Section 2.1 presents the system model under the consideration of Problem I. The main designs in developing our customized slice-level transmission protocol and the corresponding E2E VR video transmission process are discussed in Section 2.2. In Section 2.3, key transport parameters in the proposed transmission protocol are determined via theoretical analysis. Performance evaluation of the proposed transmission protocol is presented in Section 2.4, and a summary for this chapter is presented in Section 2.5.

In Chapter 3, the considered system model of Problem II is described in Section 3.1. Section 3.2 presents our proposed E2E performance analytical model for video packet services over a video streaming slice with layered encoding. Simulation results are discussed in Section 3.3, and a summary for this chapter is given in Section 3.4. In Chapter 4, the considered system model for Problem III is provided in Section 4.1. Section 4.2 first presents the user viewing QoE characterization and then the problem formulation. The proposed PSO-based solution is presented in Section 4.3. Simulation results are discussed in Section 4.4, and a summary for this chapter is given in Section 4.5. Finally, concluding remarks are drawn and future research directions are discussed in Chapter 5.

Chapter 2

Customized Transmission Protocol for Tile-based 360° VR Video Streaming over Core Network Slices

Tile-based 360° VR video streaming suffers from inevitable FoV prediction errors driven by viewing behavior dynamics (i.e., constant head movements). Meanwhile, encoded video tiles have various properties in terms of transmission priority, deadline, and reliability requirement. Thus, a supporting transmission protocol is imperative for efficient VR video packet transmission while adapting to network and viewing behavior dynamics (e.g., random packet loss).

Existing studies on protocol design conduct prioritized transmission scheduling among requested video tiles at the user side or at the server side. For example, at the user side, considering the importance of video smoothness over video quality, BL tiles are always downloaded first until a preset upper bound of playback buffer occupancy is reached [23]. When a viewer’s future FoV is not correctly predicted, new instant EL tiles for the updated FoV are immediately requested before the predictive EL tiles [24]. At the server side, different schedulers are developed to schedule and transmit requested video tiles to a video client. The proposed schedulers determine and update the *scheduling priorities* of the requested video tiles and schedule the transmission according to their scheduling priorities. The scheduling priority is usually determined by considering each requested video tile’s priority, playback deadline, remaining size, and current network conditions [25, 26, 27]. In addition, the stream multiplexing and scheduling features that are naturally supported in the QUIC protocol can be leveraged to achieve prioritized transmissions

among requested video tiles [28, 29]. In a QUIC connection, multiple video tiles can be concurrently requested via different QUIC streams. Different scheduling policies can be flexibly designed at the server side to determine how streams are multiplexed and scheduled for different transmission purposes. For example, when viewers suddenly rotate their heads, QUIC allows the server to quickly send urgent video tiles with a high priority to mitigate the negative impacts due to missing tiles (e.g., motion sickness) [30]. QUIC also provides flexible stream-level transmission control through sending signaling frames such as RESET_STREAM and STOP_SENDING frames to adapt to viewing behavior dynamics. Most existing works focus on improving protocol operations at end hosts to enhance the video transmission performance, which may lead to slow responsiveness to network and viewing behavior dynamics.

Under the SDN/NFV framework in a slice-based core network, flexible and prompt transmission control can be realized with in-network awareness. Specifically, SDN switches, as configured by an SDN controller, can achieve early congestion detection and throttle the sending rate by modifying the *Receive Window* field in a Transmission Control Protocol (TCP) acknowledgement (ACK) packet or by selectively discarding packets of different traffic flows to trigger fast retransmission [31, 32]. With the NFV technology, customized protocol functionalities can be deployed at in-network nodes to realize in-network transmission control for enhancing the E2E transmission performance of a particular service [33, 34]. For example, for delay-sensitive and loss-tolerant 2D video streaming service, a selective caching function is deployed at an ingress node of a 5G core network to adaptively cache EL packets for alleviating network congestion with limited video quality degradation [35]. The cached EL packets are resumed transmissions when network conditions are better.

We consider an SDN/NFV-based E2E transmission network where a VR video (streaming) slice is deployed between each pair of ingress and egress nodes in the core network. The QUIC protocol is adopted as the base transport protocol for E2E video transmission. For the tile-based 360° VR video streaming service, several research issues regarding transport protocol design should be investigated. First, various video tile properties should be accommodated in protocol operations. Optional header fields are thus needed in the QUIC packet header to indicate each VR video packet's properties. Video tiles with disparate properties should not be transmitted over the same QUIC stream(s), and only data of video tiles with the same properties should be multiplexed in a QUIC packet. Correspondingly, the mapping relations between requested video tiles and QUIC streams should be determined, and how to conduct streaming multiplexing and QUIC packet assembly needs to be revisited. Second, a slice-level transmission scheme should be designed to support enhanced video transmission over a VR video slice, where differentiated packet loss recov-

ery mechanisms are required for BL and EL packets. In addition, when viewers suddenly rotate heads causing FoV prediction errors, some requested video tiles for the previously predicted FoV become outdated. En-route VR video packets may contain outdated video data which should be dropped to save transmission resources.

In this chapter, we propose a customized slice-level transmission protocol based on QUIC. The QUIC protocol is tailored to accommodate the characteristics of tile-based VR video streaming. An explicit one-to-one mapping between requested EL tiles and QUIC streams is established, and the behaviors of stream multiplexing and QUIC packet assembly at the server side are modified accordingly. Two customized protocol functionalities are designed, including packet filtering and caching-based packet retransmission, to filter out outdated video data in prompt response to viewing behavior dynamics and to perform efficient packet retransmission with disparate transmission reliability requirements. A slice-level packet header is designed to support enhanced slice-based VR video packet transmission with the proposed protocol functionalities. Simulation results demonstrate the effectiveness of our proposed transmission protocol in achieving short average video segment downloading time and high average video segment quality.

2.1 System Model

2.1.1 Network Model

An SDN/NFV-based E2E transmission network is considered to support (on-demand) 360° VR video streaming service delivered from remote VR video servers to video clients. As shown in Fig. 2.1, Geographically adjacent video clients connect via different base stations (e.g., gNodeBs) to an edge node (e.g., an egress edge switch of its connected core network). Egress and ingress edge switches (referred to as nodes) are located in between the RAN domain and the core network and in between the core network and the Internet, respectively. Ingress and egress nodes host advanced traffic management functions and are deployed by a mobile operator that provides network support for the provisioning of the 360° VR video streaming service. SDN switches including ingress and egress nodes are physically (may not be directly) connected to the SDN/NFV controller. The logical centralized SDN/NFV controller in the control plane has a global view over the underlying core network, and communicates with programmable SDN switches including ingress and egress nodes in the data plane via southbound protocols such as the OpenFlow [2]. Under the SDN/NFV architecture, multiple virtual networks or network slices are instantiated over the shared core network for supporting diversified services. Specifically, a VR video

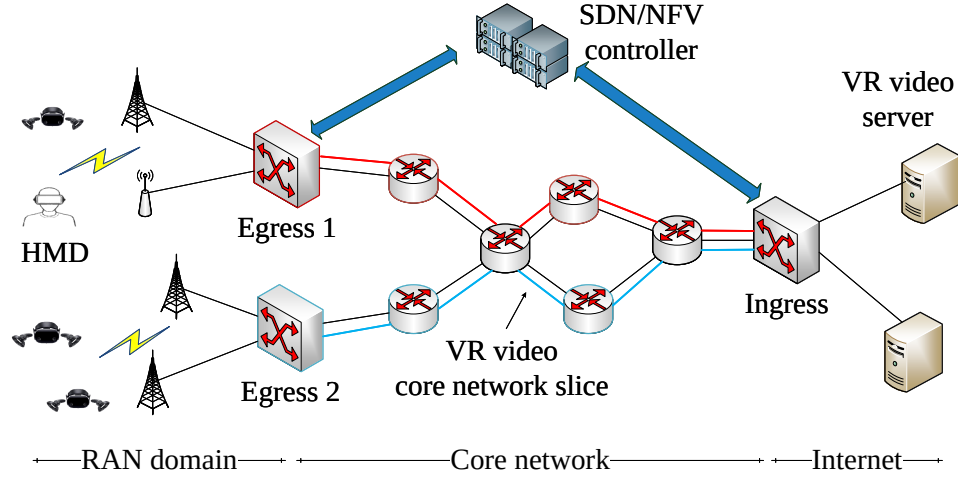


Figure 2.1: The SDN/NFV-based E2E transmission network scenario.

slice is deployed between each pair of ingress and egress nodes in the core network for supporting aggregated VR video traffic. For each VR video slice, a dedicated virtual network topology (assumed linear for simplicity) is configured by the SDN/NFV controller [36], where $W - 1$ intermediate SDN switches $\{s_1, \dots, s_{W-1}\}$ between a pair of ingress (s_0) and egress (s_W) nodes are interconnected by W transmission links, denoted by $\{l_0, \dots, l_{W-1}\}$. Virtual network resources including link transmission resources and processing resources are reserved at physical links and programmable switches along the route for data transmission and processing. Customized protocol functionalities are enabled at network nodes within each slice to achieve finer-grained VR video service provisioning.

E2E VR video transmission traverses three network segments: (i) from a VR video server on the Internet to an ingress node, (ii) over a VR video slice, and (iii) from the egress node of the VR video slice to a video client. Here, we focus on the core network and aim to develop a slice-level transmission protocol for supporting enhanced VR video data transmission which operates over a VR video (core network) slice. We adopt QUIC as our protocol design base for E2E video data transmission due to its superiority in achieving reduced E2E connection establishment latency and in stream multiplexing with flexible stream-level transmission control [37].

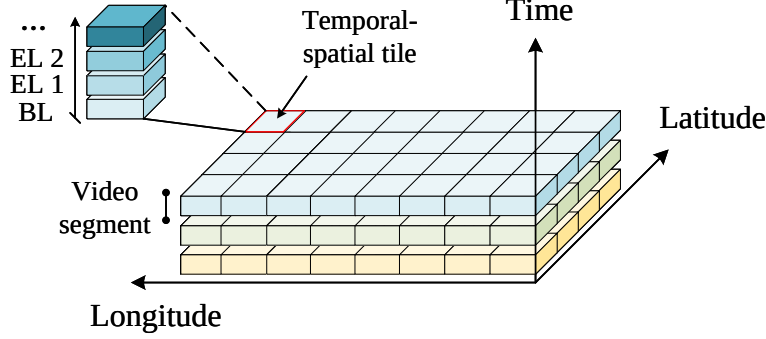


Figure 2.2: Tile-based layered VR video encoding.

2.1.2 Video Traffic Model

As shown in Fig. 2.2, a VR video is temporally divided into a sequence of video segments, each with a playback time of typically 2-10 seconds. Each video segment is spatially partitioned into multiple non-overlapping video tiles. Each temporal-spatial video tile is encoded into multiple interdependent layers for adaptive streaming under network transmission rate variations. Specifically, each video tile is encoded into one BL and multiple ELs. BL tiles ensure video playback smoothness and provide basic video quality. Each BL packet is required to be delivered with high reliability, and BL packet losses result in video rebuffering. EL video tiles provide improved video quality and can be decoded only if the corresponding BL tiles are decoded. EL packets can be discarded with low reliability for congestion alleviation, without affecting the video playback smoothness. Therefore, BL packets have a higher transmission priority and reliability requirement than EL packets. The scalable and layered video tile encoding provides additional flexibility in response to network and viewing behavior dynamics in small timescales due to the disparate transmission reliability requirements of BL and EL packets [35].

Suppose that encoded VR videos are stored on remote servers on the Internet. Video clients progressively download video content from remote servers on a video segment basis. For each segment downloading, an HMD selectively requests a set of video tiles with different numbers of encoding layers, based on factors such as the estimated throughput, tracked head/viewpoint movement trajectory, and playback buffer occupancy, etc. An HMD sends additional urgent requests after sudden head movements during watching a video segment and/or when FoV prediction errors are identified. Here, we consider that

each VR video server streams full-view BL for video robustness and EL tiles covering a client's FoV.

2.1.3 Main Protocol Functionalities

To support enhanced VR video transmission by accommodating various VR video packet properties and by adapting to viewing behavior dynamics, two customized protocol functionalities are proposed.

1) *Packet Filtering*: When an FoV prediction error occurs due to sudden head movements, previously requested video content may deviate from the updated predicted FoV. In this case, additional urgent EL tiles should be requested. As en-route VR video packets may contain outdated EL video data for streaming the previously predicted FoV, a packet filtering function is designed to promptly remove outdated video data that is no longer needed by clients, which saves transmission resources while adapting to viewing behavior dynamics. Given the deployment and time costs for packet filtering operations, we consider that the packet filtering function is enabled at only the ingress and egress nodes of a VR video slice. The ingress and egress nodes are equipped with the capability of higher-layer protocol header parsing to identify video data packets containing outdated video data.

2) *Caching-based Packet Retransmission*: We consider that the ingress node has a caching buffer for aggregated VR video traffic of each VR video slice. The ingress node temporarily stores copies of VR video packets sent but not acknowledged in the caching buffer for possible retransmissions. The egress node detects any random BL/EL packet loss occurring in the core network and triggers retransmission from the ingress node by using the cached packet copy in the caching buffer, instead of from a remote server to achieve lower packet retransmission delay. In addition, due to the disparate transmission reliability requirements, deadline-violated BL packets are not dropped, and lost BL packets are retransmitted to avoid video rebuffering. Deadline-violated EL packets should be directly dropped without retransmissions. Therefore, separate packet loss detection and retransmission mechanisms are designed for BL and EL packets, respectively, where the capability of differentiating EL packet losses due to deadline violations from those due to random link failures is realized at the egress node.

In addition, due to the limited capacity, caching buffer release is necessary to avoid buffer overflow. We consider two ways of releasing the cached packet copies in the caching buffer of the ingress node: event-driven and time-based. Specifically, caching buffer release can be triggered upon receiving a retransmission request by the ingress node (i.e., event-driven). If no retransmission request reception from the egress node for a certain time, a

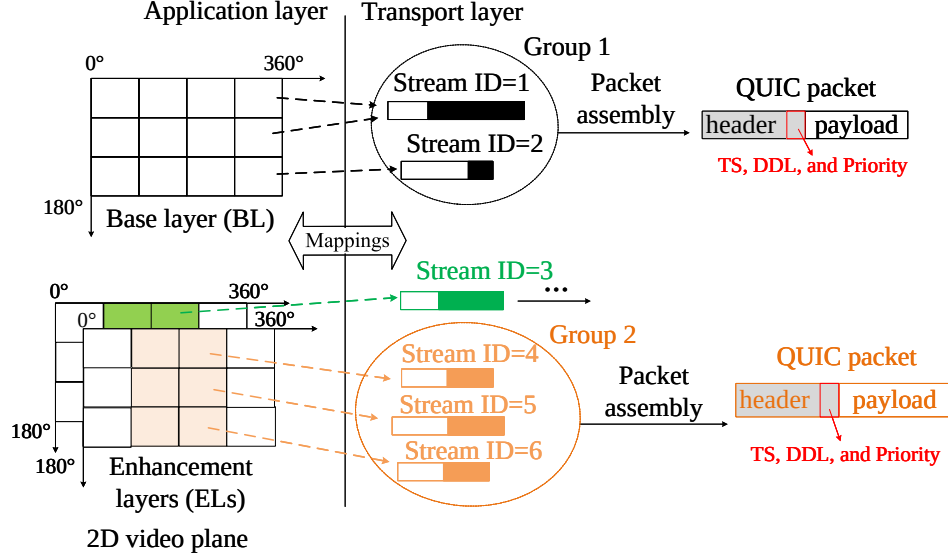


Figure 2.3: Tailored QUIC with tile-to-stream mappings.

timeout timer is set at the ingress node to periodically release the cached packet copies in the caching buffer (i.e., time-based). The detailed designs and workflows of the proposed two protocol functionalities are discussed in the next section.

2.2 Protocol Customization for VR Video Transmission

In this section, we elaborate on our main designs in developing the customized slice-level transmission protocol for supporting enhanced VR video data transmission and describe the E2E VR video transmission process with our proposed protocol.

2.2.1 Tailored QUIC

To support finer-grained QoS for 360° VR video streaming by accommodating various VR video packet properties (i.e., transmission priority, deadline, and reliability requirement) and adapting to viewer viewing behavior dynamics, we first revisit the QUIC protocol and tailor it from the following three aspects:

- Mapping relations between streams in a QUIC connection and requested video tiles by a client (or an HMD) are established. As shown in Fig. 2.3, a one-to-one mapping between each EL tile and each QUIC stream is established, and a separate set of streams is used to transmit full-view BL. This is achieved by using one stream to carry only one EL tile request and using different streams to carry BL tile requests. Assume the mapping relations between streams and video tiles are known and kept at both the HMD and the server communicating with each other. With the established tile-to-stream mappings, operations to a specific video tile are realized by controlling the data transmission over its corresponding stream, which is helpful for adapting to viewing behavior dynamics caused by head movements.
- Multiple stream groups, each of which consists of the streams corresponding to the requested video tiles of the same (BL/EL) layer, are formed for QUIC packet assembly. One stream group is selected each time to assemble a QUIC packet where round-robin is conducted among the streams within the same group for filling each STREAM frame in a QUIC packet. The approach proposed in [26] can be applied to decide which stream group is selected to assemble a QUIC packet where a *block* in [26] represents a stream group in our case. By doing so, STREAM frames (or video data) assembled in a QUIC packet come from the requested video tiles with the same properties. Video data of different encoding layers is assembled into different types of video packets, including regular BL/EL packets and urgent EL packets; thus, the same set of operations can be enforced on each type of packets.
- Three optional fields, *Timestamp (TS)*, *Deadline (DDL)*, and *Priority*, are added to the QUIC packet header to indicate each packet's transmission priority (i.e., a BL or an EL packet) and deadline [38]. Specifically, the TS and DDL fields are used to record the time when a packet is generated and its transmission deadline, respectively. For uplink transmission, the Priority field of each video request packet is set to 1 for requesting regular video tiles after a video segment downloading is finished and is set to 2 for requesting urgent EL tiles when sudden head movements and/or FoV prediction errors occur. For downlink transmission, the Priority field of a BL packet is set to 1, and for an EL packet from the $(n - 1)$ th EL, its priority is set to n ($n > 1$).

Note that a video server can be aware of the usage of optional header fields when a user initiates a streaming request and establishes an E2E QUIC connection with the server. The video server independently conducts prioritized transmission scheduling among different stream groups. The tile-to-stream mappings between the server and the user are

maintained and updated along progressive video tile requests until the streaming session is completed.

2.2.2 Connection Establishment

Before video data transmission begins, an E2E QUIC connection needs to be established between an HMD and a VR video server. The SDN/NFV controller with a global view over the physical (core) network can check path availability for two-way communications between a pair of ingress and egress nodes. It configures a network slice and activates customized protocol functionalities (e.g., packet filtering) during the connection establishment phase to support VR video transmission over a VR video slice.

Therefore, different from the typical connection establishment process specified in [37] where signaling messages and data packets (if any) are exchanged directly between an end-device and a server in the data plane, E2E QUIC connection between an HMD and a VR video server in our case is established with the SDN/NFV controller assistance. Specifically, as shown in Fig. 2.4, the signaling and data packets (if any) exchanged between the HMD and the VR video server during the connection establishment phase, such as the Initial packet with ClientHello/ServerHello, are intercepted by the ingress and egress nodes and are encapsulated into OpenFlow packets (e.g., Packet_In or Packet_Out). These encapsulated packets go through the SDN/NFV controller in the control plane for processing.

After an E2E connection between an HMD and a VR video server is established, the ingress node acts on behalf of the HMD and communicates with the VR video server for packet transmissions on the Internet. Any packet loss on the Internet is detected and retransmitted following any state-of-the-art mechanism proposed based on QUIC [37]. Similarly, the egress node acts on behalf of the VR video server and interacts with the HMD for packet transmissions in the RAN domain. The egress node retransmits any lost packet that happens in the RAN domain. The ingress node interacts with the egress node to regulate the packet transmissions over a VR video slice in the core network. The egress node detects any packet loss and triggers the retransmission from the ingress node.

2.2.3 VR Video Data Transmission

1) VR Video Content Request

As illustrated earlier, each HMD requests video content on a segment basis. Specifically, it concurrently requests a subset of video tiles of different encoding layers for a video

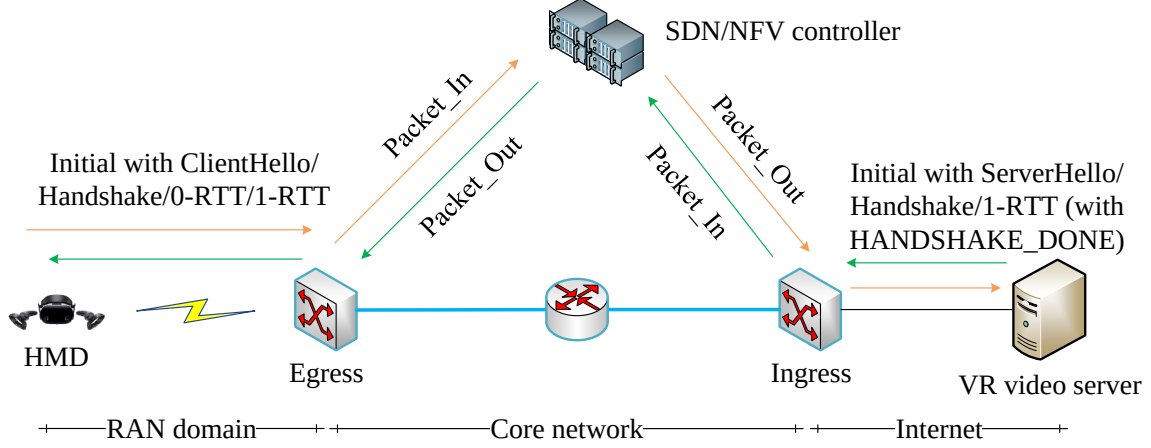


Figure 2.4: SDN/NFV-based E2E QUIC connection establishment.

segment. All video tile requests for one segment are multiplexed via different streams in a QUIC connection, as described in Subsection 2.2.1. Besides, the HMD sends additional requests for urgent EL tiles corresponding to the updated FoV after sudden head movements and/or when FoV prediction errors occur.

2) Header Conversion at Edge Switches

A slice-level packet header is designed to support the slice-level VR video packet transmission and the customized protocol functionalities presented in Subsection 2.1.3. To be compatible in terms of protocol operations with the tailored QUIC implemented at end hosts, header conversion and reversion are performed at the ingress/egress node, such as through the tunneling technique [39]. A slice-level packet header format is given in Fig. 2.5 which includes three important fields, i.e., *Slice ID*, *Packet Number Space*, and *Slice-level Packet Number*. The Slice ID field is used for slice identification and packet forwarding, upon which data packets are transmitted along the pre-configured routing path of the indicated core network slice. The fields of Packet Number Space and Slice-level Packet Number are designated to support the proposed customized protocol functionalities. Specifically, the Packet Number Space field is introduced where two separate packet number spaces are used to differentiate between BL and EL packets with disparate transmission reliability requirements. The Packet Number Space field achieves logical isolation and thus allows for respective operations (e.g., packet retransmission) to BL/EL packets. BL/EL packets sent by the ingress node are sequentially numbered in their respective packet number space through the Slice-level Packet Number field to maintain ordered slice-level packet trans-

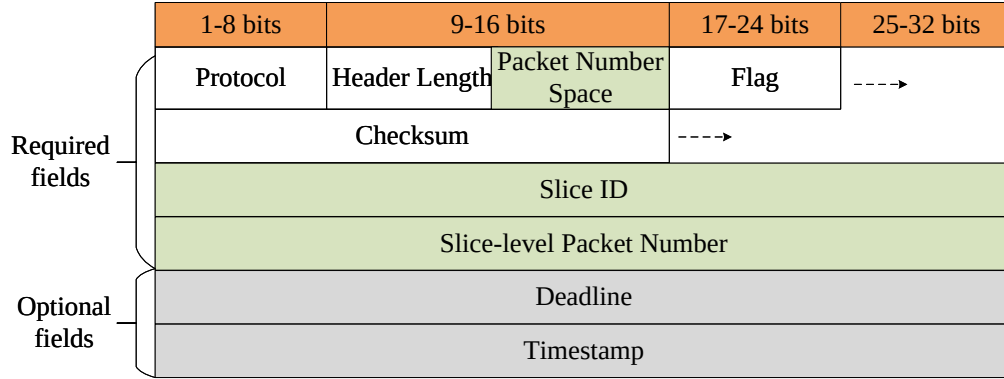


Figure 2.5: The slice-level packet header format.

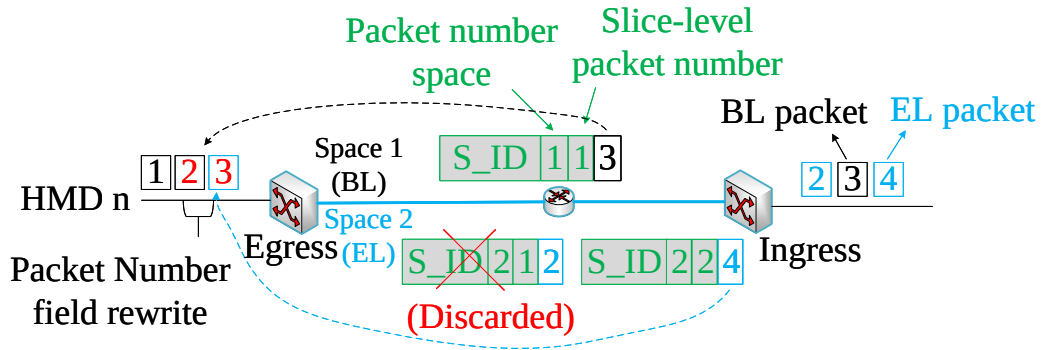


Figure 2.6: Video packet transmission over a VR video slice.

mission. This field is also used for efficient packet loss detection at the egress node and caching-based packet retransmission at the ingress node.

3) Slice-level VR Video Packet Transmission

As shown in Fig. 2.6, based on the designed slice-level packet header, when VR video packets from (remote) servers arrive at the ingress node, header conversion takes place as described above. Intermediate switches along the routing path of the VR video slice transmit packets according to the first-in-first-out (FIFO) principle and directly discard any EL packet that violates its transmission deadline. Once VR video packets reach the egress node, header reversion is performed. Due to possible deadline-violated EL packet dropping, outgoing VR video packets from the egress node may be out-of-order (OFO).

Type (i) = 0x05	Stream ID (i)	Application Protocol Error Code (i)
-----------------	---------------	-------------------------------------

Figure 2.7: The STOP_SENDING frame format.

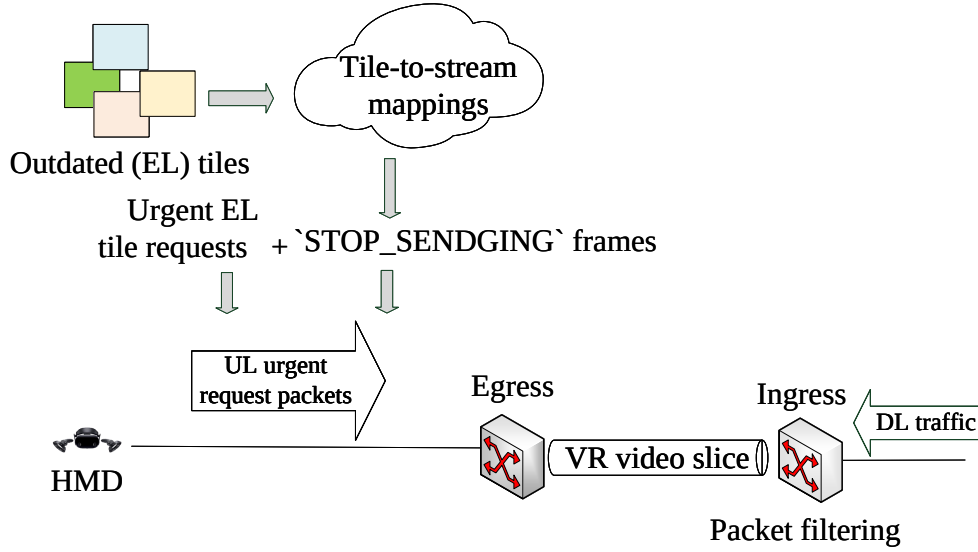


Figure 2.8: The workflow of video packet filtering.

To address this, each reverted VR video packet is renumbered by modifying the *Packet Number* field in the QUIC packet header. Specifically, the egress node records for each E2E connection the packet number of the most recently sent packet, i.e., the packet sent with the largest packet number. Then, for each reverted video packet to be transmitted, the modified packet number is the largest packet number recorded plus one.

4) Video Packet Filtering

The packet filtering functionality is implemented by leveraging flexible stream-level transmission control provided by the QUIC protocol, where the STOP_SENDING frames are sent by video clients to cease the corresponding stream data transmission of outdated EL tiles, based on the established tile-to-stream mappings in Subsection 2.2.1. The STOP_SENDING frame format is given in Fig. 2.7. The *Stream ID* field carries the ID of a stream whose STREAM frames need to be filtered and discarded, and the *Application Protocol Error Code* field contains an application-specified reason for stopping send-

ing/transmitting the STREAM frames of the indicated stream, which is FoV prediction error in the considered 360° VR video streaming case. Since head movements are tracked by HMDs, stream/video tile data transmission cessation is triggered by an HMD to begin the packet filtering operations. The detailed workflow of video packet filtering is shown in Fig. 2.8 and described as follows:

- When sudden head movements occur and cause FoV prediction errors, the set of outdated EL tiles corresponding to the previously predicted FoV is determined by the HMD. Then, the streams to be ceased are determined based on the established tile-to-stream mappings, and the corresponding STOP_SENDING frames are generated by the HMD. The generated STOP_SENDING frames may be multiplexed with urgent EL tile requests to form uplink urgent request packets. For uplink packet transmission between a pair of egress and ingress nodes, a strict priority-based scheduling policy is applied at the egress node where packets with a higher priority (i.e., the Priority field set to 1) are preemptively scheduled for transmission, while packets with the same priority are scheduled following the FIFO principle [40].
- When the egress and ingress nodes receive the uplink urgent request packets, the Stream IDs in STOP_SENDING frames are extracted for performing packet filtering on received downlink video packets. Specifically, STREAM frames with the same IDs as those extracted from the STOP_SENDING frames are ruled out from packets upon reception while other (useful) STREAM frames stay.
- When the server receives the urgent request packets, the outdated streams are terminated, and corresponding sending buffers are cleared out. In addition, STREAM frames containing video data of the requested urgent EL tiles are multiplexed and assembled into downlink urgent EL packets, as described in Subsection 2.2.1.

Note that with the established tile-to-stream mapping relations, operations to specific BL/EL tiles are realized through operating the corresponding streams. Thus, no information from the application layer is required. In addition, en-route video data of outdated EL tiles due to FoV prediction errors is filtered out, and the sending of outdated streams is terminated at the server, in response to viewing behavior dynamics for improving the transmission efficiency.

