

Copyright
by
Chao Chen
2013

The Dissertation Committee for Chao Chen
certifies that this is the approved version of the following dissertation:

**Adaptive Video Transmission over Wireless Channels
with Optimized Quality of Experiences**

Committee:

Robert W. Heath Jr., Supervisor

Alan C. Bovik, Supervisor

Constantine Caramanis

Sanjay Shakkottai

Gustavo de Veciana

Lili Qiu

**Adaptive Video Transmission over Wireless Channels
with Optimized Quality of Experiences**

by

Chao Chen, B.E., M.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2013

Dedicated to my parents,
Xiaoqiu Chen and Lianzhen Yang.

Acknowledgments

First and foremost, I express my deepest gratitude to my supervisor Professors Alan C. Bovik and Professor Robert W. Heath Jr., for their excellent guidance, patience, and continuous support. Professor Bovik led me into a whole new world of visual quality assessment, and Professor Heath encouraged me to work out quality metrics that are meaningful and tractable for wireless network design. Their acute vision profoundly inspired me, and they gave me a great deal of freedom on my research. Their trust and encouragement were fundamental to all my accomplishments at UT.

I thank my teacher and friend, Prof. Gustavo de Veciana. He included me into his biweekly meeting with his students, which greatly broadened my view and inspired many of my research ideas. I also thank my committee members, Professor Constantine Caramanis, Professor Lili Qiu, and Professor Sanjay Shakkottai. Their invaluable comments and suggestions greatly improved the quality of my research. I owe a great debt of gratitude to my coworkers Lark Kown Choi and Xiaoqing Zhu. Without Lark, I would not have the “Lark” to finish the subjective study successfully. Xiaoqing provided excellent suggestions, which greatly enhanced the value of our work from the perspective of industry.

My heartfelt thanks also go to my colleagues in the VAWN project at

UT: Zheng Lu, Vinay Joseph, and Sarabjot Singh. Their excellent research work set great examples for me. And I thank my lab mates in WSIL: Salam Akoum, Ramya Bhagavatula, Omar El Ayach, Behrang Nosrat Makouei, Kien Trung Truong, Amin Abdel Khalek, Ahmed Alkhateeb, Tianyang Bai, Insoo Hwang, Steven Peters, Namyoon Lee, Ken Pesyna, Andrew Thornburg, and Vutha Va. I will always remember the surprise birthday cake they gave to me. It also has been a great pleasure to work with my colleagues in LIVE lab: Michele Saad, Che-Chun Su, Anish Mittal, Greg Freeman, Ming-Jun Chen, Gautam Muralidhar, Rajiv Soundararajan, Anush Moorthy, Sina Jahanbin, Janice Pan, and Yang Liu.

My life at UT would never have been so exciting without the “Chinese Basketball Club”: Hongbo Si, Chao Jia, Ping Xia, Yudong Chen, Jiaming Xu, Yuhuan Du, Yingzhe Li, Yicong Wang, Yan Cai, Yingxi Liu, and Tong Zhao. I am also very grateful for my friends: Yuxin Chen, Jing Lin, Xiaohu Shen, and Aibo Tian. Without their generous help, I would not have adapted so well to my life in the United States.

Last, but definitely not least, I thank my parents, Xiaoqiu Chen and Lianzhen Yang, for their love and relentless support throughout my education. I thank my mother for her sacrifices for the family. She believed in me and gave me great encouragement at all stages of my endeavors. My father is not only a man of paternal love but also a perfect model of a scientist. His passion for and diligence and persistence in his research work all shaped me since my childhood. I thank him and my mother for making our family full of wonders.

Adaptive Video Transmission over Wireless Channels with Optimized Quality of Experiences

Publication No. _____

Chao Chen, Ph.D.
The University of Texas at Austin, 2013

Supervisors: Robert W. Heath Jr.
Alan C. Bovik

Video traffic is growing rapidly in wireless networks. Different from ordinary data traffic, video streams have higher data rates and tighter delay constraints. The ever-varying throughput of wireless links, however, cannot support continuous video playback if the video data rate is kept at a high level. To this end, adaptive video transmission techniques are employed to reduce the risk of playback interruptions by dynamically matching the video data rate to the varying channel throughput. In this dissertation, I develop new models to capture viewers' quality of experience (QoE) and design adaptive transmission algorithms to optimize the QoE. The contributions of this dissertation are threefold.

First, I develop a new model for the viewers' QoE in rate-switching systems in which the video source rate is adapted every several seconds. The model is developed to predict an important aspect of QoE, the time-varying

subjective quality (TVSQ), i.e., the up-to-the-moment subjective quality of a video as it is played. I first build a video database of rate-switching videos and measure TVSQs via a subjective study. Then, I parameterize and validate the TVSQ model using the measured TVSQs. Finally, based on the TVSQ model, I design an adaptive rate-switching algorithm that optimizes the time-averaged TVSQs of wireless video users.

Second, I propose an adaptive video transmission algorithm to optimize the Overall Quality (OQ) of rate-switching videos, i.e., the viewers' judgement on the quality of the whole video. Through the subjective study, I find that the OQ is strongly correlated with the empirical cumulative distribution function (eCDF) of the video quality perceived by viewers. Based on this observation, I develop an adaptive video transmission algorithm that maximizes the number of video users who satisfy given constraints on the eCDF of perceived video qualities.

Third, I propose an adaptive transmission algorithm for scalable videos. Different from the rate-switching systems, scalable videos support rate adaptation for each video frame. The proposed adaptive transmission algorithm maximizes the time-averaged video quality while maintaining continuous video playback. When the channel throughput is high, the algorithm increases the video data rate to improve video quality. Otherwise, the algorithm decreases the video data rate to buffer more videos and to reduce the risk of playback interruption. Simulation results show that the performance of the proposed algorithm is close to a performance upper bound.

Table of Contents

Acknowledgments	v
Abstract	vii
List of Tables	xii
List of Figures	xiv
Chapter 1. Introduction	1
1.1 Wireless Video Communication	1
1.2 Adaptive Video Transmission	4
1.3 Challenges in Adaptive Video Transmission	8
1.4 Thesis Statement	11
1.5 Contributions of This Dissertation	11
1.6 Notation	14
1.7 Organization	15
Chapter 2. Modeling and Optimizing the Time-Varying Subjective Quality for Rate-switching Systems	16
2.1 Proposed TVSQ Prediction Method	17
2.1.1 Overview of the TVSQ Prediction Method	18
2.1.2 Model Structure	19
2.2 Subjective Study for Model Parameterizations	21
2.2.1 Video Database Construction	21
2.2.2 Subjective Experiments	30
2.3 Model Parametrization	34
2.3.1 Model Parameter Training	34
2.3.2 Model Order Selection	39
2.4 Model Evaluation and Analysis	42

2.4.1	Performance Evaluation	42
2.4.2	Cross Validation	51
2.4.3	Model Analysis and Simplification	52
2.5	TVSQ-optimized Rate-Switching Algorithm	56
2.5.1	System Model	56
2.5.2	Adaptive Transmission Algorithm Design	61
2.5.3	Simulation Results	64
2.6	Summary	66
Chapter 3.	Adaptive Video Transmission with Overall Quality Constraints	68
3.1	Proposed Overall Quality Metric	69
3.2	The System Model for Wireless Video Transmission	71
3.2.1	System Overview	71
3.2.2	Channel Model and Rate-Quality Model	72
3.2.3	Constraints on the Quality of Experience	74
3.3	Adaptive Transmission without Quality Expectation	77
3.3.1	Rate Adaptation Algorithm	77
3.3.2	Admission Control Strategy	81
3.3.3	Online Algorithm for Threshold Optimization	84
3.3.4	Simulation Results	88
3.4	Adaptive Transmission with Quality Expectation	90
3.4.1	Rate Adaptation and Admission Control	92
3.4.2	Threshold Optimization Algorithm	93
3.4.3	Simulation Results	97
3.5	Summary	99
Chapter 4.	Adaptive Transmission of Stored Scalable Videos	100
4.1	Introduction	101
4.2	System Model	103
4.2.1	System Overview	104
4.2.2	Video Codec Configuration	104
4.2.3	Rate-Quality Model	106

4.2.4	Streaming Setup	108
4.3	Markov Decision Process-Based Model	110
4.3.1	Scheduling Policy and State Space	110
4.3.2	Optimization Objective	115
4.3.3	Finite State Problem Formulation	115
4.3.4	Policy Optimization	119
4.3.5	Performance Upper Bound	121
4.3.6	Performance Evaluation	124
4.4	Near-optimal Heuristic On-line Scheduling Algorithm	125
4.4.1	Channel Model Simplification	126
4.4.2	Layer Selection	127
4.4.3	Resource Allocation Among GOPs	129
4.4.4	Performance Evaluation	132
4.5	Summary	134
Chapter 5.	Conclusions	135
5.1	Summary of the Dissertation	135
5.2	Future work	137
Appendices		145
Appendix A.	Instructions for Subjective Study	146
Appendix B.	Gradient Calculation for Model Identification	147
Appendix C.	Proof for Theorem 3.2.1	149
Appendix D.	Proof of Theorem 3.3.1	150
Appendix E.	Transition Probability	153
Appendix F.	Computation of t_μ and $\tilde{\mathbb{P}}_\mu$	157
Appendix G.	Simulation Settings in Chapter 4	160
Bibliography		163

List of Tables

1.1	The advantages and disadvantages of the adaptive video transmission techniques based on rate-switching and scalable video coding.	9
2.1	A brief description of the video clips in the rate-switching video database.	22
2.2	Optimized parameters of the proposed HW model.	43
2.3	The performance of the TVSQ prediction model on the TVSQ database	43
2.4	Results of the F-test performed on the prediction errors. Each entry in the table is a symbol of “0”, “-”, or “1”. A symbol value of “1” indicates that the statistical performance of the method in the row is superior to that of the method in the column. A symbol value of “0” indicates that the statistical performance of the method in the row is inferior to that of the method in the column and - indicates that the statistical performance of the method in the row is equivalent to that of the method in the column. M1 through M6 are “PSNR without HW”, “SSIM without HW”, “RRED without HW”, “PSNR with HW”, “SSIM with HW”, and “RRED with HW” respectively	51
2.5	Results of leave-one-out cross-validations. Here $\{n_1, \dots, n_2\}$ denotes the set of video sequences with sequence numbers from n_1 to n_2	52
2.6	Optimized parameters of the proposed HW model with linear output function.	54
2.7	Performance of the TVSQ prediction model when the output nonlinearity is replaced with a linear function	55
2.8	Notations used in the system model of the TVSQ-optimized wireless video transmission system.	61
3.1	The correlation of different metrics with the OQ measured in the subjective study	71
3.2	Frequently used notations in Chapter 3	76

4.1	Frequently Used Notations in Chapter 4.	109
4.2	The performance of the near-optimal policy in SSIM-predicted DMOS. $f^d = 5$	125
4.3	The performance of the near-optimal policy in SSIM-predicted DMOS. $f^d = 3$	125
G.1	The encoding parameters and rate-quality model parameters of the tested sequences.	162

List of Figures

1.1	A typical wireless video transmission system. It consists of three parts: the video sources, the wireless network, and the video viewers.	2
1.2	(a) The temporal variation of the video source rate and the channel throughput. The video source rate is measured in megabytes per second of video while the channel throughput is measured in megabytes per second of time; (b) The corresponding temporal variation of receiver buffer states. The solid line plots the amount of video buffered at the receiver, which is measured in seconds of video. The dashed line plots the amount of video that has been played back. Once all the buffered video has been played out, the playback process is interrupted.	5
1.3	(a) The video source rate is adapted to match the variation of channel throughput. (b) The video data buffered at the receiver is increasing linearly. Playback interruption is avoided.	6
1.4	(a) Video source rate adaptation using bitrate switching. (b) Video source rate adaptation using scalable video coding. The dashed arrows indicate the encoding order between the quality layers and video frames.	7
1.5	At any time t , the optimal video source rate depends on the buffer state. The buffer state, together with the selected video source rate, then determines the buffer state at $t + 1$	10
2.1	Proposed paradigm for TVSQ prediction. The STSQ of each second is predicted using the RRED index. An dynamic system model is then employed to predict the TVSQ of each second.	18
2.2	Proposed Hammerstein-Wiener model for TVSQ prediction. The core of the HW model is a linear filter. At the input and output of the HW model, two non-linear static functions are employed to model the non-linearities in the human response.	20
2.3	Sample frames of the video clips involved in the subjective study. The abbreviations of the names of the videos can be found in Table. 2.1.	23
2.4	The construction of the reference videos. The abbreviation of the names of the clips can be found in Table. 2.1.	24

2.5	(a) The STSQ of each compressed version of the reference video is shown in different colors. (b) A example of the designed target video quality $q^{\text{tgt}}(t)$ and the actual video quality $q^{\text{st}}(t)$ of the video sequence used in my database.	26
2.6	The design of the target STSQs. The durations of each segment d_1, d_2, \dots are realizations of $\bar{D}(1), \bar{D}(2), \dots$. The STSQ levels q_1, q_2, \dots are realizations of $\bar{Q}(1), \bar{Q}(2), \dots$	28
2.7	The user interface used in the subjective study. A continuous scale sliding bar is displayed near the bottom of the screen. The subject could continuously move the bar via a mouse to express his/her judgment of the video quality as each video is played.	31
2.8	The function $U_\nu(x, \epsilon)$ with different configurations of the parameter ν . The three plots shows $U_{0.5}(x, \epsilon)$, $U_1(x, \epsilon)$, and $U_5(x, \epsilon)$, respectively.	36
2.9	The (a) Lipschitz quotient and (b) Description length of the proposed TVSQ prediction model at different model order r	41
2.10	The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 1 of 5.	44
2.11	The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 2 of 5.	45
2.12	The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 3 of 5.	46
2.13	The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 4 of 5.	47
2.14	The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 5 of 5.	48
2.15	The performance comparison between the proposed TVSQ prediction method and the baselines. The length of the bars shows the linear correlation coefficients achieved by the TVSQ prediction methods. The 95% confidence interval is also shown for reference.	50
2.16	Input and output nonlinearities of the HW model parameterized on the TVSQ database. The input nonlinearity is concave and the output nonlinearity is close to a linear function.	53
2.17	The architecture of the considered wireless network. The video sources are stored in the content delivery networks. The proxy is co-located with the base station.	57
2.18	The impulse response of the linear filter in (2.2). It is seen that all the tabs are positive, which justifies the optimality of the proposed rate adaptation algorithm.	64

2.19	The average TVSQ of the proposed rate-switching algorithm. Two baseline are selected for comparison. The first baseline algorithm maximizes the sum of transmission rate in each slot. The second baseline algorithm maximizes the sum of STSQs in each slot.	66
3.1	The absolute values linear correlation coefficient (LCC) between $F^{(2)}(x)$ and the mean opinion scores when different x is selected.	70
3.2	The proposed OQ-constrained video streaming system. The admission control algorithm decide to admit or reject a newly arrived video user to the network. The rate adaptation algorithm determines the video rate to each admitted video user.	73
3.3	The simulated $g(k)$ and $e(k)$. $g(k)$ is the probability that a video user's OQ constraints are satisfied when the threshold is k . $e(k)$ is the probability that a video user is admitted into the network but its OQ constraints are not satisfied.	85
3.4	(a) The 2 nd -order eCDFs of the video users when $\alpha = 12$ and Algorithm 4 is employed for rate adaptation. (b) The 2 nd -order eCDFs of the video users when $\alpha = 12$ and rate vector is adapted to maximize $\sum_{u \in \mathcal{U}^{\text{av}}(t)} q_u^{\text{st}}(t)$	89
3.5	(a) Simulation results of the proposed admission control policy under different channel scaling parameters. Each data point on the figure is obtained by simulating 2000 video user arrivals. The Y-axis indicates the percentage of users who satisfy the OQ constraints among all video users that arrive to the network. (b) The convergence of Algorithm 6 when the scaling parameter is $\alpha = 6$	91
3.6	(a) The probability of admitted Type-1 video users whose OQ constraints are violated. (b) The probability of admitted Type-2 video users whose OQ constraints are violated. (c) The probability of video users whose OQ constraints are satisfied. All results are shown in percentage.	95
3.7	Simulation results of the proposed admission control policy under different channel scaling parameters.	97
3.8	The updated threshold vector \mathbf{k}^n 's of Algorithm 7 in the first 50 iterations are shown in (a). The dashed box is shown in (b) to illustrate more details. The contours of $g(\mathbf{k})$ is also shown on the figure for reference.	98

4.1	Dynamic scheduling system for wireless video transmission. The scheduler is located at the the transmitter. Scheduling decisions are based on the channel and receiver buffer state fed backed by the receiver.	105
4.2	Encoder prediction structure when $L = 3$. The prediction order is indicated by arrows. The data unit index (f, ℓ) is also shown on each data unit.	105
4.3	An illustration of the rate-quality function $q_f(z_f)$ for the f^{th} frame. The rate-quality function $q_f(z_f)$ is piecewise constant and right-continuous (solid). Its concave envelope $\hat{q}_f(z_f)$ is also shown (dotted).	107
4.4	An illustration of the receiver buffer state when $F^{\text{GOP}} = 8$, $L = 3$. $\mathbf{v}_t^{\text{pre}} = (4, 2, 1)$, $\mathbf{v}_t^I = (6, 2)$, $\mathbf{v}_t^{\text{post}} = (3, 3, 1)$	113
4.5	The dynamics of the system Π_μ and the corresponding simplified system $\tilde{\Pi}_\mu$	118
4.6	Given different relationship between $d(\mathbf{s})$ and $\Gamma(\ell, \mathbf{s})$, the proportions of states corresponding to different $L^{\text{sch}}(\mathbf{s})$ are shown in different colors. Results are obtained under Rayleigh fading channels with different Doppler shifts (5Hz in (a) and 3Hz in (b)) and are calculated on 5 different video sequences (“bus”, “foreman”, “flower”, “mobile”, “Paris”).	130
4.7	Performance comparison of different scheduling algorithms. The STSQ is measured in DMOS which is predicted by MS-SSIM using equation (4.13).	133
5.1	The relationship between MOS and QoE. Because a higher MOS implies a better QoE, QoE is an increasing function of MOS. But a larger difference in MOS (see ΔMOS_1 and ΔMOS_2) does not imply a larger difference in QoE (see ΔQoE_1 and ΔQoE_2).	138
5.2	A generic QoE model for natural viewing environment. It maps the environment features and the video-quality features to the QoE features.	141
5.3	An adaptive video transmission system. Video packets are fetched from the server and buffered by the base station. Then, the packets are forwarded to the users.	142
5.4	The decentralized rate adaptation algorithm. The base station signals the price of network resources to each user. Each user then determines its rate by solving an optimization problem independently.	144

Chapter 1

Introduction

If a picture is worth a thousand words, a video is worth a million. Individuals share video clips on social networks to exchange ideas. Television stations produce video programs to attract subscribers. Companies buy video advertisements to expand their businesses. Videos are undoubtedly the most important and effective carriers of information. In recent years, the ubiquitous mobile services have further exploded the demands of communicating videos over wireless networks. According to Cisco Visual Networking Index [28], mobile video traffic will increase sixteen-fold and count for 66% of the mobile traffic by the year 2017. Therefore, designing wireless video transmission systems that provide good Quality of Experiences (QoE) for video users is of critical importance. In this chapter, I introduce the background of wireless video communication systems, review existing adaptive video transmission techniques, and outline the contributions of this dissertation.

1.1 Wireless Video Communication

The structure of a typical wireless video transmission system is shown in Fig. 1.1. It consists of three parts: the video sources, the wireless network,

and the video viewers.

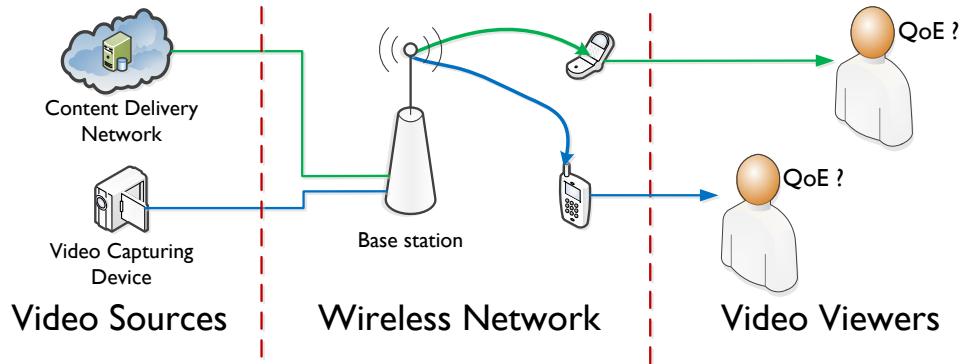


Figure 1.1: A typical wireless video transmission system. It consists of three parts: the video sources, the wireless network, and the video viewers.

Video sources are the video files to be transmitted. The video sources can be classified into two types: pre-stored videos and real-time videos. Pre-stored videos are created by content providers and then stored on video servers in Content Delivery Networks (CDNs). The videos streamed from Netflix [4] or YouTube [6] are pre-stored videos. Since the entire video file is available at the time of transmission, at any moment, the video content to be delivered in the future is known. Real-time videos are transmitted while the videos are being captured. Live videos and telephony videos belong to this category. Different from pre-stored videos, the contents of real-time videos are only partially known at the time of transmission. The transmitter only knows the part that has already been captured. The video data to be delivered in the future is not known.

Video files contain huge amount of data. Hence, video data are compressed by video encoders before transmission. The compressed video files are then encapsulated into video packets for delivery. Most existing video encoders such as H.264/AVC [87] and H.265/HEVC [80] are based on lossy compression techniques, which introduce distortions when compressed. Usually, the video encoder provides an interface to control the level of distortion. A lower level of distortion is accompanied with a larger video file size. In the following, I refer to the video file before compression as the pristine video.

Wireless networks deliver compressed video packets to video viewers. Typically, a wireless network consists of a backhaul network, a base station, and a wireless down-link. The video data packets are first transmitted to the base station through a backhaul network. Then, the base station forwards the packets to the wireless devices. Finally, the video packets buffered at the memory of wireless devices are decoded and played back to video viewers. Usually, the backhaul network has a very high transmission capacity, but the capacity of the wireless downlink is much lower. The mobility of wireless devices also causes variations in wireless channel capacity. Therefore, the wireless down-link is the major bottleneck of wireless video transmission.

The video viewers judge the Quality of Experience (QoE) of the received videos. The QoE of a video can be measured through a subjective study in which a subject scores the video according to his or her satisfaction with respect to the received video. The QoE can be affected by the content of the transmitted video, the viewer’s personal preference, and the perceptual

quality of the video [30, 37, 40]. This dissertation focuses on improving the QoE by optimizing the design of the wireless networks, which can neither control the video contents nor affect viewers' preferences. Thus, I focus on the QoE of typical video contents for typical users. To this end, this dissertation employs a *normalized QoE* that averages out the impact of video contents and viewers' preferences. The normalized QoE can be measured by computing the Difference Mean Opinion Score (DMOS) from subjective studies. The way to compute DMOS is specified by the ITU standard [37]. In the following, I use the terms normalized QoE and QoE interchangeably.