2.2.4 Packet Loss Recovery

Differentiated packet loss detection and retransmission mechanisms are required for BL and EL packets, due to their disparate transmission reliability requirements. Owing to

the Packet Number Space field in the designed slice-level packet header (see Fig. 2.5) and the separate packet number space used for BL/EL packet transmission, we design different packet loss recovery schemes for BL and EL packets respectively in the following.

BL Packet Retransmission

Since BL and EL packets are numbered and sent in order in their respective packet number space, packet loss due to random link failure (referred to as random packet loss) can be detected by the egress node based on the slice-level packet numbers of two consecutively received BL/EL packets where OFO packet arrival indicates a random packet loss¹.

As shown in Fig. 2.9(a), when a random BL packet loss is detected, the egress node sends a *Retransmission Request (RR)* packet to the ingress node and starts a timer for detecting any RR packet loss. Each RR packet contains three important fields: 1) Flag, a packet type indicator, 2) Packet Number Space, for either BL or EL packets, and 3) *Requested Packet Number*, the slice-level packet number of a lost packet that needs retransmission. This field is in the Optional fields of the designed slice-level packet header (see Fig. 2.5). When the ingress node receives an RR packet, it retransmits the lost packet using the previously cached packet copy in the caching buffer, as introduced in Subsection 2.1.3.

EL Packet Retransmission

Since deadline-violated EL packets are directly discarded without retransmissions, the egress node should be equipped with the capability of differentiating between EL packet losses due to random link failure and due to transmission deadline violation. To achieve this, we propose that each intermediate switch generates a dummy packet in place of consecutively discarded EL packets due to deadline violation, as shown in Fig. 2.9(b). In each dummy packet, the slice-level packet number of the first dropped EL packet is recorded in the Slice-level Packet Number field of the slice-level packet header. The number of consecutively dropped EL packets is put in the Optional fields. Besides, the egress node detects a random EL packet loss via OFO packet reception as in the case of BL packets.

When a random EL packet loss is detected by the egress node, similarly, it sends an RR packet to the ingress node, while EL packet losses due to transmission deadline violation,

¹When a VR video slice between a pair of ingress and egress nodes has a multi-path virtual network topology, a packet scheduling algorithm needs to be implemented at the ingress node where BL and/or EL packets are managed to be sent over different paths according to their discrepancies in terms of link capacity and packet delay, to ensure in-order packet reception at the egress node [41, 42].

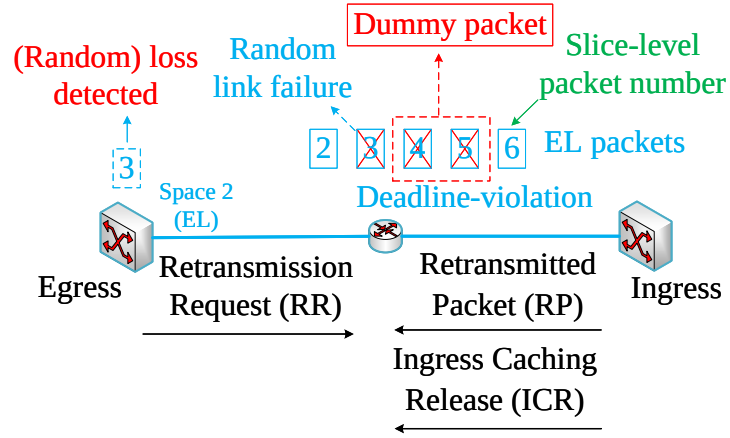
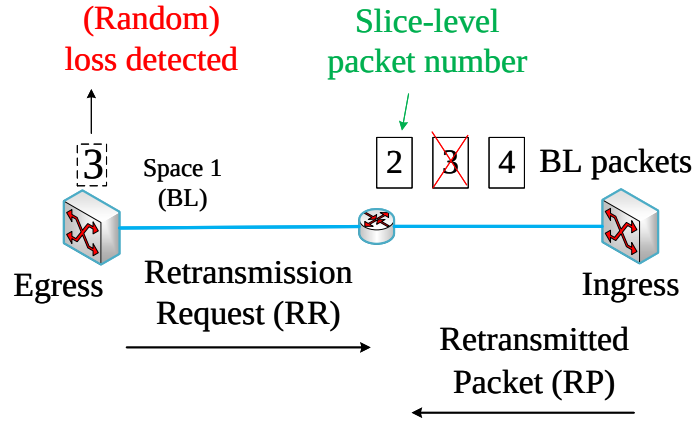


Figure 2.9: Caching-based BL/EL packet loss recovery

indicated by dummy packets, are neglected. Then, the ingress node retransmits the lost EL packet if its copy is still stored in the caching buffer upon receiving the RR packet. Note that a lost EL packet is worth retransmission only if its transmission deadline is not exceeded. If the requested EL packet copy is already discarded from the caching buffer due to deadline violation, the ingress node responds with an *Ingress Caching Release (ICR)* packet to avoid meaningless retransmission requests from the egress node. In an ICR packet, the smallest slice-level packet number among the cached EL packet copies in the caching buffer is recorded in the optional field of *Minimum (Slice-level) EL Packet Number* in the slice-level packet header. When the egress node receives an ICR packet, all the randomly lost EL packets with slice-level packet numbers smaller than the value indicated in the Minimum (Slice-level) EL Packet Number field no longer need retransmissions.

Caching Buffer Release

The caching buffer of the ingress node needs to be released regularly to avoid overflow. Consequently, we use event-driven and time-based methods for caching buffer release. For the event-driven method, a received RR packet is used as a trigger to release the cached BL/EL packet copies. Specifically, take BL packets as an example, when the ingress node receives an RR packet requesting retransmission of the BL packet with slice-level packet number i , all the cached BL packet copies with slice-level packet numbers smaller than i are released. For the time-based method, in the case of no random BL/EL packet loss happening in the core network for a certain period of time, a timeout timer is set at the ingress node to periodically release the cached packet copies. In addition, due to the low transmission reliability requirement of EL packets, the ingress node can proactively discard deadline-violated EL packet copies in the caching buffer without notifying the egress node.

The proposed caching-based packet retransmission scheme is efficient in terms of reduced signaling overhead (i.e., ACK frames) for packet loss detection, adaptiveness to disparate transmission reliability requirements, and better usage of link transmission resources due to deadline-violated EL packet dropping with no retransmission.

2.3 Transport Parameter Calculation

The workflow of caching buffer release at the ingress node is shown in Fig. 2.10, including the event-driven method based on a received RR packet and the time-based method by setting up a periodic timer. Specifically, during the timer initialization phase, the ingress node transmits BL/EL packets received from remote servers on the Internet for the first

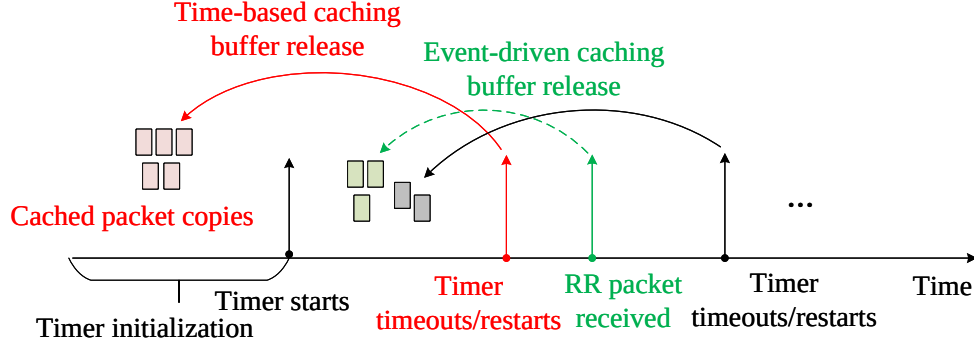


Figure 2.10: The workflow of event-driven and time-based caching buffer releases.

time and stores their copies in the caching buffer. Then, the ingress node starts the periodic timer while continuing to receive packets and put their copies in the caching buffer. Every time the timer times out, the ingress node releases the packet copies that are stored before the timer starts from the caching buffer. The periodic timer restarts when it times out.

There are two key transport parameters to be determined in the proposed caching-based packet retransmission scheme. First, the duration of the timer initialization phase and the cycle of the periodic timer need to be carefully decided, such that the cached packet copies in the caching buffer are released after they are received by the egress node, and the caching resources are efficiently utilized. If a *'release-before-arrival'* event happens to a packet that is randomly lost during the transmission, the lost packet has to be retransmitted from a remote server instead of from the ingress node, leading to a large packet retransmission delay. Second, considering the two caching buffer release methods, a minimum caching buffer size can be derived such that, on average, copies of arriving packets at the ingress node can be put into the caching buffer without causing buffer overflow². Since each RR packet must be received during a cycle of the periodic timer, the minimum caching buffer size required for the event-driven (caching buffer) release method is smaller than that for the time-based release method. Therefore, in this section, we focus on the periodic caching buffer release (i.e., time-based) and derive the minimum caching buffer sizes (in packet) required for BL and EL packets.

Specifically, we consider the *'best case'* where there are no deadline-violated and outdated EL packets being discarded during the slice-level VR video packet transmission. In

²Extra caching resources need to be assigned against the burstiness of traffic arrivals, which is out of the scope of this work.

fact, the amount of outdated EL video data to be filtered out and the number of additionally requested urgent EL tiles depend on the FoV prediction error, which affects the aggregated EL traffic arrival rate at the ingress node. Thus, without loss of generality, we consider the aggregated BL and EL traffic of the same VR video slice as two independent Poisson processes [43]. For the time-based caching buffer release method, the minimum required caching buffer size, the duration of the timer initialization phase, and the cycle of the periodic timer should be determined based on the estimated average E2E packet delay for traversing a VR video slice, as discussed in the following.

2.3.1 E2E Packet Transmission Analysis

As illustrated in Subsection 2.1.1, we consider that a VR video slice between a pair of ingress (s_0) and egress nodes (s_W) in the core network has a linear topology consisting of $W + 1$ nodes interconnected by W transmission links. The link capacity (in packet/s) of node s_i is denoted by μ_i . We denote the aggregated BL and EL packet arrival rates at the ingress node by λ^b and λ^e , respectively.

Packet transmission at the first node (i.e., the ingress node) is modeled as an $M/D/1$ queuing system with the total packet arrival rate $\lambda_0 = \lambda^b + \lambda^e$ and transmission rate μ_0 . The average packet delay at the first node is given by [44]

$$T_0 = \left(1 + \frac{1}{2} \cdot \frac{\rho_0}{1 - \rho_0}\right) \cdot \frac{1}{\mu_0} \quad (2.1)$$

where $\rho_0 = \frac{\lambda_0}{\mu_0}$ is the traffic intensity at the first node, indicating the fraction of time the system is busy.

Next, we focus on the second node. Packet transmission at the second node is closely correlated to its preceding node regarding link transmission rate and packet departure process. Specifically, if the transmission rate of the second node is no less than that of the first node, i.e., $\mu_0 \leq \mu_1$, an arriving packet at the second node is immediately transmitted before the arrival of the next packet. Thus, packet delay at the second node, T_1 , consists of only packet transmission delay, i.e., $T_1 = \frac{1}{\mu_1}$, and packet departure process at the second node remains the same as that at the first node. On the other hand, if $\mu_0 > \mu_1$, newly arriving packets may wait in the second node's transmission queue when a packet is being transmitted. In this case, both packet queuing and transmission delays exist and should be considered.

Let random variable Z_0 be the inter-departure time of successive packets departing from the first node. If a departing packet sees a nonempty queue, then $Z_0 = \frac{1}{\mu_0}$. Otherwise,

$Z_0 = \xi + \frac{1}{\mu_0}$, where ξ represents the time interval from the departure of a packet to the next packet arrival. Due to the memoryless property of Poisson arrivals, ξ has the same exponential distribution with parameter λ_0 as the packet inter-arrival time. It is seen that packet departures from the first node follow a mixed process alternating between Poisson and deterministic processes. Due to the level crossing property and the fact that the Poisson Arrivals See Time Averages (PASTA) property holds for an M/D/1 queuing system [44], the steady-state probability of queue length seen by a departing packet is the same as that seen by an arriving packet or at an arbitrary time [45]. Hence, the mean and variance of packet inter-departure time at the first node, Z_0 , are derived as

$$\begin{aligned} E[Z_0] &= \rho_0 \cdot \frac{1}{\mu_0} + (1 - \rho_0) \cdot E\left[\xi + \frac{1}{\mu_0}\right] = \frac{1}{\lambda_0} \\ D[Z_0] &= E[(Z_0 - E[Z_0])^2] = \frac{1}{(\lambda_0)^2} - \frac{1}{(\mu_0)^2}. \end{aligned} \quad (2.2)$$

We can observe from Eq. (2.2) that, when the link transmission rate μ_0 is large, packet departures from the first node approach to a Poisson process. When λ_0 increases to approach μ_0 , i.e., when the queuing system is heavily loaded, packet departures from the first node approach to a deterministic process, which is consistent with the preceding analysis. To achieve the independence between two consecutive transmission nodes for analysis tractability, under the assumption of a large transmission rate, we approximate the packet departures from a node (or the packet arrivals at its subsequent node) as a Poisson process. Therefore, when $\mu_0 > \mu_1$, the packet delay at the second node, T_1 , can be calculated according to Eq. (2.1) with traffic intensity $\rho_1 = \frac{\lambda_0}{\mu_1}$ and transmission rate μ_1 .

Finally, the E2E (BL/EL) packet delay for traversing a VR video slice in the core network is given by

$$T^E = \sum_{i=0}^{W-1} T_i \quad (2.3)$$

where T_i is the packet delay for passing through the i -th node along the route of the VR video slice.

For the time-based caching buffer release, the cycle of the periodic timer should be set based on the estimated E2E packet delay (i.e., Eq. (2.3)). Specifically, the timer cycle should be at least equal to or greater than the estimated E2E packet delay, such that the cached packet copies in the caching buffer are released after the egress node has successfully received them. In that case, any random BL/EL packet loss happening during the slice-level packet transmission is retransmitted from the ingress node instead of from a remote server on the Internet.

2.3.2 Minimum Required Caching Buffer Size

Let k_I and k_c be the initialization phase duration and the cycle of the periodic timer, respectively. As shown in Fig. 2.10, k_I is the time interval that the ingress node transmits packets received from remote servers for the first time. If $k_I < k_c$, in order to ensure that the set of BL packet copies for each periodic caching buffer release is released after they are successfully received by the egress node, the minimum caching buffer size required for BL packets is $\lceil \lambda^b (2k_c) \rceil$. However, the caching resource utilization during the initialization phase is low. If $k_I > k_c$, in order to ensure that the packet copies corresponding to the set of BL packets transmitted by the ingress node during the initialization phase can be stored at the caching buffer and be released after they reach the egress node, the minimum required caching buffer size for BL packets is $\lceil \lambda^b (k_I + k_c) \rceil$. However, in this case, caching resources are not efficiently utilized after the cached packet copies corresponding to the set of BL packets transmitted during the initialization phase are released. Therefore, the optimal initialization phase duration is $k_I^{opt} = k_c$, and the minimum caching buffer size required for BL packets, C^b , is given by

$$C^b = \lceil \lambda^b (2k_c) \rceil, k_c \geq T^E. \quad (2.4)$$

Similarly, for EL packets, due to the low transmission reliability requirement, deadline-violated EL packets are directly released from the caching buffer. Let d^e be the average transmission deadline of arriving EL packets at the ingress node. Thus, the minimum required caching buffer size for EL packets, C^e , is given by

$$C^e = \lceil \min(\lambda^e (2k_c), \lambda^e d^e) \rceil. \quad (2.5)$$

The minimum total caching buffer size required for both BL and EL packets is given by

$$C^{min} = C^b + C^e. \quad (2.6)$$

To verify the performance of the proposed caching-based packet retransmission scheme with the caching buffer size set according to Eq. (2.6), we consider a VR video slice with a linear topology which consists of 8 transmission nodes including the ingress and egress nodes (i.e., $W = 7$). The link transmission rates are set as [400, 400, 300, 300, 200, 200, 200] in packet/s, which includes both of the cases when there is or is no queuing delay at a node. The BL and EL traffic arrival rates are $\lambda^b = 100$ and $\lambda^e = 50$ in packet/s, respectively. We set the cycle of the periodic timer by the estimated E2E packet delay. The results show that the caching miss ratio caused by ‘*release-before-arrival*’ packets is 0.83%, which means that

Table 2.1: Main simulation parameters

Parameters	Value
The number of video segments	15
Video segment duration	2 s
The number of encoded packets per video tile	10
QUIC packet size	1200 bytes
Video tiling layout	4×8
FoV	$100^\circ \times 100^\circ$
FoV prediction error (in terms of pitch and yaw)	$\mathcal{N}(x^\circ, 10^\circ), x = [0 : 15]$
Link random loss rate	$[0.5 : 0.5 : 8]\%$
Congestion interval duration	$[0 : 0.5 : 7]$ s

most of the cached packet copies are released after they are received by the egress node. A near-optimal packet retransmission delay is achieved, compared to the optimal case when the caching miss ratio is 0. Therefore, the proposed caching-based packet retransmission scheme with the periodic timer cycle and the caching buffer size set based on Eq. (2.3) and Eq. (2.6) performs well. Any randomly lost BL/EL packet is highly-likely retransmitted by the ingress node using the corresponding packet copy in the caching buffer.

2.4 Performance Evaluation

In this section, we evaluate the effectiveness of our proposed transmission protocol based on real data traces³ [46]. The main simulation parameters are given in Table 2.1.

2.4.1 Data Preprocessing

The selected data traces contain viewing orientations of 50 subjects watching ten 360° VR videos from YouTube with Oculus Rift DK2 being the HMD. The ERP is adopted for spherical VR video projection and storage. We choose the ‘Shark Shipwreck’ video and extract its first 30s for experiments. The viewing trajectories are given in radians in

³<https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour>

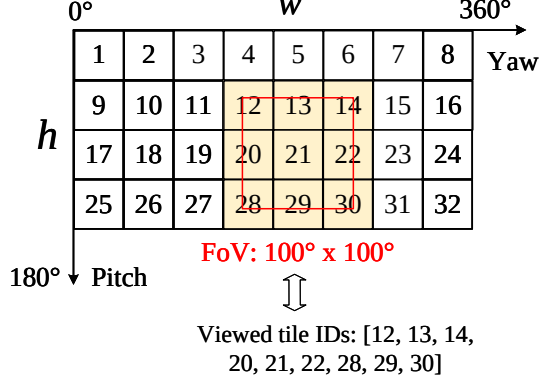


Figure 2.11: The considered video tiling layout in the (projected) 2D video plane.

terms of yaw (from $-\pi$ to π) and pitch (from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$). In the ERP-formatted 2D video plane, we first conduct sphere-to-plane coordinate transformation based on Eq. (2.7) and Eq. (2.8), which maps the yaw (θ_i) and pitch (φ_i) of a spherical video to the horizontal ($w_i \in [0^\circ, 360^\circ]$) and vertical ($h_i \in [0^\circ, 180^\circ]$) coordinates in the 2D video plane.

$$\theta_i = w_i \cdot \frac{2\pi}{W} - \pi, W = 360^\circ \quad (2.7)$$

$$\varphi_i = h_i \cdot \frac{\pi}{H} - \frac{\pi}{2}, H = 180^\circ. \quad (2.8)$$

In addition, we consider a 4×8 tiling layout, as shown in Fig. 2.11. The panoramic video scene is partitioned into 32 video tiles, each of which covers a $45^\circ \times 45^\circ$ (square) view span and is indexed in raster-scan order. We consider an FoV of $100^\circ \times 100^\circ$ and map a specific FoV to the video tile IDs it covers. Finally, according to [47], with a short prediction step (e.g., 3s), FoV prediction errors for either yaw or pitch can be reasonably assumed to follow a Gaussian distribution. Therefore, we add Gaussian noises to the selected data traces to generate the corresponding noised traces with FoV prediction errors for simulation purposes.

2.4.2 Simulation Settings

The considered network scenario in the simulation is shown in Fig. 2.12. Three video clients download video segments from remote VR video servers on the Internet where QUIC are

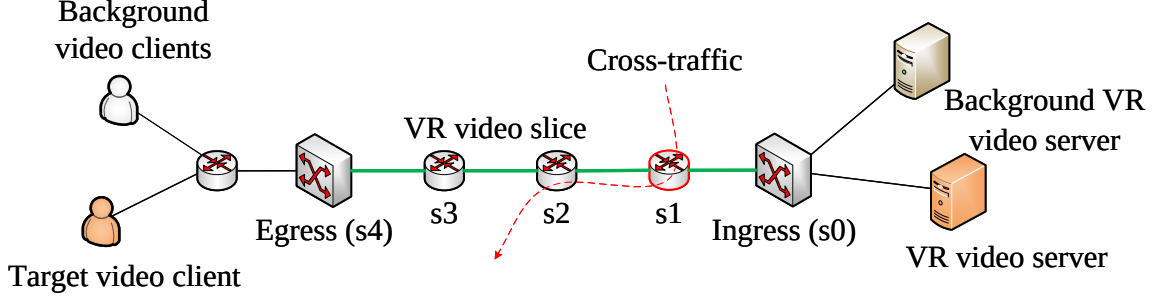


Figure 2.12: The considered network scenario in the simulation.

implemented with *aiouic*⁴. The aggregated video traffic traverses a VR video slice between a pair of ingress and egress nodes in the core network. Network congestion may happen when different network slices or service traffic pass through the same path and share the transmission resources. In the evaluation, we throttle the link transmission capacity (in packet/s) of intermediate switch s_1 available to the considered VR video slice to represent the case of network congestion due to cross-traffic from other services.

The duration of each video segment is set to 2 seconds [10], and each video segment is encoded into one BL and one EL. Each video client requests video content on a segment basis, and is considered to progressively prefetch 1 video segment ahead only. Specifically, for each video segment downloading, a video client requests full-view BL and only the EL tiles covered by the FoV. In view of the decoding dependency between BL and EL tiles, we consider that BL packets are sent before EL packets by the VR video servers. For simplicity, we assume each video client suddenly rotates head at most once in the middle of watching a video segment. If a video client suddenly rotates head, additional urgent EL tiles corresponding to the updated predicted FoV and the current FoV (after the head rotations) are requested for FoV prediction correction and current viewing experience compensation. The outdated EL tiles corresponding to the previously predicted FoV are filtered out due to viewpoint prediction errors.

2.4.3 Performance Metrics

We compare our proposed transmission protocol with QUIC under different network conditions. For brevity, we denote the proposed transmission protocol without deadline-violated

⁴<https://github.com/aiortc/aiouic>

EL packet dropping, packet filtering functionality, and caching-based packet retransmission scheme by Proposed wo. dropping, Proposed wo. filtering, and Proposed wo. caching, respectively. Four performance metrics are considered in the evaluation [35]:

- 1) Segment downloading time: The time to receive all the BL tiles of a video segment, indicating the time when a video segment is ready to play smoothly with basic quality;
- 2) Segment quality: The ratio of the number of expected EL tiles timely delivered to the total number of expected EL tiles requested for a video segment, given that the corresponding BL tiles have already been received. Thus, the maximum video segment quality is 2;
- 3) Packet retransmission delay: The waiting time for receiving a retransmitted packet;
- 4) Goodput ratio: The number of received packets that contribute to the video segment quality, including all BL packets and timely-delivered expected EL packets, over the total number of packets received by a video client.

”Note that in this work, we focus on slice-level transmission protocol customization for tile-based 360° VR video streaming. Video streaming performance is evaluated from a transmission (or QoS) perspective rather than an application (or QoE) perspective, i.e., how many requested and expected BL/EL packets or tiles can be timely delivered. The streaming performance evaluated from the transmission perspective can be mapped to that from the application perspective, which is out of the scope of this work.”

2.4.4 Simulation Results

Fig. 2.13(a) and Fig. 2.13(b) show the average segment downloading time and quality with link capacity, respectively. As the link capacity of each transmission node increases, the average segment downloading time reduces, and the average segment quality improves, where our proposed transmission protocol achieves better performance compared to the benchmarks. Originally, QUIC is a reliable transport protocol that ensures reliable packet delivery. However, deadline-violated and outdated EL packets do not improve video quality⁵, which wastes link transmission resources and aggravates network congestion instead. In our proposed transmission protocol, useless EL packets for streaming a video segment are directly discarded, which does not affect the BL/EL packet transmissions of subsequent

⁵We use ‘useless EL packets’ hereafter to refer to deadline-violated and outdated EL packets for brevity.

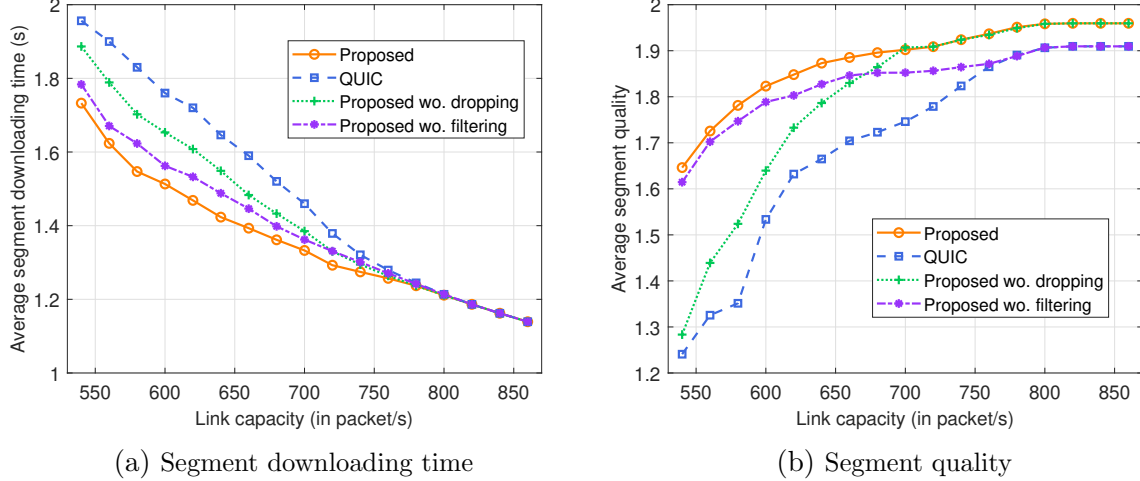
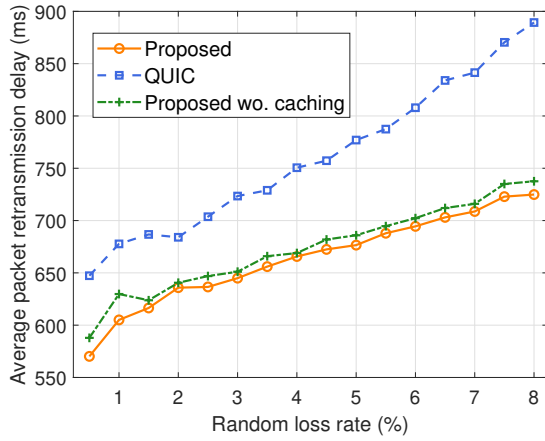


Figure 2.13: Average segment downloading time and quality vs. link capacity.

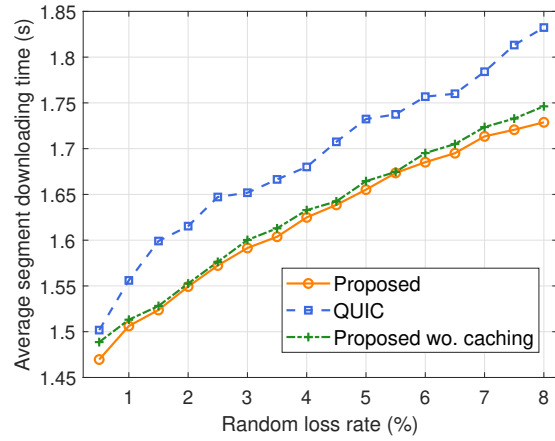
segments, while link transmission resources are saved to transmit those expected packets that may still be able to be timely delivered to the target video client in a relatively congestion-mild transmission environment. Hence, our proposed transmission protocol achieves better performance than the QUIC protocol.

For Proposed wo. filtering, deadline-violated EL packets are discarded without affecting the BL tile downloading of the next video segment, thus achieving a smaller average segment downloading time than the Proposed wo. dropping. In terms of average segment quality, as link capacity is small, there are many deadline-violated EL packets which account for the majority of useless EL packets. The Proposed wo. filtering scheme thus achieves better average segment quality than the Proposed wo. dropping. As link capacity increases, the number of deadline-violated EL packets is small. In this case, filtering out outdated EL packets brings larger segment quality improvement. For Proposed wo. filtering, part of the saved transmission resources is still used to transmit outdated EL packets which account for the majority of useless EL packets when link capacity is large. Therefore, the Proposed wo. dropping scheme achieves higher average segment quality than Proposed wo. filtering.

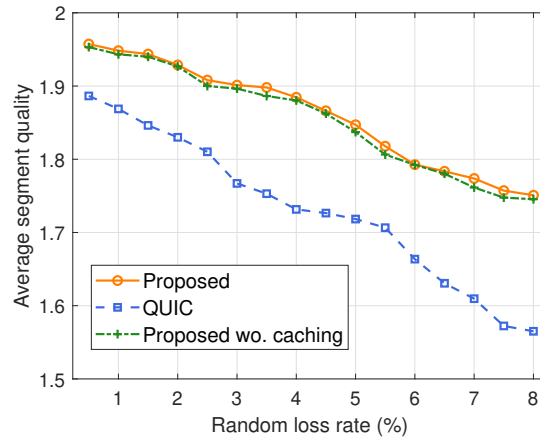
Fig. 2.14(a) - Fig. 2.14(c) show the performance of average packet retransmission delay, average segment downloading time, and average segment quality when random packet loss occurs at each hop along the route of the VR video slice. The mean of FoV prediction error is set as 5° . It can be seen that as random loss rate increases, the average packet retransmission delay and segment downloading time increase, and the average segment



(a) Packet retransmission delay



(b) Segment downloading time



(c) Segment quality

Figure 2.14: Average packet retransmission delay, segment downloading time, and segment quality vs. random packet loss rate.

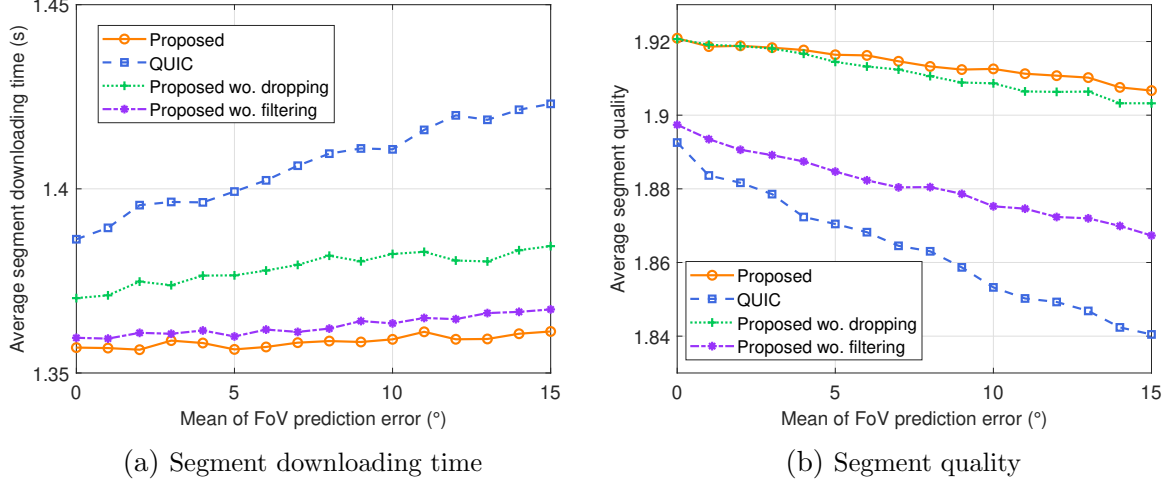


Figure 2.15: Average segment downloading time and quality vs. FoV prediction error.

quality decreases, while our proposed transmission protocol outperforms the benchmarks. As random loss rate increases, more BL/EL packets are lost and re-injected into the VR video slice for retransmissions. Retransmitted BL packets lead to increased segment downloading time, while retransmitted EL packets degrade segment quality due to deadline violations. With the proposed caching-based packet retransmission scheme, randomly lost BL/EL packets are retransmitted by the ingress node using the cached packet copies in the caching buffer, instead of by the remote servers. Thus, a smaller packet retransmission delay is achieved. Besides, deadline-violated and outdated EL packets are discarded without being further transmitted, including the retransmitted ones. On the contrary, in QUIC, all lost BL/EL packets including the deadline-violated ones are retransmitted from the remote servers. Therefore, our proposed transmission protocol outperforms the QUIC protocol. In addition, when the proposed caching-based packet retransmission scheme is deactivated (i.e., Proposed wo. caching), randomly lost BL/EL packets are retransmitted in a less congested transmission environment compared to QUIC since useless EL packets are directly dropped, which, resultingly, achieves better performance.

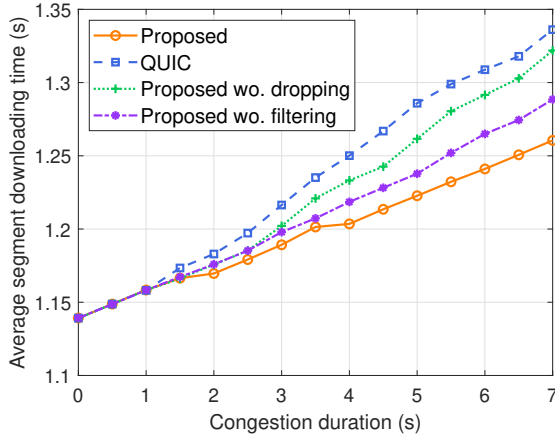
Fig. 2.15(a) and Fig. 2.15(b) show the average segment downloading time and quality with the mean of FoV prediction error. When the mean of FoV prediction error increases, the predicted FoV deviates more from the real FoV, and the corresponding sets of EL tiles covered by the predicted and the real FoVs differ more significantly from each other. Correspondingly, there will be more urgent expected EL packets to be delivered and more outdated EL packets to be filtered out. It can be seen from Fig. 2.15 that the average

segment downloading time increases and the average segment quality reduces as FoV prediction deviation becomes larger, while our proposed transmission protocol outperforms the benchmarks. In the proposed transmission protocol, useless EL packets for each video segment downloading are directly discarded without affecting the subsequent segment's BL/EL packet transmissions, and more urgent expected EL packets can be timely delivered to the target video client. For QUIC, link transmission resources are wasted for reliably transmitting those useless EL packets. Hence, our proposed transmission protocol achieves better performance.

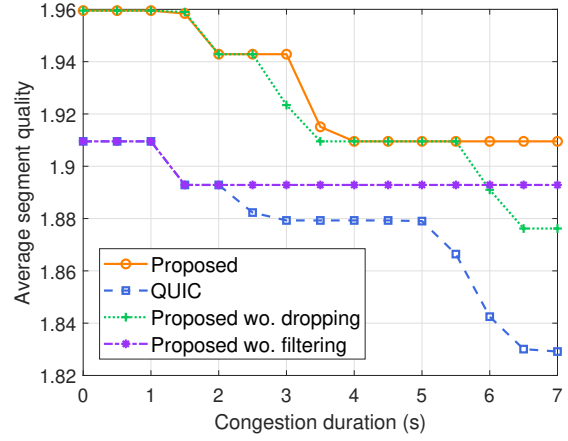
In addition, for Proposed wo. filtering, deadline-violated EL packets are discarded, which thus has minor impacts on the subsequent segment's BL tile downloading. On the other hand, as FoV prediction deviation increases, the number of outdated EL packets is larger. In this case, packet filtering operations obtain bigger segment quality improvement. For the Proposed wo. filtering, part of the saved transmission resources due to deadline-violated EL packet dropping is still consumed to transmit outdated rather than expected EL packets. Therefore, compared to the Proposed wo. dropping, the Proposed wo. filtering scheme achieves a lower average segment downloading time and quality.

Fig. 2.16(a) - Fig. 2.16(c) show the average segment downloading time, quality, and average goodput ratio with congestion duration, respectively. Specifically, starting from the 10th second in the evaluation, we throttle the link transmission rate of s_1 to 540 packet/s. We can see from Fig. 2.16 that the average segment downloading time increases and the average segment quality decreases with congestion duration, and our proposed transmission protocol outperforms the benchmarks, showing a lower sensitivity to network congestion. Besides, our proposed transmission protocol maintains a very high average goodput ratio, indicating high transmission efficiency.

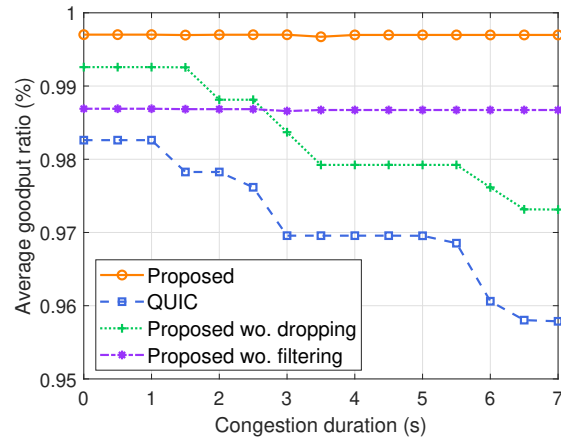
When the transmission network is congested, it takes longer time to receive all the BL tiles of a video segment, and more expected EL packets exceed their transmission deadlines, thus leading to a longer segment downloading time and degraded segment quality. In the proposed transmission protocol, useless EL packets are directly dropped without affecting the subsequent segment's BL/EL packet transmissions. For QUIC, useless EL packets are still reliably transmitted, which aggravates network congestion and hinders the timely delivery of subsequent BL and expected EL packets, thus leading to a larger average segment downloading time and lower average segment quality. In addition, for the Proposed wo. filtering, when congestion duration is short, the number of deadline-violated EL packets is small, and part of link capacity is taken up to timely deliver outdated EL packets rather than urgent expected EL packets. Thus, the Proposed wo. filtering achieves lower average segment quality than the Proposed wo. dropping. When congestion duration increases, the number of deadline-violated EL packets is large, some of which are likely outdated



(a) Segment downloading time



(b) Segment quality



(c) Goodput ratio

Figure 2.16: Average segment downloading time, segment quality, and goodput ratio vs. congestion duration.

EL packets. In that case, the Proposed wo. filtering achieves higher average segment quality than the Proposed wo. dropping. In addition, goodput ratio loss may come from deadline-violated EL packets when deadline-violated EL packet dropping is not enabled or from outdated EL packets when packet filtering functionality is not activated. With our proposed transmission protocol, almost all the EL packets received by the target video client are expected and timely-delivered, thus maintaining a very high average goodput ratio.

2.5 Summary

In this chapter, we have presented a customized slice-level transmission protocol based on QUIC for tile-based 360° VR video streaming. Various properties of video tiles are supported by tailoring the QUIC protocol, where explicit mapping relations between requested video tiles and QUIC streams are established. Two customized protocol functionalities including packet filtering and caching-based packet retransmission are proposed which remove outdated EL video data due to FoV prediction errors in prompt response to viewing behavior dynamics and efficiently perform retransmissions of lost BL and EL packets with disparate transmission reliability requirements. Key transport parameters are determined via analytical modeling. Simulation results demonstrate the effectiveness of the proposed transmission protocol. The proposed slice-level transmission protocol can also be applied to slice-based and layer-encoded (on-demand) video streaming. Slice-based and layer-encoded video streaming can be viewed as a special case of tile-based 360° VR video streaming, where the FoV of a user is fixed without head rotations, and there are no peripheral BL tiles outside the FoV to be transmitted.

Chapter 3

E2E Performance Modeling for Slice-Based Video Streaming with Layered Encoding

In order for virtual network deployment optimization and fine-grained QoS provisioning, an E2E performance analytical model is essential. It is used to evaluate the performance of slice-based video streaming with layered encoding, including E2E packet delay, deadline violation probability, and throughput. Deadline violation probability is a critical performance metric to evaluate, as packets delivered within an average delay bound may not always improve video playback smoothness or quality due to deadline violations. Deadline-violated BL packets result in the decoding failure of other timely-delivered EL packets, while deadline-violated EL packets (if not discarded) do not enhance video quality. In addition, the analytical model is also helpful in determining key transport parameters of customized protocol functionalities implemented in the video streaming slice.

Many works have conducted performance analysis of slice-based service provisioning to ensure that the QoS requirements are met in the long run. Some works focus on the RAN while others focus on the core network [48, 49, 50, 51]. For RAN slicing, it is usually modeled as a single-hop packet service process. For core network slicing scenario, a network slice is described as an SFC embedded over a physical network. A service traffic flow passes through a sequence of VNFs of the SFC in a multi-hop manner. For either a single-hop or a multi-hop case, existing analytical approaches can be categorized into stochastic network calculus (SNC)-based modeling and queuing theory-based modeling [52].

With the assumption of Poisson traffic arrivals, queuing theory-based approaches (e.g.,

M/M/1/K, M/D/1/K) are used to derive closed-form steady-state probability distribution of queue length (or packet delay) of a service node in many cases, and certain performance metrics are analyzed, e.g., average packet delay [53, 54, 55]. For a multi-hop setting, existing works model a packet service as a tandem of independent queues, and the service process at each hop is modeled as an M/M/1 or M/D/1 queuing system [56, 57, 45, 58]. Then, the E2E packet delay is calculated as a summation of per-hop packet delays. Different from queuing theory-based approaches, SNC-based approaches aim at a non-asymptotic performance bound on packet delay in the form of $Probability[delay > budget] \leq violation\ probability$ by characterizing the (affine) arrival and service envelopes [48, 59, 60]. It is able to deal with various types of packet arrivals and services, and provides a conservative estimation of such delay bound, which may deviate from the real measurements, at the expense of obtaining the exact distribution of queue dynamics and average delay performance [48, 51].

For slice-based and layer-encoded video streaming service, the impact of disparate service reliability requirements of BL and EL packets needs further investigation, which makes the E2E performance modeling, especially the analysis of deadline violation probability, challenging to perform. The packet service performance at one service node is challenging to analyze due to deadline-violated EL packet dropping. BL and EL packets may experience different average packet delays. The steady-state probability distribution of queue length at the node is difficult to obtain, thus making it challenging to determine the packet deadline violation probability. In addition, queuing dynamics at one node are closely correlated to those at its preceding node, which makes the E2E performance modeling even more challenging. The packet arrival process at one node is equivalent to the packet departure process at its preceding node, which is difficult to analyze especially when deadline-violated EL packets are directly discarded without being further forwarded, leading to reduced throughput. Therefore, the correlations between packet services at two consecutive nodes need to be investigated in the E2E performance analytical modeling.

In this chapter, we present an E2E service performance modeling for layer-encoded video traffic traversing a video streaming slice in the core network, where the disparate service reliability requirements of BL and EL packets are considered. Specifically, each NFV node along the routing path of the video streaming slice is split into two consecutive logical nodes, one for packet processing and the other for packet transmission, with respective packet service rate. The E2E packet processing and transmission over the video streaming slice is abstracted as a multi-hop packet service process with various service reliability requirements. Based on the analysis of the correlations between two consecutive nodes, a segment-based analysis framework is proposed for E2E service performance modeling. A segment consists of several consecutive nodes along the routing path of the video

streaming slice where the packet service rate of the first node is no greater than those of the rest of nodes in the segment. Within each segment, a two-stage queuing model is established to obtain the approximate steady-state probability distribution of queue length at the first node, upon which the BL/EL packet delay, deadline violation probability, and throughput at the segment are derived. In addition, the departure process from a segment is discussed based on an approximated M/D/1 queuing system, which is approximated as a Poisson process under the assumption of a large packet service rate of the first node in the segment. The independence between two consecutive segments is then achieved for analysis tractability, upon which the E2E performance metrics are derived. Extensive simulation results demonstrate the accuracy of our proposed performance analytical model and its effectiveness such as in transport parameter determination.

3.1 System Model

3.1.1 Network Model

We consider an SDN/NFV-based core network, as shown in Fig. 3.1. With the SDN/NFV controller, multiple SFCs are embedded in the core network to create different network slices for supporting diverse services (e.g., video streaming service), each of which consists of a sequence of NFV nodes and SDN switches interconnected in a predefined order. The virtual network topology of each core network slice (assumed linear for simplicity) is configured where dedicated virtual network resources including processing and link transmission resources are reserved on NFV nodes and physical links along the route for data processing and transmission, respectively. Customized protocol functionalities (e.g., retransmission [34] and selective caching [35]) can be deployed on certain network nodes (e.g., an NFV node or an SDN edge switch) within a core network slice for supporting fine-grained transmission control and enhanced service provisioning. Here, we consider that a network slice for supporting the video streaming service is deployed between each pair of ingress and egress edge switches in the core network¹.

3.1.2 Video Traffic Model

We consider a video streaming service with layered encoding. Specifically, each video stored in a remote server on the Internet is temporally divided into a sequence of video segments

¹For simplicity, each network switch is also referred to as a node.

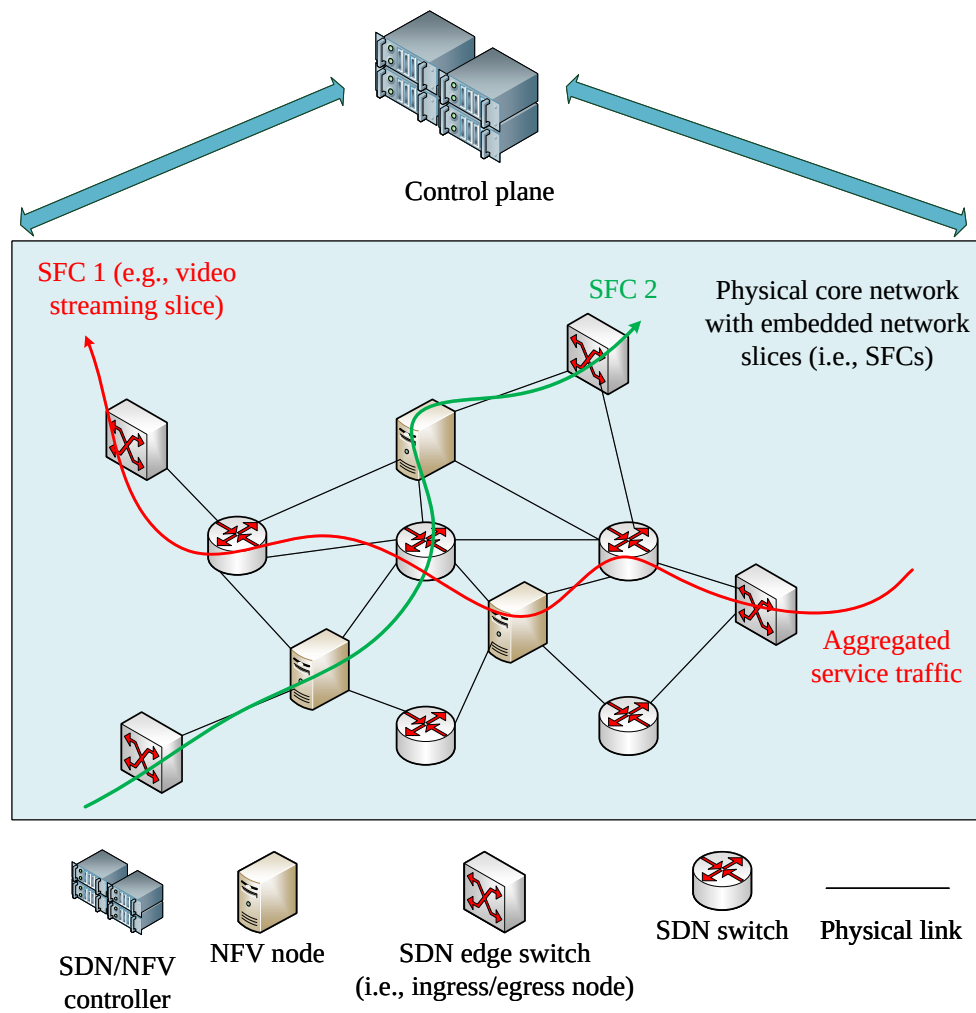


Figure 3.1: An SDN/NFV-based core network with embedded network slices.

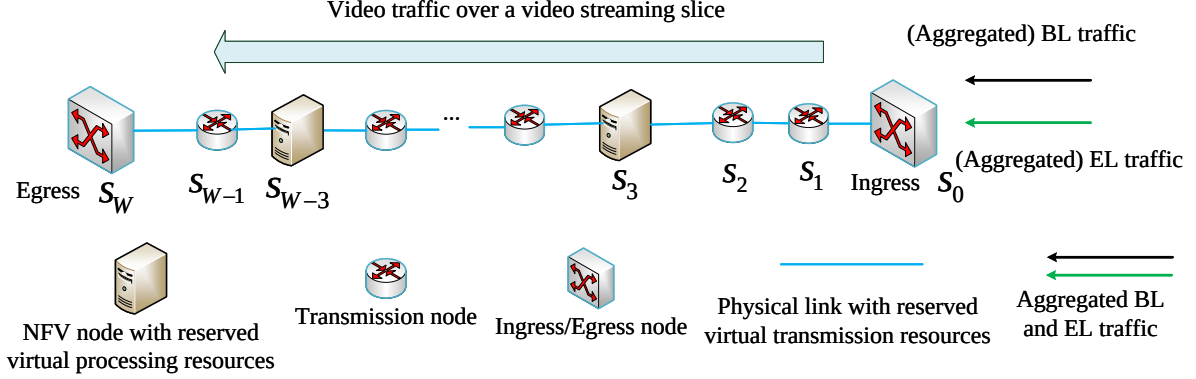


Figure 3.2: The virtual routing path of a video streaming slice with aggregated BL and EL traffic.

(or chunks), each with a playback time of 1-10 seconds [10]. Each video chunk is encoded into multiple interdependent layers, including one BL and several ELs, to support adaptive video streaming. BL ensures video robustness with basic video quality, while EL enhances video quality. An EL can be decoded only if the corresponding BL and all lower-layer ELs are received and decoded. Therefore, EL packets are optional in terms of video decoding for smooth video playback. BL packets have a higher service reliability requirement than EL packets, indicating that BL packets need to be reliably delivered to avoid long video rebuffering time, while deadline-violated EL packets that do not improve video quality should be discarded to save processing/transmission resources.

The (virtual) routing path of a video streaming network slice is shown in Fig. 3.2. For on-demand video streaming, each user progressively requests and downloads video content on a video chunk basis. That is, each user sends chunk requests at a certain time interval. The probability of a user sending a chunk request within a small time interval is small. Besides, a video streaming slice is deployed to serve a large number of users who send chunk requests independently. According to the Poisson Limit Theorem [61, 62], the aggregated requests from multiple users served by the same video streaming slice can be approximated as a Poisson process. Thus, the BL and EL traffic flows entering the same video streaming slice are considered as two independent Poisson arrival processes at the ingress node [43, 48, 55, 53], with average packet arrival rates denoted by λ_b and λ_e in packet/s, respectively. The virtual routing path between the ingress (s_0) and egress (s_W) nodes consists of $(W + 1)$ nodes including NFV nodes for data packet processing and transmission nodes for packet forwarding. Denote the packet processing/transmission rate

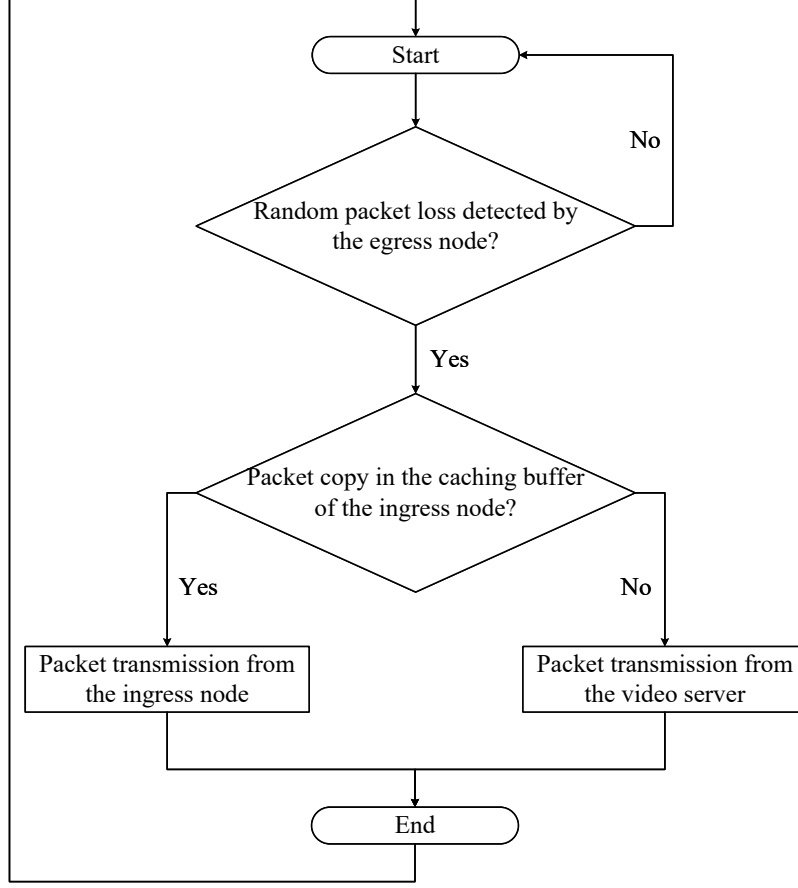


Figure 3.3: The flowchart of caching-based packet retransmission.

of NFV/transmission node i ($i = 0, 1, \dots, W$) by μ_i in packet/s. At each node, packets are transmitted/processed based on the FIFO principle, and the remaining service deadline of each EL packet is checked. Deadline-violated EL packets are directly discarded without being further forwarded.

3.1.3 Caching-based Packet Retransmission Functionality

To support enhanced layer-encoded video packet services, we propose a customized caching-based packet retransmission functionality to achieve efficient packet retransmission. Specifically, we consider that the ingress node has an additional caching buffer for aggregated

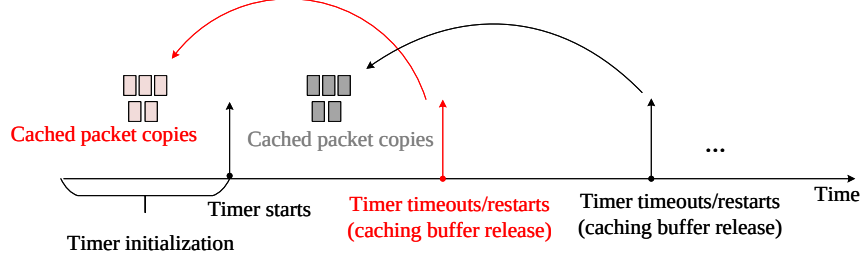


Figure 3.4: The workflow of time-based caching buffer release.

video traffic of each video streaming slice which temporarily stores video packets sent but not acknowledged for possible retransmissions. Besides, the egress node is considered to be equipped with the capability to detect any packet loss due to random link failures (referred to as random packet loss) and differentiate from the EL packet losses due to service deadline violations². The flowchart of the proposed caching-based packet retransmission is shown in Fig. 3.3. When the egress node detects a random BL/EL packet loss, it sends a retransmission request to the ingress node. The ingress node retransmits the lost packet by retrieving the corresponding packet copy in the caching buffer. If the requested packet copy is unavailable at the ingress node, the lost packet is then retransmitted by a remote server with a longer packet retransmission delay.

Due to limited caching resources, the caching buffer at the ingress node should be regularly released to avoid buffer overflow. We consider a time-based caching buffer release mechanism. Specifically, a timeout timer is set at the ingress node to periodically release the cached packet copies in the caching buffer. The workflow of time-based caching buffer release is shown in Fig. 3.4. The timer initialization phase is the time interval where the ingress node transmits BL/EL packets received from remote servers for the first time and stores their packet copies in its caching buffer. Then, the ingress node starts the periodic timer and continues putting the copies of subsequently received packets into the caching buffer. Every time the periodic timer times out, the ingress node releases the cached packet copies that are stored before the timer starts from the caching buffer. The periodic timer restarts when it times out (or after each caching buffer release).

In order to balance between the performance of the proposed caching-based packet retransmission scheme and the caching resource utilization, two key transport parameters

²Due to the low service reliability requirement, deadline-violated EL packets do not need to be retransmitted.

need to be determined. The first key transport parameter is the cycle of the periodic timer, which needs to be set based on the estimated average E2E BL/EL packet delay³, such that, on average, the cached packet copies in the caching buffer are released after the original packets are successfully received by the egress node. If the cycle of the periodic timer is smaller than the estimated average E2E BL/EL delay, video packets transmitted by the ingress node will arrive at the egress node after the corresponding cached packet copies are released from the caching buffer (referred to as ‘*release-before-arrival*’ packets), and any lost packets will be retransmitted from remote servers on the Internet, compromising the performance of the proposed caching-based packet retransmission scheme. Correspondingly, a minimum required caching buffer size (in packets) for BL/EL packets, as the other key transport parameter, needs to be decided, with which copies of arriving packets at the ingress node can be put into the caching buffer instead of being dropped due to buffer overflow⁴. If the caching buffer size is set to be larger than the minimum required value, the caching resources are not efficiently utilized.

Suppose we obtain the average E2E BL and EL packet delays for traversing the video streaming slice between a pair of ingress and egress nodes, denoted by T_b and T_e respectively. Let k_I and k_c be the duration of the timer initialization phase and the cycle of the periodic timer, respectively. Based on the above discussions, we have $k_I = k_c$, $k_c = T_b$ for BL packets, and $k_c = T_e$ for EL packets. Thus, the minimum required caching buffer size for BL packets, $C_{\min}^b$, is given by

$$C_{\min}^b = \lceil \lambda_b \cdot 2T_b \rceil \quad (3.1)$$

where $\lceil \cdot \rceil$ is the ceiling function. Besides, Let $\bar{d}^e$ be the average service deadline of arriving EL packets at the ingress node. Since deadline-violated EL packets are directly released from the caching buffer due to a low service reliability requirement, the minimum required caching buffer size for EL packets, $C_{\min}^e$, is given by

$$C_{\min}^e = \lceil \min(\lambda_e \cdot 2T_e, \lambda_e \bar{d}^e) \rceil. \quad (3.2)$$

Based on Eqs. (3.1) - (3.2), we obtain the minimum required total caching buffer size for both BL and EL packets, denoted by $C_{\min} = C_{\min}^b + C_{\min}^e$.

³The time for the egress node to detect a random packet loss until the ingress node receives the retransmission request is neglected

⁴Additional caching resources need to be assigned against the burstiness of packet arrivals, which is out of the scope of this work.

3.2 E2E Service Performance Modeling

In this section, we discuss our E2E performance modeling for (aggregated) video traffic with layered encoding delivered over a video streaming slice. The analytical model determines the average E2E packet delay, the average E2E deadline violation probability, and the average E2E throughput, while considering the discrepant service reliability requirements of BL and EL packets. For any NFV node along the routing path of the video streaming slice, arriving packets are first queued for processing, and the processed packets are queued for transmission to the subsequent node. Accordingly, each NFV node is split into two consecutive logical nodes, one for packet processing and the other for transmission. In the following, any processing or transmission node is referred to as service node s_i ($i = 0, 1, \dots, W'$), where $(W' + 1)$ is the total number of nodes composing the video streaming slice with the consideration of the NFV node logic splitting. The E2E service over the video streaming slice is abstracted as a multi-hop packet service system with various service reliability requirements, where each hop/node is either a processing or transmission node. Here, we suppose each (abstracted) node s_i along the considered video streaming slice has a deterministic service rate μ_i (i.e., constant service time), allocated and reserved by the SDN/NFV controller at the time of slice deployment.

3.2.1 Segment-Based Analysis Framework

During multi-hop packet service, the queuing dynamics of each intermediate node are closely correlated to the packet service rate or packet departure process of its preceding node. Specifically, for any intermediate node between the ingress and egress nodes, s_i ($i > 0$), the packet arrival process is a slightly delayed version of the packet departure process at its preceding node s_{i-1} (due to propagation delay). In addition, given the packet service rates of nodes, if the packet service rate μ_i of s_i is greater than or equal to that of s_{i-1} , i.e., $\mu_i \geq \mu_{i-1}$, a packet arriving at s_i is immediately served before the next packet arrival. Note that the case of $\mu_i > \mu_{i-1}$ can happen when extra resources are sliced and reserved on node i to accommodate short-term traffic load fluctuations for enhancing the E2E service performance. There is no packet queuing at s_i but only deterministic packet service delay $\frac{1}{\mu_i}$, and the packet departure process at s_i statistically remains the same as that at s_{i-1} . Whether a packet exceeds its service deadline when passing through s_i directly depends on its remaining deadline and the queue length when it arrives at s_{i-1} . If $\mu_i < \mu_{i-1}$, newly arrived packets at s_i may wait in the service queue of s_i when a packet is being served. In this case, both packet queuing and service delays exist.

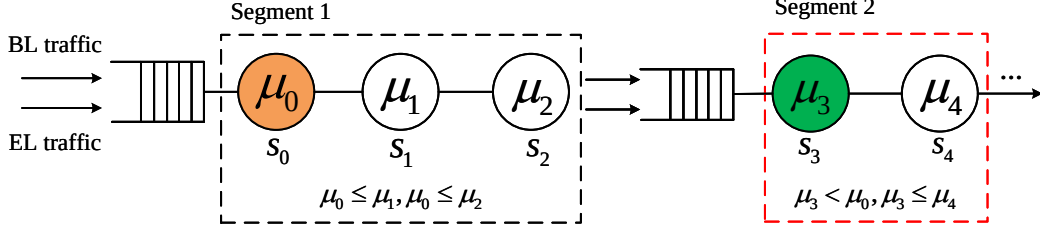


Figure 3.5: A segment-based analysis framework.