1.2 Adaptive Video Transmission

In video transmission systems, the video is played back to the viewer while it is being transmitted. Once all the video data buffered at the receiver has been played out, the playback process is interrupted and the viewers' QoE is damaged. In wireless channels, the channel throughput for each user varies over time [29]. If the data rate of the video source does not match the varying channel throughput, playback interruptions may happen. This is illustrated in Fig. 1.2. When the transmission is initiated, the receiver first waits for a while until a certain amount of video has been buffered. Then, the receiver starts playing back the buffered video at the rate of 1 second of video per second of time (see the dashed line in Fig. 1.2(b)). The amount of video data buffered at the receiver depends on the channel throughput and the video source rate. If the channel throughput is higher than the video source rate, the amount of

received video data increases more rapidly than the amount of video data that is being played. Thus the amount of buffered video increases. Otherwise, if the channel throughput falls below the video source rate, the amount of data that can be played at the receiver decreases. Once there is no data to be played, the playback process is interrupted (see the solid line in Fig. 1.2(b)).

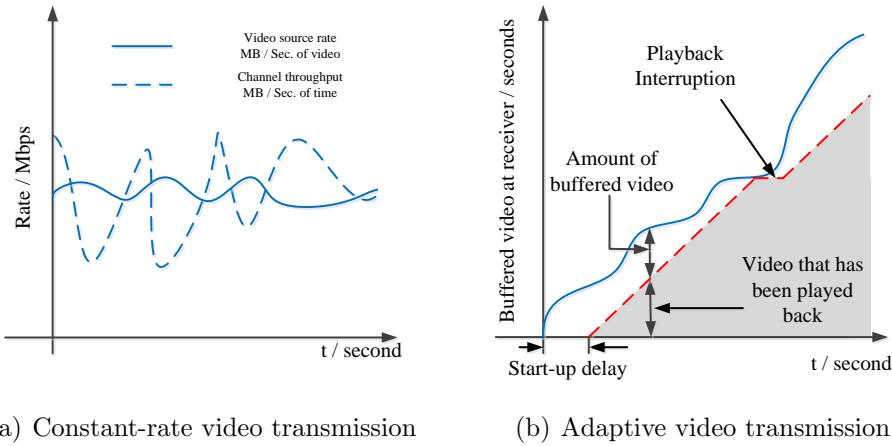


Figure 1.2: (a) The temporal variation of the video source rate and the channel throughput. The video source rate is measured in megabytes per second of video while the channel throughput is measured in megabytes per second of time; (b) The corresponding temporal variation of receiver buffer states. The solid line plots the amount of video buffered at the receiver, which is measured in seconds of video. The dashed line plots the amount of video that has been played back. Once all the buffered video has been played out, the playback process is interrupted.

To address the QoE degradation caused by playback interruption, adaptive video transmission techniques have been proposed to match the video source rate with the varying channel throughput. As illustrated in Fig. 1.3, if the video source rate is dynamically adjusted to match the channel throughput

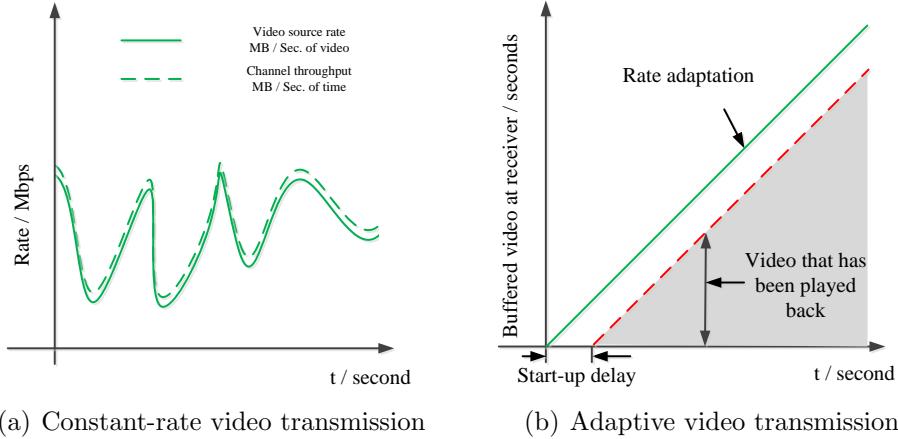
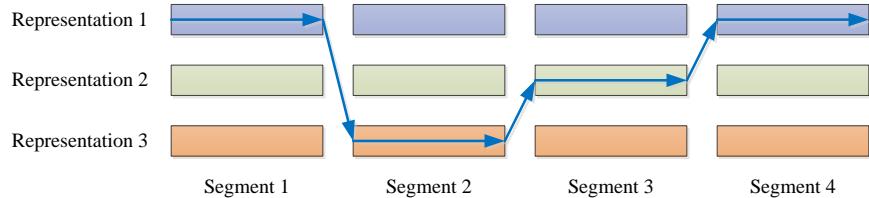


Figure 1.3: (a) The video source rate is adapted to match the variation of channel throughput. (b) The video data buffered at the receiver is increasing linearly. Playback interruption is avoided.

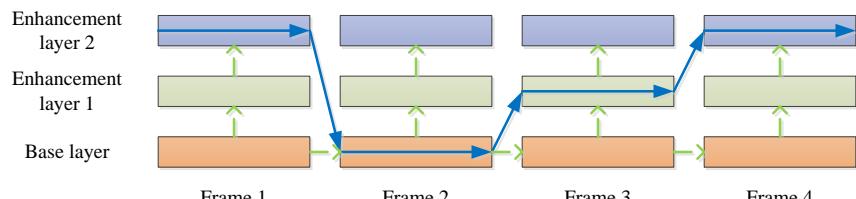
(see Fig. 1.3(a)), the video data that can be played at the receiver will not be drained and playback interruption can be avoided (see Fig. 1.3(b)).

To implement the idea of adaptive video transmission, two types of techniques are proposed. The first type of adaptive video transmission technique is based on the idea of bitrate switching (see Fig. 1.4(a)) [8, 53, 55, 62]. The video source in a bitrate switching system is compressed into multiple representations at different bitrates. Each representation is then partitioned into segments, each several seconds long. At any moment, the adaptation algorithm can dynamically select a segment from an appropriate representation to download such that the selected video source rate matches the channel capacity. The second type of adaptive video transmission technique employs scalable video coding (SVC) [42, 71]. As shown in Fig. 1.4(b), each video frame in a

scalable video stream contains a base layer and several enhancement layers. If the throughput is low, the transmitter can choose to transmit the base layer only, which provides a moderate, but acceptable, degree of visual quality at the receiver. If the channel throughput increases, the transmitter can transmit one, or more, enhancement layers to further improve the visual quality. The wireless transmitter can adapt data rates by selectively scheduling video data associated with various layers for transmission.



(a) Bitrate switching



(b) Scalable video coding

Figure 1.4: (a) Video source rate adaptation using bitrate switching. (b) Video source rate adaptation using scalable video coding. The dashed arrows indicate the encoding order between the quality layers and video frames.

I compare both categories of techniques in Table 1.1. The video source rate in rate-switching can only be adapted at the boundary of video segments.

A typical video segment is several seconds long. Thus, rate-switching is more suitable to combat throughput fluctuations in slower time scales. In wireless channels, the channel throughput varies on both slow and fast time scales. Therefore, rate-switching systems need to buffer several seconds of video before the video is played back. Although this reduces the risk of playback interruption caused by fast channel variations, a long start-up delay is introduced. On the contrary, using SVC, video source rates can be adapted for each frame. Therefore, SVC can be used to combat fast channel variations and requires shorter start-up delays.

To implement rate-switching, all representations of the video must be stored at the video source. For scalable video transmission, storing a single scalable video file is sufficient and thus requires less storage space. The scalable video streams, however, usually have worse rate-quality performances. That is, at a given video source rate, a multi-layer scalable video stream has worse quality than a single-layer representation encoded at the same rate. To achieve the same visual quality, SVC-based adaptive video transmission methods impose a higher load on the wireless channel. As storage space costs continue to decrease, rate-switching is a more cost-effective alternative to SVC-based methods if longer start-up delay is tolerable.

1.3 Challenges in Adaptive Video Transmission

Designing QoE-optimized adaptive video transmission systems is challenging. The foremost challenge is our limited understanding of the QoE of the rate-

Table 1.1: The advantages and disadvantages of the adaptive video transmission techniques based on rate-switching and scalable video coding.

	Rate-switching	Scalable video coding
Adaptation time-scale	per segment	per frame
Start-up delay	big	small
Storage space	big	small
Rate-quality performance	good	bad

adaptive videos. As the optimization objective in the design of adaptive wireless video transmission systems, QoE is measured via subjective studies, which are very labor intensive and time consuming. Thus, it is impractical to optimize the QoE of video transmission systems through extensive subjective studies. Although several algorithms have been proposed to predict QoE, they are all designed for constant-rate videos [20, 65, 72, 77, 85]. In an adaptive video transmission system, the varying video source rate causes video quality fluctuations. The behavioral responses of viewers to such quality fluctuation may affect their judgement of the QoE. Unfortunately, the mechanism with which the quality fluctuation affects QoE is not well understood. As will be shown in this dissertation, the existing video quality assessment algorithms fail to take into account the effect of quality fluctuations and thus cannot efficiently predict QoE.

Aside from optimizing the QoE, another target of adaptive video transmission is to avoid playback interruptions. In rate-switching systems where a longer start-up delay is tolerable, the system can buffer several seconds of

video before the playback starts. Thus, the risk of playback interruptions is negligible. But for SVC-based systems, which are designed to adapt to small time-scale channel variation, the tolerable start-up delay is usually small. The choice of video source rates not only depends on the channel throughput but also on the receiver buffer state. In particular, when the amount of buffered video is small, the adaptation algorithm should conservatively choose a lower source rate to reduce the risk of playback interruptions by buffering more video ahead of time. If a lot of video has been buffered, the adaptation algorithm can aggressively choose a large source rate to improve video quality. Therefore, the state of the receiver buffer affects the video source rate. The video source rate, in turn, affects the buffer state in the future. As illustrated in Fig. 1.5, optimizing the source rate adaptation policy is equivalent to solving a sequential decision problem, which is quite complicated.

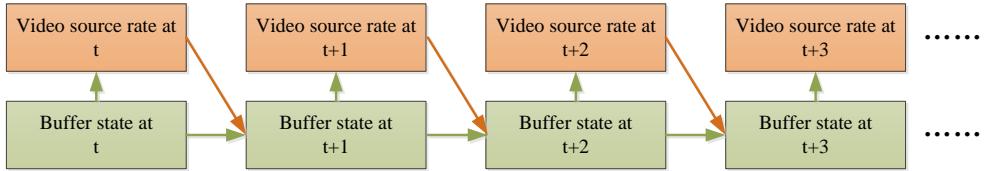


Figure 1.5: At any time t , the optimal video source rate depends on the buffer state. The buffer state, together with the selected video source rate, then determines the buffer state at $t + 1$.

In this dissertation, I first develop QoE models for rate-switching systems and design rate-switching algorithms based on the QoE models. Then, I propose an adaptive scalable video transmission algorithm that not only opti-

mizes the time-averaged video quality but also avoids playback interruptions by jointly considering the channel state and the receiver buffer state.

1.4 Thesis Statement

Accurate and tractable prediction models for the QoE of the videos transmitted over wireless networks facilitate the design of adaptive transmission algorithms that optimize the QoE while maintaining continuous video playback.

1.5 Contributions of This Dissertation

In this dissertation, I make three contributions that address three different yet related problems in wireless video transmission.

The first and second contributions both target rate-switching video systems but focus on two different aspects of the QoE. The first contribution aims at modeling the *Time-Varying Subjective Quality* (TVSQ), which is the up-to-the-moment subjective quality of a video as it is played. To develop the TVSQ model, I first conduct extensive subjective studies on rate-switching videos. From the subjective studies, two important observations are made. The first observation is that the video viewers exhibit a hysteresis effect [73], whereby the TVSQ at a particular moment depends on the viewing experience before the moment. The second observation is that, because of the nonlinearity in human behavioral responses to quality variations, the sensitivity of the

TVSQ to quality variations is not a constant. I propose a TVSQ model to capture both the hysteresis effect and the nonlinearities. Then, as an application of the model, I design a rate-switching algorithm that optimizes the time-averaged TVSQ of wireless video users.

The second contribution is aimed at optimizing another important aspect of QoE, the overall subjective quality (OQ). Different from TVSQ, the OQ of a video is the viewers' quality judgement after the whole video has been played. In the subjective studies, I find that the OQ of rate-switching videos is strongly correlated with the statistical distribution of the time-varying video qualities. Thus, I develop an OQ metric based on the cumulative distribution function of video qualities. Then, I propose a rate adaptation algorithm and an admission control strategy to maximize the number of video users in a wireless down-link who satisfy given constraints on OQ.

The third contribution of the dissertation is to optimize the time-averaged quality of the videos transmitted over SVC-based video systems while maintaining continuous playback. Using a Markov decision process (MDP) formulation, I incorporate the receiver buffer state into my adaptation algorithm design. In particular, I model the dynamics of the channel and buffer states using a Markov chain and reduce the problem of adaptive video transmission to a tractable Markov decision problem over a finite-state space. Based on the adaptive transmission policies derived from the MDP formulation, I devise an online algorithm that requires easily measurable knowledge of the channel dynamics only, and is thus viable in practice. Simulation results show that the

performance of the algorithm is close to a performance upper bound.

The three contributions of this dissertation are summarized as follows:

- I propose a new prediction model for TVSQ that can be used to design QoE-optimized rate-switching algorithms (partially reported in [22, 23]).
 1. I build a video database of long-duration rate-switching videos and conduct a subjective study to measure the TVSQs of each video.
 2. I develop a prediction model for TVSQ that captures the hysteresis effects and nonlinearities of human behavioral responses to video quality variations.
 3. I propose an online rate-switching algorithm that maximizes the sum of the time-averaged TVSQs of all users sharing a wireless down-link. The algorithm is simple to implement and has provable optimality.
- I propose an OQ metric and develop an adaptive video transmission algorithm for OQ optimization (partially reported in [24]).
 1. I propose a novel OQ metric based on the second-order empirical cumulative distribution function of the varying video quality. The metric has strong correlation with the OQs measured in the subjective study.
 2. I develop an online adaptive video transmission algorithm that maximizes the number of users satisfying given OQ constraints.
 3. I design an admission control strategy that blocks the users who can not satisfy the OQ constraints.

- I develop an adaptive transmission algorithm for stored scalable videos (partially presented in [22, 25]).
1. I propose an MDP formulation from which the optimal adaption policy for scalable video transmission can be derived.
 2. I propose an online adaptation algorithm based on the insights taken from the development of the MDP-based adaptation policies.
 3. I justify the near optimality of the online adaptation algorithm by comparing its simulated performance with a performance upper bound derived in this dissertation.

1.6 Notation

I use the following notation throughout this dissertation. Lower-case symbols such as a denote scalar variables and bold face symbols such as \mathbf{a} denote vectors. The notation \mathbf{a}^T is the transpose of \mathbf{a} . The function $\nabla_{\mathbf{a}} f(\mathbf{a}, \mathbf{b})$ denotes the gradient of the multivariate function $f(\mathbf{a}, \mathbf{b})$ with respect to variable \mathbf{a} . Random variables are denoted by uppercase letters such as A . Calligraphic symbols such as \mathcal{A} denote sets while $|\mathcal{A}|$ is the cardinality of \mathcal{A} . The set of positive integers is denoted by \mathbb{N}^+ . Let $\{x(t), t \in \mathbb{N}^+\}$ denote a discrete time series. The notation $(x)_{t_1:t_2}$ denotes the column vector $(x(t_1), x(t_1 + 1), \dots, x(t_2))^T$. Finally, I denote the zero-padded convolution of $(x)_{t_1:t_2}$ and $(y)_{t_1:t_2}$ by $(x)_{t_1:t_2} * (y)_{t_1:t_2}$.

1.7 Organization

The rest of the dissertation is organized as follows. Chapter 2 introduces the TVSQ prediction model and the corresponding TVSQ-optimized adaptive transmission algorithm. Chapter 3 introduces the rate control algorithm and the admission control algorithm for OQ optimization. Chapter 4 presents the adaptive transmission algorithms for stored scalable videos. Chapter 5 concludes the dissertation by drawing some conclusions and making suggestions for future work.

Chapter 2

Modeling and Optimizing the Time-Varying Subjective Quality for Rate-switching Systems

Although rate-switching provide flexibility in rate adaptation, designing rate-adaptation methods that could optimize end-users' Quality of Experience (QoE) is difficult since the relationship between the served bitrate and the users' viewing experience is not well understood. In particular, when the video bitrate is changed, the served video quality also varies. If the impact of the quality variation on QoE is not accurately predicted, the rate adaptation method cannot provide the optimal QoE for the users.

One important indicator of QoE is the *time-varying subjective quality* (TVSQ) of the viewed videos. This is *a time series or temporal record of viewers' judgments of the quality of the video as it is being played and viewed*. The TVSQ depends on many elements of the video including spatial distortions and temporal artifacts [72, 92]. Further more, human viewers exhibit a hysteresis [73] or recency [64] "after effect", whereby the TVSQ of a video at a particular moment depends on the viewing experience before the moment. Another important property of the TVSQ is its nonlinearity. In particular, the sensitivity of the TVSQ to quality variation is not a constant. This prop-

erty should be utilized in rate adaptation. For example, when the TVSQ of a viewer is insensitive to quality variations, the rate-controller could reserve some transmission resources (such as power) by reducing the bitrate without lowering the TVSQ. The reserved resources could then be used to increase the bitrate and thus improve the TVSQ.

This chapter study how to model and optimize the TVSQ of rate-switching video systems. I first build a database of rate-switching videos. Then I measure the TVSQ of the videos through a subjective study. Using the measured TVSQs, I develop a model for TVSQ prediction. Based on this prediction model, I devise a rate-switching algorithm that optimizes the time-averaged TVSQs of the users sharing a wireless down-link.

The reminder of this chapter is organized as follows: Section 2.1 introduces the proposed TVSQ prediction method. Section 2.2 explains the subjective study. The methodologies used for model parametrization is explained in Section 2.3. The proposed model is evaluated and analyzed in Section 2.4. Finally, the TVSQ-optimized adaptive video transmission algorithm is presented in Section 2.5.

2.1 Proposed TVSQ Prediction Method

In this section, I first overview the proposed TVSQ prediction method in Section 2.1.1. Then, in Section 2.1.2, I describe the core of the proposed model, a dynamic system model that captures the human behavioral responses to quality variations.

2.1.1 Overview of the TVSQ Prediction Method

I propose to predict TVSQ in two steps. The two steps capture the spatial-temporal characteristics of the video and the hysteresis effects in human behavioral responses, respectively.

First, I partition quality-varying videos into one second long video chunks and predict the short-time subjective quality (STSQ) of each chunk. Unlike TVSQ, which is a time series, the STSQ is *a scalar prediction of viewers' subjective judgment of a short video's overall perceptual quality*. A STSQ prediction model such as those in [65, 72, 77, 83, 85] operates by extracting perceptually relevant spatial and temporal features from short videos then uses these to form predictions of STSQ. Hence, STSQ contains useful, but incomplete evidence about TVSQ. I use the RRED index [77] to predict STSQs because of its excellent quality prediction performance and fast computational speed. Second, I input the computed STSQs to a dynamic system model that mimics the hysteresis effects and nonlinearity in human behavior. In the following, I discuss the design of the dynamic system model.

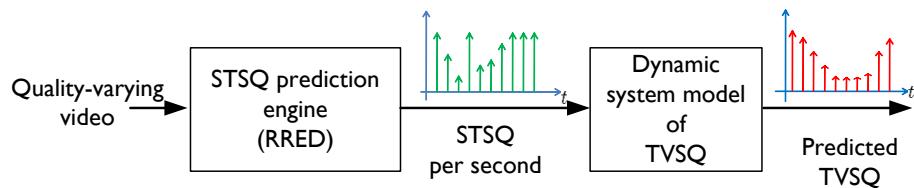


Figure 2.1: Proposed paradigm for TVSQ prediction. The STSQ of each second is predicted using the RRED index. An dynamic system model is then employed to predict the TVSQ of each second.

2.1.2 Model Structure

Due to the hysteresis effect of human behavioral responses to quality variations, the TVSQ at a moment depends on the viewing experience prior to the current moment. A dynamic system model can be used to capture the hysteresis effect using the “memory” of the system state. The simplest type of dynamic system is a linear filter. The human vision system, however, is non-linear in general. Although introducing intrinsic nonlinearities into the dynamic system model could help to capture those nonlinearities¹, the dynamic system would become too complicated to provide guidance on the design of TVSQ-optimized rate-adaptation algorithms. Therefore, I employ a Hammerstein-Wiener (HW) model [46], which captures the nonlinearity with extrinsic nonlinear functions. The model is illustrated in Fig. 2.2. The core of the HW model is a linear filter (see [46]) which is intended to capture the hysteresis. At the input and output of the HW model, two non-linear static functions are employed to model potential non-linearities in the human response. I call these two functions input nonlinearity and output nonlinearity, respectively.

Input nonlinearity: Denoting by $q^{st}(t)$ the STSQ at time t , I employ the following sigmoid function to model the input nonlinearity

$$x(t) = \beta_3 + \beta_4 \frac{1}{1 + \exp(-(\beta_1 q^{st}(t) + \beta_2))}, \quad (2.1)$$

¹We say a nonlinear system has intrinsic nonlinearity if its current system state is a nonlinear function of the previous system state and input. Otherwise, we say the system has extrinsic nonlinearity.

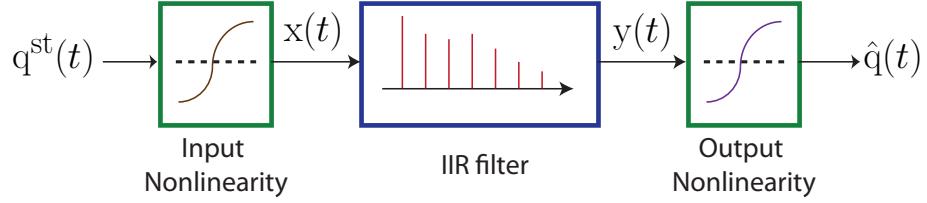


Figure 2.2: Proposed Hammerstein-Wiener model for TVSQ prediction. The core of the HW model is a linear filter. At the input and output of the HW model, two non-linear static functions are employed to model the non-linearities in the human response.

where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_4)^\top$ are model parameters and $x(t)$ is the input to the linear filter.

Linear filter: The linear filter has the following form:

$$\begin{aligned} y(t) &= \sum_{d=0}^r b_d x(t-d) + \sum_{d=1}^r f_d y(t-d) \\ &= \mathbf{b}^\top(x)_{t-r:t} + \mathbf{f}^\top(y)_{t-r:t-1}, \end{aligned} \tag{2.2}$$

where the parameter r is the model order and the coefficients $\mathbf{b} = (b_0, \dots, b_r)^\top$ and $\mathbf{f} = (f_1, \dots, f_r)^\top$ are model parameters to be determined. At any time t , the model output $y(t)$ depends not only on the previous r seconds of the input $x(t)$, but also on the previous r seconds of $y(t)$ itself. Thus this filter has an infinite impulse response (IIR). I employ this model rather than a finite impulse response (FIR) filter because the IIR filter can model the long-term impact of quality variations with a lower model order and thus using fewer parameters. To train a parameterized model, the size of the training data set increases exponentially with the number of the parameters [46]. Therefore, it is easier to train an IIR model. A drawback of the IIR filter (2.2) is its

dependency on its initial state. Specifically, to compute $(y)_{t>r}$, the initial r seconds of output $(y)_{1:r}$ need to be known. But $(y)_{1:r}$ is the TVSQ of the user, which is unavailable. Actually, I can show that this unknown initial condition only has negligible impact on the performance of the proposed model. A more detailed analysis is presented in Section 2.4.