We propose a segment-based analysis framework, as shown in Fig. 3.5. Given the allocated packet service rates of nodes along the routing path of the video streaming slice, the E2E multi-hop packet service is grouped into several sequential segments. We denote by μ^g the service rate of the first node in segment g . Segment g consists of a consecutive sequence of nodes within which the packet service rates of the rest of nodes except the first one are no less than the first node's packet service rate, i.e., μ^g . Besides, we have $\mu^{g+1} < \mu^g$ for two consecutive segments g and $g+1$. Let λ_b^g and λ_e^g be the average packet arrival rates of BL and EL traffic arriving at segment g , respectively, and the total packet arrival rate at segment g is thus given by $\lambda^g = \lambda_b^g + \lambda_e^g$. Note that λ^g may not always be equal to λ^{g+1} for two consecutive segments g and $g+1$ due to possible deadline-violated EL packet dropping. In addition, let $\bar{d}^{b,g}$ and $\bar{d}^{e,g}$ be the average (remaining) service deadlines of BL and EL packets arriving at segment g , respectively. Within each segment, packet service performance at the first node is critical to analyze since there are only deterministic packet service delays at the other nodes, which is discussed in the following.

3.2.2 Service Performance Modeling at The First Segment

Given the average service deadline of EL packets arriving at the first segment, $\bar{d}^{e,1}$, the corresponding queue length bound for EL packets that can successfully pass through the first node is given by $N_e^1 = \lceil \bar{d}^{e,1} \mu^1 \rceil$. When the queue length of the first node increases to the bound, N_e^1 , arriving EL packets are dropped before passing through the first node due to service deadline violations. In that case, only BL packets with a higher service reliability requirement can pass through. The state transition diagram of queue length at the first node is shown in Fig. 3.6. It can be seen that the transition rate from state i to state $i+1$ is $\lambda_i = \lambda^1 = \lambda_b^1 + \lambda_e^1$ for $i < N_e^1$, as both arriving BL and EL packets can be served through the first node. The transition rate from state i to state $i+1$ is $\lambda_i = \lambda_b^1$ for $i \geq N_e^1$, as only arriving BL packets can pass through the first node while

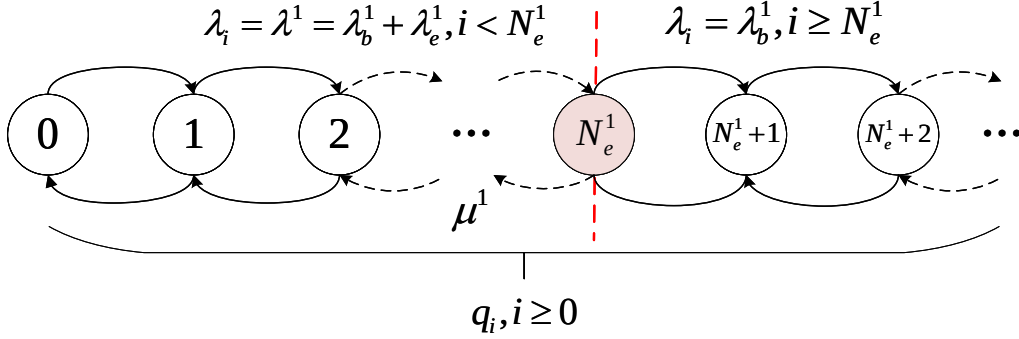


Figure 3.6: The state transition diagram of queue length at the first node in the first segment.

arriving EL packets will be discarded due to service deadline violations. Therefore, packet service at the first node is a special state-dependent M/D/1 queuing system where the packet arrival rate is state-dependent only at state $i = N_e^1$, and the packet service rate is not state-dependent and keeps the same for all states (i.e., μ^1). Let the steady-state probability distribution of queue length at the first node (at arbitrary times) be denoted by $\{q_i, i \geq 0\}$. Note that $\{q_i, i \geq 0\}$ is difficult to obtain, which involves complex Laplace Transforms (LTs) and recursive computations, and the related steady-state averages such as the mean sojourn time are challenging to derive [63]. Therefore, according to the state transition diagram shown in Fig. 3.6, we propose an approximate approach to obtain the steady-state probability distribution of queue length at the first node, denoted by $\{\hat{q}_i, i \geq 0\}$.

Specifically, we consider that packet services at the first node go through two sequential stages, as shown in Fig. 3.7. At stage one when the queue length of the first node is smaller than N_e^1 , both BL and EL packets can pass through the first node. At stage two when the queue length of the first node is larger than or equal to N_e^1 , only arriving BL packets are admitted to the queue while an arriving EL packet is discarded before being served due to service deadline violation. The stage one queuing is modeled as an M/D/1/ N_e^1 system with packet arrival rate of λ^1 and service rate of μ^1 . The steady-state probability distribution

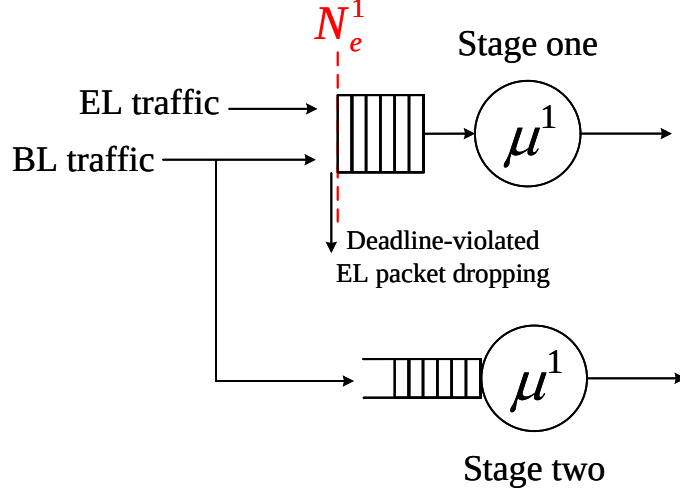


Figure 3.7: Two-stage queuing at the first node of the first segment.

of queue length, denoted by $P_j, j = 0, 1, \dots, N_e^1$, is given by [64]

$$\begin{aligned}
 P_0 &= \frac{1}{1 + \rho_1^1 b_{N_e^1 - 1}^1} \\
 P_{N_e^1} &= 1 - \frac{b_{N_e^1 - 1}^1}{1 + \rho_1^1 b_{N_e^1 - 1}^1} \\
 P_j &= \frac{b_j - b_{j-1}}{1 + \rho_1^1 b_{N_e^1 - 1}^1}, j = 1, \dots, N_e^1 - 1
 \end{aligned} \tag{3.3}$$

where $\rho_1^1 = \frac{\lambda^1}{\mu^1}$ is the traffic intensity for stage one queuing. The coefficients $\{b_n, n \geq 0\}$ are given by $b_n = \sum_{k=0}^n P_k \left((k-n) \frac{1}{\mu^1} \right)$ with $P_k(t) = \frac{(\lambda^1 t)^k}{k!} e^{-\lambda^1 t}$. In addition, the stage two queuing is modeled as an M/D/1 system with packet arrival rate of λ_b^1 and service rate of μ^1 . The steady-state probability distribution of queue length, denoted by $\pi_n, n \geq 0$, is given by [65]

$$\begin{aligned}
 \pi_0 &= 1 - \rho_2^1 \\
 \pi_1 &= (e^{\rho_2^1} - 1) (1 - \rho_2^1) \\
 \pi_n &= (1 - \rho_2^1) \left[e^{n\rho_2^1} + \sum_{k=1}^{n-1} (-1)^{n-k} e^{k\rho_2^1} \right. \\
 &\quad \left. \left(\frac{(k\rho_2^1)^{n-k}}{(n-k)!} + \frac{(k\rho_2^1)^{n-k-1}}{(n-k-1)!} \right) \right], n \geq 2
 \end{aligned} \tag{3.4}$$

where $\rho_2^1 = \frac{\lambda_b^1}{\mu^1}$ is the traffic intensity for stage two queuing.

With Eqs. (3.3) - (3.4), the approximate steady-state probability distribution of queue length at the first node, i.e., $\{\hat{q}_i, i \geq 0\}$, is obtained as

$$\hat{q}_i = \begin{cases} P_i, i < N_e^1 \\ P_{N_e^1} \pi_{i-N_e^1}, i \geq N_e^1. \end{cases} \quad (3.5)$$

Remark 1. In our approximate approach, the state transitions from $N_e^1 + i$ to $N_e^1 + (i+1)$ for $i \geq 0$ (see Fig. 3.6) are neglected, leading to approximation errors. When the packet service rate of the first node, μ^1 , increases, such that the queue length is seldomly larger than N_e^1 , more accurate approximations of $\{q_i, i \geq 0\}$ can be obtained. In addition, since the packet arrival process (or rate) is not state-dependent in each queuing stage, the PASTA property holds for both the modeled $M/D/1/N_e^1$ system and the modeled $M/D/1$ system. The approximated steady-state probability distribution at arbitrary times, $\{\hat{q}_i, i \geq 0\}$, is also the approximated steady-state probability distribution at arrival epochs.

Next, we derive the average BL/EL packet delay, deadline violation probability, and throughput for traversing the first segment. Let $\tilde{T}^1$ denote the total service time for a packet passing through the rest of nodes except the first one in the first segment. If there are i packets (including the one that is in service) at the first node, then the time for the i th packet to traverse the first segment, denoted by T_i^1 , is given by

$$T_i^1 = \begin{cases} 0, i = 0 \\ (i-1) \frac{1}{\mu^1} + \tilde{T}^1, i > 0 \end{cases} \quad (3.6)$$

where the expected residual service time of the packet in service when there are i packets in the system, denoted by $E[R_i]$, is neglected in the case $i > 0$. With the approximated steady-state probability distribution of queue length at the first node, $\{\hat{q}_i, i \geq 0\}$, there is a one-to-one mapping between $\hat{q}_i$ and T_i^1 for each $i \geq 0$.

The average BL/EL packet delay for traversing the first segment, denoted by $T_\theta^1, \theta \in \{b, e\}$, is given by

$$T_e^1 = \tilde{T}^1 + T^{M/D/1/N_e^1} + P_{N_e^1} \left((N_e^1 - 1) \frac{1}{\mu^1} \right) (\pi_0 - 1) \quad (3.7)$$

and

$$T_b^1 = P_{N_e^1} T^{M/D/1} + T^{M/D/1/N_e^1} + \tilde{T}^1 \quad (3.8)$$

where $T^{M/D/1/N_e^1}$ and $T^{M/D/1}$ are the average packet delays of the considered $M/D/1/N_e^1$ system for stage one queuing and the considered $M/D/1$ system for stage two queuing,

Table 3.1: The values of N_e^1 and j^e for different packet service rates of the first node μ^1 in the example segment

μ^1	162	164	166	168	170	172	174	176	178
N_e^1	17	17	17	17	17	18	18	18	18
j^e	15	15	16	16	16	16	16	16	17

respectively, given by [44]

$$T^{M/D/1/N_e^1} = \frac{1}{\lambda^1} \frac{1 + \rho_1^1 b_{N_e^1-1}}{b_{N_e^1-1}} \frac{N_e^1 + N_e^1 \rho_1^1 b_{N_e^1-1} - \sum_{k=0}^{N_e^1-1} b_k}{1 + \rho_1^1 b_{N_e^1-1}} \quad (3.9)$$

and

$$T^{M/D/1} = \left(1 + \frac{1}{2} \frac{\rho_2^1}{1 - \rho_2^1}\right) \frac{1}{\mu^1}. \quad (3.10)$$

In terms of deadline violation probability, taking EL packets as an illustrative example, we first find $j^e = \min_j \{T_j^1 > \bar{d}^{e,1}\}$. The queue length bound for EL packets at the first node that can successfully traverse the first segment is thus $j^e - 1$. Then, the deadline violation probability of EL packets when traversing the first segment, denoted by V_e^1 , is calculated by

$$V_e^1 = V_{e,A}^1 + V_{e,B}^1 = \left(1 - \sum_{j=0}^{j^e-2} \hat{q}_j\right) + \hat{q}_{j^e-2} a^b \quad (3.11)$$

where $a^b = \frac{\lambda_b^1}{\lambda_b^1 + \lambda_e^1}$ is the probability of a BL packet arriving at the first segment before an EL packet. The EL packet deadline violation probability at the first segment, V_e^1 , consists of two parts, i.e., $V_{e,A}^1$ and $V_{e,B}^1$, which represent the probability of queue length (of the first node) no less than $j^e - 1$ and the probability of observing $j^e - 2$ packets in the queue by an arriving BL packet, respectively. The deadline violation probability of BL packets when traversing the first segment, denoted by V_b^1 , can be calculated similarly according to Eq. (3.11) with j^e and a^b replaced by $j^b = \min_j \{T_j^1 > \bar{d}^{b,1}\}$ and $a^e = \frac{\lambda_e^1}{\lambda_b^1 + \lambda_e^1}$, respectively.

We take a segment consisting of four nodes with packet service rates $[\mu^1, 200, 200, 200]$ as an example. We set $\lambda_b^1 = 100$, $\lambda_e^1 = 50$, $\bar{d}^{b,1} = 105$ ms, and $\bar{d}^{e,1} = 100$ ms. The values of N_e^1 and j^e for different μ^1 are listed in Table 3.1. We notice that, due to the

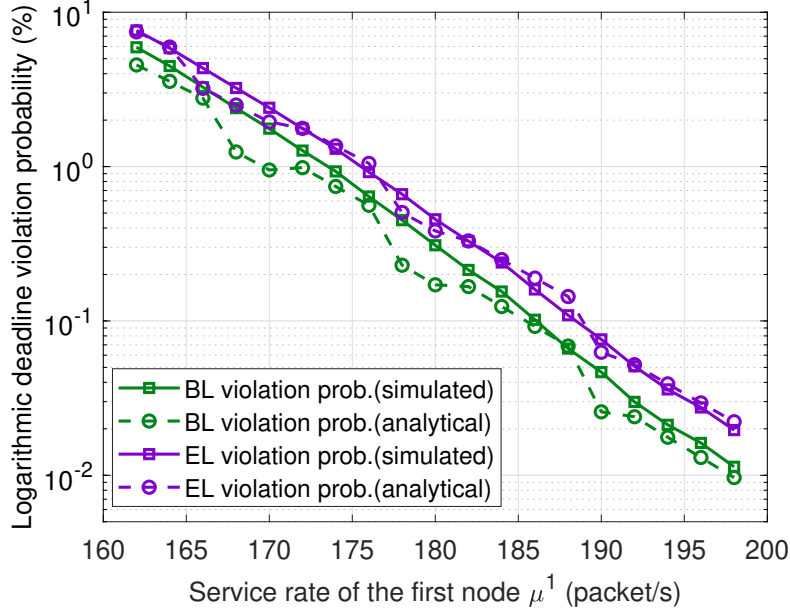


Figure 3.8: Deadline violation probability (in log-scale) of BL and EL packets at the example segment with varying packet service rates of the first node (before approximations).

$\lceil \cdot \rceil$ operator, N_e^1 does not change or increase as μ^1 increases, and the changing points of N_e^1 are different from those of j^e , which leads to the relative estimation error between the estimated V_e^1 by Eq. (3.11) and the real value oscillating (the same for V_b^1), as shown in Fig. 3.8. To address this and obtain a more accurate deadline violation probability estimate, we make some approximations based on the above observations. Specifically, let (N_e^1, j^e) and $(N_e^{1'}, j^{e'})$ be the tuples corresponding to two different packet service rates of the first node in the first segment, denoted by μ^1 and $\mu^{1'}$ where $\mu^{1'} = \mu^1 + \Delta_\mu$ (Δ_μ set as 2 packet/s in the simulations). Due to term $\tilde{T}^1$ in Eq. (3.6), intuitively, j^e should be smaller than N_e^1 . Thus, if $j^e = N_e^1$, we let $j^e = j^e - 1$. Besides, if $j^{e'} - j^e > 0$ while $N_e^{1'} = N_e^1$, we let $j^{e'} = j^{e'} - 1$. The deadline violation probability estimations after the approximations are presented in the next section.

Finally, due to the disparate service reliability requirements, deadline-violated EL packets are directly discarded without being further transmitted. Therefore, the throughputs of BL and EL traffic after traversing the first segment, denoted by $\lambda_{t,b}^1$ and $\lambda_{t,e}^1$ respectively,

are given by

$$\begin{aligned}\lambda_{t,b}^1 &= \lambda_b^1 \\ \lambda_{t,e}^1 &= \lambda_e^1 (1 - V_e^1)\end{aligned}\quad (3.12)$$

Thus, the total throughput, λ_t^1 , at the first segment is $\lambda_t^1 = \lambda_{t,b}^1 + \lambda_{t,e}^1$. In addition, the updated average service deadline of BL/EL packets arriving at the second segment is $\bar{d}^{\theta,2} = (\bar{d}^{\theta,1} - T_\theta^1)^+$, $\theta \in \{b, e\}$ where $(x)^+ = \max(0, x)$.

3.2.3 Service Performance Modeling Between Segments

Next, we analyze the packet departure process at the first segment (also the packet arrival process at the second segment), which is non-trivial due to two main reasons. First, the steady-state probability distribution of queue length seen by departing packets from stage one queuing of the first node, denoted by $q_m, m = 0, 1, \dots, N_e^1 - 1$, differs from the steady-state probability distribution of queue length at arbitrary times, i.e., $P_j, j = 0, 1, \dots, N_e^1$. Second, certain EL packets may exceed their service deadlines and be discarded when passing through the other nodes in the first segment rather than the first node. To address this, we adopt an approximate approach to analyze the packet departure process at the first segment. Specifically, from the viewpoint of departing packets from the first segment, stage one queuing of the first node can be approximated by an M/D/1 system with the packet arrival rate being λ_t^1 and service rate being μ^1 , such that the probability of queue length of the first node no less than j^e is small and can be neglected. Then, according to the PASTA property of Poisson arrivals and the level crossing property [44], the steady-state probability distribution of queue length seen by departing packets in the approximated M/D/1 system is the same as that seen by arrival epochs or at arbitrary times.

Based on the approximate M/D/1 system, let random variable Y^1 be the inter-departure time of successive packets departing from the first segment. If a departing packet, k , sees a nonempty queue, then $Y^1 = \frac{1}{\mu^1}$. Otherwise, $Y^1 = X^1 + \frac{1}{\mu^1}$, where X^1 represents the time interval from the departure of packet k to the arrival of packet $k + 1$. Due to the memoryless property of Poisson arrivals, X^1 follows the same exponential distribution as the packet inter-arrival time with parameter λ_t^1 . Therefore, it is seen that packet departure from the first segment is a mixed process alternating between deterministic and Poisson processes. The mean and variance of the packet inter-departure time, Y^1 , are given by

$$\begin{aligned}E[Y^1] &= \rho_t^1 \frac{1}{\mu^1} + (1 - \rho_t^1) E\left[X^1 + \frac{1}{\mu^1}\right] = \frac{1}{\lambda_t^1} \\ D[Y^1] &= E\left[(Y^1 - E[Y^1])^2\right] = \frac{1}{(\lambda_t^1)^2} - \frac{1}{(\mu^1)^2}\end{aligned}\quad (3.13)$$

where $\rho_t^1 = \frac{\lambda_t^1}{\mu^1}$. From Eq. (3.13), when the packet service rate of the first node, i.e., μ^1 , is large, packet departures from the first segment approach to a Poisson process. Besides, when λ_t^1 increases to approach to μ^1 , i.e., when the queuing system is heavily loaded, packet departures from the first segment approach to a deterministic process, which is aligned with the preceding analysis.

In order to achieve the independence between two consecutive segments for analysis tractability, under the assumption of a large packet service rate of the first node in the first segment, i.e., μ^1 , we approximate the packet departures from the first segment as a Poisson process with rate parameter λ_t^1 . Note that a large packet service rate of the first node in each segment, except the last one, is a necessary condition for E2E performance analytical modeling. Under this condition, the segment-level performance metrics of different segments composing the video streaming slice can be analyzed following the same logic. The E2E performance metrics are then attainable. From the perspective of a network operator (or slice orchestrator), the packet service rate of each node within a network slice can be flexibly configured as long as the service requirements are satisfied.

3.2.4 Average E2E Performance

Based on the proposed segment-based analysis framework, we derive the (average) BL/EL packet delay, deadline violation probability, and throughput at the first segment. The independence between two consecutive segments is achieved under the assumption of a large packet service rate of the first node in the preceding segment. For a video streaming slice in the core network which consists of N_s segments, the average E2E packet delay, T_θ , average E2E deadline violation probability, V_θ , and average E2E throughput, $\lambda_{t,\theta}$, of BL/EL packets are given by

$$\begin{aligned} T_\theta &= \sum_{g=1}^{N_s} T_\theta^g, \theta \in \{b, e\} \\ V_\theta &= 1 - \prod_{g=1}^{N_s} (1 - V_\theta^g), \theta \in \{b, e\} \\ \lambda_{t,\theta} &= \begin{cases} \lambda_b^1, \theta = b \\ \lambda_e^1 (1 - V_e), \theta = e \end{cases} \end{aligned} \quad (3.14)$$

where T_θ^g and V_θ^g denote the average BL/EL packet delay and deadline violation probability at the g th segment, respectively⁵.

⁵Note that when the number of segments composing a video streaming slice N_s increases, the cumulative effect of successive approximations over multiple segments in the E2E performance analytical modeling

The E2E performance metrics given in Eq. (3.14) depend only on the number of segments composing a video streaming slice, i.e., N_s , which incurs a time complexity of $O(N_s)$.

3.2.5 Discussion

In the E2E performance analytical modeling, the key feature of layer-encoded video streaming, i.e., the distinct service reliability requirements between BL and EL packets, is captured. However, the inter-dependence between different ELs, that is, an EL can only be decoded if the corresponding BL and all lower-layer ELs are successfully decoded, is not explicitly considered. In that case, EL packets of multiple ELs may also have different levels of service reliability requirements. If any EL packet of an EL is discarded due to deadline violation, the EL packets of higher ELs should also be discarded for transmission efficiency.

The proposed analytical model can be used for performance evaluation of a video streaming slice. Given the characteristics of aggregated BL and EL traffic flows (i.e., packet arrival rates λ_b and λ_e) and the slice resource configuration (i.e., the reserved packet service rates of nodes within the video streaming slice), the following key steps should be followed:

- *Step 1:* Determine the segmentation of the video streaming slice according to the slice resource configuration, as discussed in Subsection 3.2.1;
- *Step 2:* Calculate the segment-level performance metrics for each segment based on Eqs. (3.7), (3.8), (3.11) and (3.12);
- *Step 3:* Calculate the E2E performance metrics based on Eq. (3.14).

The proposed analytical model can also be helpful for slice resource reservation. Given the characteristics of aggregated BL and EL traffic flows and the specific service requirements in terms of average packet delay and/or deadline violation probability, a network operator (or slice orchestrator) can configure slice resource starting from the case where there is only one segment composing the video streaming slice, and utilize the proposed analytical model to evaluate the E2E performance until the service requirements are met. In the next section, two examples are presented to show the effectiveness of the proposed E2E analytical model for practical applications.

can lead to larger accuracy gaps.

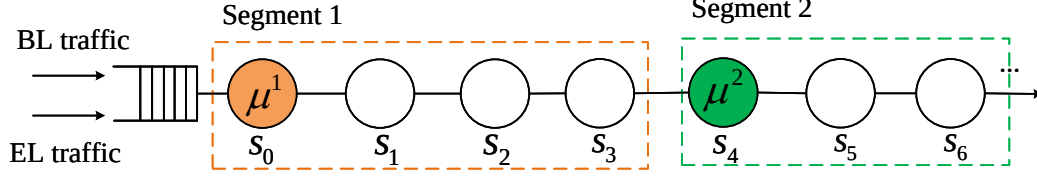


Figure 3.9: The considered virtual network topology of a video streaming slice in the simulation.

Finally, when the virtual network topology of a video streaming slice comprises multiple physical paths between the ingress and egress nodes, the proposed performance analytical model can be extended to accommodate this scenario. To achieve this, the statistics of BL/EL packet forwarding at each common node of different paths need to be known, i.e., the fraction of aggregated BL/EL packet arrivals at the common node that are forwarded towards each path. The aggregated BL/EL traffic forwarded towards different paths can be considered as a set of classified Poisson processes. Then, the proposed performance analytical model can be independently applied to different physical paths, upon which the E2E performance of aggregated BL/EL traffic entering the video streaming slice can be obtained.

3.3 Simulation Results

In this section, simulation results are presented to verify the accuracy of our proposed performance analytical model for E2E layer-encoded video packet services over a video streaming slice. The considered virtual network topology is shown in Fig. 3.9 which consists of two segments. Within each segment, each node is either a processing node or a transmission node with a deterministic packet service rate. In addition, aggregated BL and EL traffic flows at the ingress node (s_0) are considered as two independent Poisson processes with packet arrival rates of $\lambda_b^1 = 100$ and $\lambda_e^1 = 50$ packet/s [66]. The average service deadlines of BL and EL packets at the first segment are set as $\bar{d}^{b,1} = 105$ ms and $\bar{d}^{e,1} = 100$ ms [23] [67, 68, 69]. All simulations are carried out with SimPy 4.0.1 and Python 3.10.10⁶. In the evaluation, we validate the proposed performance analytical model on a segment basis. For each segment, we vary the packet service rate of the first node (e.g., μ^1

⁶Please refer to <https://simpy.readthedocs.io/en/latest/index.html> for more details.

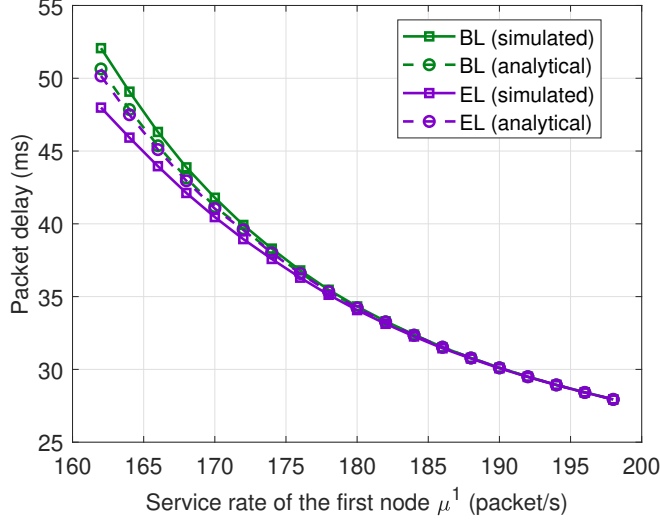


Figure 3.10: Packet delay at the first segment with varying packet service rates of the first node μ^1 .

or μ^2) to compare the performances under different traffic load conditions. Then, the accuracy of E2E performance analysis is verified. In addition, the effectiveness of the proposed performance analytical model is shown with two application examples, where M/M/1 and M/D/1, two widely used mathematical models in existing works [57, 45], are adopted as benchmarks.

3.3.1 Validation of The Proposed Performance Analytical Model

Fig. 3.10 shows the (average) BL and EL packet delays at the first segment with varying μ^1 where the packet service rates of the nodes in the first segment are $[\mu^1, 200, 200, 200]$ in packet/s. We can see that close estimations of BL and EL packet delays are achieved, especially when μ^1 is large. For small values of μ^1 , the queuing system of the first node is heavily loaded, and there are more BL packets on stage two queuing, leading to larger BL packet delay than EL packet delay. For a large μ^1 , packet services at the first node are more closely seen as an M/D/1/ N_e^1 system with a large value of N_e^1 and a very small value of $P_{N_e^1}$. Thus, BL and EL packets have a close packet delay, and more accurate packet delay estimations are achieved (see Eq. (3.7)). The deadline violation probabilities (in log-scale) of BL and EL packets at the first segment with varying μ^1 are shown in Fig. 3.11. It can be seen that the proposed performance analytical model achieves close BL/EL packet deadline

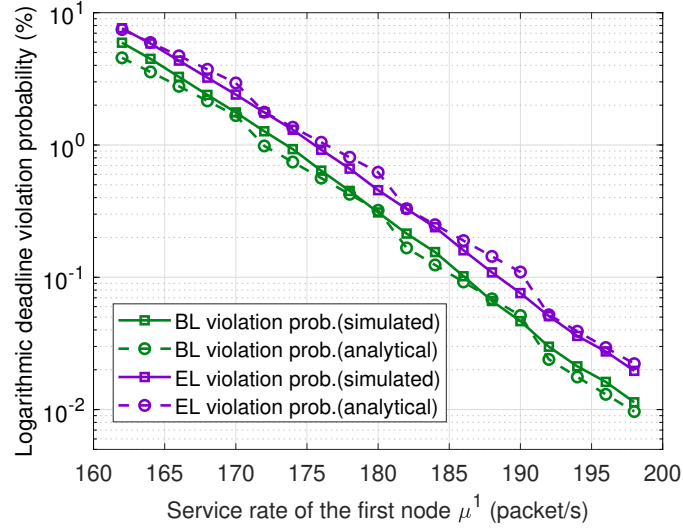


Figure 3.11: Deadline violation probability at the first segment with varying packet service rates of the first node μ^1 .

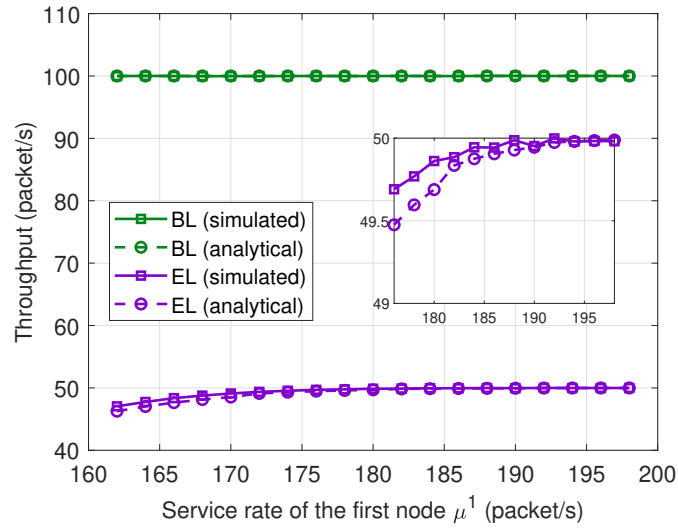


Figure 3.12: Throughput at the first segment with varying packet service rates of the first node μ^1 .