Output nonlinearity: I set the output functions to be

$$\hat{q}(t) = \gamma_3 + \gamma_4 \frac{1}{1 + \exp(-(\gamma_1 y(t) + \gamma_2))}, \quad (2.3)$$

where $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_4)^T$ are model parameters and \hat{q} is the predicted TVSQ.

Let $\boldsymbol{\theta} = (\mathbf{b}^T, \mathbf{f}^T, \boldsymbol{\beta}^T, \boldsymbol{\gamma}^T)^T$ be the parameters of the proposed HW model, and let \hat{q} be regarded as a function both of time t and parameter $\boldsymbol{\theta}$. Thus, in the following, I explicitly rewrite \hat{q} as $\hat{q}(t, \boldsymbol{\theta})$. In the following, I explain the database and subjective study that are used to parameterize the model.

2.2 Subjective Study for Model Parameterizations

In this section, I first describe the construction of the database in Section 2.2.1. Then, I present the details of the subjective study in Section 2.2.2.

2.2.1 Video Database Construction

I seek to be able to predict the TVSQ using the STSQ. To better understand the relationship between TVSQ and STSQ, I broadly sample the space of STSQ sequences and measure the resulting TVSQ sequences through a subjective study. Using the following 5 steps, I construct 18 quality-varying videos such

Table 2.1: A brief description of the video clips in the rate-switching video database.

Name	Abbreviation	Description
Fountain	ft	Still camera, shows a fountain.
Turtles	tu	Still camera, a girl is feeding turtles.
Stick	st	Still camera, a man is waving a stick.
Bulldozer	bu	Camera span, a man is driving a bulldozer.
Singer&girl	sg	Camera zoom, a man is singing to a girl.
Volleyball	vo	Still camera, shows a volleyball game.
Dogs	do	Camera span, two dogs play near a pool.
Singer	si	Camera zoom, a singer is singing a song.

that the STSQs vary randomly across time.

1. I select 8 high quality, uncompressed video clips with different content.

These clips have spatial resolution of 720p (1280×720) and frame rate of 30fps. A short description of these clips is provided in Table 2.1. The content is chosen to represent a broad spectrum of spatial and temporal complexity (see sample frames in Fig. 2.3).

2. Using the video clips selected in the first step, I construct 3 reference videos, which are used to generate quality-varying videos in the subjective study. Each reference video is 300 seconds long and is constructed by concatenating 5 or 6 different clips (see Fig. 2.4). I construct the reference video in this way because long videos with monotonous content can be boring to subjects. This could adversely impact the accuracy of the TVSQ measured in the subjective study.



(a) ft



(b) tu



(c) st



(d) bu



(e) sg



(f) vo



(g) do



(h) si

Figure 2.3: Sample frames of the video clips involved in the subjective study. The abbreviations of the names of the videos can be found in Table. 2.1.

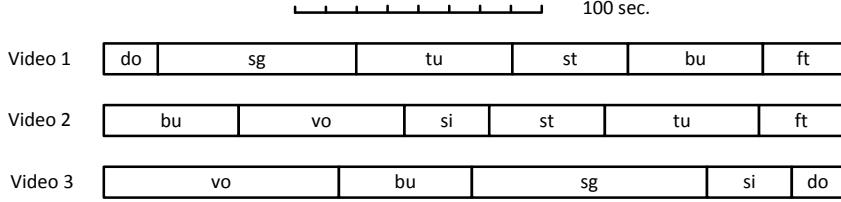


Figure 2.4: The construction of the reference videos. The abbreviation of the names of the clips can be found in Table. 2.1.

3. For each reference video, I generate 28 distorted versions. Specifically, I encode every reference video sequence into 28 constant bitrate streams using the H.264 encoder in [12] and then decode them. To achieve a wide range of video quality exemplars, the encoding bitrates are chosen from hundreds of Kbps to several Mbps.
4. I partition every distorted version into 1 second long video chunks and predict their STSQs with the computationally efficient and perceptually accurate RRED index [77]. Let the RRED index of the t^{th} chunk in the ℓ^{th} distorted version of the k^{th} reference video be denoted by $q_{\ell,k}^{\text{rred}}(t)$, where $t \in \{1, \dots, 300\}$, $\ell \in \{1, \dots, 28\}$, and $k \in \{1, 2, 3\}$. Then the Difference Mean Opinion Score (DMOS) of the STSQ for each chunk is predicted via logistic regression:

$$q_{\ell,k}^{\text{dmos}}(t) = 16.4769 + 9.7111 \log \left(1 + \frac{q_{\ell,k}^{\text{rred}}(t)}{0.6444} \right). \quad (2.4)$$

The regression model in (4.13) is obtained by fitting a logistic mapping from the RRED index to the DMOSs on the LIVE Video Quality Assessment Database [40]. Here, $q_{\ell,k}^{\text{dmos}}(t)$ ranges from 0 to 100 where lower values

indicate better STSQ. To represent STSQ more naturally, so that higher numbers indicate better STSQ, I employ a Reversed DMOS (RDMOS) as follows:

$$q_{\ell,k}^{\text{rdmos}}(t) = 100 - q_{\ell,k}^{\text{dmos}}(t). \quad (2.5)$$

Broadly speaking, a RDMOS of less than 30 on the LIVE databases [40] indicates bad quality, while scores higher than 70 indicate excellent quality. As an example, Fig. 2.5(a) plots $q_{\ell,k}^{\text{rdmos}}(t)$ for all of the distorted versions of the first reference video. Clearly, their STSQ covers a wide range of RDMOSs.

5. Finally, for each reference video, I construct 6 quality-varying videos by concatenating the video chunks selected from different distorted versions. For the k^{th} reference video, I first design 6 target STSQ sequences $(q_{j,k}^{\text{tgt}})_{1:300}$, where $j = 1, \dots, 6$ to simulate the typical quality variation patterns in rate-switching videos. Then, I construct 6 quality-varying videos such that their STSQs approximate the target sequences. Specifically, I construct the t^{th} chunk of the j^{th} quality-varying video by copying the t^{th} chunk in the $\ell_{t,j,k}^*$ -th distorted version, where

$$\ell_{t,j,k}^* = \arg \min_{\ell} |q_{j,k}^{\text{tgt}}(t) - q_{\ell,k}^{\text{rdmos}}(t)|. \quad (2.6)$$

Denoting the STSQ of the t^{th} chunk in the obtained video by $q_{j,k}^{\text{st}}(t)$, I have

$$q_{j,k}^{\text{st}}(t) = q_{\ell_{t,j,k}^*}^{\text{rdmos}}(t). \quad (2.7)$$

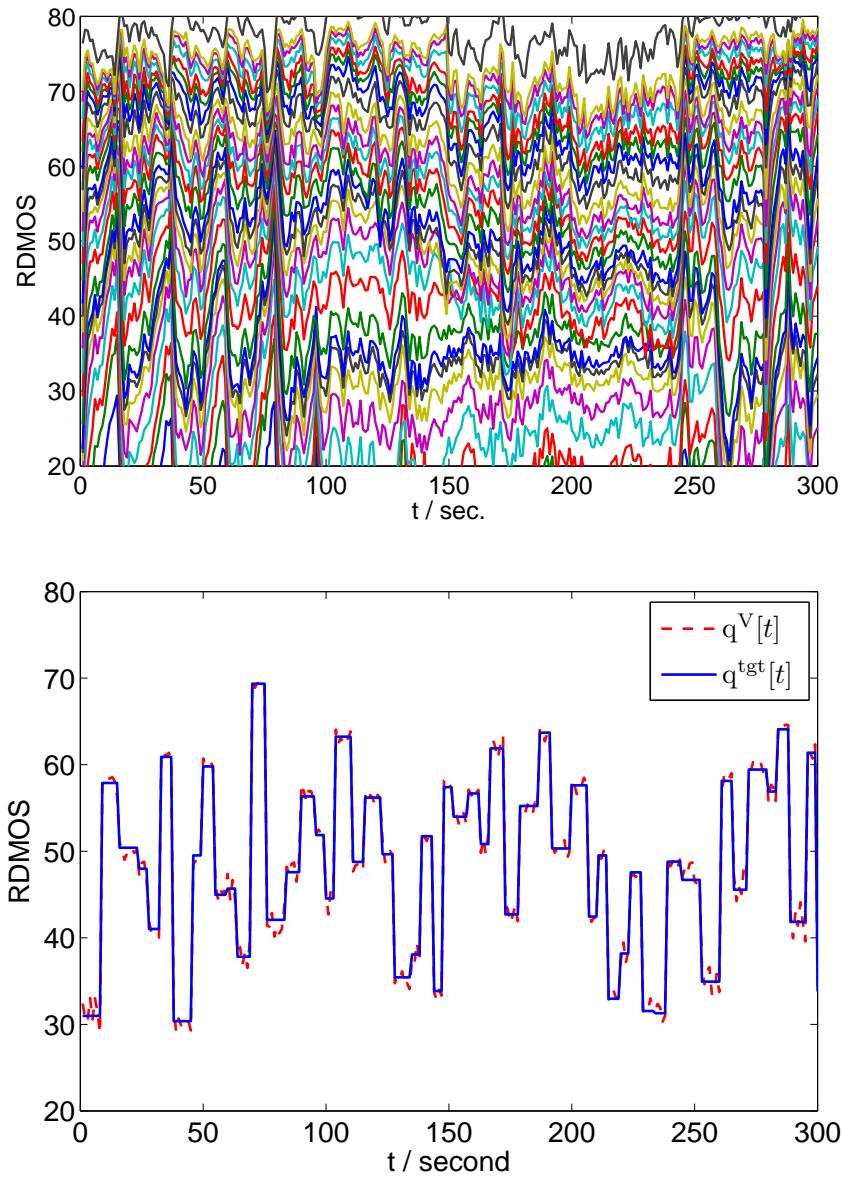


Figure 2.5: (a) The STSQ of each compressed version of the reference video is shown in different colors. (b) A example of the designed target video quality $q^{tgt}(t)$ and the actual video quality $q^V(t)$ of the video sequence used in my database.

As can be seen in Fig. 2.5, since the RDMOS scale is finely partitioned by the RDMOSs of the compressed versions, the error between the obtained STSQ $q_{j,k}^{\text{st}}(t)$ and the target STSQ $q_{j,k}^{\text{tgt}}(t)$ is small. In all, I construct $3 \times 6 = 18$ quality-varying videos.

With this procedure, we are able to manipulate the pattern of quality variations in the test video sequences using the target video quality sequence $(q_{j,k}^{\text{tgt}})$. Next, I describe the design of $(q_{j,k}^{\text{tgt}})$.

To obtain a good TVSQ prediction model for rate-switching videos, I design the target video quality $(q_{j,k}^{\text{tgt}})_{1:300}$ such that the generated quality-varying videos can roughly simulate the STSQs of rate-switching videos. In rate-switching protocols such as those described in [8, 53, 55, 62], videos are encoded into multiple representations at different bitrates. Each representation is then partitioned into segments, each several seconds long. The client dynamically selects a segment of a representation to download. Therefore, I model $(q_{j,k}^{\text{tgt}})_{1:300}$ as a piece-wise constant time-series. Specifically, I generate $(q_{j,k}^{\text{tgt}})_{1:300}$ using two independent random processes. The first random process $\{D(n) : n = 1, 2, \dots\}$ simulates the length of the video segments. The second random process $\{Q(n) : n = 1, 2, \dots\}$ simulates the STSQs of segments. The sequence $(q_{j,k}^{\text{tgt}})_{1:300}$ is constructed as a series of constant-value segments where the durations of the segments are given by $D(n)$ and the RDMOSs of the segments are given by $Q(n)$ (see Fig. 2.6).

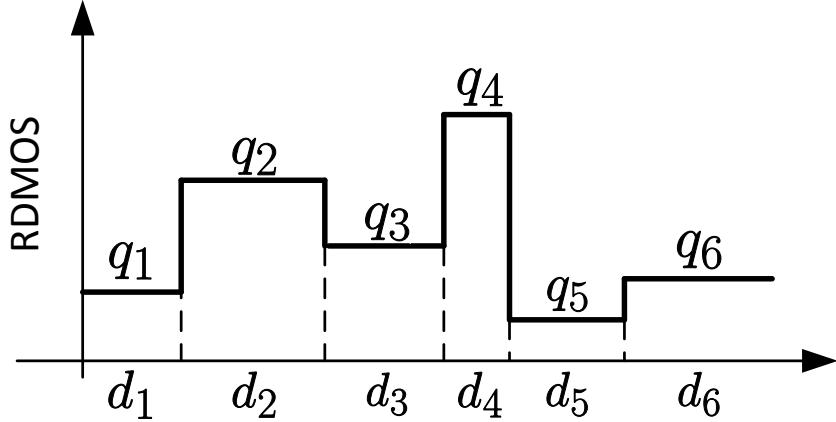


Figure 2.6: The design of the target STSQs. The durations of each segment d_1, d_2, \dots are realizations of $D(1), D(2), \dots$. The STSQ levels q_1, q_2, \dots are realizations of $Q(1), Q(2), \dots$.

In rate-switching systems, the duration of segments can be flexibly chosen by the service provider. Shorter durations allow more flexibility for rate adaptation when the channel condition is varying rapidly. I focus on applications where the lengths of the segments are less than 10 seconds. In a subjective experiment, there is always a delay or latency between a change in STSQ and a subject's response. During the experimental design, I found that if the video quality varied too quickly, subjects could not reliably track their judgments of quality to the viewed videos. Specifically, when the video quality changes, a subject may take 1-2 seconds to adjust his/her opinion on the TVSQ. If the quality is adapted frequently, the quality variations that occur during this adjustment process can annoy the subject and thus reduce the accuracy of the measured TVSQs. Thus, I restrict the length of each segment

to be at least 4 seconds, which is comfortably longer than the subjects' latency and short enough to model quality variations in adaptive video streaming. In sum, I let the random process $\{D(n) : n = 1, 2, \dots\}$ take values from the set $\{4, 5, 6, 7, 8, 9, 10\}$.

The distribution of STSQs of a rate-switching video depends on many factors including the encoding bitrates, the rate-quality characteristics, the segmentation of each representation, the channel dynamics, and the rate adaptation strategy of the client. I try to sample uniformly from among all possible patterns of STSQ variations. Therefore, I design the random processes $D(n)$ and $Q(n)$ as i.i.d. processes, which tend to traverse all possible patterns of quality variations. Also, I design the distributions of $D(n)$ and $Q(n)$ to “uniformly” sample all possible segment lengths and STSQ levels, respectively. To this end, I let $D(n)$ take values in the set $\{4, 5, 6, 7, 8, 9, 10\}$ with equal probability. Similarly, I design the distribution of $Q(n)$ such that the sample values of $Q(n)$ would be distributed as if we were uniformly sampling the videos in the LIVE database, because that set of videos is carefully chosen to represent a wide range of perceptually separated STSQ [40]. The RDMOSs of videos in the LIVE database are distributed as approximately obeying a normal distribution $\mathcal{N}(50, 10^2)$ [40]. Therefore, I let the distribution of $Q(n)$ be $\mathcal{N}(50, 10^2)$. Since almost all of the RDMOSs in the LIVE database fall within the range $[30, 70]$, I truncate $Q(n)$ to $[30, 70]$.

2.2.2 Subjective Experiments

I conduct a subjective study to measure the TVSQs of the quality-varying videos in my database. The study is completed at the LIVE subjective testing lab at The University of Texas at Austin. The videos in my database are grouped into 3 sessions. Each session includes one of the three reference videos and the 6 quality-varying videos generated from the reference video. The videos in each session are each viewed and scored by 25 subjects. One of the quality-varying videos is used as a training sequence. The other six videos, including 5 quality-varying videos and the reference video, are used for subjective study.

I develop a user interface for the subjective study using the Matlab XGL toolbox [17]. The user interface ran on a Windows PC with an Intel Xeon 2.93GHz CPU and a 24GB RAM. The XGL toolbox interfaces with ATI Radeon X300 graphics card on the PC to precisely display video frames without latencies or frame drops, by loading each video into memory before display. Video sequences are displayed to the viewers on a 58 inch Panasonic HDTV plasma monitor at a viewing distance of about 4 times the picture height. During the play of each video, a continuous scale sliding bar is displayed near the bottom of the screen. Similar to the ITU-R ACR scale [37], the sliding bar is marked with five labels: “Bad”, “Poor”, “Fair”, “Good”, and “Excellent”, equally spaced from left to right. The subject could continuously move the bar via a mouse to express his/her judgment of the video quality as each video is played. The position of the bar is sampled and recorded automatically in

real time as each frame is displayed (30 fps). No mouse clicking is required in the study. Fig. 2.7 shows the subjective study interface including a frame of a displayed video.

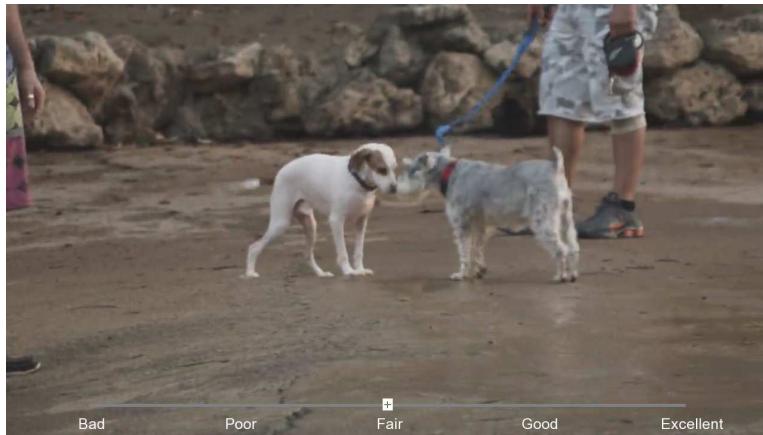


Figure 2.7: The user interface used in the subjective study. A continuous scale sliding bar is displayed near the bottom of the screen. The subject could continuously move the bar via a mouse to express his/her judgment of the video quality as each video is played.

During the training period, each subject first reads instructions describing the operation of the user interface (see Appendix A), then practices on the training sequence. The subject then starts rating the test videos (reference video and five quality-varying videos) shown in random order. The subjects are unaware of the presence of the reference video.

Denote the average score assigned by the i^{th} subject to the frames of the t^{th} chunk of the j^{th} quality-varying video in the k^{th} session by $c_{i,j,k}(t)$. Let the score assigned to the reference video be denoted by $c_{i,k}^{\text{ref}}(t)$. I offset the

impact of video content on the TVSQs of the test videos using

$$c_{i,j,k}^{\text{offset}}(t) = 100 - (c_{i,k}^{\text{ref}}(t) - c_{i,j,k}(t)). \quad (2.8)$$

I denote the length of the test videos by T and the number of quality-varying videos in each session by J. In my experiment, we have T = 300 and J = 5. Note that the subjects deliver their quality judgments in real-time as the test video is being displayed. To avoid distracting the subjects from viewing the video, we did not require them to use the full scale of the sliding bar. Moreover, such an instruction may tend to bias the recorded judgments from their natural response. Thus, each subject was allowed to freely deploy the sliding bar when expressing their judgments of TVSQ. To align the behavior of different subjects, paralleling to prior work such as [45, 50, 69, 74, 76], we normalize ($c_{i,j,k}^{\text{offset}}$) by computing the Z-scores [82] as follows:

$$\begin{aligned} m_{i,k} &= \frac{1}{J} \frac{1}{T} \sum_{j=1}^J \sum_{t=1}^T c_{i,j,k}^{\text{offset}}(t); \\ \sigma_{i,k}^2 &= \frac{1}{JT - 1} \sum_{j=1}^J \sum_{t=1}^T (c_{i,j,k}^{\text{offset}}(t) - m_{i,k})^2; \\ z_{i,j,k}(t) &= \frac{c_{i,j,k}^{\text{offset}}(t) - m_{i,k}}{\sigma_{i,k}}. \end{aligned} \quad (2.9)$$

In (2.9), the values of $m_{i,k}$ and $\sigma_{i,k}$ are respectively the mean and the variance of the scores assigned by the i^{th} subject in the k^{th} session. The value of $z_{i,j,k}(t)$ is the normalized score. Let I denote the number of subjects. We have I = 25. Then for the t^{th} second of the j^{th} test video, compute the average and standard

deviation of the Z-scores assigned by the subjects

$$\begin{aligned}\mu_{j,k}(t) &= \frac{1}{I} \sum_{i=1}^I z_{i,j,k}(t); \\ \eta_{j,k}^2(t) &= \frac{1}{I-1} \sum_{i=1}^I (z_{i,j,k}(t) - \mu_{j,k}(t))^2.\end{aligned}\tag{2.10}$$

If $z_{i,j,k}(t) > \mu_{j,k}(t) + 2\eta_{j,k}(t)$ or $z_{i,j,k}(t) < \mu_{j,k}(t) - 2\eta_{j,k}(t)$, I mark $z_{i,j,k}(t)$ as an outlier because the Z-score given by subject i deviates far from the Z-scores given by the other subjects. The outliers are excluded and the Z-scores are recomputed using (2.9). Let $\mathcal{O}_{j,k,t}$ denote the set of subjects who assigned outlier Z-scores to the t^{th} chunk of the j^{th} video in the k^{th} session. The averaged Z-score of the TVSQ for the t^{th} chunk is then

$$\bar{z}_{j,k}(t) = \frac{1}{I - |\mathcal{O}_{j,k,t}|} \sum_{i \notin \mathcal{O}_{j,k,t}} z_{i,j,k}(t).\tag{2.11}$$

The 95% confidence interval of the average Z-scores is

$$\bar{z}_{j,k}(t) \pm 1.96\eta_{j,k}(t)/\sqrt{I - |\mathcal{O}_{j,k,t}|}.$$

I found that the values of the averaged Z-scores all lie in the range $[-4, 4]$. Therefore, I map $\bar{z}_{j,k}(t)$ to the range $[-100, 100]$ using the following formula:

$$q_{j,k}^{\text{tv}}(t) = \frac{\bar{z}_{j,k}(t) + 4}{8} \times 100.\tag{2.12}$$

Correspondingly, the 95% confidence interval of TVSQ is $q_{j,k}^{\text{tv}}(t) \pm \epsilon_{j,k}(t)$, where

$$\epsilon_{j,k}(t) = \frac{1.96\eta_{j,k}(t)/\sqrt{I - |\mathcal{O}_{j,k,t}|} + 4}{8} \times 100.\tag{2.13}$$

In all, I measured the TVSQ for $N = 3 \times 5 = 15$ quality-varying videos. In the following, I replace the subscript (j, k) with a subscript $1 \leq n \leq N$ to index the quality-varying videos and denote by $q_n^{\text{tv}}(t)$ and $\epsilon_n(t)$ the measured TVSQ and the confidence interval of the n^{th} video, respectively. Similarly, I denote the STSQ of the n^{th} video predicted by the RRED index [77] as $q_n^{\text{st}}(t)$.

2.3 Model Parametrization

To find the optimal HW model for TVSQ prediction, I need to determine two things: the model order r and the model parameter $\boldsymbol{\theta}$. In the following, I first show how to optimize the model parameter $\boldsymbol{\theta}$ for given model orders in Section 2.3.1. Then, in Section 2.3.2, I introduce the method for model order estimation.

2.3.1 Model Parameter Training

In this section, I discuss how to optimize the model parameter $\boldsymbol{\theta}$ such that the error between the measured TVSQ and the predicted TVSQ can be minimized. In system identification and machine learning, the mean square error (MSE) is the most widely used error metric. Denoting the predicted TVSQ of the n^{th} video by $\hat{q}_n(t, \boldsymbol{\theta})$, the MSE is defined as

$$\frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (\hat{q}_n(t, \boldsymbol{\theta}) - q_n^{\text{tv}}(t))^2.$$

The MSE always assigns a higher penalty to a larger estimation error. For the purposes of tracking TVSQ, however, once the estimated TVSQ deviates far

from the measured TVSQ, the model fails. There is no need to penalize a large error more than another smaller, yet still large error. Furthermore, since the $q_n^{tv}(t)$ is just the average subjective quality judgment, the confidence interval of TVSQ $\epsilon_n(t)$ (see the definition in (2.13)) should also be embodied in the error metric to account for the magnitude of the estimation error. I choose to use the outage rate, also used in [41], as the error metric. Specifically, the outage rate of a TVSQ model is defined as the frequency that the estimated TVSQ deviates by at least twice the confidence interval of the measured TVSQ. More explicitly, outage rate can be written as

$$E(\boldsymbol{\theta}) = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \mathbb{1}(|\hat{q}_n(t, \boldsymbol{\theta}) - q_n^{tv}(t)| > 2\epsilon_n(t)), \quad (2.14)$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Gradient-descent parameter search algorithms are commonly used for model parametrization. In my case, however, the gradient of the indicator function $\mathbb{1}(\cdot)$ in (2.14) is zero almost everywhere and thus the gradient algorithm cannot be applied directly. To address this difficulty, I approximate the indicator function $\mathbb{1}(|x| > 2\epsilon)$ by a penalty function

$$U_\nu(x, \epsilon) = h(x, \nu, -2\epsilon) + (1 - h(x, \nu, 2\epsilon)), \quad (2.15)$$

where $h(x, a, b) = 1 / (1 + \exp(-a(x + b)))$ is a logistic function. In Fig. 2.8, I plot $U_\nu(x, \epsilon)$ with different configurations of the parameter ν . It can be seen that, as $\nu \rightarrow \infty$, $U_\nu(x, \epsilon)$ converges to $\mathbb{1}(|x| > 2\epsilon)$. The outage rate $E(\boldsymbol{\theta})$ can

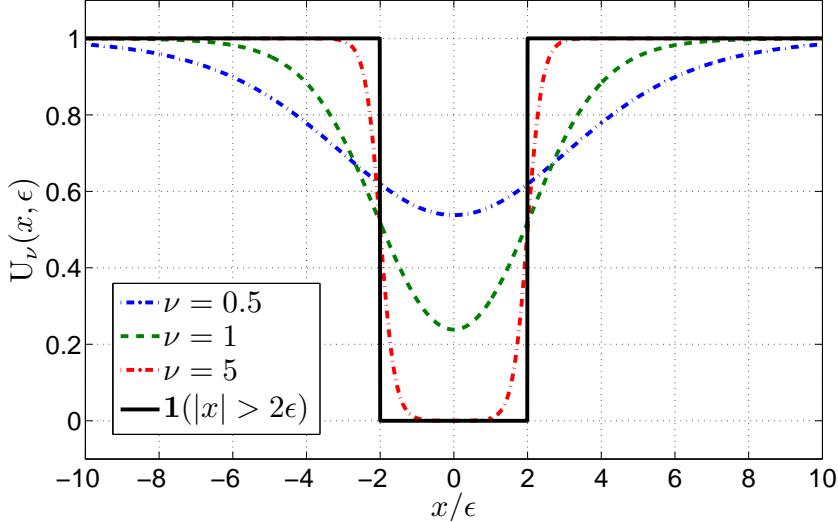


Figure 2.8: The function $U_\nu(x, \epsilon)$ with different configurations of the parameter ν . The three plots shows $U_{0.5}(x, \epsilon)$, $U_1(x, \epsilon)$, and $U_5(x, \epsilon)$, respectively.

thus be approximated by $E(\boldsymbol{\theta}) = \lim_{\nu \rightarrow \infty} E_\nu^{\text{apx}}(\boldsymbol{\theta})$, where

$$E_\nu^{\text{apx}}(\boldsymbol{\theta}) = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T U_\nu(\hat{q}_n(t, \boldsymbol{\theta}) - q_n^{\text{tv}}(t), \epsilon(t)). \quad (2.16)$$

The iterative algorithm used for model parameter identification is described in Algorithm 1. In the i^{th} iteration, a gradient-descent search algorithm is applied to minimize $E_\nu^{\text{apx}}(\boldsymbol{\theta})$. The obtained parameter $\boldsymbol{\theta}^i$ is then used as the starting point for the gradient-descent search in the $(i+1)^{\text{th}}$ iteration. At the end of each iteration, I increase the parameter ν by $\nu := 1.2\nu^2$. Using this algorithm, I gradually modify the penalty function $U_\nu(x, \epsilon)$ to $\mathbb{1}(|x| > 2\epsilon)$ such that the estimated TVSQ is forced into the confidence interval of the measured

²The choice of the multiplicative factor 1.2 is to balance the efficiency and accuracy of the algorithm. Any number less than 1.2 gives rise to similar performance.

Algorithm 1 Parameter optimization algorithm

Require: $q_n^{\text{st}}(t)$, $q_n^{\text{tv}}(t)$, $\epsilon_n(t)$, $i = 1$, and $\nu = 0.8$

```
1: while  $\nu < 20$  do
2:    $\boldsymbol{\theta}^i = \arg \min_{\boldsymbol{\theta}} E_{\nu}^{\text{apx}}(\boldsymbol{\theta})$  via gradient-descent search starting from  $\boldsymbol{\theta}^{i-1}$ .
3:    $i := i + 1$ 
4:    $\nu := 1.2\nu$ 
5: end while
```

TVSQ. Note that when $\nu \geq 20$, $U_{\nu}(x, \epsilon)$ is very close to $\mathbb{1}(|x| > 2\epsilon)$. Hence, I terminate the iteration when $\nu \geq 20$ ³.

The gradient-descent mechanism in Algorithm 1 is described by Algorithm 2. The algorithm contains two loops. In the outer loop, $\boldsymbol{\theta}$ is moved along the direction of negative gradient $-\nabla_{\boldsymbol{\theta}} E_{\nu}^{\text{apx}}(\boldsymbol{\theta})$ with a step-size ω . I terminate the loop when the decrement of the cost function between consecutive loops is less than a small threshold δ . On my database, I find that setting $\delta = 10^{-5}$ is sufficient. The inner loop of Algorithm 2 is a standard backtracking line search algorithm (see [15]), which determines an appropriate step-size ω . To calculate the gradient $\nabla_{\boldsymbol{\theta}} E_{\nu}^{\text{apx}}(\boldsymbol{\theta})$, we have

$$\begin{aligned} & \nabla_{\boldsymbol{\theta}} E_{\nu}^{\text{apx}}(\boldsymbol{\theta}) \\ &= \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \left[\frac{dU_{\nu}(x, \epsilon_n(t))}{dx} \Bigg|_{x=\hat{q}_n(t, \boldsymbol{\theta}) - q_n^{\text{tv}}(t)} \right] \nabla_{\boldsymbol{\theta}} \hat{q}_n(t, \boldsymbol{\theta}). \end{aligned} \tag{2.17}$$

In (2.17), $\frac{dU_{\nu}(x, \epsilon)}{dx}$ can be directly derived from (2.15). The calculation of $\nabla_{\boldsymbol{\theta}} \hat{q}_n(t, \boldsymbol{\theta})$ is not straightforward since the dynamic model has a recurrent

³Since $E(\boldsymbol{\theta})$ is not a convex function of $\boldsymbol{\theta}$, gradient-descent can only guarantee local optimality.

structure. Specifically, the input-output relationship of the HW model can be written as:

$$\hat{q}_n(t, \boldsymbol{\theta}) = g\left(\boldsymbol{\theta}, (\hat{q}_n)_{t-r:t-1}, (q_n^{\text{st}})_{t-r:t}\right), \quad (2.18)$$

where the function $g(\cdot)$ is the combination of (2.2), (2.1), and (2.3). The model output $\hat{q}_n(t, \boldsymbol{\theta})$ depends not only on $\boldsymbol{\theta}$ but also on previous system outputs $(\hat{q}_n)_{t-1:t-r}$, which depend on $\boldsymbol{\theta}$ as well ⁴. Denoting by θ_i the i^{th} component of $\boldsymbol{\theta}$, differentiating both side of (2.18), we have

$$\frac{\partial \hat{q}_n(t, \boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial g}{\partial \theta_i} + \sum_{d=1}^r \frac{\partial g}{\partial \hat{q}_n(t-d, \boldsymbol{\theta})} \frac{\partial \hat{q}_n(t-d, \boldsymbol{\theta})}{\partial \theta_i}. \quad (2.19)$$

Because of the structure of (2.19), computing $\frac{\partial \hat{q}_n(t, \boldsymbol{\theta})}{\partial \theta_i}$ is equivalent to filtering $\frac{\partial g}{\partial \theta_i}$ by a filter with a transfer function

$$H(z) = \frac{1}{1 - \sum_{d=1}^r \frac{\partial g(\cdot)}{\partial \hat{q}_n(t-d, \boldsymbol{\theta})} z^{-d}}, \quad (2.20)$$

If $\boldsymbol{\theta}$ is not appropriately chosen, the filter $H(z)$ can be unstable. The computed gradient $\frac{\partial \hat{q}_n(t, \boldsymbol{\theta})}{\partial \theta_i}$ could diverge as t increases. I prove in Appendix B that, if the root radius⁵ $\rho(\mathbf{f})$ of the polynomial $z^r - \sum_{d=1}^r f_d z^{r-d}$ is less than 1, the filter $H(z)$ is stable. Therefore, in the line search step Algorithm 2, I always choose a small enough step-size ω such that the condition $\rho(\mathbf{f}) < 1$ is satisfied (see line 3-5 in Algorithm). For further details about the calculation of $\frac{\partial \hat{q}_n(t, \boldsymbol{\theta})}{\partial \theta_i}$, see Appendix B.

⁴Here (\hat{q}_n) also depends on (q^{st}) . But in parameter training, (q^{st}) is treated as a known constant.

⁵The root radius of a polynomial is defined as the maximum radius of its complex roots.

2.3.2 Model Order Selection

Using Algorithm 1, I find the optimal parameter $\boldsymbol{\theta}$ for a given model order r .

In this section, I discuss how to select the model order. First, I estimate a possible range of model orders by inspecting the correlation between the input and output of the HW model, i.e., $(q^{st})_{1:T}$ and $(q^{tv})_{1:T}$. Then, I determine the model order in the estimated range using the principle of Minimum Description Length, which I describe in more detail below.

The TVSQ at any time depends on the previous viewing experience.

In the proposed TVSQ model (2.18), I have employed

$$\phi_r(t) = \left((q^{st})_{t-r:t}^\top, (q^{tv})_{t-r:t-1}^\top \right)^\top$$

as the model input to capture the previous viewing experience. Thus, identifying the model order r is essentially estimating how much previous viewing experience is relevant to the current TVSQ. In [34], the Lipschitz quotient is

Algorithm 2 Gradient-descent algorithm

Require: $q^{st}(t)$, $q^{tv}(t)$, $\epsilon(t)$, ν , and $j = 1$

- 1: **while** $E_\nu^{\text{apx}}(\boldsymbol{\theta}^{j-1}) - E_\nu^{\text{apx}}(\boldsymbol{\theta}^j) \geq 10^{-5}$ **do**
 - 2: $\Delta\boldsymbol{\theta} := -\nabla_{\boldsymbol{\theta}} E_\nu^{\text{apx}}(\boldsymbol{\theta}^j)$
 - 3: **while** $E_\nu^{\text{apx}}(\boldsymbol{\theta}^j + \omega\Delta\boldsymbol{\theta}) > E_\nu^{\text{apx}}(\boldsymbol{\theta}^j) - 0.1\omega\|\Delta\boldsymbol{\theta}\|_2^2$ or $\rho(\mathbf{f}) \geq 1$ **do**
 - 4: $\omega := 0.7\omega$
 - 5: **end while**
 - 6: $\boldsymbol{\theta}^{j+1} := \boldsymbol{\theta}^j + \omega\Delta\boldsymbol{\theta}$
 - 7: $j := j + 1$
 - 8: **end while**
-

proposed to quantify the relevance of ϕ_r by

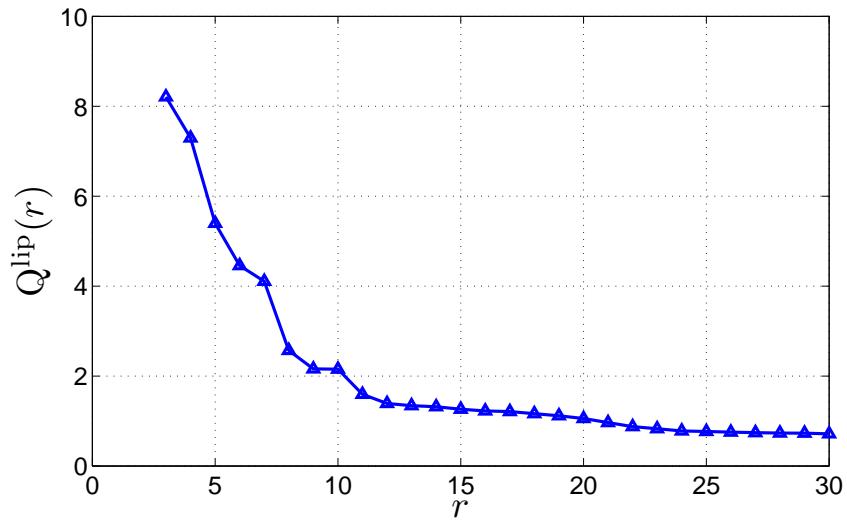
$$Q^{\text{lip}}(r) = \max_{1 \leq t_1 < t_2 \leq T} \left(\frac{|q^{\text{tv}}(t_1) - q^{\text{tv}}(t_2)|}{\|\phi_r(t_1) - \phi_r(t_2)\|_2} \right). \quad (2.21)$$

A large $Q^{\text{lip}}(r)$ implies that a small change in ϕ_r could cause a significant change in q^{tv} and thus ϕ_r is relevant to TVSQ. Conversely, if $Q^{\text{lip}}(r)$ is small, the model order r may be larger than necessary. Using $Q^{\text{lip}}(r)$, we can estimate the necessary model order. In Fig. 2.9(a), I plot the Lipschitz quotients for different values of r . It can be seen that, as the model order increases, the corresponding Lipschitz quotient decreases significantly when r is less than 10. This means the viewing experience over the previous 10 seconds is closely related to the TVSQ. Therefore, the model order r should be at least 10.

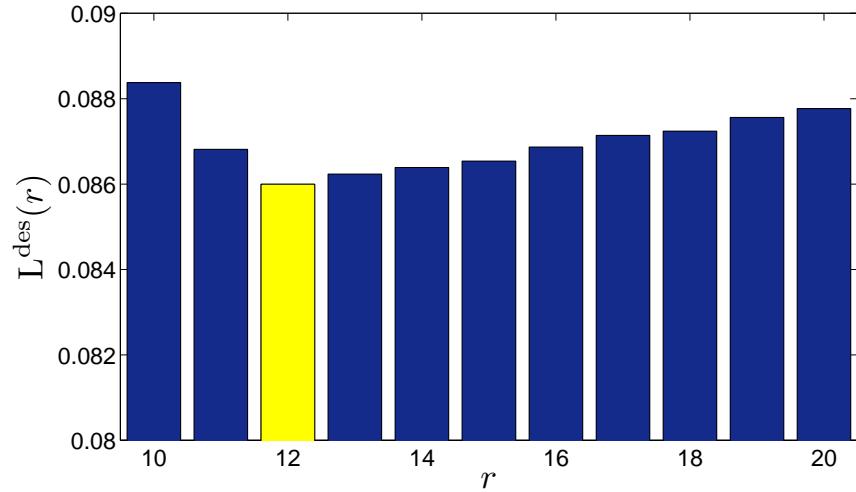
According to the parameterizations of the HW model in (2.2), (2.1), and (2.3), models of lower order are special cases of the model of higher order. Therefore, in principle, the higher the order, the better performance can be achieved by the model. A large model order, however, may result in over-fitting the model to the training dataset. To select an appropriate order for the HW model, I employed the Minimum Description Length (MDL) criterion, which is widely used in the realm of system identification [11][46]. The description length of an r -order model is defined in [46] as

$$L^{\text{des}}(r) = E(\theta_r^*) \left(1 + (2r + 1) \frac{\log(N(T - r))}{N(T - r)} \right), \quad (2.22)$$

where θ_r^* is the model parameter of the r -order model determined through Algorithm 1. The first multiplicative term in (2.22), which is defined in (2.14)



(a) Lipschitz quotient



(b) Description length

Figure 2.9: The (a) Lipschitz quotient and (b) Description length of the proposed TVSQ prediction model at different model order r .

as the outage rate, represents the ability of a model to describe the data. The second multiplicative term increases with the number of parameters ($2r + 1$) and decreases with the size of training set $N(T - r)$. Thus, this term roughly indicate the whether the training set is sufficiently large for training a r -order model. The definition of (2.22) balances the accuracy and the complexity of the model. In Fig. 3, I plot the description lengths of the proposed models under different model orders. It is seen that the minimum description length is achieved at $r = 12$. Therefore, I select $r = 12$.

2.4 Model Evaluation and Analysis

In this section, I first evaluate and validate the proposed TVSQ model in Section 2.4.1 and Section 2.4.2. Then, in Section 2.4.3, I analyze several important properties of the model. Based on the analysis, I simplify the model to a simpler form that can be used for my adaptive transmission algorithm design.

2.4.1 Performance Evaluation

I train the model parameters using my database via Algorithm 1. The optimized parameters of the 12-order HW model are listed in Table 2.2.

In Table 2.3, I list the outage rate of the trained model on all of the 15 test videos. The average outage rate is 8.06%. This means that the model can accurately predict 91.94% of the TVSQs in the database. Furthermore, in Table 2.3, I also calculated the linear correlation coefficient and the Spearman's

Table 2.2: Optimized parameters of the proposed HW model.

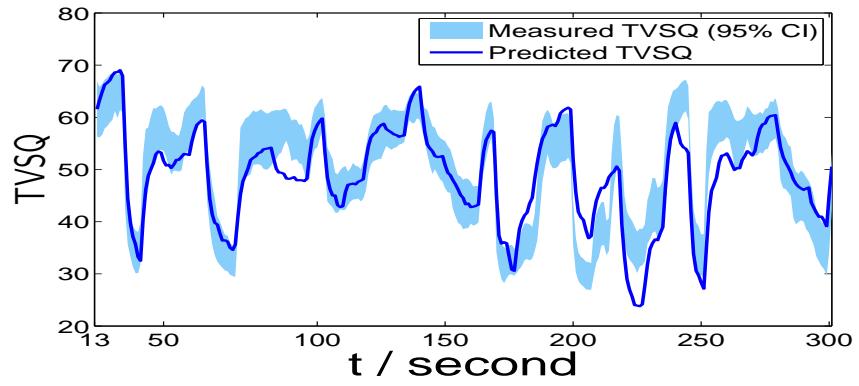
Input	$\beta = (0.0548, 1.9232, -150.0104, 179.9914)$	
IIR filter	$\sum_{d=1}^r f_d z^{-d}$	$0.0218z^{-1} + 0.0261z^{-2} + 0.0278z^{-3} + 0.0280z^{-4} + 0.0232z^{-5} + 0.0175z^{-6} + 0.0171z^{-7} + 0.0172z^{-8} + 0.0155z^{-9} + 0.0163z^{-10} + 0.0174z^{-11} + 0.0177z^{-12}$
	$\sum_{d=0}^r b_d z^{-d}$	$0.0127z^0 + 0.1119z^{-1} + 0.0765z^{-2} + 0.0405z^{-3} + 0.0163z^{-4} + 0.0110z^{-5} + 0.0192z^{-6} + 0.0058z^{-7} - 0.0023z^{-8} - 0.0019z^{-9} + 0.0042z^{-10} + 0.0023z^{-11} - 0.0054z^{-12}$
Output	$\gamma = (0.1411, -0.7397, 9.9907, 99.9957)$	

rank correlation coefficient between the predicted TVSQ and the measured TVSQ values. The average linear correlation and rank correlation achieved by my model is 0.885 and 0.880, respectively.

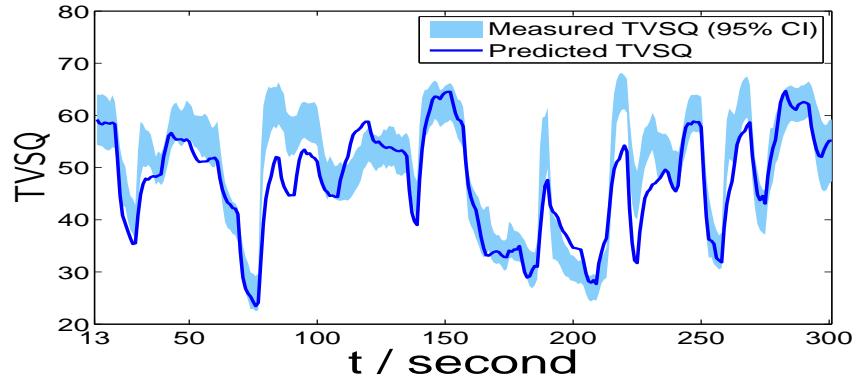
Table 2.3: The performance of the TVSQ prediction model on the TVSQ database

	#1	#2	#3	#4	#5	#6	#7	#8
outage rate(%)	12.15	11.46	9.38	18.06	9.72	4.17	10.76	8.33
linear correlation	0.868	0.897	0.862	0.785	0.919	0.936	0.859	0.896
rank correlation	0.881	0.857	0.875	0.814	0.897	0.943	0.872	0.901
	#9	#10	#11	#12	#13	#14	#15	mean
outage rate(%)	8.33	8.33	3.82	7.64	1.74	6.25	0.69	8.06
linear correlation	0.845	0.863	0.938	0.898	0.892	0.916	0.906	0.885
rank correlation	0.833	0.859	0.911	0.899	0.870	0.927	0.866	0.880

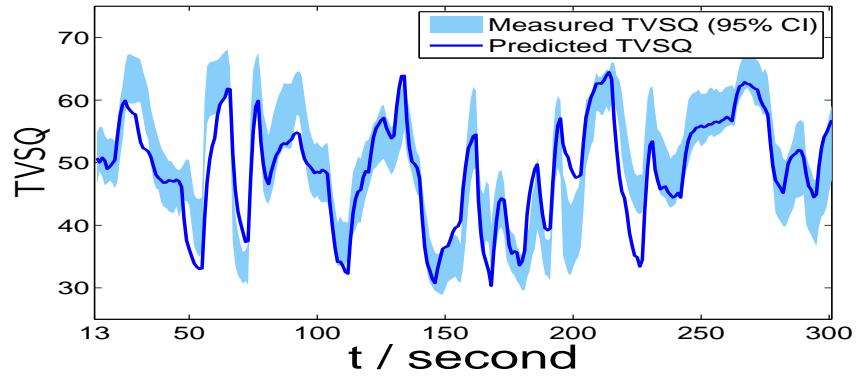
In Fig. 2.10 - 2.14, I plot the predicted TVSQs and the 95% confidence interval of the measured TVSQs. The proposed model effectively tracked the measured TVSQs of the 15 quality-varying videos.