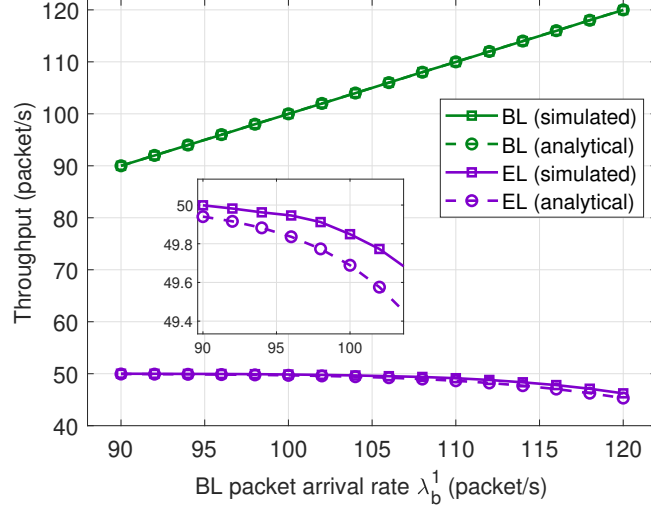


Figure 3.13: Throughput at the first segment with varying BL packet arrival rates λ_b^1 .

violation probability estimation, even for large values of μ^1 . Besides, compared to Fig. 3.8, the relative estimation errors between the analytical results and the simulation results are reduced, which proves the effectiveness of the approximations made in Subsection 3.2.2⁷. The throughputs of BL and EL packets at the first segment with varying μ^1 are shown in Fig. 3.12. We see the analytical results match with the simulation results well, and the estimation error reduces as μ^1 increases due to a smaller V_e^1 . In addition, due to the high service reliability requirement, BL packets are reliably transmitted, and thus the BL packet throughput equals to its packet arrival rate (i.e., $\lambda_{t,b}^1 = \lambda_b^1$). On the other hand, due to the low service reliability requirement, deadline-violated EL packets that do not improve video quality are discarded. Thus, the EL packet throughput is no greater than its arrival rate (i.e., $\lambda_{t,e}^1 \leq \lambda_e^1$). The results presented in Fig. 3.10 - Fig. 3.12 verify the effectiveness of the established two-stage queuing in Subsection 3.2.2 as a service modeling approximation of the first node, and also the accuracy of our proposed performance analytical model.

Furthermore, the BL/EL packet throughput at the first segment with varying BL packet arrival rate (λ_b^1) is shown in Fig. 3.13. The packet service rates of the nodes in the first segment are set as [180, 200, 200, 200] packet/s. We can see that due to the high service

⁷Please note that the non-smoothness of analytical results (in Figs. 3.11, 3.15, and 3.19) is caused by the $\lceil \cdot \rceil$ operator in determining the queue length bound N_e^1 for stage one queuing at the first node and by the approximations we made for obtaining a more accurate deadline violation probability estimate (see the explanations for Fig. 3.8).

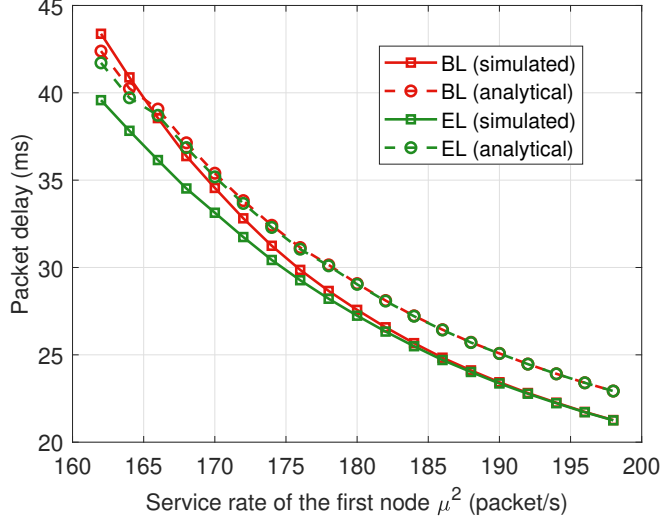


Figure 3.14: Packet delay at the second segment with varying packet service rates of the first node μ^2 .

reliability requirement, the BL packet throughput increases linearly with its arrival rate. On the other hand, as λ_b^1 increases, the inter-arrival time of successive BL packets is smaller. More BL packets enter stage one queuing while the queue length bound, N_e^1 , for EL packets is unchanged. Less EL packets can be served through the first node without exceeding their service deadlines. Hence, the EL packet throughput decreases with λ_b^1 .

The packet delays, deadline violation probabilities (in log-scale), and throughputs of BL and EL packets at the second segment with varying μ^2 are demonstrated in Fig. 3.14 - Fig. 3.16, respectively. The packet service rates of the nodes in the first segment are set as $[300, 300, 300, 300]$ packet/s while the service rates of the nodes in the second segment are $[\mu^2, 200, 200]$ packet/s. We can see that due to smaller packet queuing and service delays, the BL/EL packet delay and deadline violation probability decrease with μ^2 . Besides, the BL packet throughput maintains the same as its arrival rate (i.e., $\lambda_{t,b}^2 = \lambda_b^2 = \lambda_b^1$) due to the high service reliability requirement, while the EL packet throughput gradually approaches to its arrival rate (λ_e^2) as μ^2 increases due to a smaller V_e^2 . The close estimations of BL/EL packet delay, deadline violation probability, and throughput using our proposed performance analytical model compared to the simulation results validate the independence approximation between two consecutive segments on the premise of a large μ^1 . As discussed in Subsection 3.2.3, when μ^1 is large enough, the packet departure process at the first segment (or the packet arrival process at the second segment) is approximated as a Poisson

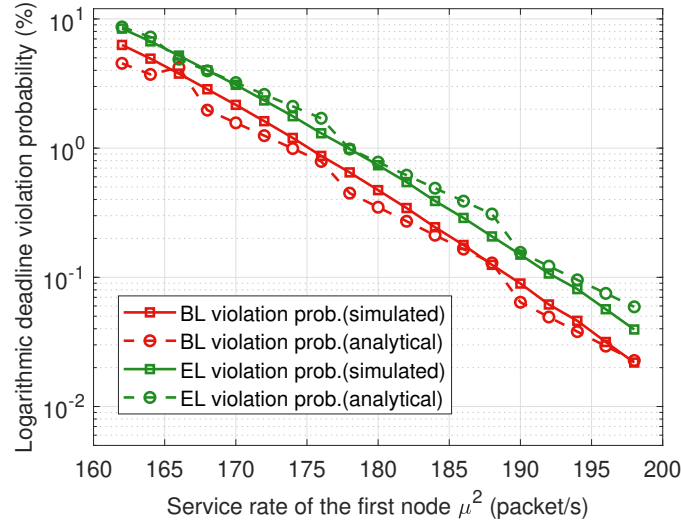


Figure 3.15: Deadline violation probability at the second segment with varying packet service rates of the first node μ^2 .

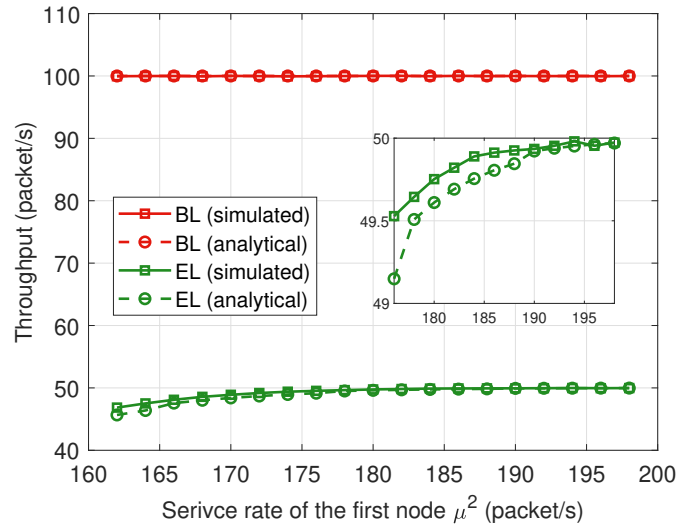


Figure 3.16: Throughput at the second segment with varying packet service rates of the first node μ^2 .

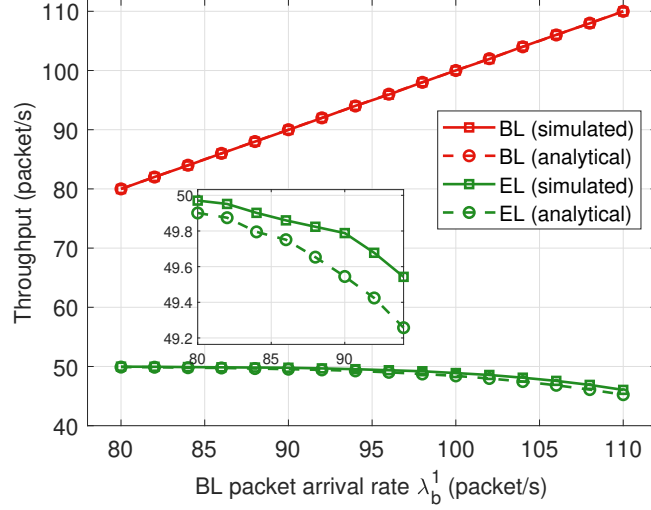


Figure 3.17: Throughput at the second segment with varying BL packet arrival rates λ_b^1 .

process. Thus, the simulation results obtained for the second segment are similar to those for the first segment. In addition, the BL/EL packet throughput at the second segment with varying BL packet arrival rate (λ_b^1) is shown in Fig. 3.17, where the packet service rates of the nodes in the second segment are set as $[170, 200, 200]$ packet/s. Similar results as in Fig. 3.13 are observed where the BL packet throughput increases linearly with its arrival rate, while the EL packet throughput decreases with λ_b^1 , as more BL packets join stage one queuing which has a finite and fixed queue length bound, i.e., N_e^2 .

Finally, the performances of E2E packet delays, deadline violation probabilities (in log-scale), and throughputs of BL and EL packets with varying μ^2 are evaluated in Fig. 3.18 - Fig. 3.20, respectively. The packet service rates of the nodes composing the considered virtual network topology (see Fig. 3.9) of the video streaming slice are set as $[300, 300, 300, 300, \mu^2, 200, 200]$ in packet/s. The close analytical results compared to the simulation results indicate the accuracy of our proposed performance analytical model which bases on the segment-based analysis framework in Subsection 3.2.1 for slice-based E2E video packet service with layered encoding.

3.3.2 Effectiveness of The Proposed Analytical Model

In the following, the effectiveness of the proposed performance analytical model is shown with two application examples, where M/M/1 and M/D/1 are selected as the benchmark

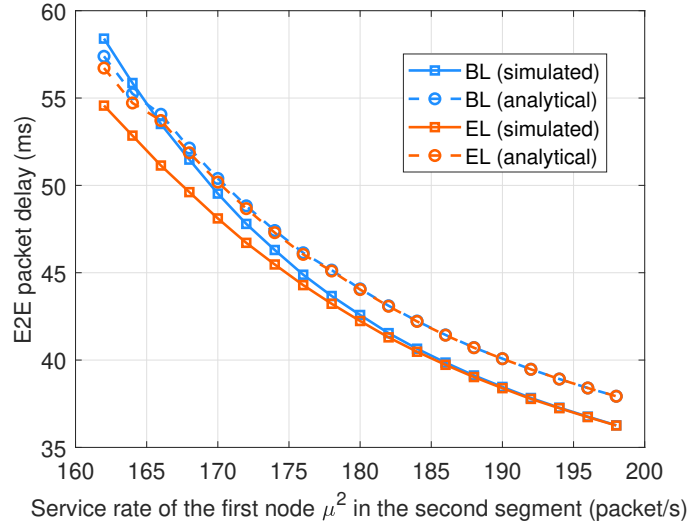


Figure 3.18: E2E packet delay with varying packet service rates of the first node in the second segment μ^2 .

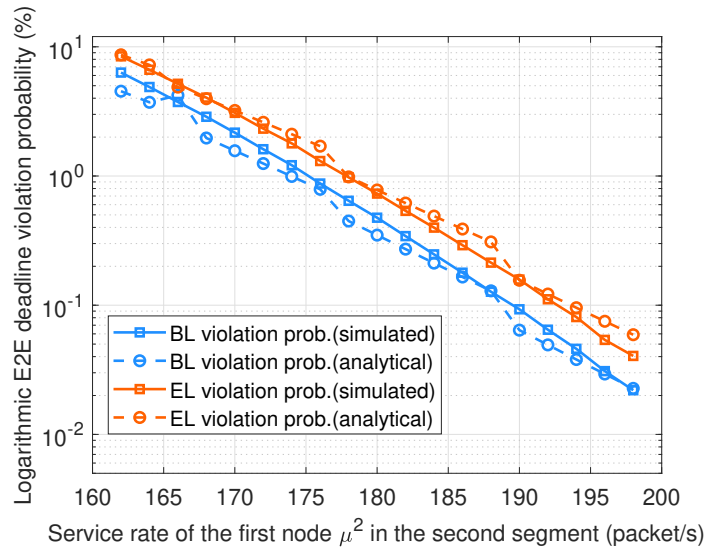


Figure 3.19: E2E deadline violation probability with varying packet service rates of the first node in the second segment μ^2 .

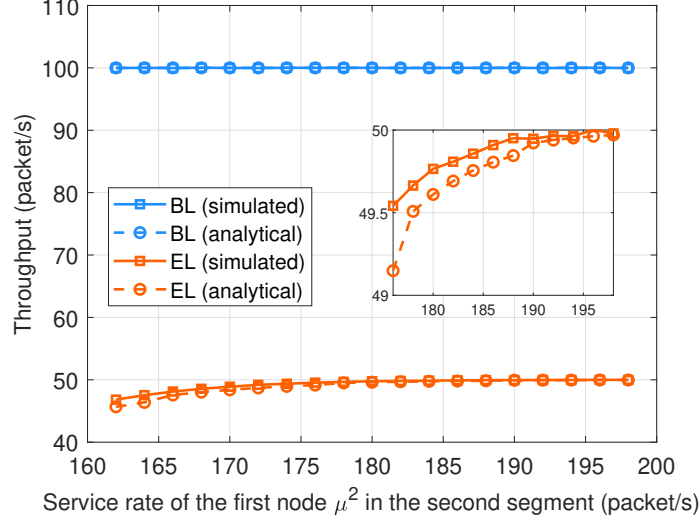
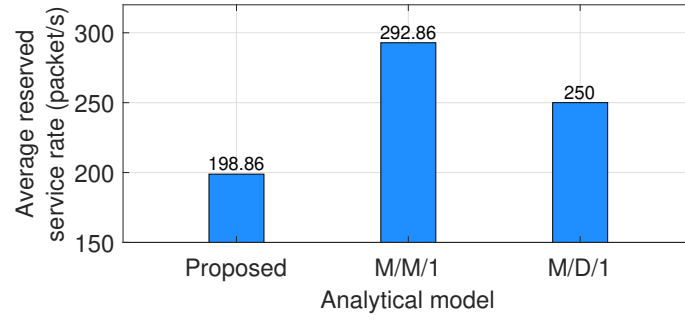


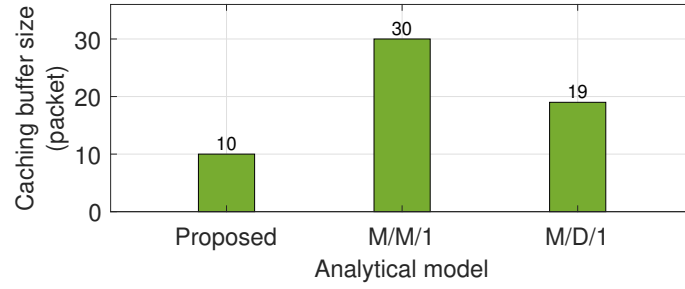
Figure 3.20: E2E throughput with varying packet service rates of the first node in the second segment μ^2 .

models. First, the proposed performance analytical model can be applied for slice resource reservation. We consider the virtual network topology in Fig. 3.9 for layer-encoded video streaming service and suppose a target E2E delay of 49 ms for both BL and EL packets. For the proposed performance analytical model, we assume that the virtual network topology consists of only one segment for simplicity. A feasible node packet service rate configuration to achieve the target BL/EL delay is [180, 212, 200, 200, 200, 200, 200] packet/s. In the benchmark models, packet service at each node is considered independent, and the differentiated service reliability requirements between BL and EL packets are neglected. The target E2E BL/EL delay is equally divided to each node, i.e., a per-node delay target of 7 ms, based on which the packet service rate reserved on each node is determined accordingly. The average reserved packet service rate on each node in the considered virtual network topology is shown in Fig. 3.21(a). Compared to the benchmark models, our proposed performance analytical model achieves more accurate estimations of BL and EL packet delays. Thus, fewer resources (i.e., packet service rate) are required for each node to achieve the target E2E packet delay using the proposed analytical model.

In addition, the proposed performance analytical model can be utilized to determine the key transport parameters in the customized caching-based packet retransmission functionality for supporting enhanced video packet delivery, including the cycle of the periodic timer and the minimum required caching buffer size for time-based caching buffer release.



(a) Average reserved packet service rate on each node



(b) Minimum required caching buffer size for BL packets

Figure 3.21: Average reserved per-node packet service rate and the minimum required caching buffer size determined using different analytical models.

Table 3.2: The performance of the caching-based packet retransmission scheme

Caching miss ratio (%)	1.4597
Caching buffer utilization (%)	82.0005
Packet retransmission delay (ms)	49.2115
Optimal packet retransmission delay (ms)	49.0655

Specifically, we set the packet service rates of the nodes in the considered virtual network topology as $[180, 212, 200, 200, 200, 200, 200]$ packet/s and consider only BL packets for an example. The packet delay between a remote video server on the Internet and the ingress node is set to 10 ms. The cycle of the periodic timer is set by the estimated E2E BL packet delay, and the minimum required caching buffer size for BL packets is determined according to Eq. (3.1). Fig. 3.21(b) shows the minimum required caching buffer sizes (for BL packets) calculated using different analytical models. The performance of the caching-based packet retransmission scheme with the caching buffer size set based on our proposed performance analytical model is given in Table 3.2. It is seen that compared to the benchmark models, fewer caching resources are required with our proposed performance analytical model due to more accurate E2E BL packet delay estimation. Meanwhile, the proposed caching-based packet retransmission scheme performs well and achieves low caching miss ratio caused by ‘*release-before-arrival*’ BL packets, high caching buffer utilization, and near-optimal packet retransmission delay, compared to the optimal case when the caching miss ratio is 0.

3.4 Summary

In this chapter, we have developed an E2E performance analytical model for layer-encoded video packet service over a core network slice. A segment-based E2E analysis framework is proposed, where a two-stage queuing model is established to determine per-segment service performance with the consideration of discrepant service reliability requirements of BL and EL packets. The independence between two consecutive segments is achieved, based on which the E2E service performance is determined, including E2E BL/EL packet delay, deadline violation probability, and throughput. Simulation results demonstrate the proposed analytical model outperforms the benchmarks as well as its effectiveness in determining the sliced node service rates and transport-layer protocol parameters.

Chapter 4

Energy-Efficient Multi-User Adaptive 360° Video Streaming: A Two-Step Approach with Device Super-Resolution

Tile-based adaptive 360° video streaming suffers from transmission rate fluctuations and inevitable FoV prediction errors. A large playback buffer occupancy ensures high video playback smoothness, whereas reducing the FoV prediction accuracy due to a large prediction time gap, thereby affecting user perceived FoV quality. In contrast, it is more efficient to transmit the predictive FoV tiles with high bitrate levels when the playback buffer occupancy is small due to the high FoV prediction accuracy. However, it also increases the risk of video stalling under transmission rate variations. Thus, an adaptive streaming scheme is critical for tile-based adaptive 360° video streaming, which considers the interplay among video playback smoothness, perceived FoV quality, and FoV prediction accuracy.

There are many existing works addressing this dilemma, focusing on single-user 360° video streaming. For example, leveraging the saliency and motion maps extracted from a particular 360° video, content-aware and trajectory-based prediction approaches have been developed to achieve more accurate FoV predictions [70, 46, 71]. In addition, probabilistic FoV prediction methods have been proposed to accommodate possible FoV prediction errors [72, 47]. Extension tiles are adaptively appended to the predicted FoV based on the recent prediction accuracy to form an extended FoV [73]. On the other hand, some works focus on the playback buffer management for an accurate FoV prediction. The

maximum playback buffer length of a video player (i.e., HMD) is set to a small value (e.g., 3-5s) to guarantee high FoV prediction accuracy with a short prediction time gap. To balance between video playback smoothness and perceived FoV quality, multiple two-tier streaming schemes have been proposed [23, 24, 47, 74]. During the streaming, a user can either download a new video chunk with the lowest bitrate level (i.e., base tier) to improve the video playback smoothness or download the updated FoV tiles of a prefetched chunk with a higher bitrate level (i.e., enhancement tier) to enhance the perceived FoV quality.

Recent advancements in VSR techniques open another promising direction to further enhance the viewing QoE of 360° video users. Multiple lightweight deep neural network (DNN)-based VSR models, customized for the tile-based 360° video streaming service, have been developed [19]. Typically, to train a VSR model offline, a pair of low-resolution and high-resolution tile frames (i.e., a video tile in a frame of a 360° video) of the same video tile is used as the input and output of the VSR model as one training data sample. Each VSR model is trained to learn how to reconstruct a tile frame from a low-resolution one to a high-resolution one (e.g., 240p $\rightarrow$ 480p). During the inference stage, the reconstruction of an encoded video tile usually goes through the sequential steps of decoding, reconstruction, and re-encoding, which requires certain computations and incurs additional reconstruction time. Many works have investigated VSR-assisted adaptive 360° video streaming by leveraging the computing capability of an edge server and/or a user device [75, 76, 71, 77, 78, 79]. A video tile with a low bitrate level can be transmitted by a remote video server and then reconstructed into a target bitrate level by the edge server or the user device before being put into the playback buffer. A pair of download and reconstructed bitrate levels is adaptively determined for each video tile to balance between transmission time and reconstruction time.

In this chapter, we consider multi-user 360° video streaming in a time-slotted system and propose a two-step adaptive streaming scheme to achieve enhanced viewing QoE. Specifically, each video chunk is first prefetched based on the predictive FoV, and the FoV quality of the chunk is then enhanced at a closer-to-playback time instant based on the updated FoV with improved prediction accuracy. Comparing the predictive FoV with the updated FoV of a video chunk, two types of tiles, i.e., miss-predicted and hit tiles, are respectively enhanced. Transmission-driven and device-VSR-driven methods for chunk enhancement are proposed, which are adaptively enabled for both miss-predicted and hit tiles to achieve device energy-efficient FoV quality enhancement. In order to determine when to prefetch a new video chunk or enhance a prefetched chunk, a time-difference approach is applied to characterize the incremental QoE gain of each user at any slot, based on which the best candidate chunk is determined. A single-shot problem is then formulated for maximizing the total incremental QoE gain while minimizing the

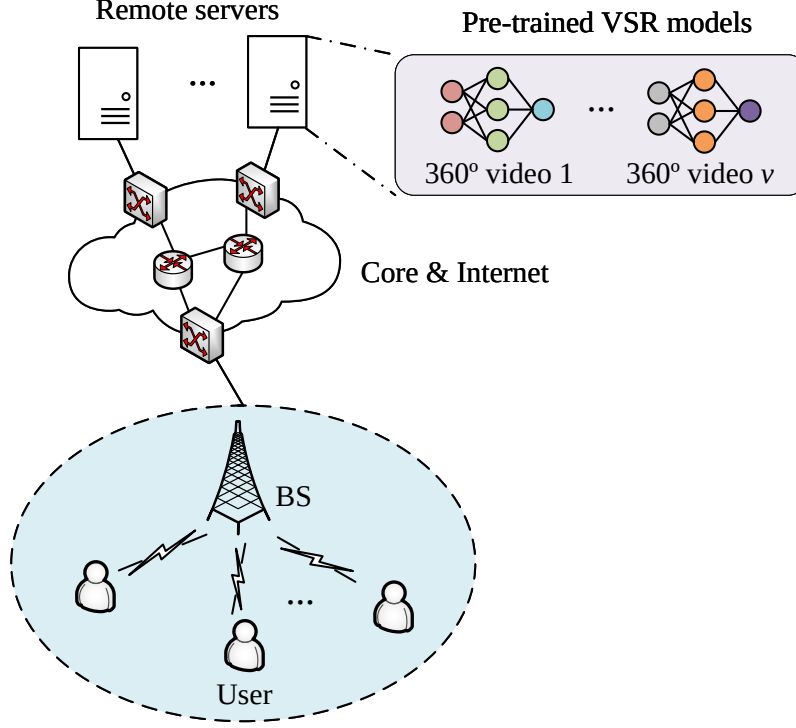


Figure 4.1: The considered network scenario.

total device energy consumption. A PSO-based solution is developed to determine the per-slot optimal decisions for different users, including bandwidth allocation, bitrate level selection, and enhancement method selection. Extensive simulation results demonstrate that our proposed solution outperforms benchmark schemes in terms of average viewing QoE, average device energy consumption, and average utility.

4.1 System Model

4.1.1 Network Scenario

We consider a multi-user on-demand 360° video streaming service, as shown in Fig. 4.1, where time is slotted with constant duration τ . Multiple video users under the coverage of a base station (BS) request 360° videos asynchronously which may have different video

lengths. Each user's streaming request is initiated at the beginning of a slot. In each slot, we consider only active users who have an ongoing 360° video streaming session. Let $\mathcal{U}_t$ be the set of users at slot t . For each user $u \in \mathcal{U}_t$, the playback buffer at slot t is denoted by $\mathcal{B}_{t,u}$, which contains the indexes of all prefetched video chunks to be played at slot t , and the playback buffer occupancy at slot t is given by $O_{t,u}$ seconds¹. The BS is pre-configured with a total radio spectrum bandwidth of Ω MHz for wireless video data transmission.

Content-aware VSR is considered to support video quality enhancement [71]. A particular DNN-based VSR model is pre-trained for each 360° video and stored in a remote server with the 360° video. The VSR models in [79, 80, 81, 82] featuring a scalable DNN structure can be adopted, which can take input with arbitrary sizes and scaling factors. When a user initiates a 360° video streaming request to a remote server, the server sends back the media presentation description (MPD) file and the pre-trained VSR model of the requested 360° video to the user. Due to the small sizes of an MPD file and a VSR model, e.g., 1 MB, the time for transmitting the MPD file and the pre-trained VSR model at the initial stage of a 360° video streaming session is neglected. Each VSR model leverages the computing capability of a user device (i.e., HMD) to reconstruct a video tile from a low-resolution (or low bitrate level) to a higher-resolution (or higher bitrate level). Early simple interpolation methods, such as Bicubic, are essentially rule-based methods. Local information (i.e., pixels) is utilized to scale up images, which fail to recover or reconstruct details. In contrast, DNN-based VSR algorithms (e.g., convolutional neural network (CNN)-based) attempt to predict and reconstruct details by learning high-level features of images. Compared to simple interpolation methods, DNN-based VSR algorithms achieve superior reconstruction performance, especially when the video content involves complex scenes and textures. For example, DNN-based VSR algorithms can achieve a peak signal-to-noise ratio (PSNR) of 40 dB [78, 83], a video multimethod assessment fusion (VMAF) score of 80 [80], or maintain high deterministic and statistical fidelity [84]. With content-aware VSR being considered, due to the small focused area of a video tile and the inherent overfitting property of DNN, near-zero training error can be achieved [75]. A pre-trained VSR model for a particular 360° video can well learn the features that map a low-resolution video tile to a higher-resolution one. Thus, we assume that a video tile with an original bitrate level r'_b achieves the same perceived quality as the same tile that is reconstructed from a lower bitrate level to the bitrate level r'_b [76, 77]. This assumption hardly holds for simple interpolation methods, which often produce blurry images. In addition, simple interpolation methods incur only a few computations, and the device energy consumption and time overhead can be neglected. Conversely, DNN-based VSR algorithms incur much

¹Throughout this chapter, we assume that all the mathematical sets are finite and their elements are sorted in an ascending order.

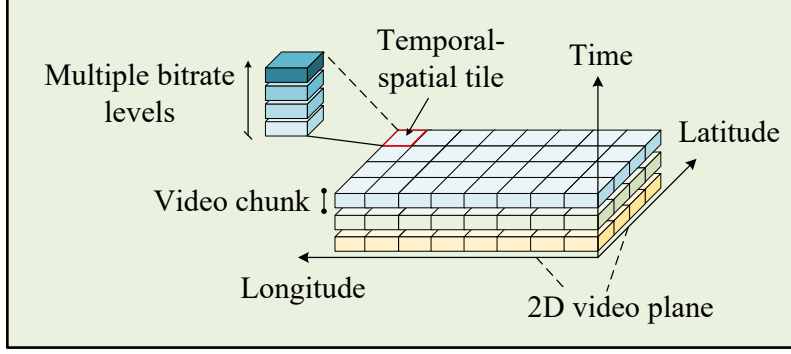


Figure 4.2: Video traffic model.

more computations and energy consumption, which need to be considered.

4.1.2 Video Traffic Model

As shown in Fig. 4.2, each 360° video is temporally divided into multiple consecutive video chunks, with each video chunk of T_c seconds. The duration of a time slot is smaller than that of a video chunk (i.e., $\tau < T_c$), and T_c is assumed to be divisible by τ . Let $\mathcal{I}_u = \{1, 2, \dots, |\mathcal{I}_u|\}$ be the set of video chunks of user u , indexed by $i \in \mathcal{I}_u$. Each video chunk is further spatially partitioned into a set of non-overlapping video tiles (e.g., 4×6), denoted by $\mathcal{J} = \{1, 2, \dots, |\mathcal{J}|\}$ and indexed by $j \in \mathcal{J}$. To support adaptive tile-based 360° video streaming, constant bitrate (CBR) format is considered for video tile encoding [74, 85]. Each video tile is encoded into multiple bitrate levels denoted by $\mathcal{R} = \{r_1, r_2, \dots, r_{|\mathcal{R}|}\}$ and indexed by $r_b \in \mathcal{R}$, corresponding to different video tile resolutions. A video tile with a higher bitrate level has a higher quality. Let $\mathcal{R}^\dagger = \{r_0\} \cup \mathcal{R}$ with $r_0 = 0$.