(a) Video #1

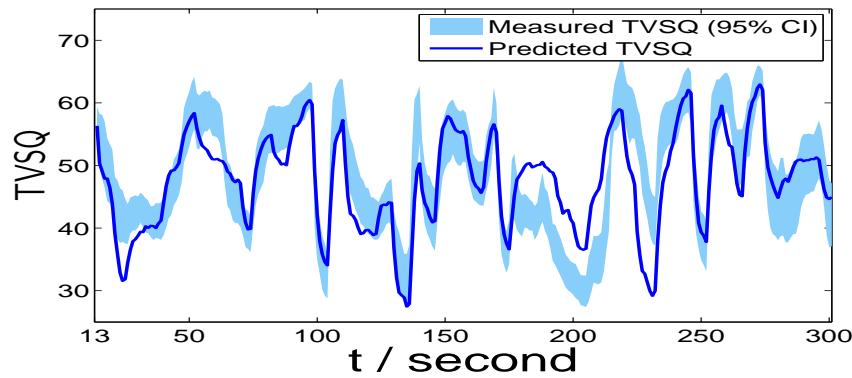


(b) Video #2

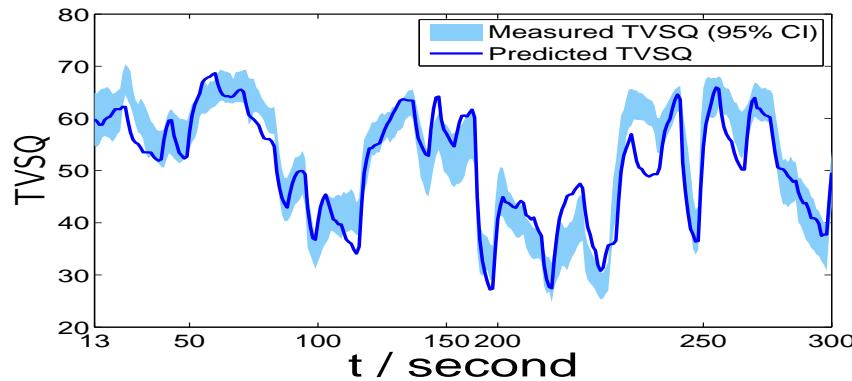


(c) Video #3

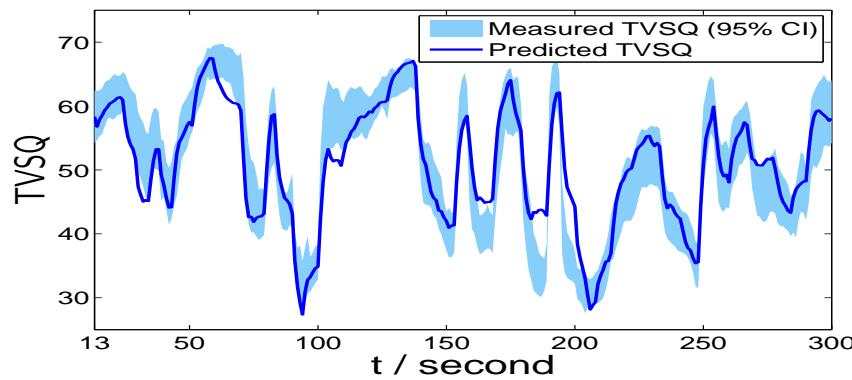
Figure 2.10: The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 1 of 5.



(a) Video #4

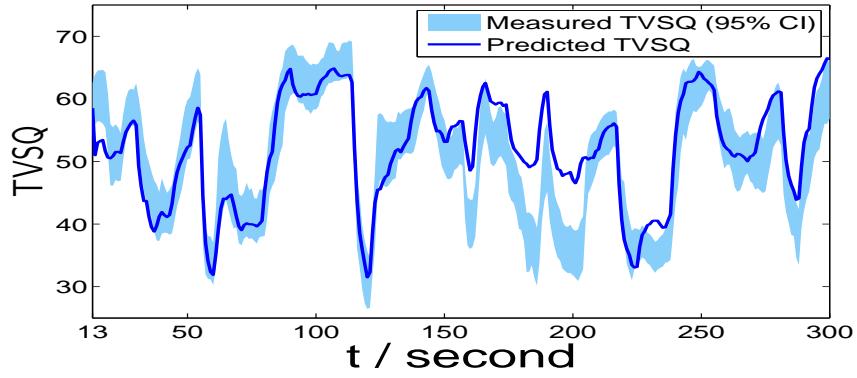


(b) Video #5

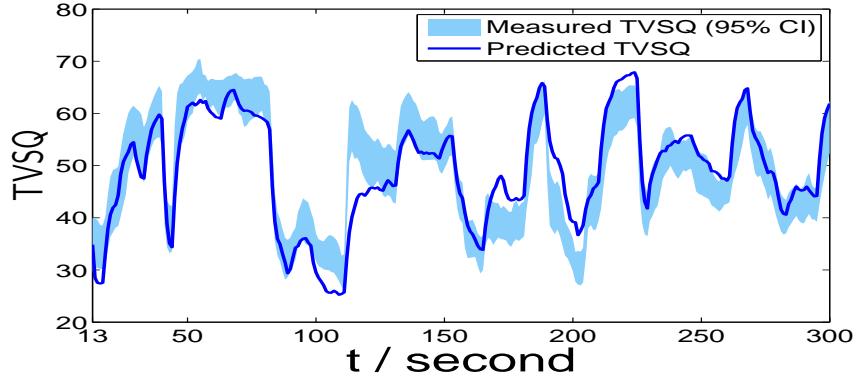


(c) Video #6

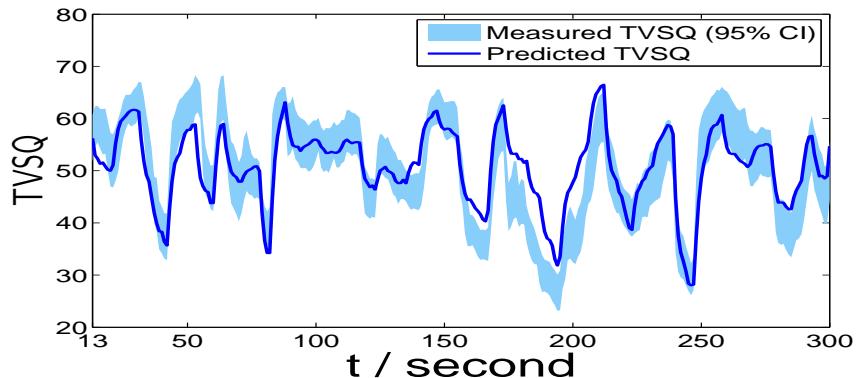
Figure 2.11: The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 2 of 5.



(a) Video #4

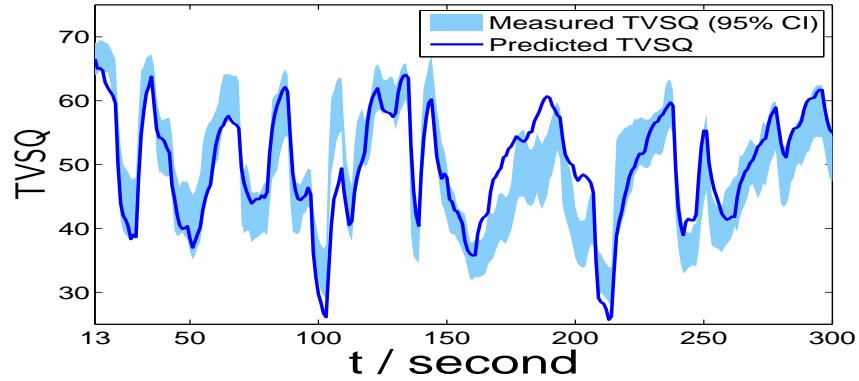


(b) Video #5

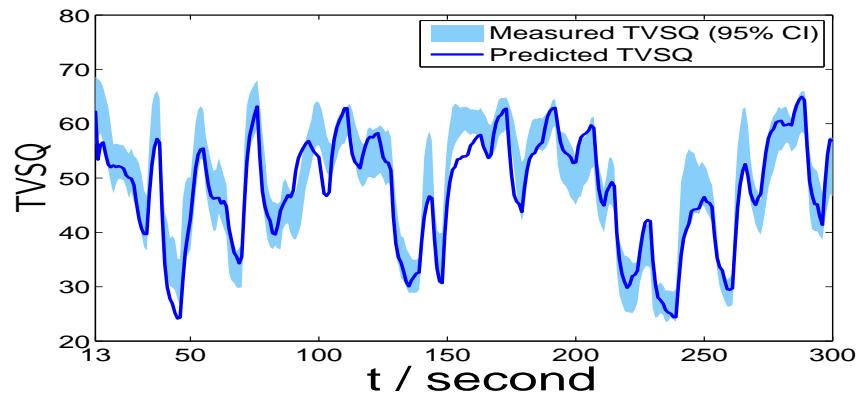


(c) Video #6

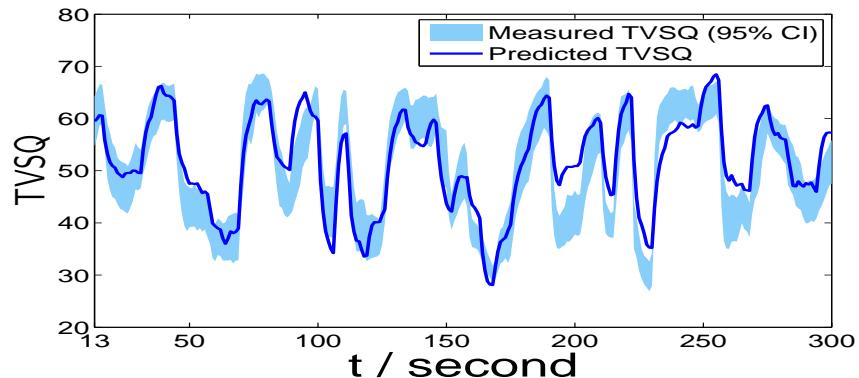
Figure 2.12: The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 3 of 5.



(a) Video #4

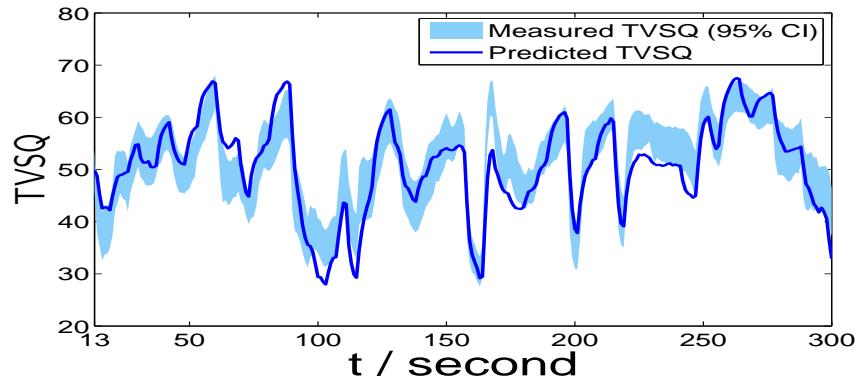


(b) Video #5

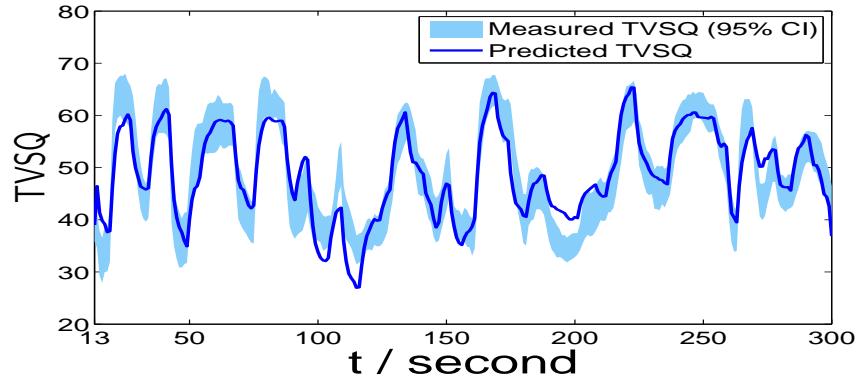


(c) Video #6

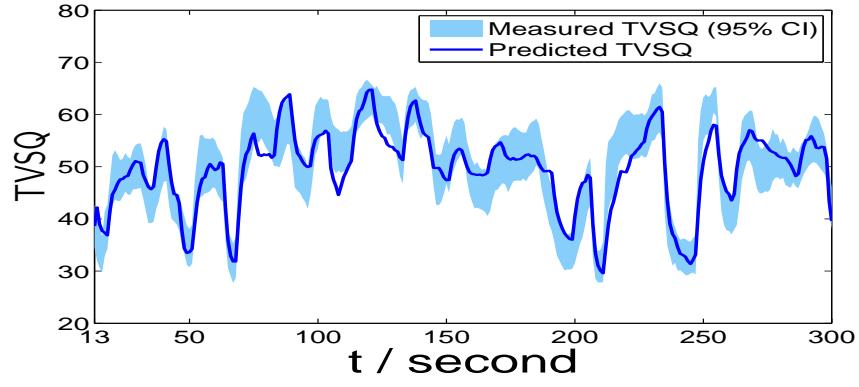
Figure 2.13: The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 4 of 5.



(a) Video #4



(b) Video #5



(c) Video #6

Figure 2.14: The predicted TVSQ and the 95% confidence interval (CI) of the TVSQs measured in the subjective study. Part 5 of 5.

The performance of our TVSQ prediction method is compared with the following baselines in Fig. 2.15.

- **PSNR without HW** The TVSQ is predicted with the average PSNR of each second of the videos.
- **SSIM without HW** The TVSQ is predicted with the average SSIM indices of each second of the videos.
- **RRED without HW** The TVSQ is predicted with the RRED indices of each second of the videos.
- **PSNR with HW** The STSQ is predicted with the average PSNR of each second of the videos. Then, the predicted STSQ is input to the Hammerstein-Wiener model for TVSQ prediction.
- **SSIM with HW** The STSQ is predicted with the average SSIM indices of each second of the videos. Then, the predicted STSQ is input to the Hammerstein-Wiener model for TVSQ prediction.

It is seen that the proposed TVSQ prediction method achieves higher linear correlation than the baselines. It is also observed that, for PSNR, SSIM, and RRED, the Hammerstein-Wiener model can significantly improve their performance in TVSQ prediction. It is mainly because the Hammerstein-Wiener model can capture the hysteresis effects of human vision system.

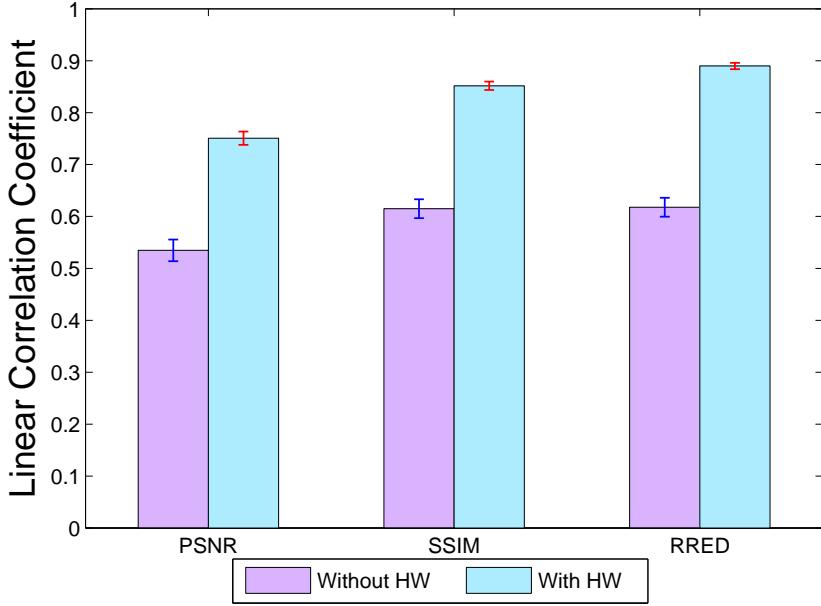


Figure 2.15: The performance comparison between the proposed TVSQ prediction method and the baselines. The length of the bars shows the linear correlation coefficients achieved by the TVSQ prediction methods. The 95% confidence interval is also shown for reference.

I evaluate the statistical significance of the gain of our method in TVSQ prediction by F-test [16]. The results are tabulated in Table 2.4. The null hypothesis is that the prediction error of the prediction error of the row and the prediction error of the column have equal variance at the 95% confidence interval. The alternative hypothesis is the variance of the row is greater (or less) than that of the column. Table 2.4 indicates which row is statistically superior (“1”), statistically equivalent (“-”) or statistically inferior (“0”) to which column. It is seen that the prediction error of my TVSQ model is less than all the baseline methods.

Table 2.4: Results of the F-test performed on the prediction errors. Each entry in the table is a symbol of “0”, “-”, or “1”. A symbol value of “1” indicates that the statistical performance of the method in the row is superior to that of the method in the column. A symbol value of “0” indicates that the statistical performance of the method in the row is inferior to that of the method in the column and - indicates that the statistical performance of the method in the row is equivalent to that of the method in the column. M1 through M6 are “PSNR without HW”, “SSIM without HW”, “RRED without HW”, “PSNR with HW”, “SSIM with HW”, and “RRED with HW” respectively

	M1	M2	M3	M4	M5	M6
M1	-	0	0	0	0	0
M2	1	-	-	0	0	0
M3	1	-	-	0	0	0
M4	1	1	1	-	0	0
M5	1	1	1	1	-	0
M6	1	1	1	1	1	-

2.4.2 Cross Validation

To rule out the risk of over-fitting the model to the TVSQ database, a leave-one-out cross-validation protocol were employed to check whether the model trained on our database is robust. Each time, the 5 videos corresponding to the same reference video were selected as the validation set and trained the model parameters on the other 10 videos. This procedure was repeated such that all the videos are included once in the validation set. The results are summarized in Table 2.3. Comparing with the models trained on the whole database, the performance of the models in the cross-validation is only slightly degraded. Therefore, the model obtained from our database appears to be robust.

Table 2.5: Results of leave-one-out cross-validations. Here $\{n_1, \dots, n_2\}$ denotes the set of video sequences with sequence numbers from n_1 to n_2 .

validation set	$\{1, \dots, 5\}$		$\{6, \dots, 10\}$		$\{11, \dots, 15\}$	
training set	$\{1, \dots, 15\}$	$\{6, \dots, 15\}$	$\{1, \dots, 15\}$	$\{1, \dots, 5, 11, \dots, 15\}$	$\{1, \dots, 15\}$	$\{1, \dots, 11\}$
outage rate (%)	12.154	13.75	7.98	10.00	4.03	5.00
linear correlation	0.866	0.860	0.881	0.875	0.910	0.903
rank correlation	0.864	0.862	0.882	0.879	0.895	0.889

In the following, I analyze several important properties of the TVSQ model.

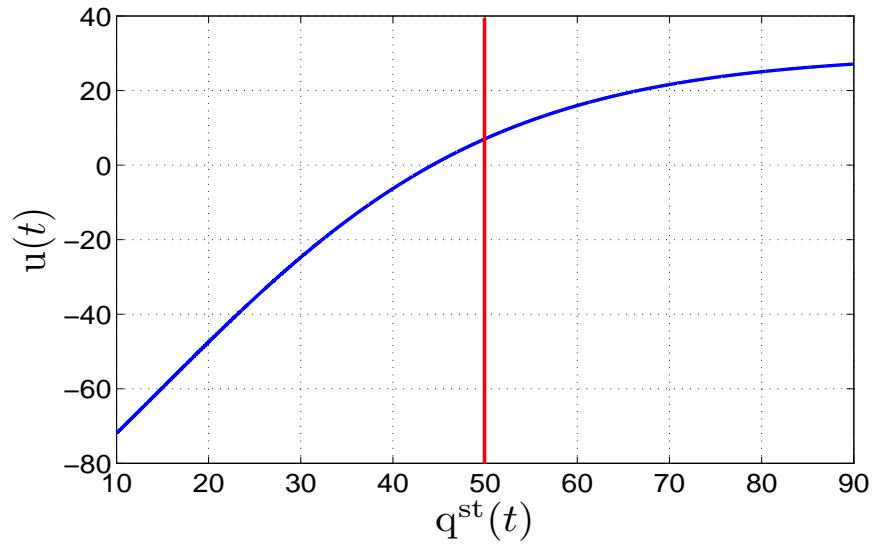
2.4.3 Model Analysis and Simplification

In Fig. 2.16(a), I plot the input nonlinearity of the TVSQ model. As the input $q^{st}(t)$ increases, the gradient of the input nonlinearity diminishes. In particular, the slope of the the input nonlinearity is much larger when $q^{st}(t) < 50$. As discussed in Section 2.2.1, an RDMOS of 50 indicates acceptable STSQ. Therefore, the concavity of the input non-linearity implies that, the TVSQ is more sensitive to quality variations when viewers are watching low quality videos.

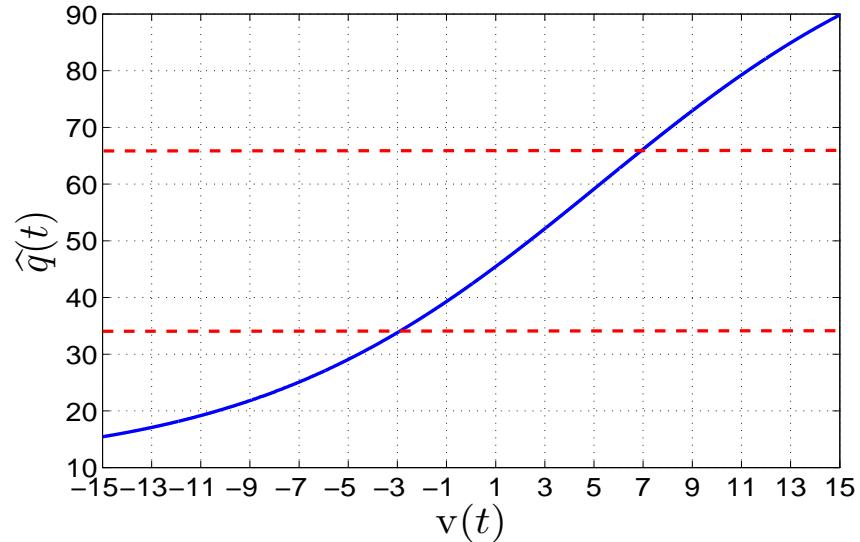
In Fig. 2.16(b), I plot the output nonlinearity of my TVSQ model. I observe that, when $30 \leq \hat{q}(t) \leq 70$, the function is almost linearly increasing with the input. This observation inspired us to further simplify the model. I replace the output nonlinearity with the following linear function

$$\hat{q}(t) = \kappa_1 y(t) + \kappa_2. \quad (2.23)$$

The model parameter trained on my TVSQ database is listed in Table 2.6.



(a) Input nonlinearity



(b) Output nonlinearity

Figure 2.16: Input and output nonlinearities of the HW model parameterized on the TVSQ database. The input nonlinearity is concave and the output nonlinearity is close to a linear function.

Table 2.6: Optimized parameters of the proposed HW model with linear output function.

Input	$\boldsymbol{\beta} = (0.0458, 1.6482, -150.0105, 179.9918)$	
IIR filter	$\sum_{d=1}^r f_d z^{-d}$	$0.1182z^{-1} + 0.0780z^{-2} + 0.0508z^{-3} + 0.0398z^{-4} + 0.0271z^{-5} + 0.0167z^{-6} + 0.0146z^{-7} + 0.0161z^{-8} + 0.0147z^{-9} + 0.0140z^{-10} + 0.0115z^{-11} + 0.0104z^{-12}$
	$\sum_{d=0}^r b_d z^{-d}$	$0.1244z^0 + 0.3792z^{-1} + 0.2710z^{-2} + 0.1450z^{-3} + 0.0789z^{-4} + 0.0368z^{-5} + 0.0197z^{-6} + 0.0124z^{-7} - 0.0007z^{-8} - 0.0098z^{-9} - 0.0075z^{-10} - 0.0073z^{-11} - 0.0123z^{-12}$
Output	$\kappa = (0.7013, 49.9794)$	

The performance of this simplified model is shown in Table 2.7. Comparing with Table 2.3, it can be seen that the outage rate is increased slightly but that the linear correlation coefficients and Spearman's rank correlation coefficients are almost the same. Hence, I conclude that the simplified model can also predict TVSQ reasonably well. An important advantage of this simplified model is its concavity. Indeed, since the input nonlinearity function is a concave function and the filter is linear, then at any time t , the mapping between $q^{st}(t)$ and $\hat{q}(t)$ is also concave. Hence, the simplified model can thus be easily incorporated into a convex TVSQ optimization problem, which can be easily solved and analyzed.

Table 2.7: Performance of the TVSQ prediction model when the output non-linearity is replaced with a linear function

	#1	#2	#3	#4	#5	#6	#7	#8
outage rate(%)	12.00	10.91	10.55	16.00	9.82	5.45	10.18	11.64
linear correlation	0.840	0.896	0.864	0.787	0.920	0.930	0.869	0.876
rank correlation	0.866	0.845	0.876	0.818	0.906	0.939	0.883	0.887
	#9	#10	#11	#12	#13	#14	#15	mean
outage rate(%)	10.55	10.18	4.00	10.91	1.45	6.91	1.09	8.78
linear correlation	0.854	0.842	0.937	0.886	0.883	0.914	0.897	0.879
rank correlation	0.851	0.840	0.916	0.890	0.853	0.935	0.853	0.877

As indicated in Section 2.1.2, the initial conditions $(y)_{1:r}$ are required to estimate TVSQ. For online video streaming applications, however, $(y)_{1:r}$ is unavailable because $(y)_{1:r}$ is given by $(q^{tv})_{1:r}$ and the latter is the TVSQ of the first r seconds of the video. The transfer function of the linear filter is

$$H(z) = \frac{\sum_{d=0}^r b_d z^{-d}}{1 - \sum_{d=1}^r f_d z^{-d}}. \quad (2.24)$$

According to classical results from system theory, if the root radius of the denominator polynomial $z^r - \sum_{d=1}^r f_d z^{r-d}$, is less than 1, the impact of the initial condition fades to 0 as $t \rightarrow \infty$ exponentially fast. Denoting by $\rho(\mathbf{f})$ the root radius of $z^r - \sum_{d=1}^r f_d z^{r-d}$, the fading speed is $\rho(\mathbf{f})^t$. Here, I define the quantity $\tau(\mathbf{f}) = -3/\ln \rho(\mathbf{f})$. Over every $\tau(\mathbf{f})$ seconds, the impact of the initial state fades to $e^{-3} \approx 5\%$ of its original level. Therefore, $\tau(\mathbf{f})$ indicates the delay before my TVSQ model starts to track TVSQ. For the model trained on the TVSQ database, I found that $\tau(\mathbf{f}) = 16.5151$ seconds. This means that my model cannot accurately predict the TVSQs of the first 16.5151 seconds of the video. For quality monitoring of long videos, this delay is tolerable.

Next, I consider a wireless network and design an adaptive transmission algorithm based on this simplified TVSQ model.

2.5 TVSQ-optimized Rate-Switching Algorithm

In this section, I propose an adaptive wireless video transmission algorithm which optimizes the TVSQs of the users sharing a wireless down-link. I first introduce the system model of the considered wireless video transmission system in Section 2.5.1. Then, in Section 2.5.2, I introduce the adaptive rate switching algorithm and prove its optimality. Section 2.5.3 evaluates the algorithm via simulations.

2.5.1 System Model

In this section, I first outline the architecture of the wireless video transmission systems. Then, I explain the channel model and the rate-quality model. Finally, I formulate my algorithm design as an optimization problem.

System Overview: In this chapter, I study a wireless network that consists of a base station and a group of users sharing the down-link (see Fig. 2.17). I assume that there are two types of users: the data users and the video users. When a data user arrives, it starts downloading data from a Content delivery network (CDN) via the base station. Upon the completion of the downloading, the data user leaves the network. When a video user arrives, it requests to watch a video that is stored at a CDN and is streamed to the user via the base station. Parallel to prior work such as [35][38][67], I assume a proxy is

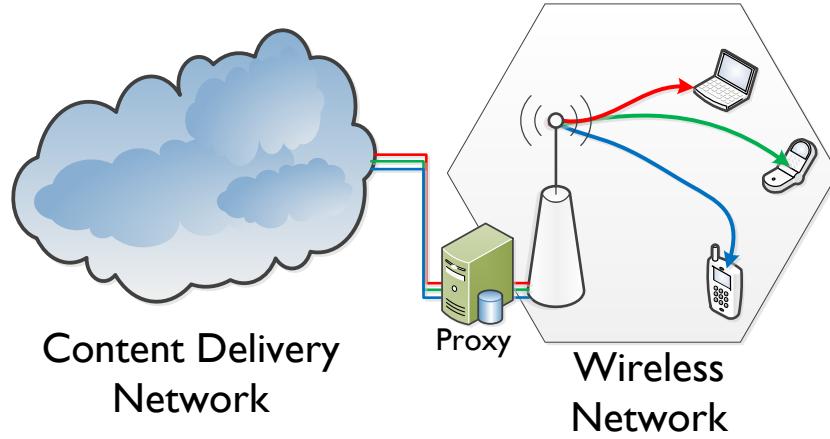


Figure 2.17: The architecture of the considered wireless network. The video sources are stored in the content delivery networks. The proxy is co-located with the base station.

co-located with the base station. I treat the data users as background traffic and the proxy is only used to control the video streams.

When a video user arrives, the proxy adapts the transmission data rates according to the varying channel conditions. In the meanwhile, the proxy matches the video source rate to the transmission data rate to ensure continuous playback. I assume the proxy operates in a time-slotted manner, where the duration of each slot is $\Delta T = 1$ second. To maintain continuous playback, in each slot, the base station delivers 1 seconds of video to each user. I assume the rate adaptation actions are conducted at the beginning of every time slot.

Channel Model For each user, I employ a binary random process $\{\mathbb{1}_u(t) : t \in \mathbb{N}^+\}$ to describe its arrival and departure. If user u has arrived to the network

before slot t and is still in the network at slot t , we have $\mathbb{1}_u(t) = 1$. Otherwise, $\mathbb{1}_u(t) = 0$. The time spent by a user in the network is the random variable $T_u = \sum_{t=1}^{\infty} \mathbb{1}_u(t)$. I assume the arrival and departure processes $\{\mathbb{1}_u(t) : t \in \mathbb{N}^+\}$ are independent across users.

I denote by $\mathcal{U}^d(t)$ the set of data users in the network at slot t . For each data user $u \in \mathcal{U}^d(t)$, I employ a random variable $K_u^d(t)$ to represent the amount of data that is downloaded in slot t . The downloading throughput is thus $R_u^d(t) = K_u^d(t)/\Delta T$. I denote by $\mathcal{U}^v(t)$ the set of video users in the network at slot t and by $k_u^v(t)$ the amount of video data that is transmitted to a video user in slot t . The channel throughput to the user is thus $r_u(t) = k_u^v(t)/\Delta T$. I assume that the proxy adapt user u 's the video source rate to $r_u(t)$ for all t . In the following, I call $\mathbf{r}(t) = (r_u(t) : u \in \mathcal{U}^v(t))$ the *video rate vector* at time t .

I assume the wireless system is a time division multiple access (TDMA) system [38][35]. Denote by $P_u(t)$ the channel throughput when user u uses the channel during slot t . Then, a video user costs $\frac{k_u^v(t)}{P_u(t)}$ seconds to download a video segment. Similarly, each data user costs $\frac{K_u^d(t)}{P_u(t)}$ seconds to download data. Since the total transmission time of all user is less than ΔT , we have $\sum_{u \in \mathcal{U}^v(t)} \frac{k_u^v(t)}{P_u(t)} + \sum_{u \in \mathcal{U}^d(t)} \frac{K_u^d(t)}{P_u(t)} \leq \Delta T$. Since $r_u(t) = k_u^v(t)/\Delta T$ and $R_u^d(t) = K_u^d(t)/\Delta T$, we have $\sum_{u \in \mathcal{U}^v(t)} \frac{r_u(t)}{P_u(t)} + \sum_{u \in \mathcal{U}^d(t)} \frac{R_u^d(t)}{P_u(t)} \leq 1$. Therefore, for TDMA systems, the set of video rate vectors that can be supported by the wireless

channel is given by

$$\mathcal{C}(t) = \{\mathbf{r} : c_t(\mathbf{r}(t)) \leq 0\}, \quad (2.25)$$

where the function $c_t(\mathbf{r}(t)) = \sum_{u \in \mathcal{U}^v(t)} \frac{r_u(t)}{P_u(t)} + \sum_{u \in \mathcal{U}^d(t)} \frac{R_u^d(t)}{P_u(t)} - 1$.

In rate-switching video streaming protocols, the video data rate can only takes values in a finite and discrete set, which corresponds to the data rates of different video representations. In my problem formulation, I relax the constraint a bit and allow $r_u(t)$ to take values in a compact interval, i.e,

$$r_u(t) \in [r_u^{\min}(t), r_u^{\max}(t)], \quad (2.26)$$

where $r_u^{\min}(t)$ and $r_u^{\max}(t)$ denote the minimum and maximum data rate available for user u , respectively. Letting $\mathcal{R}(t) = \Pi_{u \in \mathcal{U}^{\text{av}}(t)} [r_u^{\min}(t), r_u^{\max}(t)]$, equation (2.26) can be rewritten as

$$\mathbf{r}(t) \in \mathcal{R}(t), \quad \forall t. \quad (2.27)$$

In my algorithm implementation, I round up the optimal video data rate obtained under this relaxed constraint to the nearest available data rate in all representations.

Rate-TVSQ Model Denoting by $q_u^{\text{st}}(t)$ the STSQ of user u at time t , the relationship between $q_u^{\text{st}}(t)$ and $r_u(t)$ can be characterized by a rate-quality function:

$$q_u^{\text{st}}(t) = f_{u,t}^{\text{rq}}(r_u(t)), \quad (2.28)$$

where the function $f_{u,t}^{rq}(\cdot)$ depends on the characteristics of the video transmitted by user u at time t . Denoting by $q_u^{tv}(t)$ the TVSQ of user u at time t , the relationship between $q_u^{st}(t)$ and $q_u^{tv}(t)$ can be captured by the simplified TVSQ prediction model in (2.1), (2.2) and (2.23). Letting $x_u(t)$ denote the input to the linear filter of user u (see equation (2.2)), we have

$$x_u(t) = f^{in} (q_u^{st}(t)), \quad (2.29)$$

where $f^{in}(x) = \beta_3 + \beta_4 \frac{1}{1+\exp(-(\beta_1 x + \beta_2))}$. Letting $f_{u,t}^{rx}(\cdot) = f^{in} (f_{u,t}^{rq}(\cdot))$, the relationship between $r_u(t)$ and $q_u^{tv}(t)$ is given by

$$x_u(t) = f_{u,t}^{rx}(r_u(t)), \quad (2.30)$$

$$y_u(t) = \mathbf{b}^T (x_u)_{t-r:t} + \mathbf{f}^T (y_u)_{t-r:t-1}, \quad (2.31)$$

$$q_u^{tv}(t) = \kappa_1 y_u(t) + \kappa_2, \quad (2.32)$$

where $y_u(t)$ is the output of the linear filter of user u .

Problem Formulation Given a finite time horizon T , I denote by $\mathcal{U}_{1:T}^v = \cup_{t=1}^T \mathcal{U}^v(t)$ the set of users who arrive to the network before T . The goal of the my adaptive transmission algorithm is maximize the sum of time-averaged TVSQ of all users in $\mathcal{U}_{1:T}^v$ by adapting the video rate vector $\mathbf{r}(t)$, i.e.,

$$\begin{aligned} & \underset{\mathbf{r}(t), \forall 1 \leq t \leq T}{\text{maximize}} \quad \sum_{u \in \mathcal{U}_{1:T}^v} \left(\frac{1}{T} \sum_{t=1}^T (q_u^{tv}(t) \mathbb{1}_u(t)) \right) \\ & \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \quad \forall t \\ & \quad x_u(t) = f_{u,t}^{rx}(r_u(t)), \quad \forall u \in \mathcal{U}^v(t), \quad \forall t, \\ & \quad y_u(t) = \mathbf{b}^T (x_u)_{t-r:t} + \mathbf{f}^T (y_u)_{t-r:t-1}, \quad \forall u \in \mathcal{U}^v(t), \quad \forall t \\ & \quad q_u^{tv}(t) = \kappa_1 y_u(t) + \kappa_2, \quad \forall u \in \mathcal{U}^v(t), \quad \forall t. \end{aligned} \quad (2.33)$$

Table 2.8: Notations used in the system model of the TVSQ-optimized wireless video transmission system.

$\mathbb{1}_u(t)$	Indicator: u is in the network at t
T_u	Time spent by user u in the network
$\mathcal{U}^d(t)$	Data users at time t ;
$\mathcal{U}^v(t)$	Video users at time t ;
$r_u(t)$	Data rate of video user u at t
$\mathcal{C}(t)$	Rate region at t
$\mathcal{R}(t)$	Available video source rate at t
$q_u^{st}(t)$	The STSQ of user u at t
$q_u^{tv}(t)$	The STSQ of user u at t
$x_u(t)$	The input of the filter of user u at t
$y_u(t)$	The output of the filter of user u at t
$f_{u,t}^{rx}(\cdot)$	The mapping from $r_u(t)$ to $x_u(t)$

In the next section, I propose the TVSQ-optimized adaptive video transmission algorithm. The frequently used notations are summarized in Table 3.2.

2.5.2 Adaptive Transmission Algorithm Design

The proposed adaptive transmission algorithm is listed in Algorithm 3. In each slot, the algorithm greedily maximize the weighted sum of $x_u(t)$. The weight is simply the duration of each video user T_u .

The following theorem provides a condition under which the rate vector selected by Algorithm 3 is the optimal solution to problem (2.33).

Theorem 2.5.1. *Let $\{h(d) : \forall d \in \mathbb{N}^+\}$ be the impulse response of the linear filter in (2.2). If $h(d) \geq 0, \forall d$, the solution of Algorithm 3 is the optimal solution to problem (2.33).*

Algorithm 3 Adaptive video transmission algorithm for TVSQ optimization

- 1: **for** $t = 1 \rightarrow \infty$ **do**
- 2: Choose rate vector $\hat{\mathbf{r}}(t)$ that solves the problem

$$\begin{aligned} & \underset{\mathbf{r}(t)}{\text{minimize}} \quad \sum_{u \in \mathcal{U}^v(t)} \frac{1}{\bar{T}_u} \mathbf{x}_u(t) \\ & \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \\ & \quad \mathbf{x}_u(t) = \mathbf{f}_{u,t}^{\text{rx}}(\mathbf{r}_u(t)), \quad \forall u \in \mathcal{U}^v(t). \end{aligned} \quad (2.34)$$

- 3: For all $u \in \mathcal{U}^v(t)$, compute $\mathbf{y}_u(t)$ and $\hat{q}_u^{\text{tv}}(t)$:

$$\hat{\mathbf{x}}_u(t) = \mathbf{f}_{u,t}^{\text{rx}}(\hat{\mathbf{r}}_u(t)), \quad (2.35)$$

$$\hat{\mathbf{y}}_u(t) = \mathbf{b}^T (\hat{\mathbf{x}}_u)_{t-r:t} + \mathbf{f}^T (\hat{\mathbf{y}}_u)_{t-r:t-1}, \quad (2.36)$$

$$\hat{q}_u^{\text{tv}}(t) = \kappa_1 \hat{\mathbf{y}}_u(t) + \kappa_2. \quad (2.37)$$

- 4: **end for**
-

Proof. Let $\{\tilde{\mathbf{r}}(t) : \forall t\}$ be an arbitrary rate adaptation policy. Denote by $\tilde{\mathbf{x}}_u(t)$, $\tilde{\mathbf{y}}_u(t)$ and $\tilde{q}_u^{\text{tv}}(t)$ the corresponding filter input, filter output and TVSQs under rate adaptation policy $\tilde{\mathbf{r}}^v(t)$. For the optimization objective in problem (2.33), we have

$$\begin{aligned} & \sum_{u \in \mathcal{U}_{1:T}^v} \left(\frac{1}{\bar{T}_u} \sum_{t=1}^T (\hat{q}_u^{\text{tv}}(t) \mathbb{1}_u(t)) \right) \\ &= \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{q}_u^{\text{tv}}(t) \right) \\ &= \kappa_1 \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{\mathbf{y}}_u(t) \right) + \kappa_2 \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \right) \end{aligned} \quad (2.38)$$

Since $\kappa_1 > 0$ and the second term in (2.38) is constant, if I could show that $\sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{\mathbf{y}}_u(t) \right) \geq \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \tilde{\mathbf{y}}_u(t) \right)$, the theorem is proved.

Given that $y_u(t) = \sum_{d=0}^{\infty} h(d)x_u(t-d)$, we have

$$\begin{aligned} & \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{y}_u(t) \right) \\ &= \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \sum_{d=0}^{\infty} h(d) \hat{x}_u(t-d) \right) \\ &= \sum_{t=1}^T \sum_{d=0}^{\infty} h(d) \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{x}_u(t-d) \right) \end{aligned}$$

According to Algorithm 3, we always have

$$\sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{x}_u(t-d) \right) \geq \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \tilde{x}_u(t-d) \right).$$

Therefore, we have

$$\begin{aligned} & \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \hat{y}_u(t) \right) \\ &\geq \sum_{t=1}^T \sum_{d=0}^{\infty} h(d) \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \tilde{x}_u(t-d) \right) \\ &= \sum_{t=1}^T \sum_{u \in \mathcal{U}^v(t)} \left(\frac{1}{\bar{T}_u} \tilde{y}_u(t) \right), \end{aligned}$$

which completed the proof. \square

In Fig. 2.18, I plot the impulse response of the linear filter in (2.2). It is seen that all the tabs are positive. It affirms the assumptions in Theorem 2.5.1 and justifies the optimality of the proposed rate adaptation algorithm.

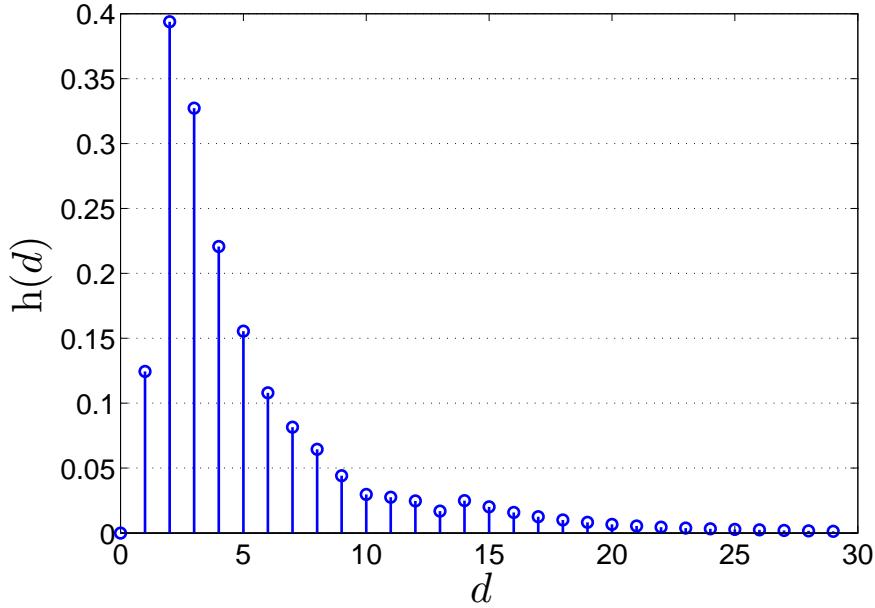


Figure 2.18: The impulse response of the linear filter in (2.2). It is seen that all the tabs are positive, which justifies the optimality of the proposed rate adaptation algorithm.

2.5.3 Simulation Results

In my simulations, I assume the duration of time slot is $\Delta T = 1$ second. The users arrive to the network as a Poisson process. Their arrival rate is $\frac{1}{10}$ users/second. The time spent by each user in the network is exponentially distributed with mean value being 200 seconds. By Little's law, at each slot, there are averagely $200/10 = 20$ users in the network. I assume that each newly arrived user can either be a data user or be a video user with equal probability. Therefore, the video users arrive to the network as a Poisson process with arrival rate $1/20$ users/second. Since video streams are typically

more than tens of seconds long, I further assume the video duration is at least 40 seconds.

To simulate the variations of the rate-quality characteristics in each video stream, I assume the rate quality function $f_{u,t}^{\text{rq}}(\cdot)$ of each slot is independently sampled from the rate-quality functions in the video database [22]. I assume the minimum and maximum available data rate for video users in (2.26) are $r_u^{\min}(t) = 302$ kbps and $r_u^{\max}(t) = 6412$ kbps, $\forall t \in \mathbb{N}^+$. For data users, the data rate R_u^d are assumed to be uniformly distributed in $[100, 300]$ kbps.