4.1.3 Two-step Adaptive Streaming Framework

Considering the interplay among video playback smoothness, perceived FoV quality, and FoV prediction accuracy in tile-based 360° video streaming, we propose a two-step adaptive streaming approach, as shown in Fig. 4.3. Each video chunk can go through two sequential steps, i.e., *chunk prefetching* and *enhancement*. In the prefetching step, a new panoramic chunk is prefetched, where the predictive FoV tiles are downloaded with a selected bitrate

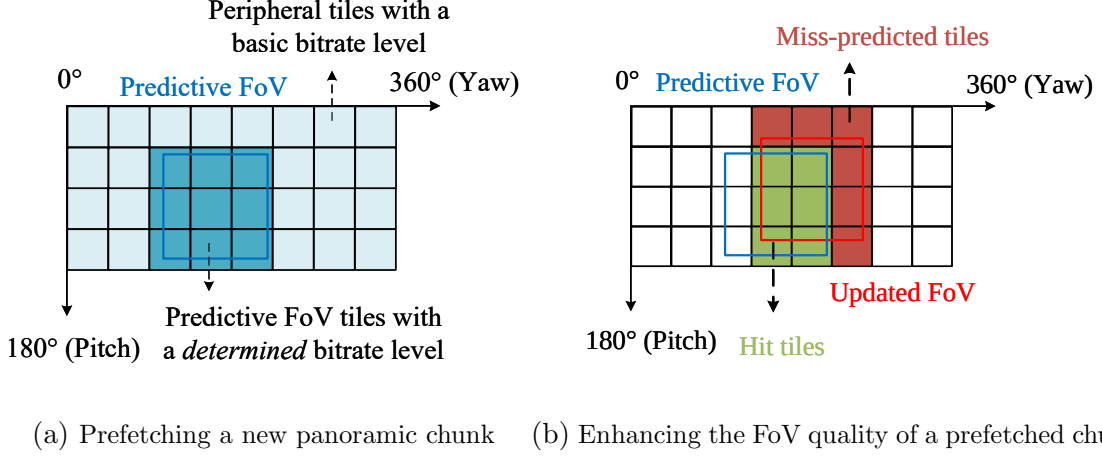


Figure 4.3: The proposed two-step adaptive streaming framework.

level, and the peripheral tiles outside the predictive FoV are downloaded with a basic bitrate level (i.e., r_1) to avoid blank screen. In the enhancement step, a prefetched chunk is selected to enhance the FoV quality based on the updated and more accurate FoV. With the two-step approach, video playback smoothness is improved by prefetching new video chunks, while the FoV quality of each prefetched chunk is enhanced with the updated FoV that is more likely to be watched by a user. For simplicity, we consider that each video chunk is enhanced at most once. Comparing the predictive FoV in the prefetching step and the updated FoV in the enhancement step, there are two types of video tiles, i.e., *miss-predicted tiles* and *hit tiles*. Miss-predicted tiles are those covered by the updated FoV but not by the predictive FoV of a video chunk and thus have a basic quality. Hit tiles are the ones covered by both the predictive and updated FoVs of a video chunk. As shown in Fig. 4.4, we consider transmission-driven and device-VSR-driven methods for the chunk enhancement. In the transmission-driven method, miss-predicted or hit tiles with an enhanced bitrate level are directly transmitted from a remote server to its intended user. In the device-VSR-driven method, the prefetched miss-predicted or hit tiles are locally reconstructed to a target enhanced bitrate level by the user device.

At each slot, a user either chooses to prefetch a new chunk based on the predictive FoV or enhances a prefetched chunk in the current playback buffer based on the updated FoV. Note that chunk prefetching or enhancement at the slot does not affect the chunk currently being played. Besides, as all tiles covering the entire video scene are downloaded when each chunk is prefetched, tiles not covered by both the predictive and updated FoVs

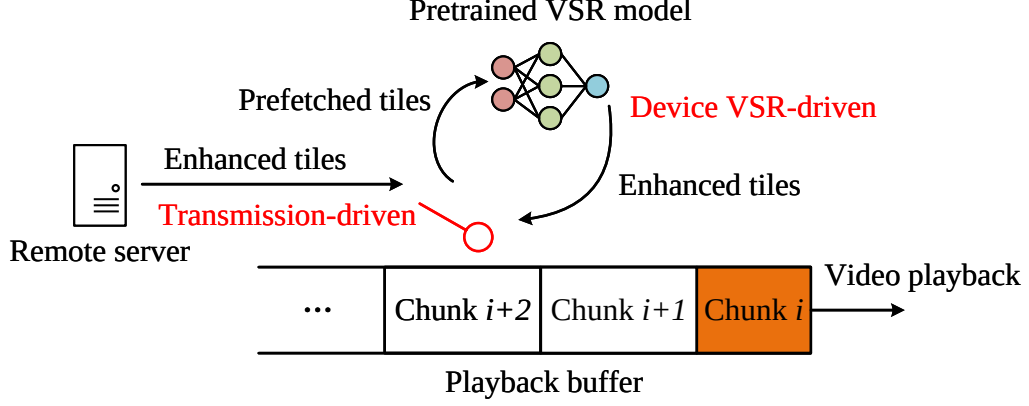


Figure 4.4: Enhancement methods.

can be used to accommodate unanticipated user viewing orientation changes. For example, when users frequently rotate their heads during playback of a chunk, the video content in any direction is at least visible without blank areas.

In order to track the streaming performance over time for a user, two auxiliary vectors are defined for user u at slot t : $\mathbf{A}_{t,u} = (A_{t,u,i}, i \in \mathcal{I}_u), u \in \mathcal{U}_t$ for the (average) FoV quality of each chunk, where $A_{t,u,i}$ denotes the FoV quality of chunk i at slot t ; $\mathbf{e}_{t,u} = (e_{t,u,i}, i \in \mathcal{I}_u), u \in \mathcal{U}_t$ for the chunk enhancement status, where $e_{t,u,i} = 1$ indicates video chunk i has been enhanced before slot t and 0 otherwise. If a new video chunk is prefetched, the chunk index is given by $i^\# = \min_{i \in \mathcal{I}_u} \{i, A_{t,u,i} = 0\}$. Let $\mathcal{J}_{t,u}^P$ be the set of predictive FoV tiles when a new video chunk is prefetched, and the assigned bitrate level of the predictive FoV tiles is denoted by $r_{t,u}^P$. Moreover, let $r_{u,i}^P$ and $\mathcal{J}_{u,i}^P$ record the bitrate level and the set of predictive FoV tiles, respectively, when video chunk i is prefetched. Thus, when video chunk $i^\#$ is prefetched, we have $r_{u,i^\#}^P = r_{t,u}^P$ and $\mathcal{J}_{u,i^\#}^P = \mathcal{J}_{t,u}^P$. Then, $A_{t+1,u,i^\#}$ is updated by

$$A_{t+1,u,i^\#} = q(r_{t,u}^P) \quad (4.1)$$

where $q(r_b) = \frac{r_b}{r_{|\mathcal{R}|}}$ [77].

When prefetched chunk $i \in \mathcal{B}_{t,u}$ in the current playback buffer is enhanced, the updated FoV of prefetched chunk i is denoted by $\mathcal{J}_{t,u,i}^E$. Comparing the predictive FoV when video chunk i is prefetched, i.e., $\mathcal{J}_{u,i}^P$, with the updated FoV, the sets of miss-predicted and hit tiles are denoted as $\mathcal{J}_{t,u,i}^M = \mathcal{J}_{t,u,i}^E \setminus \mathcal{J}_{u,i}^P$ and $\mathcal{J}_{t,u,i}^H = \mathcal{J}_{t,u,i}^E \cap \mathcal{J}_{u,i}^P$, respectively. Let $r_{t,u,i}^M$ and

$r_{t,u,i}^H$ be the enhanced bitrate levels of miss-predicted and hit tiles, respectively, and we have

$$\begin{aligned} r_{t,u,i}^M &\geq r_1 \\ r_{t,u,i}^H &\geq r_{u,i}^P. \end{aligned} \quad (4.2)$$

Then, we have $e_{t+1,u,i} = 1$, and the FoV quality of prefetched chunk i , i.e., $A_{t+1,u,i}$, is updated by

$$A_{t+1,u,i} = \frac{|\mathcal{J}_{t,u,i}^M| q(r_{t,u,i}^M) + |\mathcal{J}_{t,u,i}^H| q(r_{t,u,i}^H)}{|\mathcal{J}_{t,u,i}^E|}. \quad (4.3)$$

Moreover, let $\rho_{t,u,i}^M = \{0, 1\}$ and $\rho_{t,u,i}^H = \{0, 1\}$ be the enhancement method indicators for the miss-predicted and hit tiles of prefetched chunk i , respectively, where $\rho_{t,u,i}^M/\rho_{t,u,i}^H = 0$ for the transmission-driven method and $\rho_{t,u,i}^M/\rho_{t,u,i}^H = 1$ for the device-VSR-driven method.

4.1.4 Communication Model

At each time slot, the BS allocates orthogonal bandwidth among users to support chunk prefetching and/or transmission-driven chunk enhancement. Let $w_{t,u} = [0, 1]$, $u \in \mathcal{U}_t$ be the fraction of bandwidth allocated to user u at slot t , with the wireless transmission rate $\kappa_{t,u}$ given by

$$\kappa_{t,u} = w_{t,u} \Omega \log_2(1 + \xi_{t,u}) \quad (4.4)$$

where $\xi_{t,u}$ denotes the signal-to-noise ratio (SNR). We assume that the wireless link between the BS and a user is the bottleneck during E2E video data transmission from a remote server to the user.

The total fraction of bandwidth allocated to all users at any slot t should not exceed 1, given by

$$\sum_{u \in \mathcal{U}_t} w_{t,u} \leq 1, \forall t. \quad (4.5)$$

4.1.5 Delay Model

Next, we analyze the total experienced delay for user $u \in \mathcal{U}_t$ when a new video chunk is prefetched or when a prefetched chunk is enhanced at slot t . When a new video chunk is

$$d_{t,u}^{E,A} = (1 - \rho_{t,u,i}^H) \left(\frac{\mathbf{1}(r_{t,u,i}^M > r_1) |\mathcal{J}_{t,u,i}^M| s(r_{t,u,i}^M) + \mathbf{1}(r_{t,u,i}^H > r_{u,i}^P) |\mathcal{J}_{t,u,i}^H| s(r_{t,u,i}^H)}{\kappa_{t,u}} \right) + \rho_{t,u,i}^H (|\mathcal{J}_{t,u,i}^M| \phi_u(r_1, r_{t,u,i}^M) + |\mathcal{J}_{t,u,i}^H| \phi_u(r_{u,i}^P, r_{t,u,i}^H)) \quad (4.8)$$

$$d_{t,u}^{E,B} = \max \left[(1 - \rho_{t,u,i}^H) \frac{\mathbf{1}(r_{t,u,i}^H > r_{u,i}^P) |\mathcal{J}_{t,u,i}^H| s(r_{t,u,i}^H)}{\kappa_{t,u}}, \rho_{t,u,i}^H |\mathcal{J}_{t,u,i}^H| \phi_u(r_{u,i}^P, r_{t,u,i}^H), (1 - \rho_{t,u,i}^M) \frac{\mathbf{1}(r_{t,u,i}^M > r_1) |\mathcal{J}_{t,u,i}^M| s(r_{t,u,i}^M)}{\kappa_{t,u}}, \rho_{t,u,i}^M |\mathcal{J}_{t,u,i}^M| \phi_u(r_1, r_{t,u,i}^M) \right] \quad (4.9)$$

prefetched, the delay for transmitting the chunk from a remote server to the user, denoted by $d_{t,u}^P$, is given by

$$d_{t,u}^P = \mathbf{1}(r_{t,u}^P \neq r_0) \frac{|\mathcal{J} \setminus \mathcal{J}_{t,u}^P| s(r_1) + |\mathcal{J}_{t,u}^P| s(r_{t,u}^P)}{\kappa_{t,u}} \quad (4.6)$$

where $\mathbf{1}(a)$ is an identity function which equals 1 if condition a is true and 0 otherwise and $s(\cdot)$ returns the size in Mbits of a video tile given a bitrate level r_b with $s(r_b) = r_b T_c$. In the numerator of (4.6), the first and the second terms represent the total sizes of peripheral tiles and predictive FoV tiles, respectively.

When prefetched chunk i is enhanced, the experienced delay, denoted by $d_{t,u}^E$, depends on both the enhancement methods and the enhanced bitrate levels for the miss-predicted and hit tiles, given by

$$d_{t,u}^E = \mathbf{1}(\rho_{t,u,i}^H = \rho_{t,u,i}^M) \cdot d_{t,u}^{E,A} + \mathbf{1}(\rho_{t,u,i}^H \neq \rho_{t,u,i}^M) \cdot d_{t,u}^{E,B} \quad (4.7)$$

where $d_{t,u}^{E,A}$ and $d_{t,u}^{E,B}$ are given in (4.8) and (4.9), respectively. In (4.7) and (4.8), $d_{t,u}^{E,A}$ calculates the total delay when the miss-predicted and hit tiles are enhanced via the same method, $d_{t,u}^{E,B}$ calculates the total delay when the miss-predicted and hit tiles are enhanced via different methods in parallel, and $\phi_u(r_b, r'_b)$ represents the time for reconstructing a video tile from bitrate level r_b to a higher bitrate level r'_b by user device, where $r'_b \geq r_b$ with $\phi_u(r_b, r_b) = 0$ [79].

4.2 Problem Formulation

4.2.1 User Viewing QoE

User viewing QoE consists of three components: average perceived FoV quality, average video stall time, and average temporal quality smoothness across all video chunks of a user [86, 87, 88]. The average temporal quality smoothness captures the average perceived FoV quality variations between consecutive video chunks. Formulating user viewing QoE is challenging: First, with the proposed two-step adaptive streaming approach, each video chunk can go through the two sequential steps of prefetching and enhancement, which happen at two different time slots. Second, the average perceived FoV quality and the average temporal quality smoothness are two QoE components that are measured on the basis of video chunks, whereas the average video stall time is a QoE component measured based on the dynamics of a user's playback buffer over time. In the following, we leverage a time-difference approach to analyze each QoE component individually.

(1) Average Perceived FoV Quality

Let $Q_{t,u,i}^1, i \in \mathcal{I}_u$ be the perceived FoV quality for video chunk i of u at slot t with FoV quality $A_{t,u,i}$. The impact of FoV prediction accuracy on the FoV quality perceived by a user needs to be considered. In the chunk prefetching step, a high bitrate level can be assigned to predictive FoV tiles when the current playback buffer occupancy of a user is small, that is when the FoV prediction accuracy is high with a small prediction time gap. On the other hand, it is more effective to enhance a prefetched chunk closer to its playback time, as the updated FoV is more accurate and more likely to be the actual FoV a user will watch during the playback of the chunk. We define an effective factor vector for user u , denoted by $\mathbf{f}_{t,u} = (f_{t,u,i}, i \in \mathcal{I}_u)$, to characterize the impact of FoV prediction accuracy when a new video chunk is prefetched or a prefetched chunk is enhanced, which is updated over time. Specifically, when a new video chunk $i^\#$ is prefetched at slot t , $f_{t+1,u,i^\#}$ is updated as

$$f_{t+1,u,i^\#} = 1 - \frac{O_{t,u}}{B} \quad (4.10)$$

where B is a constant for normalization.

When prefetched chunk $i \in \mathcal{B}_{t,u}$ is enhanced, $f_{t+1,u,i}$ is updated as

$$f_{t+1,u,i} = 1 - \frac{(\mathcal{B}_{t,u}^{-1}(i) - 1)^+ T_c + \varepsilon_{t,u}}{B} \quad (4.11)$$

where $(x)^+ = \max(x, 0)$, $\varepsilon_{t,u}$ denotes the unplayed time of the video chunk currently being played, and $\mathcal{B}_{t,u}^{-1}(i)$ indicates a reverse function that returns the index of element i in $\mathcal{B}_{t,u}$.

Then, the perceived FoV quality for video chunk i is given by $Q_{t,u,i}^1 = A_{t,u,i} f_{t,u,i}$. The average perceived FoV quality, denoted by $Q_{t,u}^1$, is given by

$$Q_{t,u}^1 = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} Q_{t,u,i}^1, u \in \mathcal{U}_t. \quad (4.12)$$

(2) Average Video Stall Time

Let $Q_{t,u}^2$ be the average video stall time of user u at slot t . When a new video chunk of user u is prefetched at slot t , the video stall time at slot t is given by $(\tau - O_{t,u})^+$, and the average video stall time is updated as $Q_{t+1,u}^2 = Q_{t,u}^2 + \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^+$. The playback buffer occupancy is updated as

$$O_{t+1,u} = (O_{t,u} - \tau)^+ + T_c. \quad (4.13)$$

When prefetched chunk $i \in \mathcal{B}_{t,u}$ is enhanced, the video stall time is 0 since FoV quality enhancement does not affect the smooth playback of the chunk, thus $Q_{t+1,u}^2 = Q_{t,u}^2$. The playback buffer occupancy is updated by

$$O_{t+1,u} = (O_{t,u} - \tau)^+. \quad (4.14)$$

Finally, when neither chunk prefetching nor enhancement is performed, the video stall time at slot t is $(\tau - O_{t,u})^+$, and the average video stall time is updated by $Q_{t+1,u}^2 = Q_{t,u}^2 + \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^+$. The playback buffer occupancy is updated as $O_{t+1,u} = (O_{t,u} - \tau)^+$.

(3) Average Temporal Quality Smoothness

The average temporal quality smoothness of user $u \in \mathcal{U}_t$ at slot t , denoted by $Q_{t,u}^3$, is given by

$$Q_{t,u}^3 = \frac{1}{|\mathcal{I}_u| - 1} \sum_{i=2}^{|\mathcal{I}_u|} |Q_{t,u,i}^1 - Q_{t,u,i-1}^1|. \quad (4.15)$$

Let $G_{t,u}^1$, $G_{t,u}^2$, and $G_{t,u}^3$ represent the incremental gains in terms of different QoE components of user $u \in \mathcal{U}_t$ at slot t , respectively, given by

$$\begin{cases} G_{t,u}^1 = Q_{t+1,u}^1 - Q_{t,u}^1 \\ G_{t,u}^2 = Q_{t+1,u}^2 - Q_{t,u}^2 \\ G_{t,u}^3 = Q_{t+1,u}^3 - Q_{t,u}^3 \end{cases} \quad (4.16)$$

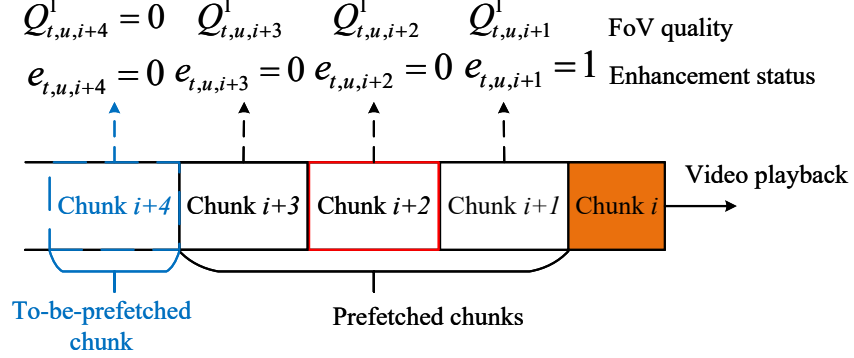


Figure 4.5: An example playback buffer of a user $u \in \mathcal{U}_t$ at slot t .

The incremental QoE gain of user $u \in \mathcal{U}_t$ at slot t is given by

$$G_{t,u} = \alpha G_{t,u}^1 - \beta G_{t,u}^2 - \gamma G_{t,u}^3 \quad (4.17)$$

where $\alpha, \beta, \gamma \in [0, 1]$ are the importance coefficients regarding different QoE components, which can be flexibly set to represent various QoE objectives². The viewing QoE of user u is then given by $\sum_t \mathbf{1}(u \in \mathcal{U}_t) G_{t,u}$. Note that for an accurate user viewing QoE model, the values of α , β , and γ must be carefully adjusted while considering many other factors, such as the video content, which is out of the scope of this work. Many previous studies have investigated QoE modeling for adaptive (360°) video streaming [89, 90].

4.2.2 Best Candidate Chunk

As a comprehensive evaluation of viewing experience, it is challenging to optimize the user viewing QoE due to dynamics of user viewing behaviors (i.e., changing viewing orientation and the number of video tiles covered by an FoV) and video streaming requests (i.e., the time instant when a streaming request is initiated and the requested 360° video). In the following, we develop a solution to enhance multi-user viewing QoE under user device energy constraints.

Fig. 4.5 shows an example playback buffer of a user $u \in \mathcal{U}_t$ at slot t , where chunk i is the video chunk currently being played, chunks $i+1$ - $i+3$ are prefetched chunks that have

²Moreover, multiple users with different QoE requirements can set user-specific importance coefficients by preference.

$$f(Q_{t+1,u,i}^1) = \alpha \frac{1}{|\mathcal{I}_u|} (Q_{t+1,u,i}^1 - Q_{t,u,i}^1) - \frac{1}{|\mathcal{I}_u| - 1} \gamma \cdot \left[\begin{aligned} &(|Q_{t+1,u,i}^1 - Q_{t+1,u,i-1}^1| - |Q_{t,u,i}^1 - Q_{t,u,i-1}^1|) e_{t,u,i-1} \\ &+ (|Q_{t+1,u,i+1}^1 - Q_{t+1,u,i}^1| - |Q_{t,u,i+1}^1 - Q_{t,u,i}^1|) e_{t,u,i+1} \end{aligned} \right] \geq 0 \quad (4.18)$$

not been played, and chunk $i + 4$ is the to-be-prefetched chunk. The prefetched chunks and the to-be-prefetched chunk are referred to as ‘*candidate chunks*’. Our main idea is to first estimate the achieved largest possible incremental QoE gain, denoted by $\Delta_{t,u,i}$, for each candidate chunk. Specifically, for each candidate chunk i , the achieved largest possible perceived FoV quality, denoted by $\hat{Q}_{t+1,u,i}^1$, is estimated based on the following two principles: 1) The incremental gain in terms of average perceived FoV quality must be no less than the penalty of average temporal quality smoothness, which is given in (4.18). Note that only the adjacent chunks that have already been enhanced before slot t are accounted for the penalty of average temporal quality smoothness, since the adjacent chunks that have not been enhanced at slot t (i.e., $e_{t,u,i} = 0$) have the chance of being enhanced in future slots. For example, in Fig. 4.5, when prefetched chunk $i + 2$ is enhanced at slot t , only the adjacent chunk $i + 1$ with $e_{t,u,i+1} = 1$ is considered in the estimation of the achieved largest possible perceived FoV quality for chunk $i + 2$; 2) For each candidate chunk i , the perceived FoV quality must satisfy $Q_{t,u,i}^1 \leq Q_{t+1,u,i}^1 \leq f_{t+1,u,i}$. Then, the perceived FoV quality of each candidate chunk i can be maximized as in $(\mathbf{P}_0)$.

$$(\mathbf{P}_0) : \max_{Q_{t+1,u,i}^1} f(Q_{t+1,u,i}^1) \quad (4.19)$$

$$\text{s.t. } Q_{t,u,i}^1 \leq Q_{t+1,u,i}^1 \leq f_{t+1,u,i} \quad (4.20)$$

$$f(Q_{t+1,u,i}^1) \geq 0. \quad (4.21)$$

Proposition 1. *The optimum of $(\mathbf{P}_0)$ is obtained at the boundary values of $Q_{t+1,u,i}^1$.*

Proof. The proof of Proposition 1 is provided in Appendix. $\square$

According to Proposition 1, the largest perceived FoV quality for each candidate chunk i of user u at slot t , i.e., $\hat{Q}_{t+1,u,i}^1$, is estimated. Then, the best candidate chunk that achieves the largest possible incremental QoE gain is given by

$$i_{t,u}^* = \operatorname{argmax}_i \Delta_{t,u,i} = \operatorname{argmax}_i \left\{ f(\hat{Q}_{t+1,u,i}^1) - \beta \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^+ \right\}. \quad (4.22)$$

If the best candidate chunk is the to-be-prefetched chunk, then chunk prefetching is performed at slot t , and the user is termed as a *prefetching user*. Otherwise, chunk enhancement is performed at slot t , and the user is termed as an *enhancement user*. Hereafter, we use i instead of i^* to indicate the best candidate chunk for brevity.

4.2.3 Problem Formulation

A single-slot problem is formulated to allocate bandwidth among multiple users, select the bitrate level of predictive FoV tiles for the best candidate chunk of each prefetching user, and determine the enhanced bitrate levels and enhancement methods of the miss-predicted and hit tiles for the best candidate chunk of each enhancement user. Let $\mathcal{U}_t^P$ and $\mathcal{U}_t^E$ be the sets of prefetching and enhancement users at slot t , respectively. The total delay for chunk prefetching or enhancement needs to be smaller than the slot length, given by

$$d_{t,u}^P \leq \tau, u \in \mathcal{U}_t^P \quad (4.23)$$

$$d_{t,u}^E \leq \tau, u \in \mathcal{U}_t^E. \quad (4.24)$$

Moreover, device energy is consumed when the best candidate chunk of an enhancement user is enhanced through the device-VSR-driven method, which is critical to consider due to an HMD's limited battery capacity. Let $E_{t,u}$ be the incremental average (normalized) device energy consumption of user u at slot t , given by

$$E_{t,u} = \frac{\rho_{t,u,i}^M |\mathcal{J}_{t,u,i}^M| \phi(r_1, r_{t,u,i}^M) + \rho_{t,u,i}^H |\mathcal{J}_{t,u,i}^H| \phi(r_{u,i}^P, r_{t,u,i}^H)}{\tau |\mathcal{I}_u|} \quad (4.25)$$

where $E_{t,u} = 0$ for any $u \in \mathcal{U}_t^P$. At each slot, the objective is to maximize the total incremental QoE gain while minimizing the total incremental average device energy consumption to achieve energy-efficient multi-user viewing QoE optimization. The single-slot

problem is formulated as

$$(\mathbf{P}_1) : \max_{\substack{w_{t,u}, r_{t,u}^P, \\ r_{t,u,i}^H, r_{t,u,i}^M, \rho_{t,u,i}^H, \rho_{t,u,i}^M}} \sum_{u \in \mathcal{U}_t^P} \Psi_u^P + \sum_{u \in \mathcal{U}_t^E} \Psi_u^E \quad (4.26)$$

$$\text{s.t. } (4.5), (4.23), (4.24)$$

$$r_{t,u}^P \in \mathcal{R}^\dagger, u \in \mathcal{U}_t^P \quad (4.27)$$

$$r_{t,u,i}^H \in \mathcal{R}, r_{t,u,i}^H \geq r_{u,i}^P, u \in \mathcal{U}_t^E \quad (4.28)$$

$$r_{t,u,i}^M \in \mathcal{R}, r_{t,u,i}^M \geq r_1, u \in \mathcal{U}_t^E \quad (4.29)$$

$$\rho_{t,u,i}^H, \rho_{t,u,i}^M = \{0, 1\}, u \in \mathcal{U}_t^E \quad (4.30)$$

where $\Psi_u^P = f(Q_{t+1,u,i}^1) - \beta \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^+$, $\Psi_u^E = f(Q_{t+1,u,i}^1) - \beta \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^+ - \delta E_{t,u}$, and $\delta = [0, 1]$ is the importance coefficient regarding device energy consumption.

Note that $(\mathbf{P}_1)$ is a mixed-integer non-linear programming (MINLP), which is NP-hard [91]. In the next section, we develop a PSO-based iterative solution to solve the problem.

4.3 PSO-based Iterative Solution

4.3.1 Problem Decomposition

Given the bandwidth allocation decisions $w_{t,u}, u \in \mathcal{U}_t$, the bitrate level and/or enhancement method selection decisions for a user are independent of those for the other users. Thus, $(\mathbf{P}_1)$ can be decomposed into multiple independent per-user subproblems, expressed as $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$, where $(\mathbf{P}_2 - 1)$ represents the subproblem for each prefetching user $u \in \mathcal{U}_t^P$ at slot t , and $(\mathbf{P}_2 - 2)$ represents the subproblem for each enhancement user $u \in \mathcal{U}_t^E$ at slot t .

$$(\mathbf{P}_2-1) : \max_{r_{t,u}^P} \Psi_u^P \quad (4.31)$$

$$\text{s.t. } (4.23), (4.27)$$

$$(\mathbf{P}_2-2) : \max_{r_{t,u,i}^H, r_{t,u,i}^M, \rho_{t,u,i}^H, \rho_{t,u,i}^M} \Psi_u^E \quad (4.32)$$

$$\text{s.t. } (4.24), (4.28), (4.29), (4.30)$$

Given $w_{t,u}, u \in \mathcal{U}_t$, the objective of $(\mathbf{P}_1)$ is obtained by solving the subproblem of each user according to the user type (i.e., prefetching or enhancement user). As both $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$ have only discrete decision variables and the size of the bitrate level set $|\mathcal{R}|$ is small (e.g., 5), the sizes of the solution spaces for $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$ are $|\mathcal{R}^\dagger|$ and $4|\mathcal{R}| \times |\mathcal{R}|$, respectively. Therefore, we enumerate all feasible combinations of decisions and find the one that obtains the maximal objective value to determine the optimal decisions of $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$.

4.3.2 Algorithm Design

We propose a PSO-based solution to solve $(\mathbf{P}_1)$, where the bandwidth allocation decisions are iteratively optimized through the PSO algorithm, and the bitrate level and/or the enhancement method selection decisions are made by solving the subproblem of each user (i.e., $(\mathbf{P}_2 - 1)$ or $(\mathbf{P}_2 - 2)$). The flowchart of the proposed PSO-based solution is shown in Fig. 4.6. Specifically, we consider a particle swarm of size N . The position of particle n , denoted by $\mathbf{x}_n = (x_{n,1}, x_{n,2}, \dots, x_{n,|\mathcal{U}_t|})$, is a multi-dimensional vector with dimension $D = |\mathcal{U}_t|$, representing a feasible bandwidth allocation among users, i.e., $\mathbf{x}_n = \{w_{t,u}, u \in \mathcal{U}_t\}$. Let K be the total number of iterations. The inertia weight is denoted by ϖ . Two acceleration coefficients, also termed as cognitive and social factors, are denoted by c_1 and c_2 , respectively. In the initialization stage of the PSO algorithm, the positions of N particles are first initialized with the Dirichlet distribution, which meets the constraint (4.5). The velocity of particle n , denoted by $\mathbf{v}_n = (v_{n,1}, v_{n,2}, \dots, v_{n,|\mathcal{U}_t|})$, is initialized with each dimension $v_{n,d} \in [v_{\min}, v_{\max}]$, where $v_{\min}$ and $v_{\max}$ are the minimal and maximal velocities, respectively. The personal best position and fitness value (i.e., the objective value of $(\mathbf{P}_1)$) of each particle are initialized using its initial position and the corresponding fitness value, respectively. The global best position and fitness value within the swarm are initialized by the particle that has the largest personal best fitness value. Let the algorithm iteration index $k = 1$.

For each iteration k , the position and velocity of particle n are updated as (*Step 1*)

$$v_{n,d}^k = \varpi^k v_{n,d}^{k-1} + c_1 \theta_1 (x_{n,d}^P - x_{n,d}^{k-1}) + c_2 \theta_2 (x_d^G - x_{n,d}^{k-1}) \quad (4.33)$$

$$x_{n,d}^k = (x_{n,d}^{k-1} + v_{n,d}^k)^+ \quad (4.34)$$

where $\theta_1, \theta_2 \in [0, 1]$ are two random numbers, $x_{n,d}^P$ is the d -th dimension of the personal best position of particle n , and x_d^G is the d -th dimension of the global best position within

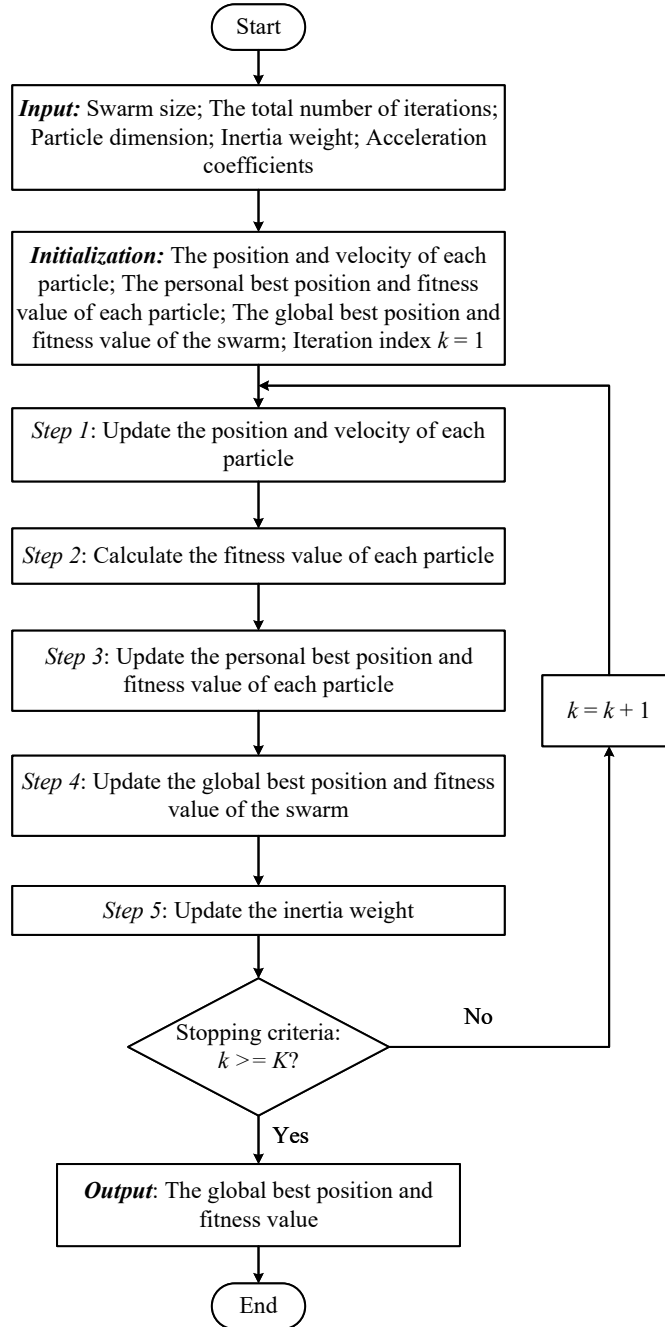


Figure 4.6: The flowchart of the proposed PSO-based solution.

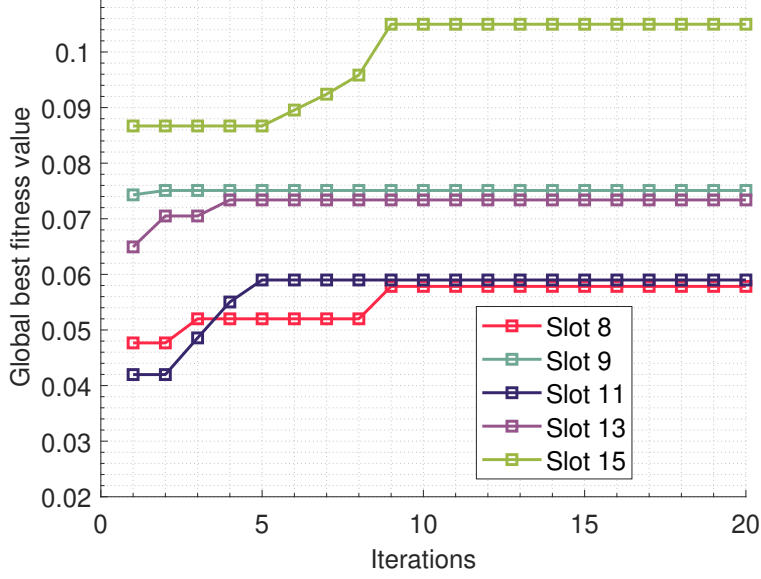


Figure 4.7: The convergence performance of the proposed PSO-based solution at sample slots.

the swarm. If $\sum_{d=1}^{|\mathcal{U}_t|} x_{n,d}^k > 1$, $x_{n,d}^k$ is projected onto $x_{n,d}^k / \sum_{d=1}^{|\mathcal{U}_t|} x_{n,d}^k$ to satisfy the constraint (4.5).

Next, the fitness value of each particle is calculated, and the personal best position and fitness value are updated (*Steps 2* and *3*). Then, the global best position and fitness value of the swarm are updated (*Step 4*). Finally, inertia weight ϖ is updated with an adaptive linear decreasing method to balance the global exploration and local exploitation capabilities of particles within the swarm (*Step 5*), given by

$$\varpi^k = \varpi_{\max} - (\varpi_{\max} - \varpi_{\min}) \frac{k}{K} \quad (4.35)$$

where $\varpi_{\min}$ and $\varpi_{\max}$ denote the minimal and maximal inertia weights, respectively. The global best position and fitness value of the swarm are output when the number of iterations reaches K .

Fig. 4.7 shows the convergence performance of the proposed PSO-based iterative solution at sample slots [92, 93]. The proposed PSO-based two-step algorithm is summarized in Algorithm 1. At slot t , the best candidate chunk of each user is first determined (*line*

Algorithm 1: PSO-based Two-Step Adaptive 360° Video Streaming

```

1 Input:  $\mathcal{U}_t, \mathcal{B}_{t,u}, O_{t,u}, \mathbf{A}_{t,u}, \mathbf{e}_{t,u}, \mathbf{f}_{t,u}, \mathcal{J}_{t,u}^P, \mathcal{J}_{t,u,i}^E, r_{u,i}^P, \mathcal{J}_{u,i}^P$ 
2 for slot  $t$  do
3   Determine the best candidate chunk of each user  $u \in \mathcal{U}_t$ 
4   Obtain the optimal bandwidth allocation decisions  $w_{t,u}, u \in \mathcal{U}_t$ 
5   Solve  $(\mathbf{P}_2 - 1)$  or  $(\mathbf{P}_2 - 2)$  for each user  $u \in \mathcal{U}_t$  based on the allocated
      bandwidth  $w_{t,u}$  and the user type
6   Update  $\mathbf{A}_{t,u}, \mathbf{e}_{t,u}, \mathbf{f}_{t,u}, r_{u,i}^P, \mathcal{J}_{u,i}^P$ 
7 Output:  $w_{t,u}, r_{t,u}^P, r_{t,u,i}^H, r_{t,u,i}^M, \rho_{t,u,i}^H, \rho_{t,u,i}^M$ 

```

3). Next, the proposed PSO-based solution is applied to obtain the optimal bandwidth allocation decisions, i.e., $w_{t,u}, u \in \mathcal{U}_t$ (line 4). Then, based on the allocated bandwidth $w_{t,u}$, the subproblem $(\mathbf{P}_2 - 1)$ or $(\mathbf{P}_2 - 2)$ is solved for user u based on the user type to obtain the optimal bitrate level and/or the enhancement method selection decisions, i.e., $r_{t,u}^P$ or $(r_{t,u,i}^H, r_{t,u,i}^M, \rho_{t,u,i}^H, \rho_{t,u,i}^M)$ (line 5). At the end of each slot, $\mathbf{A}_{t,u}, \mathbf{e}_{t,u}, \mathbf{f}_{t,u}, r_{u,i}^P$, and $\mathcal{J}_{u,i}^P$ are updated for each user based on the determined optimal decisions (line 6). For the practical implementation of Algorithm 1, at any slot, each user first determines the best candidate chunk. Then, the current streaming performance and the related information of the best candidate chunk (e.g., the set of predictive FoV tiles or the sets of miss-predicted and hit tiles) are sent from the user to the BS. The BS determines the bandwidth allocation among different users. Finally, each user determines the other decisions based on the allocated bandwidth and user type, and sends a corresponding video content request to the intended video server.

The time complexity of Algorithm 1 consists of two parts. The best candidate chunk of each user at any slot t is first independently determined (in parallel), requiring a running time of at most $O(B)$, where B is the maximum number of prefetched chunks in the playback buffer of a user. Then, the proposed PSO-based solution is applied, consuming at most $O(NK|\mathcal{U}_t||\mathcal{R}|^2)$ time. Overall, the time complexity of Algorithm 1 is $O(B) + O(NK|\mathcal{U}_t||\mathcal{R}|^2)$.

4.4 Performance Evaluation

In this section, simulation results are presented to validate the performance of our proposed two-step adaptive streaming scheme based on real data traces³ [46]. All the experiments

³<https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour>

Table 4.1: Main simulation parameters

Parameters	Values
Video chunk length T_c	1 s
Time slot length τ	0.25 s
Transmit power of the BS	43 dBm
Pass loss exponent	3.4
Noise power	10^{-13} W
Video tiling layout	4×8
FoV	$100^\circ \times 100^\circ$
Video chunk bitrate levels	[2.5, 5, 8, 16, 40] Mbps
Importance coefficients (α, β, γ)	$(\frac{2}{3}, \frac{1}{6}, \frac{1}{6})$, $(\frac{1}{6}, \frac{2}{3}, \frac{1}{6})$, or $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

are conducted using Python 3.10.0 and FFmpeg⁴. FFmpeg is leveraged for video cropping (i.e., video chunk and tile), encoding, and FoV stitching, while tile-based adaptive 360° video streaming is simulated using Python. The main simulation parameters are given in Table 4.1.

4.4.1 Data Trace Preprocessing

The selected data traces contain ten 360° videos and the viewing orientations of 50 subjects for each 360° video. The viewing trajectories are given in radian in terms of yaw (from $-\pi$ to π) and pitch (from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$).

As shown in Fig. 4.8, we consider a 4×8 tiling layout and an FoV of $100^\circ \times 100^\circ$. The panoramic video scene is spatially partitioned into 32 video tiles, each of which covers a $45^\circ \times 45^\circ$ square viewing span and is indexed in raster-scan order [20]. For a predicted viewpoint (i.e., yaw and pitch), we first conduct sphere-to-plane coordinate transformation, which maps the predicted yaw (θ_i) and pitch (φ_i) in the spherical coordinate system to the horizontal ($\omega_i \in [0^\circ, 360^\circ]$) and vertical ($h_i \in [0^\circ, 180^\circ]$) coordinates in the projected 2D video plane [94]. Then, the set of tiles covered by the predicted FoV is obtained.

⁴<https://ffmpeg.org/>

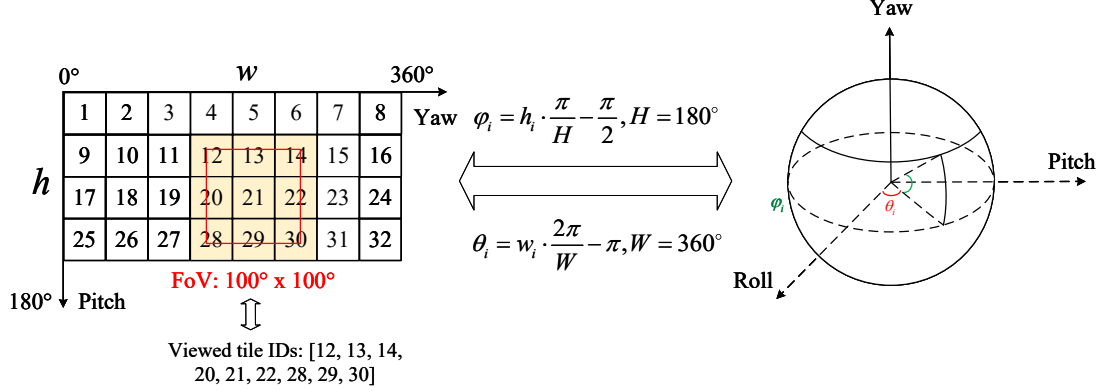


Figure 4.8: The considered video tiling layout with sphere-to-plane coordinate transformation.

4.4.2 Simulation Settings

We consider that $|\mathcal{U}|$ users are uniformly distributed within the coverage (500 m) of a BS and asynchronously request one of the two 360° videos (3840×2160) selected from the real data traces. The transmit power of the BS is 43 dBm. The channel gain between the BS and a user consists of path loss, log-normal shadow fading with a standard deviation of 8 dB, and Rayleigh-distributed fast fading with scale parameter $\frac{\sqrt{2}}{2}$ [95, 96]. Video chunk length $T_c = 1$ s, and the slot length $\tau = 0.25$ s [76, 77]. Following the recommended encoding settings by YouTube⁵, each video chunk can be encoded into five bitrate levels: 480p (2.5 Mbps), 720p (5 Mbps), 1080p (8 Mbps), 2K (16 Mbps), and 4K (40 Mbps). Each video tile consumes $\frac{1}{32}$ bitrate of a video chunk [78]. The tile reconstruction time for a given pair of bitrate levels, i.e., $\phi_u(r_b, r'_b)$, $r'_b \geq r_b$, is set according to experimental data from related works, where VSR models are offline trained with NVIDIA GPUs (e.g., V100 or GTX 1080) [75, 79]. The pretrained VSR models are applied for tile reconstruction, where the multi-threading technique is leveraged to accelerate the processing and achieve a per-tile reconstruction time at the millisecond level. Moreover, linear regression (LR) is adopted for FoV prediction using the recent past 3-s viewing orientations (i.e., pitch and yaw) at a sampling frequency of 5 Hz [74, 76]. According to empirical data, the importance coefficient regarding device energy consumption δ is set to 0.04.

We consider three existing streaming schemes as benchmarks for performance comparison, termed as Flare, SDSR, and RAM, respectively, as shown in Fig. 4.9. Specifically,

⁵<https://support.google.com/youtube/answer/1722171?hl=en>

Comparative schemes	One step	
	Device VSR assistance	Adaptive chunk enhancement
Flare	×	×
SDSR	√	×
Comparative schemes	Two-step	
	Device VSR assistance	Adaptive chunk enhancement
RAM	×	√
Proposed	√	√

Figure 4.9: The considered benchmark schemes for performance comparison.

- Flare [97]: A typical tile-based streaming scheme. Video chunks are progressively downloaded based on the predictive FoVs;
- SDSR [78]: A device-VSR-assisted streaming scheme. The predictive FoV tiles of each video chunk are first downloaded and then locally enhanced by the user device. A pair of download and enhanced bitrate levels is adaptively determined to balance transmission time with reconstruction time. Device energy consumption is not considered;
- RAM [74]: A two-layer streaming scheme. At each slot, either a new video chunk is prefetched with a basic quality or a prefetched chunk in the current playback buffer is selected to be enhanced via only the transmission-driven method.

4.4.3 Simulation Results

Fig. 4.10 - Fig. 4.12 show the performance of different schemes in terms of average viewing QoE, average (normalized) device energy consumption, and average utility, which incorporates both user viewing QoE and device energy consumption, under various QoE objectives. The number of users is 4, and the total bandwidth of the BS is 4 MHz. We consider three sets of configurations for importance coefficients (α, β, γ) to represent different QoE objectives, i.e., $(2/3, 1/6, 1/6)$ for prioritizing the average perceived FoV quality, $(1/6, 2/3, 1/6)$ for prioritizing the average video stall time, and $(1/3, 1/3, 1/3)$ for comprehensive consideration [77, 75].

Our observations on the performance comparison are as follows: First, SDSR achieves higher average viewing QoE than Flare due to the device-VSR-assisted (predictive) FoV quality enhancement when the total bandwidth of the BS is insufficient; Second, RAM achieves higher average viewing QoE than Flare due to the two-layer streaming scheme.

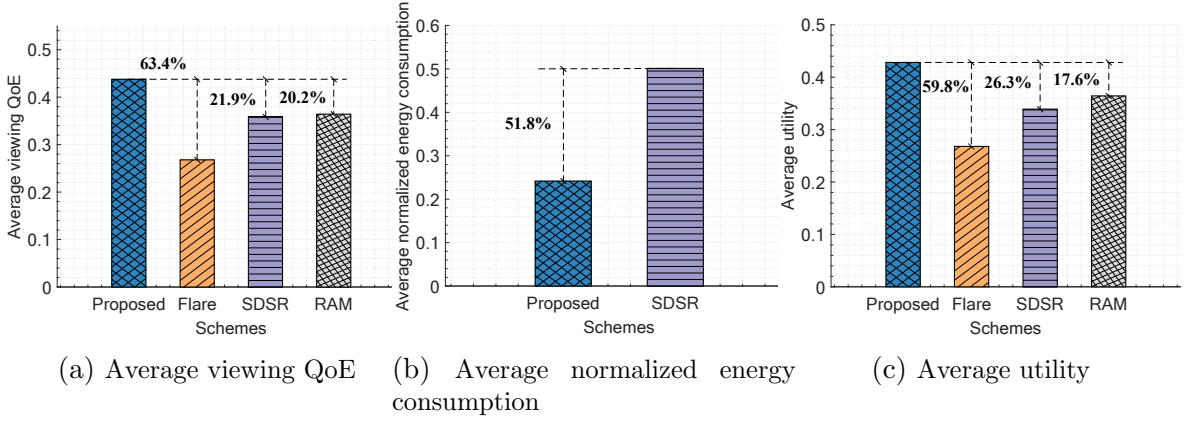


Figure 4.10: Performance of different schemes with $(\alpha, \beta, \gamma) = (2/3, 1/6, 1/6)$.

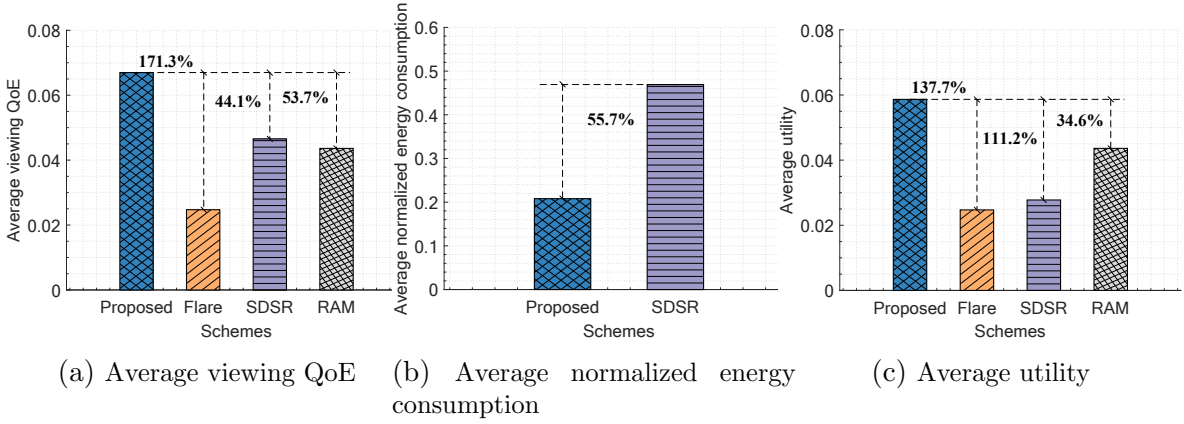


Figure 4.11: Performance of different schemes with $(\alpha, \beta, \gamma) = (1/6, 2/3, 1/6)$.

Each video chunk's FoV quality is enhanced more effectively with a more accurate updated FoV, which is limited by the insufficient total bandwidth and may incur large temporal quality variations. In addition, each video chunk is prefetched with only a basic quality while neglecting the playback buffer dynamics. Bandwidth resources may not be efficiently utilized, especially when the allocated bandwidth to a user is large and the current playback buffer occupancy is small (i.e., the FoV prediction accuracy is high); Third, our proposed two-step adaptive streaming scheme achieves the highest average viewing QoE among the benchmarks and less average device energy consumption than SDSR under different QoE objectives. The proposed scheme captures the impact of FoV prediction accuracy on user perceived FoV quality. Bandwidth resources are efficiently utilized through adaptive bandwidth allocation among users under network condition and playback buffer dynamics.

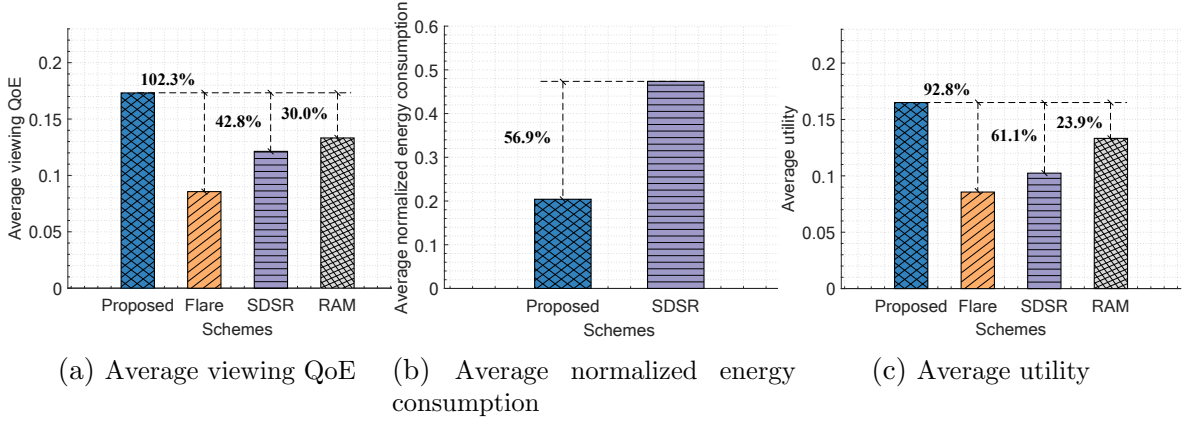
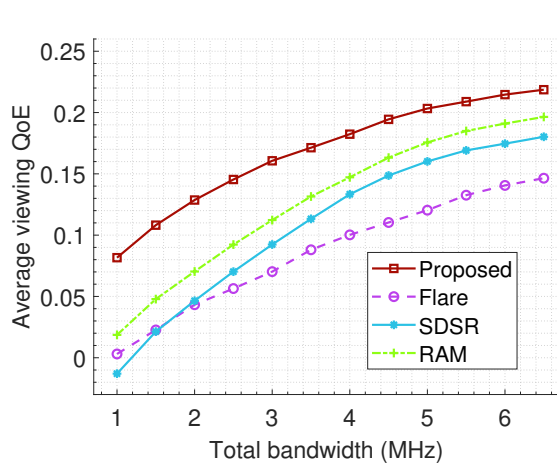


Figure 4.12: Performance of different schemes with $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$.

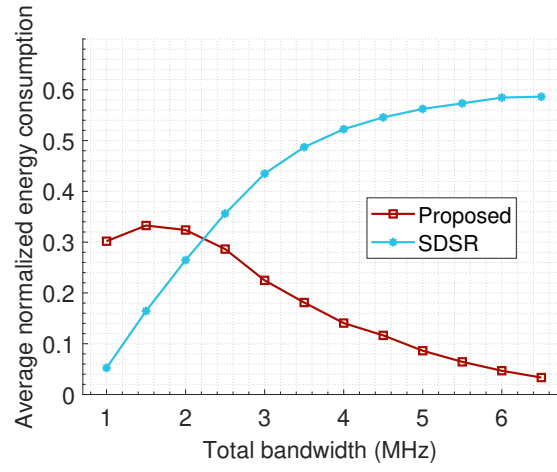
Device energy efficiency is achieved via adaptive enhancement method selection. Therefore, the proposed scheme outperforms the benchmarks and achieves the highest average utility under various QoE objectives.

Fig. 4.13(a) - Fig. 4.13(c) show the average viewing QoE, average normalized energy consumption, and average utility of different schemes with varying total bandwidth of the BS, respectively. The number of users is 3 and $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$. It can be seen that as the total bandwidth increases, the average viewing QoE and utility under different streaming schemes increase, where our proposed two-step adaptive scheme achieves the highest average viewing QoE and utility compared to the benchmarks. The average normalized energy consumption under our proposed scheme decreases with the total bandwidth, whereas the average normalized energy consumption increases with the total bandwidth under SDSR. SDSR and Flare are both one-step streaming schemes, which overlook the impact of FoV prediction accuracy on user perceived FoV quality. SDSR fully relies on the device-VSR-driven method for FoV quality enhancement without considering the device energy consumption, thus resulting in inefficient bandwidth utilization and large device energy consumption, especially when the total bandwidth is large. The performance of RAM is limited by the total bandwidth, as it fully relies on the transmission-driven method for FoV quality enhancement. Each video chunk is prefetched with only a basic quality, even if the total bandwidth is large, leading to inefficient bandwidth usage.

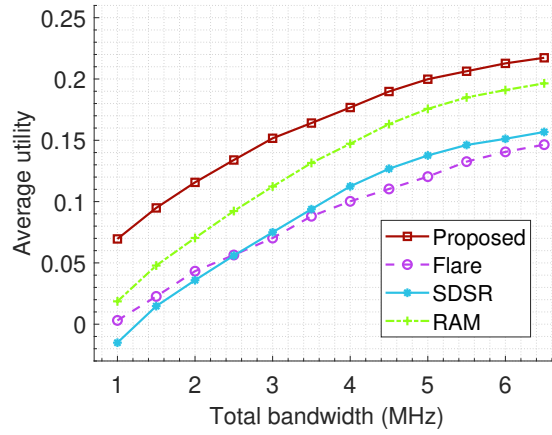
As the total bandwidth of the BS increases, a larger bitrate level is selected for predictive FoV tiles when prefetching a new video chunk, which leads to smaller reconstruction time for enhancing a video tile to a target bitrate level with the device-VSR-driven method. More video tiles are likely to be enhanced with the transmission-driven method. The computing loads for device-VSR-driven chunk enhancement are reduced. Therefore, higher



(a) Average viewing QoE



(b) Average normalized energy consumption



(c) Average utility

Figure 4.13: Performance of different schemes with varying total bandwidth of the BS.

average viewing QoE and less device energy consumption are achieved with the proposed scheme when the total bandwidth increases.

Fig. 4.14(a) - Fig. 4.14(c) show the average viewing QoE, average normalized energy consumption, and average utility of different schemes with varying numbers of users, respectively. The total bandwidth of the BS is 10 MHz and $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$. We can see that as the number of users increases, the average viewing QoE and average utility under different schemes decrease, where our proposed two-step adaptive scheme achieves the highest average viewing QoE and utility compared to the benchmarks. The average normalized energy consumption under the proposed scheme increases with the number of users and is smaller than that under SDSR. For SDSR, when the number of users increases, it costs more time, including transmission and reconstruction time, to deliver a video tile with a given pair of download and enhanced bitrate levels. A smaller pair of download and enhanced bitrate levels is selected to satisfy the delay constraint, thus leading to less device energy consumption. In addition, as discussed earlier, Flare and SDSR neglect the impact of FoV prediction accuracy on user perceived FoV quality, and RAM depends on the total bandwidth, thus resulting in worse performance than our proposed scheme. Note that the experiment can be extended to consider more than 10 users in Fig. 4.14. In that case, the running time of Algorithm 1 increases, according to the algorithm time complexity analysis in Subsection 4.3.2.

With the proposed scheme, when more users share the total bandwidth of the BS, a smaller bitrate level is selected for predictive FoV tiles when prefetching a new video chunk, which leads to larger reconstruction time for enhancing a video tile to a target enhanced bitrate level with the device-VSR-driven method. The achieved FoV quality improvement is much constrained with the transmission-driven method. More video tiles are likely to be enhanced via the device-VSR-driven method, resulting in higher computing loads. Thus, more device energy is consumed. On the other hand, the FoV quality of each video chunk is enhanced effectively based on the updated FoV with improved prediction accuracy. The best candidate chunk is determined for each user in any slot to achieve the largest total incremental QoE gain with efficient bandwidth usage. The transmission-driven and device-VSR-driven methods are adaptively enabled for FoV quality enhancement under network and playback buffer dynamics to achieve device energy efficiency. Therefore, our proposed scheme achieves higher average viewing QoE and utility than the benchmarks under different numbers of users.

To visualize the performance improvement of our proposed streaming scheme, Fig. 4.15 and Fig. 4.16 show snapshot images from the generated demo videos under the proposed scheme and Flare. Reconstructed tiles with different resolutions are scaled to the same area size for better display. Specifically, we consider two types of 360° videos: slow-paced

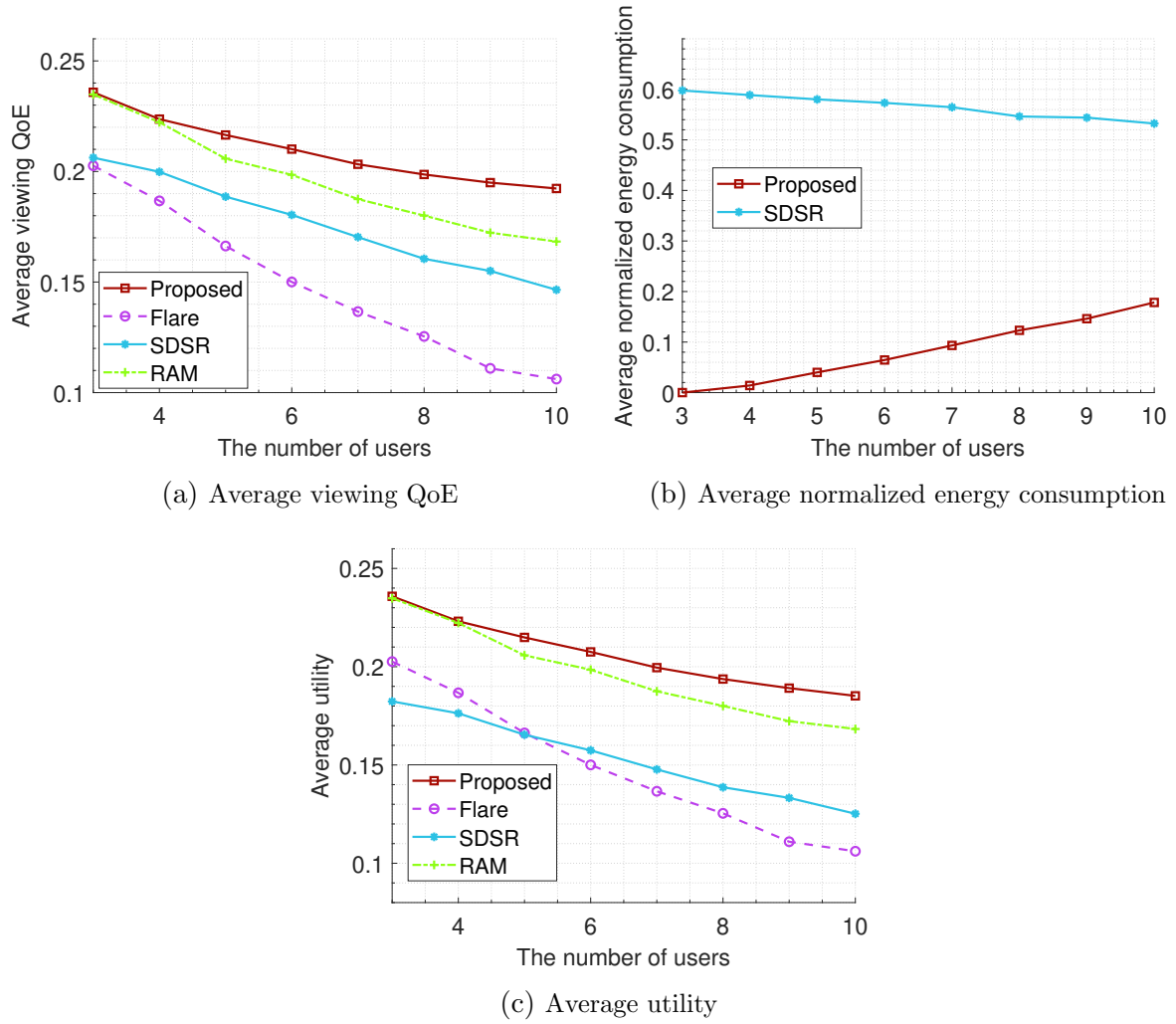


Figure 4.14: Performance of different schemes with varying number of users.