I model the channel capacity $P_u(t)$ as the product of two independent random variables, i.e., $P_u(t) = P_u^{\text{avg}} \times P_u^*(t)$. The random variable P_u^{avg} is employed to simulate the heterogeneity of channel condition across users. I assume that P_u^{avg} is uniformly distributed in $[1250\alpha, 3750\alpha]$ kbps, where the parameter α is used to scale the channel capacity in my simulations. The random variable $P_u^*(t)$ is employed to simulate the channel variation across time slots. I assume that the process of $\{P_u^*(t) : t \in \mathbb{N}^+\}$ is an i.i.d. process with $P_u^*(t)$ being uniformly distributed in $[0.5, 1.5]$.

I employ two baseline algorithms for comparison. The first baseline algorithm maximizes the sum of transmission rate in each slot, i.e., $\sum_{u \in \mathcal{U}^v(t)} r_u(t)$. The second baseline algorithm maximizes the sum of STSQs in each slot, i.e., $\sum_{u \in \mathcal{U}^v(t)} q_u^{\text{st}}(t)$. I scale the channel condition from $\alpha = 6$ to $\alpha = 16$ and simulate 250 user arrivals. The average TVSQ under the proposed algorithm and the two base line algorithms are shown in Fig. 2.19. It is seen that the proposed algorithm outperforms the two baselines uniformly under all tested

channel conditions. At the same average TVSQ of 50, the proposed algorithm requires to scale the channel to $\alpha = 7$, while the two baselines needs to scale the channel to $\alpha = 14$ and $\alpha = 8$. In other words, comparing with the two baseline algorithms, the proposed algorithm reduces the requirement for network resources by 50% and 12.5%, respectively.

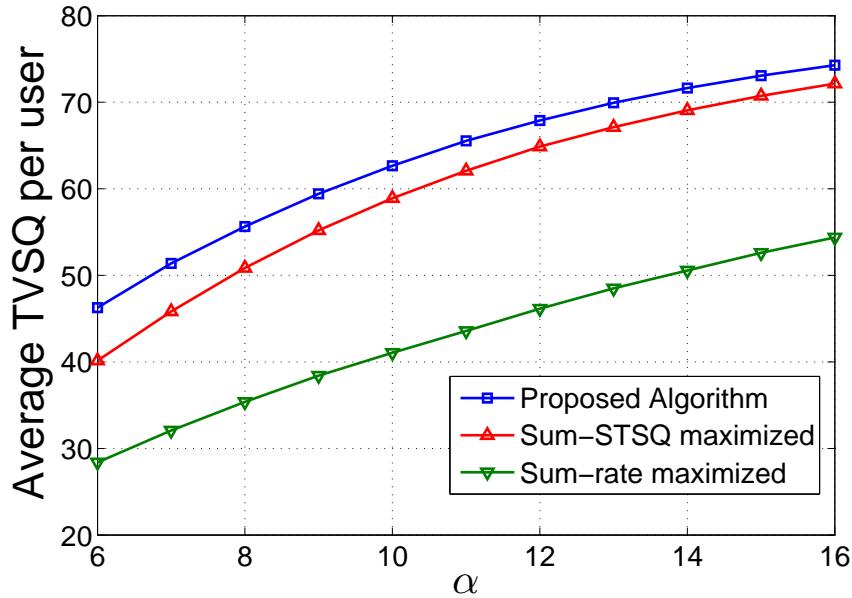


Figure 2.19: The average TVSQ of the proposed rate-switching algorithm. Two baseline are selected for comparison. The first baseline algorithm maximizes the sum of transmission rate in each slot. The second baseline algorithm maximizes the sum of STSQs in each slot.

2.6 Summary

In this chapter, I have proposed a model to predict the TVSQ of videos transmitted over rate-switching systems. The model is parameterized and

validated on a new database of quality-varying videos that simulate the true videos commonly encountered in rate-switching systems. Based on our analysis of the proposed model, the mapping from STSQ to TVSQ is not only monotone but also concave. Using these properties, I have designed an adaptive video transmission algorithm and proved its optimality in maximizing the time-averaged TVSQ.

Chapter 3

Adaptive Video Transmission with Overall Quality Constraints

In this chapter, I propose an adaptive video transmission algorithm to maximize the number of video users satisfying constraints on the Overall Quality (OQ). In most existing adaptive video transmission algorithms such as [19, 35, 36, 44, 49, 91], average video quality is employed as the proxy of OQ. But average quality does not reflect the impact of quality fluctuations on the OQ because two videos with the same average quality can have different level of quality fluctuations. Based on the subjective study discussed in Chapter 2, I propose a novel OQ metric, which achieves a strong correlation with the OQ measured in the subjective study. Then, using this OQ metric, I propose an adaptive rate-switching algorithm and an admission control algorithm that are designed to maximize the number of video users satisfying the certain OQ constraints.

In the following, I first introduces the proposed OQ metric in Section 3.1. Then, in Section 3.2, I explain the system model for the considered wireless video transmission systems. Finally, I explain the rate adaptation algorithm and the admission control algorithm in Section 3.3 and Section 3.4.

3.1 Proposed Overall Quality Metric

I propose to capture the OQ using the second order empirical cumulative distribution function (2^{nd} -order eCDF), which is defined as

$$F^{(2)}(x) = \frac{1}{T} \sum_{t=1}^T \max \{x - q^{\text{st}}(t), 0\}. \quad (3.1)$$

where T is the length of the video and $q^{\text{st}}(t)$ denotes the STSQ at time t . Since we have $\max\{x - q^{\text{st}}(t), 0\} > 0$ if and only if $q^{\text{st}}(t) < x$, the 2^{nd} -order eCDF captures for how long and to by how much the STSQ falls below x . If we interpret x as the threshold for users to judge whether the video quality is acceptable or not, then the 2^{nd} -order eCDF reflect the impact of the worst part of the video on the OQ. Since it has been recognized that the worst part of a video tends to dominate the overall quality of the whole video [10, 59, 63, 73, 88], the 2^{nd} -order eCDF could be used to predict the OQ.

Indeed, the efficacy of the the 2^{nd} -order eCDF can be justified by the experimental results of the subjective study discussed in Chapter 2. In the subjective study, I also asked each subject to feedback their judgment on the overall quality of each tested rate-switching videos (see Appendix A). Then, based on the subjects's feedback, the Mean Opinion Scores (MOSSs) of each tested video's OQ are computed following the guidelines of [37]. I calculate the linear correlation coefficient (LCC) between $F^{(2)}(x)$ and the computed MOSSs of all tested videos. In Fig 3.1, I plot the absolute value of the LCCs as a function of x . I find that, at $x^* = 37$, $F^{(2)}(x^*)$ can achieve a strong correlation of 0.8446 with MOSSs.

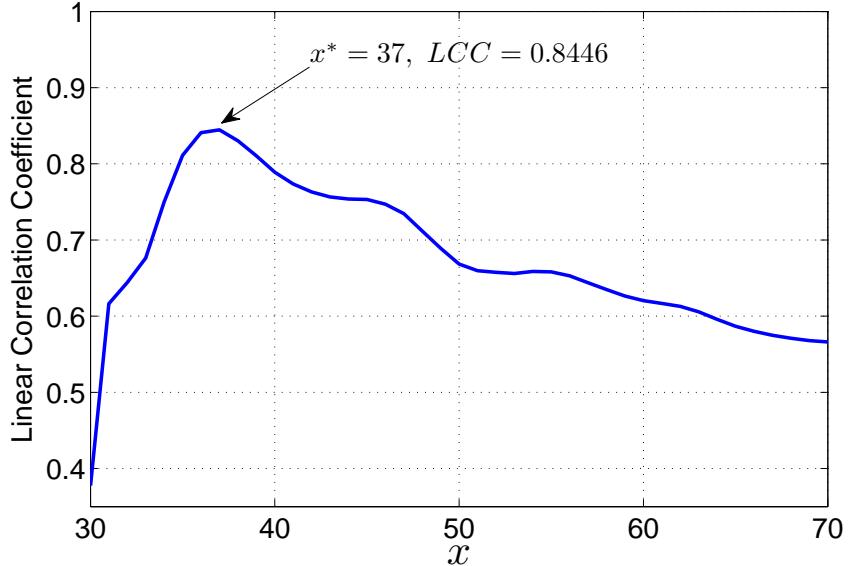


Figure 3.1: The absolute values linear correlation coefficient (LCC) between $F^{(2)}(x)$ and the mean opinion scores when different x is selected.

In Table 3.1, I compare the performance of the 2nd-order eCDF with several other OQ metrics. It is seen that the proposed metric achieves a much stronger correlation with the measured OQs. In particular, the widely used average quality metric only achieves a correlation of 0.5659. The minimum quality achieves a correlation of 0.4022. The mean-plus-variance metric used in [38] achieves a strong correlation of 0.7405. It is mainly because it captures the impact of quality variation. However, its performance is still worse than the second order eCDF.

Table 3.1: The correlation of different metrics with the OQ measured in the subjective study

OQ metrics	LCC
$\text{mean}\{(q^{\text{st}})_{1:T}\}$	0.5659
$\min_t\{(q^{\text{st}})_{1:T}\}$	0.4022
$\text{mean}\{(q^{\text{st}})_{1:T}\} + \lambda \text{var}\{(q^{\text{st}})_{1:T}\}$	0.7405
$F^{(2)}(x)$	0.8446

3.2 The System Model for Wireless Video Transmission

In this section, I first discuss the architecture of wireless networks considered in this chapter in Section 3.2.1. Then, I introduce channel model and rate-quality model in Section 3.2.2. Finally, in Section 3.2.3, I explain the two types of OQ constraints considered in this chapter.

3.2.1 System Overview

I consider the same wireless system as in Section 2.1. All the notations defined in Table 3.2 are inherited here. All users arrive to and depart from the network at random times. When a video user arrives, it requests a video that is stored at a content delivery network (CDN) and is streamed to the user via the base station. When a video is being streamed, the video data is first delivered to a receive buffer and then decoded for display. Paralleling prior work such as [35][38][67], we assume a proxy is colocated with the base station. Different from Chapter 2 where the proxy only responsible for rate adaptation, the function of the proxy is twofold. First, when a video user arrives, the proxy decides whether the user should be admitted to share the channel or not.

Second, for admitted video users, the proxy adapts the transmission data rates according to the varying channel conditions. The admission control algorithm only controls the newly arrived users and the rate adaptation algorithm do its best to satisfy the OQ constraints for admitted users. The rate adaptation algorithm accompanies the admission control algorithm by providing necessary information regarding the status of the admitted users (see Fig. 3.2).

I also assume the proxy operates in a time-slotted manner, where the duration of each slot is ΔT seconds. To maintain continuous playback, in each slot, the base station delivers ΔT seconds of video to each user. I assume the admission control and the rate adaptation actions are conducted at the beginning of every time slot. The set of video users in $\mathcal{U}^v(t)$ who are admitted to share the wireless channel is denoted by $\mathcal{U}^{av}(t)$. I define a binary variable $\mathbb{1}_u^{ad} \in \{0, 1\}$ to represent the admission control decision. If a video user u is admitted to share the channel, we have $\mathbb{1}_u^{ad} = 1$. Otherwise, we have $\mathbb{1}_u^{ad} = 0$. The set of admitted video users at t is thus $\mathcal{U}^{av}(t) = \{u \in \mathcal{U}^v(t) : \mathbb{1}_u^{ad} = 1\}$. I assume that an admission decision is made upon the arrival of a video user. Once admitted, the video user shares the channel until it leaves the network.

3.2.2 Channel Model and Rate-Quality Model

I consider the same TDMA channel model as in Chapter 2. Since, we only transmit video to admitted users, I redefine the rate vector as

$$\mathbf{r}(t) = (r_u(t) : u \in \mathcal{U}^{av}(t)).$$

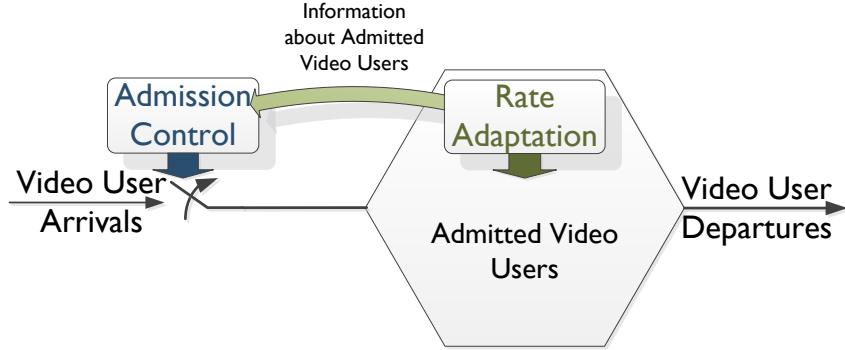


Figure 3.2: The proposed OQ-constrained video streaming system. The admission control algorithm decide to admit or reject a newly arrived video user to the network. The rate adaptation algorithm determines the video rate to each admitted video user.

The set of video rate vectors that can be supported by the wireless channel is given by

$$\mathcal{C}(t) = \{\mathbf{r} : c_t(\mathbf{r}(t)) \leq 0\}, \quad (3.2)$$

where the function $c_t(\mathbf{r}(t)) = \sum_{u \in \mathcal{U}^{\text{av}}(t)} \frac{r_u(t)}{P_u(t)} + \sum_{u \in \mathcal{U}^{\text{d}}(t)} \frac{R_u^d(t)}{P_u(t)} - 1$. Similar to Chapter 2, I confine the rate vector to the range of available video data rate, i.e.,

$$\mathbf{r}(t) \in \mathcal{R}(t), \quad \forall t. \quad (3.3)$$

I employ the following rate-quality model to predict $q_u^{\text{st}}(t)$ using the video data rate $r_u(t)$:

$$q_u^{\text{st}}(t) = a_u(t) \log(r_u(t)) + b_u(t), \quad (3.4)$$

where $q_u^{\text{st}}(t)$ is the predicted STSQ. The parameters $a_u(t)$ and $b_u(t)$ characterize the rate-quality parameters. I validated the rate-quality model in (3.4) on a video database of twenty-five different video contents [22]. The videos are first encoded and decoded with the widely used H.264 codec [12]. The STSQs of the decoded videos is then assessed using the RRED-algorithm [77]. I find that the mean square error of the video quality predicted by (3.4) is less than 1.5, which is visually negligible. Thus, in the following, I directly call $q_u^{\text{st}}(t)$ the STSQ. It should be noted that the model parameters $a_u(t)$ and $b_u(t)$ depends on the characteristics of the video content. Therefore, they vary across users and time slots.

3.2.3 Constraints on the Quality of Experience

As illustrated in Fig. 3.1, the strongest correlation is achieved at $x^* = 37$. We may interpret x^* as the users' quality expectation on the video quality, which is used by the users as a threshold in judging whether the video quality is acceptable or not. If $q_u^{\text{st}}(t)$ falls below x for a long while, $F^{(2)}(x^*)$ is large and the OQ is bad. Otherwise, $F^{(2)}(x^*)$ is small and the OQ is good. In our subjective study, all subjects view the video in an controlled environment and every subject views the videos on the same device. Broadly speaking, the value of x^* can be environment dependent. For example, viewers tend to have higher expectation for the videos shown on a laptop than the ones shown on a smart phone. Therefore, in a practical wireless network, x^* may be different across video users. Denoting by x_u^* the *quality expectation* of video user u , I

study the following two cases:

Case I: Quality expectation is unavailable. According to (3.1), the 2nd-order eCDF of user u is given by

$$F_u^{(2)}(x) = \frac{1}{T_u} \sum_{t=1}^{\infty} \max\{x - q_u^{st}(t), 0\} \mathbb{1}_u(t) \mathbb{1}_u^{av}, \quad (3.5)$$

where $\mathbb{1}_u(t) \mathbb{1}_u^{av}$ indicates whether user u is sharing the channel at slot t . If x_u^* is not known a priori, we may apply constraints on all x and for all users. In particular, I consider the following OQ constraints:

$$F_u^{(2)}(x) \leq h(x), \quad \forall x \in [0, 100], \quad \forall u \in \mathcal{U}^{av}, \quad (3.6)$$

where $h(x)$ is a function of x and $\mathcal{U}^{av} = \bigcup_{t=1}^{\infty} \mathcal{U}^{av}(t)$. In practice, we cannot apply constraints on all possible values of x in $[0, 100]$. Therefore, I consider a relaxed version of (3.6) as follows:

$$F_u^{(2)}(x^i) \leq h(x^i), \quad \forall x^i \in \mathcal{I}, \quad \forall u \in \mathcal{U}^{av}. \quad (3.7)$$

Here, \mathcal{I} is a finite set of points on $[0, 100]$. The following property of 2nd-order eCDFs shows that (3.7) can be sufficiently good in practice. Its proof is given in Appendix C.

Theorem 3.2.1. *Let $\bar{h}(x)$ be the piece-wise linear function that connects the points $\{(x^i, h(x^i)), \forall x^i \in \mathcal{I}\}$. The constraint (3.7) is equivalent to $F_u^{(2)}(x) \leq \bar{h}(x), \forall x \in [0, 100]$.*

Case II: Quality expectation is available. I also consider the case where users' phycological expectation x_u^* on video quality is somehow assessable. For

Table 3.2: Frequently used notations in Chapter 3

$\mathbb{1}_u^{\text{ad}}$	Indicator: u is admitted in the network
$\mathcal{U}^{\text{av}}(t)$	Admitted video users at time t ;
$F_u^{(2)}(x)$	The second order eCDF of user u
x_u^*	Quality expectation of user u

example, the users with different devices tend to have different expectations. Therefore, we may conduct subjective study on different devices and deduce the phycological expectation x_u^* on each type of device. Then, we can group the video users according to their respective x_u^* s and provide differentiated OQ guarantees. In particular, I define a finite set $\mathcal{G} = \{g_1, \dots, g_{|\mathcal{G}|}\}$ that represents different psychological expectations and assume that $x_u^* \in \mathcal{G}, \forall u \in \mathcal{U}^{\text{av}}$. Then, I consider the following constraints:

$$F_u^{(2)}(x_u^*) \leq h_u, \quad \forall u \in \mathcal{U}^{\text{av}}, \quad (3.8)$$

where, $x_u^* \in \mathcal{G}$ and h_u is a given OQ constraint.

In sum, the goal of the admission control algorithm and the rate adaptation algorithm is to maximize the number of users satisfying the constraints in (3.2), (3.3), (3.4), (3.7), and (3.8). In Section 3.3, I first introduce our rate adaptation algorithm and the admission control algorithm when x_u^* is unavailable. Then, in Section 3.4, I extend our rate adaptation and admission control algorithms to the case where x_u^* is available. Some of the key notations used in the rest of the chapter is given in Table 3.2.

3.3 Adaptive Transmission without Quality Expectation

In this section, I focus on the rate adaptation and admission control under the OQ constraint of (3.7). I first explain the rate-adaptation algorithm in Section 3.3.1. Then, I introduce a threshold-based admission control strategy in Section 3.3.2. An online algorithm for threshold optimization is given in Section 3.3.3. The simulation results are presented at the end of this section.

3.3.1 Rate Adaptation Algorithm

To clarify the design of our rate adaptation algorithm, I first present an off-line problem formulation, in which the future channel conditions and admission decisions are assumed to be known. Then, based on the analysis of this off-line problem, I propose our on-line rate adaptation algorithm.

If we consider a finite horizon T and assume the realization of channel conditions $\mathcal{C}(1), \dots, \mathcal{C}(T)$ are known, the rate adaptation $\mathbf{r}(t)$ should solve the following off-line feasibility problem:

$$\begin{aligned} & \underset{\mathbf{r}(t), \forall 1 \leq t \leq T}{\text{maximize}} && 1 \\ & \text{subject to:} && \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \quad \forall t \end{aligned} \tag{3.9a}$$

$$q_u^{\text{st}}(t) = a_u(t) \log(r_u(t)) + b_u(t), \quad \forall u \in \mathcal{U}^{\text{av}}(t), \quad \forall t \tag{3.9b}$$

$$F_u^{(2)}(x^i) \leq h(x^i), \quad \forall x^i \in \mathcal{I}, \quad \forall u \in \cup_{t=1}^T \mathcal{U}^{\text{av}}(t). \tag{3.9c}$$

By the definition in equation (3.5), the 2nd-order eCDF in constraint (3.9c) is determined by the whole process $q_u^{\text{st}}(1), \dots, q_u^{\text{st}}(T)$. Due to the constraints

(3.9a) and (3.9b), $q_u^{\text{st}}(t)$ depends on the rate region $\mathcal{C}(t)$. Therefore, the solution of (3.9) depends on the whole process $\mathcal{C}(1), \dots, \mathcal{C}(T)$. In practice, the future channel conditions are unavailable to rate controller. In the following, I transform problem (3.9) into a simpler form, which inspires our online rate adaptation algorithm.