(a) Proposed

(b) Flare

Figure 4.15: Cropped snapshots of the generated demo videos (Coastal pier, slow-paced) under the proposed scheme and Flare.



(a) Proposed

(b) Flare

Figure 4.16: Cropped snapshots of the generated demo videos (Mega coaster, fast-paced) under the proposed scheme and Flare.

(Coastal pier) and fast-paced (Mega coaster). The total bandwidth of the BS is 2 MHz. The number of users is 4 and $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$. The images show the cropped FoV of a particular user while watching the 360° videos. By comparing the image qualities generated by both schemes, shown in Fig. 4.15 and Fig. 4.16, we can see that with the proposed two-step adaptive streaming scheme, the user enjoys better video quality with higher-resolution FoV tiles than Flare. For better visualization, we highlight some major areas showing video quality improvements using red rectangular boxes in both figures.

4.5 Summary

In this chapter, a two-step adaptive streaming framework has been proposed to support energy-efficient multi-user enhanced 360° video viewing experience. The best candidate chunk of each user at any slot is determined based on a time-difference approach. A single-slot problem is formulated for maximizing the total incremental QoE gain while minimizing the total incremental average device energy consumption. A practical PSO-based iterative algorithm is proposed to determine the optimal bandwidth allocation, bitrate level selection, and enhancement method selection decisions for all users. Simulation results validate the performance of our proposed framework over benchmark schemes.

Chapter 5

Conclusions and Future Research Directions

5.1 Conclusions

The objective of this PhD research is to achieve E2E performance enhancement for tile-based 360° VR video streaming service, considering its characteristics including various video tile properties, transmission rate fluctuations, and inevitable FoV prediction errors driven by viewing behavior dynamics (i.e., head movements).

We first present a customized slice-level transmission protocol based on QUIC to achieve enhanced VR video packet transmission while promptly reacting to network and viewing behavior dynamics. Various video tile properties are supported by tailoring the QUIC protocol, where explicit mapping relations between requested video tiles and QUIC streams are established. Two customized protocol functionalities, including packet filtering and caching-based packet retransmission, are proposed to remove outdated EL video data due to FoV prediction errors and to perform efficient packet loss recovery with disparate transmission reliability requirements. A slice-level packet header is designed to support enhanced slice-based VR video packet transmission with the proposed protocol functionalities. Simulation results demonstrate the effectiveness of the proposed transmission protocol in achieving short average video segment downloading time and high average video segment quality.

Then, we develop an E2E performance analytical model to evaluate the performance of slice-based and layer-encoded video streaming, which is abstracted as a multi-hop packet

service (i.e., processing or transmission) process. The impact of disparate service reliability requirements of BL and EL packets is studied in the E2E analytical modeling. Based on the analysis of the correlations between two consecutive nodes, a segment-based analysis framework is proposed. A two-stage queuing model is established to determine the approximate steady-state probability distribution of queue length at the first node in each segment, based on which the segment-level performance metrics are analyzed, including E2E BL/EL packet delay, deadline violation probability, and throughput. Then, under the assumption of a large packet service rate of the first node in each segment, the independence between two consecutive segments is achieved for analysis tractability, upon which the E2E performance metrics are obtained. Simulation results demonstrate the accuracy of the proposed analytical model as well as its effectiveness in determining sliced node packet service rates and transport protocol parameters.

Finally, we consider multi-user 360° video streaming in a time-slotted system and propose a two-step adaptive streaming scheme to achieve enhanced viewing QoE. Each video chunk is first prefetched based on the predictive FoV and then enhanced at a closer-to-playtime time instant based on the updated FoV with improved prediction accuracy. Transmission-driven and device-VSR-driven methods for chunk enhancement are proposed, which are adaptively enabled for device energy-efficient FoV quality enhancement. In order to determine when to prefetch a new video chunk or enhance a prefetched chunk, a time-difference approach is applied to characterize the incremental QoE gain of each user at any slot, based on which the best candidate chunk is determined. A single-shot problem is then formulated for maximizing the total incremental QoE gain while minimizing the total incremental average device energy consumption. A PSO-based solution is developed to determine the per-slot optimal decisions for different users, including bandwidth allocation, bitrate level selection, and enhancement method selection. Simulation results demonstrate that our proposed solution outperforms benchmark schemes in terms of average viewing QoE, average device energy consumption, and average utility.

The proposed transmission protocol can be integrated into the proposed streaming scheme. The former enables efficient video packet transmission and prompt responses to network and viewing behavior dynamics (e.g., random packet loss and sudden head movements) on a small timescale, i.e., during chunk prefetching or enhancement. The latter enables adaptive video content requests on a larger timescale, i.e., adaptive switching between chunk prefetching and enhancement, along with bitrate level selection.

5.2 Future Research Directions

This PhD research can be extended in several directions.

First, considering the interdependence among the same video tiles of different ELs, that is, an EL tile can only be decoded if the BL and all lower-layer EL tiles are successfully decoded, EL packets of multiple ELs may also have different levels of service reliability requirements. If any EL packet of an EL is discarded due to service deadline violation, the EL packets of higher ELs can also be discarded for transmission efficiency. The proposed customized transmission protocol and the performance analytical model can be extended to incorporate the impact of decoding dependency among multiple ELs to make them more practical for supporting layer-encoded video streaming service.

Second, in the proposed E2E performance analytical model, the packet service rate of each node within a video streaming slice is considered static, without considering the traffic load dynamics of other service traffic traversing the same physical path as the video streaming slice. In the planning stage of a (core) network slice, dedicated (virtual) network resources are reserved to support service provisioning with QoS satisfaction in the long run, while QoS isolation among different slices for different services is ensured. The amount of reserved network resources for a slice is determined by a network operator based on the QoS requirements and traffic characteristics of the service. In the operational stage, the reserved network resources of different embedded network slices are shared in real-time with multiplexing gain. Considering the dynamic traffic loads of different services, the actual packet service rates of nodes within the video streaming slice can vary [45]. Specifically, the actual packet service rates can be larger than those reserved in the planning stage, especially when the sliced network resources for other service traffic are not fully used. Thus, the proposed E2E performance analytical model can be extended to incorporate the impact of node packet service rate variations due to cross-traffic load dynamics of other network services.

Third, more advanced and fine-grained FoV prediction models can be developed to further enhance the performance of the proposed customized transmission protocol and two-step adaptive streaming scheme. For example, personalized FoV prediction models that differentiate head movement patterns among users can achieve more accurate FoV predictions. In addition, scalable FoV prediction models are desirable. They can take historical viewing trajectories of arbitrary sizes as input and predict future FoVs with different prediction time gaps, thereby better adapting to playback buffer and user viewing behavior dynamics. A scalable and more accurate FoV prediction model enhances the performance of the proposed packet filtering functionality, ensuring that only outdated EL

tiles are removed without loss of useful video data. It can also help achieve more efficient user viewing QoE optimization.

Fourth, advanced learning-based algorithms that consider temporal correlations during the streaming of multiple users can be developed for optimal decision-making in the proposed two-step adaptive streaming scheme. Due to the asynchronous streaming requests, the number of users at each slot may not be the same. Besides, the playback buffers of different users at the same slot may be dynamic, i.e., the number of prefetched chunks to be enhanced and the existence of the to-be-prefetched chunk. All of these make a general learning-based solution very complex and time-consuming. Both state and action spaces change over slots, and the feasible actions for each user also depend on its current state. The proposed PSO-based algorithm (i.e., Algorithm 1 in Chapter 4) is essentially a heuristic approach. It effectively improves multi-user viewing QoE with device energy efficiency. At the same time, it greatly reduces the complexity of the studied problem, which can also provide valuable insights for developing learning-based solutions.

Finally, the proposed two-step adaptive streaming scheme can be extended to consider an edge-device collaborative network architecture. An edge server can cache some (popular) video tiles and deliver them to the intended users with reduced latency [98, 99]. The computing capability of an edge server can be leveraged in two ways. On one hand, when multiple users request the same 360° video, instead of caching multiple bitrate level versions of a video tile, the video tile with a high bitrate level can be cached at the edge server and is transcoded into various lower bitrate level versions to save the caching space with additional transcoding delay [100, 101, 102]. On the other hand, the edge server can reconstruct a video tile from a low bitrate level to a higher bitrate level via the VSR technique to enhance its quality with extra reconstruction time and larger wireless transmission time due to the larger size of the reconstructed video tile.

References

- [1] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, “Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.
- [3] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, “End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework,” *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65–74, 2018.
- [4] H. Li, Z. Kong, Y. Chen, L. Wang, Z. Lu, X. Wen, W. Jing, and W. Xiang, “Slice-based service function chain embedding for end-to-end network slice deployment,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3652–3672, 2023.
- [5] M. Dai, G. Sun, H. Yu, and D. Niyato, “Maximize the long-term average revenue of network slice provider via admission control among heterogeneous slices,” *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 745–760, 2024.
- [6] 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.
- [7] S. Z. A. Ansari, V. K. Shukla, K. Saxena, and B. Filomeno, “Implementing virtual reality in entertainment industry,” in *Proc. Int. Conf. Cyber Intelligence and Information Retrieval*, 2022, pp. 561–570.

- [8] J. Kim, K. Kim, and W. Kim, “Impact of immersive virtual reality content using 360-degree videos in undergraduate education,” *IEEE Transactions on Learning Technologies*, vol. 15, no. 1, pp. 137–149, 2022.
- [9] M. Intelligence, “Virtual reality (VR) market–growth, trends, Covid-19 impact, and forecasts (2021–2026),” 2021.
- [10] J. van der Hooft, H. Amirpour, M. T. Vega, Y. Sanchez, R. Schatz, T. Schierl, and C. Timmerer, “A tutorial on immersive video delivery: From omnidirectional video to holography,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1336–1375, 2023.
- [11] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, “Pano: Optimizing 360 video streaming with a better understanding of quality perception,” in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 394–407.
- [12] Z. Huang, P. Yang, N. Zhang, F. Lyu, Q. Li, W. Wu, and X. S. Shen, “QoE-driven mobile 360 video streaming: Predictive view generation and dynamic tile selection,” in *Proc. IEEE/CIC International Conference on Communications in China (ICCC)*, 2021, pp. 1113–1118.
- [13] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, “Gaze prediction in dynamic 360 immersive videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5333–5342.
- [14] Q. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang, and T. Q. S. Quek, “Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming,” *IEEE Transactions on Multimedia*, vol. 24, pp. 1529–1544, 2022.
- [15] 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 *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1190–1198.
- [16] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, “Overview of SHVC: Scalable extensions of the high efficiency video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, 2015.
- [17] M. Torres Vega, C. Liaskos, S. Abadal, E. Papapetrou, A. Jain, B. Mouhouche, G. Kalem, S. Ergüt, M. Mach, T. Sabol *et al.*, “Immersive interconnected virtual

and augmented reality: A 5G and IoT perspective,” *Journal of Network and Systems Management*, vol. 28, pp. 796–826, 2020.

- [18] H. Wang, H. Dong, and A. E. Saddik, “Tile-weighted rate-distortion optimized packet scheduling for 360° virtual-reality video streaming,” *IEEE Intelligent Systems*, vol. 39, no. 4, pp. 60–72, 2024.
- [19] A. A. Baniya, T.-K. Lee, P. W. Eklund, and S. Aryal, “A survey of deep learning video super-resolution,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 4, pp. 2655–2676, 2024.
- [20] Y. Wei, Q. Ye, K. Qu, W. Zhuang, and X. Shen, “Customized transmission protocol for tile-based 360° VR video streaming over core network slices,” *IEEE/ACM Transactions on Networking*, vol. 33, no. 1, pp. 340–354, 2025.
- [21] Y. Wei, Q. Ye, K. Qu, W. Zhuang, and X. Shen, “E2E performance modeling for slice-based video streaming with layered encoding,” *IEEE Transactions on Networking*, to appear.
- [22] Y. Wei, Q. Ye, W. Zhuang, and X. Shen, “Energy-efficient multi-user adaptive 360° video streaming: A two-step approach with device video super-resolution,” *IEEE Transactions on Network Science and Engineering*, under revision.
- [23] 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.
- [24] 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*, vol. 22, no. 9, pp. 2366–2381, 2020.
- [25] H. Shi, Y. Cui, F. Qian, and Y. Hu, “DTP: Deadline-aware transport protocol,” in *Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019*, 2019, pp. 1–7.
- [26] C. Zhou, W. Wu, D. Yang, T. Huang, L. Guo, and B. Yu, “Deadline and priority-aware congestion control for delay-sensitive multimedia streaming,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4740–4744.
- [27] D. Su, L. Cui, L. Zhang, Y. Suo, and Y. Qiu, “Rate adaptation and block scheduling for delay-sensitive multimedia applications,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4828–4832.

- [28] H. K. Ravuri, M. T. Vega, J. Der Van Hooft, T. Wauters, and F. De Turck, “Adaptive partially reliable delivery of immersive media over QUIC-HTTP/3,” *IEEE Access*, vol. 11, pp. 38 094–38 111, 2023.
- [29] C. Cui, Y. Lu, S. Li, J. Li, and Z. Ruan, “DASH+: Download multiple video segments with stream multiplexing of QUIC,” in *Proc. 2022 Tenth International Conference on Advanced Cloud and Big Data (CBD)*, pp. 66–72.
- [30] S.-C. Yen, C.-L. Fan, and C.-H. Hsu, “Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol,” in *Proceedings of the 24th ACM Workshop on Packet Video*, 2019, pp. 7–12.
- [31] Y. Lu, Z. Ling, S. Zhu, and L. Tang, “SDTCP: Towards datacenter TCP congestion control with SDN for IoT applications,” *Sensors*, vol. 17, no. 1, p. 109, 2017.
- [32] Y. Xu, S. Shukla, Z. Guo, S. Liu, A. S.-W. Tam, K. Xi, and H. J. Chao, “RAPID: Avoiding TCP incast throughput collapse in public clouds with intelligent packet discarding,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1911–1923, 2019.
- [33] M.-H. Wang, L.-W. Chen, P.-W. Chi, and C.-L. Lei, “SDUDP: A reliable UDP-based transmission protocol over SDN,” *IEEE Access*, vol. 5, pp. 5904–5916, 2017.
- [34] J. Chen, Q. Ye, W. Quan, S. Yan, P. T. Do, P. Yang, W. Zhuang, X. Shen, X. Li, and J. Rao, “SDATP: An SDN-based traffic-adaptive and service-oriented transmission protocol,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 756–770, 2019.
- [35] S. Yan, Q. Ye, and W. Zhuang, “Learning-based transmission protocol customization for VoD streaming in cybertwin-enabled next-generation core networks,” *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 326–16 336, 2021.
- [36] F. Zou, Y. Wang, and Y. Liu, “A multipath routing approach for tile-based virtual reality video streaming based on sdn,” in *Proc. IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2021, pp. 560–565.
- [37] J. Iyengar and M. Thomson, “RFC 9000 QUIC: A UDP-based multiplexed and secure transport,” *Omtermet Emgomeeromg Task Force*, 2021.

- [38] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, “The quic transport protocol: Design and internet-scale deployment,” in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.
- [39] B. Davie and J. Gross, “A stateless transport tunneling protocol for network virtualization (STT),” Internet Engineering Task Force, Internet-Draft, Apr. 2016. [Online]. Available: <https://datatracker.ietf.org/doc/draft-davie-stt/08/>
- [40] W. Bai, L. Chen, K. Chen, D. Han, C. Tian, and H. Wang, “PIAS: Practical information-agnostic flow scheduling for commodity data centers,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 1954–1967, 2017.
- [41] W. Wei, J. Han, Y. Xing, K. Xue, J. Liu, and R. Zhuang, “MP-VR: An MPTCP-based adaptive streaming framework for 360-degree virtual reality videos,” in *Proc. ICC-IEEE International Conference on Communications*, 2021, pp. 1–6.
- [42] W. Yang, P. Dong, L. Cai, and W. Tang, “Loss-aware throughput estimation scheduler for multi-path TCP in heterogeneous wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 3336–3349, 2021.
- [43] M. Lecci, M. Drago, A. Zanella, and M. Zorzi, “An open framework for analyzing and modeling XR network traffic,” *IEEE Access*, vol. 9, pp. 129 782–129 795, 2021.
- [44] D. Bertsekas and R. Gallager, *Data networks*. Athena Scientific, 2021.
- [45] Q. Ye, W. Zhuang, X. Li, and J. Rao, “End-to-end delay modeling for embedded VNF chains in 5G core networks,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 692–704, 2018.
- [46] 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.
- [47] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, “360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 315–323.
- [48] O. Adamuz-Hinojosa, V. Sciancalepore, P. Ameigeiras, J. M. Lopez-Soler, and X. Costa-Pérez, “A stochastic network calculus (SNC)-based model for planning B5G uRLLC RAN slices,” *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1250–1265, 2022.

- [49] S. A. Hashemian and F. Ashtiani, “Analytical modeling and improvement of interference-coupled RAN slicing,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 13 472–13 486, 2024.
- [50] K. Qu, W. Zhuang, X. Shen, X. Li, and J. Rao, “Dynamic resource scaling for VNF over nonstationary traffic: A learning approach,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 648–662, 2021.
- [51] Q. Xu, J. Wang, and K. Wu, “Learning-based dynamic resource provisioning for network slicing with ensured end-to-end performance bound,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 28–41, 2020.
- [52] H. Wang, Y. Wu, G. Min, and W. Miao, “A graph neural network-based digital twin for network slicing management,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1367–1376, 2022.
- [53] D. Ayepah-Mensah, G. Sun, G. O. Boateng, S. Anokye, and G. Liu, “Blockchain-enabled federated learning-based resource allocation and trading for network slicing in 5G,” *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 654–669, 2024.
- [54] I. Afolabi, J. Prados-Garzon, M. Bagaa, T. Taleb, and P. Ameigeiras, “Dynamic resource provisioning of a scalable E2E network slicing orchestration system,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2594–2608, 2020.
- [55] H. H. Esmat and B. Lorenzo, “Self-learning multi-mode slicing mechanism for dynamic network architectures,” *IEEE/ACM Transactions on Networking*, vol. 32, no. 2, pp. 1048–1063, 2024.
- [56] J. Li, W. Shi, Q. Ye, N. Zhang, W. Zhuang, and X. Shen, “Multiservice function chain embedding with delay guarantee: A game-theoretical approach,” *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 219–11 232, 2021.
- [57] R. Gouareb, V. Friderikos, and A.-H. Aghvami, “Virtual network functions routing and placement for edge cloud latency minimization,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2346–2357, 2018.
- [58] S. Kamath, S. Singh, and M. S. Kumar, “Multiclass queueing network modeling and traffic flow analysis for SDN-enabled mobile core networks with network slicing,” *IEEE Access*, vol. 8, pp. 417–430, 2020.

- [59] Z. Wang, J. Zhang, and T. Huang, “Determining delay bounds for a chain of virtual network functions using network calculus,” *IEEE Communications Letters*, vol. 25, no. 8, pp. 2550–2553, 2021.
- [60] S. Ma, X. Chen, Z. Li, and Y. Chen, “Performance evaluation of URLLC in 5G based on stochastic network calculus,” *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1182–1194, 2021.
- [61] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, “Internet traffic tends toward Poisson and independent as the load increases,” *Nonlinear estimation and classification*, pp. 83–109, 2003.
- [62] S. U. Pillai, *Probability, random variables, and stochastic processes*. McGraw-Hill, 2002.
- [63] H. Abouee-Mehrizi and O. Baron, “State-dependent M/G/1 queueing systems,” *Queueing Systems*, vol. 82, no. 1, pp. 121–148, 2016.
- [64] O. Brun and J.-M. Garcia, “Analytical solution of finite capacity M/D/1 queues,” *Journal of Applied Probability*, vol. 37, no. 4, pp. 1092–1098, 2000.
- [65] K. Nakagawa, “On the series expansion for the stationary probabilities of an M/D/1 queue,” *Journal of The Operations Research Society of Japan*, vol. 48, no. 2, pp. 111–122, 2005.
- [66] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, “Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach,” *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 272–288, 2021.
- [67] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, “BOLA: Near-optimal bitrate adaptation for online videos,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.
- [68] B. Cheng, M. Wang, X. Lin, and J. Chen, “Context-aware cognitive QoS management for networking video transmission,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1422–1434, 2021.
- [69] O. S. Peñaherrera-Pulla, C. Baena, S. Fortes, E. Baena, and R. Barco, “KQI assessment of VR services: A case study on 360-video over 4G and 5G,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5366–5382, 2022.

- [70] Z. Ye, Q. Li, X. Ma, D. Zhao, Y. Jiang, L. Ma, B. Yi, and G.-M. Muntean, “VRCT: A viewport reconstruction-based 360 video caching solution for tile-adaptive streaming,” *IEEE Transactions on Broadcasting*, vol. 69, no. 3, pp. 691–703, 2023.
- [71] J. Shi, L. Pu, X. Yuan, Q. Gong, and J. Xu, “Sophon: Super-resolution enhanced 360 video streaming with visual saliency-aware prefetch,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3124–3133.
- [72] 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*, vol. 32, no. 3, pp. 1607–1623, 2021.
- [73] D. V. Nguyen, H. T. Tran, A. T. Pham, and T. C. Thang, “An optimal tile-based approach for viewport-adaptive 360-degree video streaming,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 29–42, 2019.
- [74] H. Zhang, Y. Ban, Z. Guo, K. Chen, and X. Zhang, “RAM360: Robust adaptive multi-layer 360° video streaming with Lyapunov optimization,” *IEEE Transactions on Multimedia*, vol. 25, pp. 4225–4239, 2023.
- [75] J. Chen, M. Hu, Z. Luo, Z. Wang, and D. Wu, “SR360: Boosting 360-degree video streaming with super-resolution,” in *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2020, pp. 1–6.
- [76] L. Zhang, H. Zhou, H. Wang, and L. Cui, “TBSR: Tile-based 360° video streaming with super-resolution on commodity mobile devices,” in *Proc. IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, pp. 501–510.
- [77] J. Zeng, X. Zhou, and K. Li, “Toward high-quality low-latency 360° video streaming with edge-client collaborative caching and super-resolution,” *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 29 020–29 034, 2024.
- [78] B. Chai, J. Chen, Z. Luo, Z. Wang, M. Hu, Y. Zhou, and D. Wu, “SDSR: Optimizing metaverse video streaming via saliency-driven dynamic super-resolution,” *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 4, pp. 978–989, 2023.
- [79] H. Guo, F. Wang, W. Zhang, Y. Zhu, L. Cui, J. Liu, F. R. Yu, and L. Zhang, “Joint adaptation for mobile 360-degree video streaming and enhancement,” *IEEE Transactions on Mobile Computing*, early access, Apr. 1, 2025, doi: 10.1109/TMC.2025.3555322.

- [80] A. Zhang, Q. Li, Y. Chen, X. Ma, L. Zou, Y. Jiang, Z. Xu, and G.-M. Muntean, "Video super-resolution and caching—An edge-assisted adaptive video streaming solution," *IEEE Transactions on Broadcasting*, vol. 67, no. 4, pp. 799–812, 2021.
- [81] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.
- [82] Y. Zhang, Y. Zhang, Y. Wu, Y. Tao, K. Bian, P. Zhou, L. Song, and H. Tuo, "Improving quality of experience by adaptive video streaming with super-resolution," in *IEEE INFOCOM - IEEE Conference on Computer Communications*, 2020, pp. 1957–1966.
- [83] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. IEEE INFOCOM - IEEE Conference on Computer Communications*, 2020, pp. 1977–1986.
- [84] W. Zhou and Z. Wang, "Quality assessment of image super-resolution: Balancing deterministic and statistical fidelity," in *Proceedings of the 30th ACM international conference on multimedia*, 2022, pp. 934–942.
- [85] H. Zhao, B. Zheng, S. Yuan, H. Zhang, C. Yan, L. Li, and G. Slabaugh, "CBREN: Convolutional neural networks for constant bit rate video quality enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4138–4149, 2021.
- [86] S. Kumar, J. Jin, Y.-N. Dong *et al.*, "Seer: Learning-based 360° video streaming for MEC-equipped cellular networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3308–3319, 2023.
- [87] J. Zeng, X. Zhou, and K. Li, "MADRL-based joint edge caching and bitrate selection for multicategory 360 video streaming," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 584–596, 2023.
- [88] X. Tan, S. Wang, X. Xu, Q. Zheng, J. Yang, and S. Chen, "DACOD360: Deadline-aware content delivery for 360-degree video streaming over MEC networks," *IEEE Transactions on Multimedia*, vol. 26, pp. 4168–4182, 2024.
- [89] X. Liu, J. Yan, Z. Wan, Y. Fang, and Z. Wang, "A quality-of-experience database for adaptive omnidirectional video streaming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 5, pp. 949–963, 2023.

- [90] Z. Duanmu, W. Liu, D. Chen, Z. Li, Z. Wang, Y. Wang, and W. Gao, “A bayesian quality-of-experience model for adaptive streaming videos,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 18, no. 3s, pp. 1–24, 2023.
- [91] S. Burer and A. N. Letchford, “Non-convex mixed-integer nonlinear programming: A survey,” *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, 2012.
- [92] D. Tarekegn Nigatu, T. Gemechu Dinka, and S. Lulseged Tilahun, “Convergence analysis of particle swarm optimization algorithms for different constriction factors,” *Frontiers in Applied Mathematics and Statistics*, vol. 10, p. 1304268, 2024.
- [93] H. Huang, J. Qiu, and K. Riedl, “On the global convergence of particle swarm optimization methods,” *Applied Mathematics & Optimization*, vol. 88, no. 2, p. 30, 2023.
- [94] Y. Wei, Q. J. Ye, K. Qu, W. Zhuang, and X. S. Shen, “Transmission protocol customization for on-demand tile-based 360° VR video streaming,” in *Proc. 2024 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1069–1074.
- [95] N. Gao, G. Liu, M. Feng, X. Hua, and T. Jiang, “Non-orthogonal multiple access enhanced scalable 360-degree video multicast,” *IEEE Transactions on Multimedia*, vol. 26, pp. 8488–8503, 2024.
- [96] X. Ye, K. Qu, W. Zhuang, and X. Shen, “Accuracy-aware cooperative sensing and computing for connected autonomous vehicles,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8193–8207, 2023.
- [97] 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*, 2018, pp. 99–114.
- [98] T. Yang, Z. Tan, Y. Xu, and S. Cai, “Collaborative edge caching and transcoding for 360° video streaming based on deep reinforcement learning,” *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 551–25 564, 2022.

- [99] Y. Jin, J. Liu, F. Wang, and S. Cui, “Ebublio: Edge-assisted multiuser 360° video streaming,” *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 408–15 419, 2023.
- [100] P. Maniotis, E. Bourtsoulatze, and N. Thomos, “Tile-based joint caching and delivery of 360° videos in heterogeneous networks,” *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2382–2395, 2020.
- [101] M. Li, J. Gao, C. Zhou, X. Shen, and W. Zhuang, “User dynamics-aware edge caching and computing for mobile virtual reality,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 5, pp. 1131–1146, 2023.
- [102] H. Xiao, C. Xu, Z. Feng, R. Ding, S. Yang, L. Zhong, J. Liang, and G.-M. Muntean, “A transcoding-enabled 360 VR video caching and delivery framework for edge-enhanced next-generation wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1615–1631, 2022.

APPENDICES

Appendix A

Proof of Proposition 1

For each candidate chunk i , $Q_{t,u,i}^1$, $Q_{t+1,u,i-1}^1$, $Q_{t,u,i-1}^1$, $Q_{t+1,u,i+1}^1$, and $Q_{t,u,i+1}^1$ are known as constants, with $Q_{t+1,u,i-1}^1 = Q_{t,u,i-1}^1$ and $Q_{t+1,u,i+1}^1 = Q_{t,u,i+1}^1$. For brevity, let $Q_{t+1,u,i}^1 = x$, $Q_{t+1,u,i-1}^1 = Q_{t,u,i-1}^1 = C_1$, $Q_{t,u,i}^1 = C_2$, $Q_{t+1,u,i+1}^1 = Q_{t,u,i+1}^1 = C_3$, and (4.18) is rewritten as

$$f(x) = \alpha \frac{1}{|\mathcal{I}_u|} (x - C_2) - \frac{1}{|\mathcal{I}_u|-1} \gamma [(|x - C_1| - |C_2 - C_1|) e_{t,u,i-1} + (|C_3 - x| - |C_3 - C_2|) e_{t,u,i+1}]. \quad (\text{A.1})$$

We first consider the case when $e_{t,u,i-1} = e_{t,u,i+1} = 1$. In this case, we have

$$f(x) = f_A(x) - f_B(x) = \alpha \frac{1}{|\mathcal{I}_u|} (x - C_2) - \frac{1}{|\mathcal{I}_u|-1} \gamma \left[(|x - C_1| - |C_2 - C_1|) + (|C_3 - x| - |C_3 - C_2|) \right] \quad (\text{A.2})$$

with $f'_A(x) = \alpha \frac{1}{|\mathcal{I}_u|} > 0$. Next, we analyze the monotonicity of $f_B(x)$, which has the following two cases.

(1) $C_1 \leq C_3$

If $C_1 \leq x \leq C_3$, $f'_B(x) = 0$ and $f'(x) = \alpha \frac{1}{|\mathcal{I}_u|} > 0$. If $x \leq C_1$, $f'_B(x) = -\frac{2}{|\mathcal{I}_u|-1} \gamma$ and $f'(x) = \alpha \frac{1}{|\mathcal{I}_u|} + \frac{2}{|\mathcal{I}_u|-1} \gamma > 0$. If $x \geq C_3$, $f'_B(x) = \frac{2}{|\mathcal{I}_u|-1} \gamma$ and $f'(x) = \alpha \frac{1}{|\mathcal{I}_u|} - \frac{2}{|\mathcal{I}_u|-1} \gamma$, which can be either larger than, smaller than, or equal to 0. Therefore, when $C_1 \leq C_3$, $f(x)$ monotonically increases with x when $x \leq C_3$ and monotonically increases or decreases with x when $x \geq C_3$. Considering the domain of x , i.e., $C_2 \leq x \leq f_{t+1,u,i}$, the maximum of $f(x)$ is obtained by $\max(f(C_2), f(f_{t+1,u,i}))$ with $f(C_2) = 0$.

(2) $C_1 \geq C_3$

Similar monotonicity analysis can be applied to the case when $C_1 \geq C_3$, which is omitted for brevity.

When $e_{t,u,i-1} = 0$ and/or $e_{t,u,i+1} = 0$, it is a special case of $f(x)$. Similar monotonicity analysis can be applied, which is omitted for brevity.