We note that (3.9) is a convex problem. Hence, there exists a group of lagrangian multipliers $\lambda_u^i \geq 0$ for the constraints in (3.9c) such that the optimal rate vector of (3.9) can be obtained by solving the following problem [70]:

$$\begin{aligned} & \underset{\mathbf{r}(t), \forall 1 \leq t \leq T}{\text{minimize}} \quad \sum_{u \in \cup_{t=1}^T \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{J}} \lambda_u^i (\mathbf{F}_u^{(2)}(x^i) - h(x^i)) \\ & \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \forall t \\ & q_u^{\text{st}}(t) = a_u(t) \log(r_u(t)) + b_u(t), \forall u \in \mathcal{U}^{\text{av}}(t), \forall t \end{aligned} \quad (3.10)$$

For an admitted video user $u \in \cup_{t=1}^T \mathcal{U}^{\text{av}}(t)$, we have $\mathbb{1}_u^{\text{av}} = 1$. The term each $\mathbf{F}_u^{(2)}(x) - h(x^i)$ in (3.10) can be rewritten as:

$$\begin{aligned} & \mathbf{F}_u^{(2)}(x) - h(x^i) \\ &= \left[\frac{1}{\tau_u} \sum_{t=1}^T \max\{x^i - q_u^{\text{st}}(t), 0\} \mathbb{1}_u(t) \mathbb{1}_u^{\text{av}} - h(x^i) \right] \Big|_{\mathbb{1}_u^{\text{av}}=1} \\ &= \sum_{t=1}^T \left[\frac{1}{\tau_u} (\max\{x^i - q_u^{\text{st}}(t), 0\} - h(x^i)) \mathbb{1}_u(t) \right] \\ &= \sum_{t=1}^T s_u^i(t), \end{aligned} \quad (3.11)$$

where the function $s_u^i(t)$ is defined as:

$$s_u^i(t) := \frac{1}{\tau_u} (\max\{x^i - q_u^{\text{st}}(t), 0\} - h(x^i)) \mathbb{1}_u(t). \quad (3.12)$$

Substituting (3.11) to (3.10), we have

$$\begin{aligned}
& \sum_{u \in \cup_{t=1}^T \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{I}} \lambda_u^i (F_u^{(2)}(x^i) - h(x^i)) \\
&= \sum_{u \in \cup_{t=1}^T \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{I}} \lambda_u^i \left(\sum_{t=1}^T s_u^i(t) \right) \\
&= \sum_{t=1}^T \left(\sum_{u \in \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{I}} \lambda_u^i s_u^i(t) \right),
\end{aligned} \tag{3.13}$$

Therefore, the problem (3.10) becomes:

$$\begin{aligned}
& \underset{\mathbf{r}(t), \forall 1 \leq t \leq T}{\text{minimize}} \quad \sum_{t=1}^T \left(\sum_{u \in \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{I}} \lambda_u^i s_u^i(t) \right) \\
& \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \quad \forall t \\
& q_u^{\text{st}}(t) = a_u(t) \log(r_u(t)) + b_u(t), \quad \forall u \in \mathcal{U}^{\text{av}}(t), \forall t
\end{aligned}$$

Note that the problem (3.14) does not involve variables depending on the whole process of $q_u^{\text{st}}(t)$. Thus, (3.14) can be solved by minimize the weighted sum $\sum_{u \in \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{I}} \lambda_u^i s_u^i(t)$ in every slot. In other words, if we were be able to estimate the Lagrangian multiplier λ_u^i , then we can solve (3.9) by choosing rate vector $\mathbf{r}(t)$ as the solution of the following problem:

$$\begin{aligned}
& \underset{\mathbf{r}(t)}{\text{minimize}} \quad \sum_{u \in \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{I}} \lambda_u^i s_u^i(t) \\
& \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \\
& q_u^{\text{st}}(t) = a_u(t) \log(r_u(t)) + b_u(t), \quad \forall u \in \mathcal{U}^{\text{av}}(t),
\end{aligned} \tag{3.14}$$

We know that the Lagrangian multiplier λ_u^i indicates the difficulty in satisfying the constraint $F_u^{(2)}(x^i) \leq h(x^i)$ [15]. Inspired by the prior work in [78] and [56], I employ a virtual queue to capture such difficulty. Denote by t_u^{in}

and t_u^{out} the slot when user u joins and leaves the network, respectively. For each admitted user u and each $x^i \in \mathcal{I}_u$, I define the virtual queue $\Lambda_u^i(t)$ as:

$$\Lambda_u^i(t) = \begin{cases} 0 & t = t_u^{\text{in}} \\ \max\{\Lambda_u^i(t-1) + s_u^i(t), 0\} & t_u^{\text{in}} < t < t_u^{\text{out}}. \end{cases} \quad (3.15)$$

It is seen from (3.11) that, if the summation of $s_u^i(t)$ is large, the constraint $F_u^{(2)}(x^i) \leq h(x^i)$ is difficult to be satisfied. By (3.15), we have $\Lambda_u^i(t) \approx \Lambda_u^i(t-1) + s_u^i(t) \approx [\Lambda_u^i(t-2) + s_u^i(t-1)] + s_u^i(t) \approx \dots \approx \sum_{t=t_u^{\text{in}}}^t s_u^i(t)$. Therefore, the virtual queue captures the cumulative summation of $s_u^i(t)$ up to slot t . Hence, the virtual queue reflects the level of difficulty in satisfying $F_u^{(2)}(x^i) \leq h(x^i)$. Actually, for the special case where user set $\mathcal{U}^{\text{av}}(t)$ is fixed for all t , it can be proved that the virtual queue asymptotically approaches λ_u^i as $T \rightarrow \infty$ [78]. Hence, I replace the Lagrangian multipliers in (3.14) with virtual queue $\Lambda_u^i(t)$ and our online rate adaptation algorithm is summarized in Algorithm 4.

In every slot, the algorithm maximizes the sum of $\Lambda_u^i(t-1)s_u^i(t) = \Lambda_u^i(t-1)\frac{1}{T_u}(\max\{x^i - q_u^{\text{st}}(t), 0\} - h(x^i))$. Thus, the users with larger virtual $\Lambda_u^i(t-1)$ queues tend to be allocated with more network resource, which helps them to satisfy their OQ constraints. Also, the users with shorter video lengths T_u have higher priority in resource allocation. This is because the algorithm is designed to maximize the number of users who satisfy the OQ constraints. The users with shorter video length consume less network resource thus should have higher priority in resource allocation.

Next, I introduce the admission control algorithm.

Algorithm 4 Online algorithm for video data rate adaptation

- 1: **for** $t = 1 \rightarrow \infty$ **do**
- 2: Choose rate vector $\mathbf{r}(t)^*$ that solves the problem

$$\begin{aligned} & \underset{\mathbf{r}(t)}{\text{minimize}} \quad \sum_{u \in \mathcal{U}^{\text{av}}(t)} \sum_{x^i \in \mathcal{J}} \Lambda_u^i(t-1) s_u^i(t) \\ & \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \\ & \quad q_u^{\text{st}}(t) = a_u(t) \log(r_u(t)) + b_u(t), \quad \forall u \in \mathcal{U}^{\text{av}}(t), \end{aligned} \tag{3.16}$$

- 3: For $\forall u \in \mathcal{U}^{\text{av}}(t), \forall x^i \in \mathcal{J}$, update virtual queues with

$$\Lambda_u^i(t) = \max\{\Lambda_u^i(t-1) + s_u^i(t), 0\}.$$

- 4: **end for**
-

3.3.2 Admission Control Strategy

Since a video stream typically has high data rate and thus consumes a large amount of network resources, the arrival and departure of a single video user have significant impacts on other video users' OQ. The admission control algorithm is designed to identify the video users who tend to consume too much network resources and then block them. As have been discussed in Algorithm 4, the resource allocation in each slot is determined by the solution of the optimization problem (3.16). Therefore, we may estimate the OQ of a newly arrived user by solving (3.16) as if the user had already been admitted. Based on this idea, I propose a threshold-based admission control strategy, which is summarized in Algorithm 5. For each newly arrived video user u^{new} , I first estimate its video quality $q_{u^{\text{new}}}^{\text{st}}$ by solving the optimization problem (3.19), which is similar to the optimization problem (3.16). Then, I compare

$q_{u^{\text{new}}}^{\text{st}}$ with a threshold k . If $q_{u^{\text{new}}}^{\text{st}}$ is larger than k , it is admitted to the network.

Otherwise, it is rejected.

The optimization problem (3.19) is different from (3.16) in the following three aspects. First, to predict the long-term performance of the new video user u^{new} , I replaced the instantaneous rate-quality parameters $a_u(t)$ and $b_u(t)$ in (3.16) with the average rate-quality parameters \bar{a} and \bar{b} (see the second step in Algorithm 5):

$$\begin{aligned}\bar{a}_u &= \frac{1}{\tau_u} \sum_{j=1}^{\tau_u} a_u(j), \\ \bar{b}_u &= \frac{1}{\tau_u} \sum_{j=1}^{\tau_u} b_u(j),\end{aligned}\tag{3.20}$$

Second, I replace the instantaneous channel capacity $P_u(t)$ in (3.16) with the expected channel capacity (see the third step in Algorithm 5)

$$\bar{P}_u = \mathbb{E}[P_u(t)].\tag{3.21}$$

Third, for the newly arrived user, I initialize the virtual queue of the new user with the average virtual queues of all existing users (see the fourth step in Algorithm 5), i.e.,

$$\Lambda_{u^{\text{new}}}^i(t-1) \leftarrow \frac{1}{|\mathcal{U}^{\text{av}}(t-1)|} \sum_{u \in \mathcal{U}^{\text{av}}(t-1)} \Lambda_u^i(t-1), \quad \forall i \in \mathcal{I}.\tag{3.22}$$

In (3.20) and (3.21), the rate-quality parameter $a_u(\cdot)$ and $b_u(\cdot)$ and the expected average channel capacity $\mathbb{E}[P_u(t)]$ is assumed to be available. For the stored video streaming systems, the videos are pre-encoded. Thus, I assume the rate-quality characteristics for the whole video stream are known. Also, the expected channel capacity $\mathbb{E}[P_u(t)]$ can be estimated using the time-average

Algorithm 5 Admission control when x^* is not known.

Inputs: Threshold k , Admitted user $\mathcal{U}^{\text{av}}(t-1)$, new user u^{new}

- 1: Initialize video user set $\mathcal{U}^{\text{av+}} \leftarrow \mathcal{U}^{\text{av}}(t-1) \cap \{u^{\text{new}}\}$
- 2: Estimate rate-quality parameters for $\forall u \in \mathcal{U}^{\text{av+}}$:

$$\begin{aligned}\bar{a}_u &\leftarrow \sum_{j=1}^{\mathsf{T}_u} a_u(j)/\mathsf{T}_u, \\ \bar{b}_u &\leftarrow \sum_{j=1}^{\mathsf{T}_u} b_u(j)/\mathsf{T}_u.\end{aligned}\tag{3.17}$$

- 3: For all users $u \in \mathcal{U}^{\text{av+}}$, estimate channel capacity by $\bar{P}_u \leftarrow \mathbb{E}[\mathsf{P}_u(t)]$
- 4: Initialize virtual queue for user u^{new} :

$$\Lambda_{u^{\text{new}}}^i(t-1) \leftarrow \frac{1}{|\mathcal{U}^{\text{av}}(t-1)|} \sum_{u \in \mathcal{U}^{\text{av}}(t-1)} \Lambda_u^i(t-1), \quad \forall i \in \mathcal{I}\tag{3.18}$$

- 5: Define variables $\mathbf{r} = (r_u^v, u \in \mathcal{U}^{\text{av+}})$. Estimate the video quality of the new user by solving $q_{u^{\text{new}}}^{\text{st}}$ by solving the optimization problem

$$\begin{aligned}& \underset{\mathbf{r}}{\text{minimize}} \quad \sum_{u \in \mathcal{U}^{\text{av+}}} \sum_{x^i \in \mathcal{J}} \Lambda_u^i(t-1) s_u^i(t) \\ & \text{subject to: } \mathbf{r} \in \mathcal{C} \cap \mathcal{R}, \\ & \quad q_u^{\text{st}} = \bar{a}_u \log(r_u) + \bar{b}_u, \quad \forall u \in \mathcal{U}^{\text{av+}}\end{aligned}\tag{3.19}$$

where the set \mathcal{C} is given by

$$\left\{ \mathbf{r} : \sum_{u \in \mathcal{U}^{\text{av+}}} \frac{r_u}{P_u} \leq 1 - \sum_{u \in \mathcal{U}^{\text{d}}(t)} \mathbb{E} \left[\frac{\mathsf{R}_u(t)}{\mathsf{P}_u(t)} \right] \right\}$$

and the set \mathcal{R} is given by

$$\mathcal{R} = \Pi_{u \in \mathcal{U}^{\text{av+}}} \left[\frac{1}{\mathsf{T}_u} \sum_{j=1}^{\mathsf{T}_u} r_u^{\min}(j), \frac{1}{\mathsf{T}_u} \sum_{j=1}^{\mathsf{T}_u} r_u^{\max}(j) \right].$$

- 6: **if** $q_{u^{\text{new}}}^{\text{st}} > k$ **then**
 - 7: admit user u^{new} , i.e., $\mathcal{U}^{\text{av}}(t) \leftarrow \mathcal{U}^{\text{av}}(t-1) \cup \{u^{\text{new}}\}$
 - 8: **else**
 - 9: reject user u^{new} , i.e., $\mathcal{U}^{\text{av}}(t) \leftarrow \mathcal{U}^{\text{av}}(t-1) \cup \emptyset$
 - 10: **end if**
-

of the previous channel conditions. As have been discussed in Section 3.3.1, the virtual queue $\Lambda_u^i(t)$ captures the difficulty for an admitted video user to satisfy the OQ constraints. Thus, the users with large virtual queues tend to be allocated with more network resources. In other words, the virtual queues determines the priority in resource allocation. Since it is difficult to estimate the length of virtual queue before the new user is admitted, I simply initialize the virtual queue of the newly arrived user with the average virtual queues of all existing video users. In this way, I actually estimate the video quality $q_{u^{\text{new}}}^{\text{st}}$ when a average priority is assigned to the new user. Next, I introduce an algorithm that optimizes the threshold k in Algorithm 5.

3.3.3 Online Algorithm for Threshold Optimization

Denote by $g(k)$ the probability that a video user's OQ constraints are satisfied when the threshold is k . Also, denote by $e(k)$ the probability that a video user is admitted into the network but its OQ constraints are not satisfied. Our goal is to find out the best threshold k^* which maximizes $g(k)$. I have conducted extensive simulations under different channel condition and OQ constraints. From all the simulation results, I have made the following observations:

1. As the threshold k increases, $e(k)$ decreases and there always exists a threshold \hat{k} , such that

$$\begin{cases} e(k) > 0, & \text{if } k < \hat{k} \\ e(k) = 0, & \text{if } k \geq \hat{k} \end{cases} \quad (3.23)$$

2. The function $g(k)$ is always maximized at $k^* = \hat{k}$.

As an example, I simulate the admission control algorithm and plot the function $e(k)$ and $g(k)$ in Fig. 3.3 (see simulation settings in Section 3.3.4). It is seen that $g(k)$ is maximized at around $k^* = 60$, which also satisfies (3.23). Therefore, to find the optimal threshold k^* , it is sufficient to find a threshold \hat{k} that satisfy the (3.23).

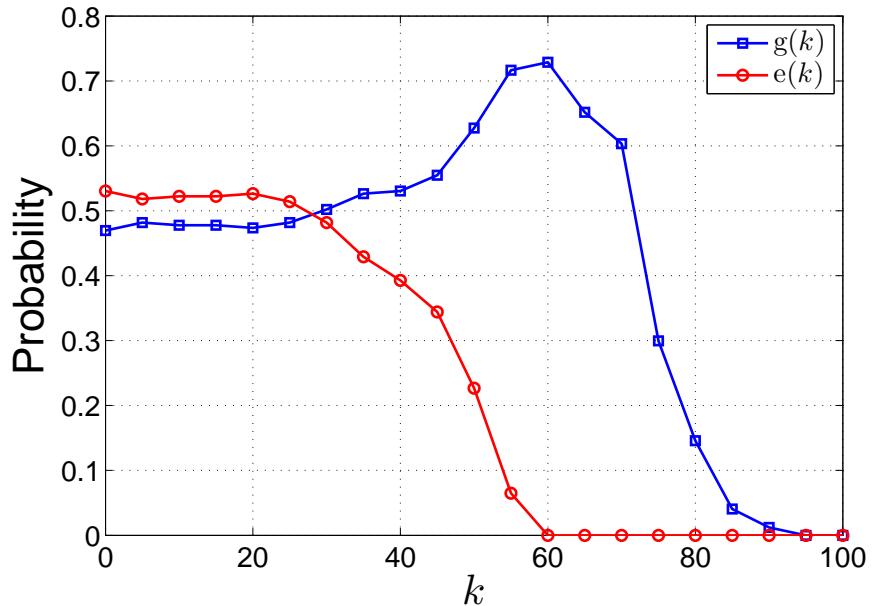


Figure 3.3: The simulated $g(k)$ and $e(k)$. $g(k)$ is the probability that a video user's OQ constraints are satisfied when the threshold is k . $e(k)$ is the probability that a video user is admitted into the network but its OQ constraints are not satisfied.

I propose an iterative algorithm that automatically adjust the threshold to k^* . The algorithm is summarized in Algorithm 6. In each iteration, the algorithm observes the 2nd-order eCDFs of L video users who have been admitted into the network since the end of the last iteration (see Step 3-12 in

Algorithm 6). Then the algorithm updates the threshold with

$$k^{n+1} = k^n + \epsilon^n y^n, \quad (3.24)$$

where k^n denote the admission control threshold in the n^{th} iteration (see Step 17 in Algorithm 6). The value $y^n \in \{-1, 1\}$ determines whether to increase or to decrease the threshold. The number $\epsilon^n > 0$ is the updating step-size. If the algorithm observes a video user whose 2nd-order eCDF violates the OQ constraints, then we know that $e(k^n) > 0$ and $k^n < k^*$. Therefore, the algorithm increases the threshold by setting $y^n = 1$. Otherwise, if all the L video users satisfy the constraints, the threshold is possibly larger than k^* . Thus the algorithm decreases the threshold by setting $y^n = -1$. The updating step-size ϵ^n is given by

$$\epsilon^n = \epsilon^0 / m, \quad (3.25)$$

where u counts the number of sign changes in the series of $\{y^1, \dots, y^n\}$ (see Step 13-15) and ϵ_0 is the initial step-size. Here, u is introduced to accelerate the convergence of the algorithm. The reason is explained as follows: If k^n is far from k^* , the sign of y^n does not change frequently and u increases slowly. Thus, the step-size ϵ^n stays large and k^n is moved towards k^* quickly. When k^n is moved to a small neighborhood of k^* , the sign of y^n changes frequently and thus u increases rapidly. Therefore, the step-size ϵ^n decreases to zero quickly, which makes the k^n converge.

In the following, I analyze the convergence of Algorithm 6. Based on our observations from the simulations, I make the following assumption:

Algorithm 6 Online algorithm for optimal threshold searching when x^* is not known.

Inputs: $L = 100$, $k^0 = 0$, $n = 0$, $m = 1$, $\epsilon^0 = 10$, and $y^0 = 1$

```

1: while 1 do
2:    $n \leftarrow n + 1$ ;  $w \leftarrow 0$ ;  $u = 1$ 
3:   while  $w = 0$  and  $u \leq L$  do
4:     if  $\exists i \in \mathcal{I}$  such that  $F_u^{(2)}(x^i) > h(x^i)$  then
5:        $w \leftarrow 1$ 
6:     end if
7:   end while
8:   if  $w \neq 0$  then
9:      $y^n \leftarrow 1$ 
10:   else
11:      $y^n \leftarrow -1$ 
12:   end if
13:   if  $y^n \neq y^{n-1}$  then
14:      $m \leftarrow m + 1$ 
15:   end if
16:    $\epsilon^n = \epsilon^0/m$ ;
17:   Update threshold with

```

$$k^{n+1} = k^n + \epsilon^n y^n \quad (3.26)$$

18: **end while**

Assumption 1. The function $e(k)$ is a continuous function of k , which satisfies (3.23) with $\hat{k} = k^*$. The function $e(k)$ is strictly decreasing on $[0, k^*]$.

I define $p^L(k)$ as the probability that all the L admitted video users in a iteration satisfy the OQ constraints. Since increasing the threshold k would block more users and thus reserve more network resources to the admitted users, I assume that $p^L(k)$ is an continuously increasing function of k . Also, according to our Assumption 1, when $k > k^*$, all the admitted users satisfy the

OQ constraints and thus $p^L(k) = 1$. Thus, I have the following assumption about $p^L(k)$:

Assumption 2. The function $p^L(k)$ is a continuous and increasing function of k . For $\forall k > k^*$, we have $p^L(k) = 1$. Furthermore, I assume that there exists a constant $M > 0$ such that, $|p^L(k) - p^L(k')| \geq M|k - k'|$ for all $k' < k$ and $p^L(k) < 1$.

The following theorem assures that, if L is sufficiently large, k^n converges to an arbitrarily small neighborhood of k^* as $n \rightarrow \infty$. Its proof is given in Appendix D.

Theorem 3.3.1. *Let $\delta > 0$ be an arbitrarily small number. If Assumption 1 and Assumption 2 are satisfied and $L \geq \frac{-\log 2}{\log(1-e(k^*-\delta))}$, then k^n converges as $n \rightarrow \infty$ and $\lim_{n \rightarrow \infty} k^n \in [k^* - \delta, k^*]$.*

3.3.4 Simulation Results

I employ the same simulation setting as in Section 2.5.3. I apply constraints on the 2nd-order eCDF on the set of $\mathcal{I} = \{30, 40, 50, 60, 70\}$. Correspondingly, for the x^i being 30, 40, 50, 60, and 70, I let the constraints $h(x^i)$ be 0.7, 1.0, 3.0, 7.0 and 15.0, respectively.

I first set the scaling parameter $\alpha = 12$ and simulate Algorithm 4 without admission control. The simulation is terminated until 50 users have arrived and departed the network. I plot the 2nd-order eCDFs of the video users in Fig. 3.4(a). It is seen that, using Algorithm 4, the 2nd-order eCDFs of

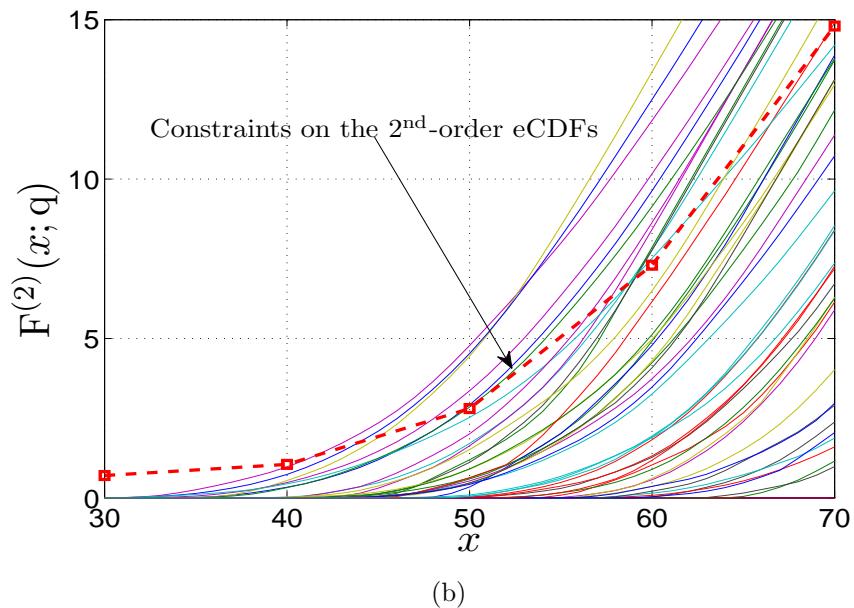
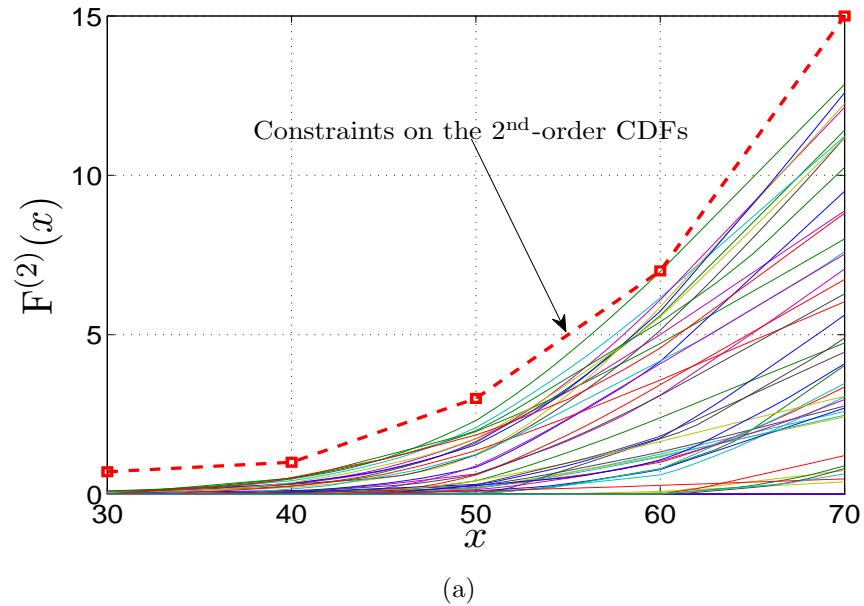


Figure 3.4: (a) The 2nd-order eCDFs of the video users when $\alpha = 12$ and Algorithm 4 is employed for rate adaptation. (b) The 2nd-order eCDFs of the video users when $\alpha = 12$ and rate vector is adapted to maximize $\sum_{u \in \mathcal{U}^{\text{av}}(t)} q_u^{\text{st}}(t)$.

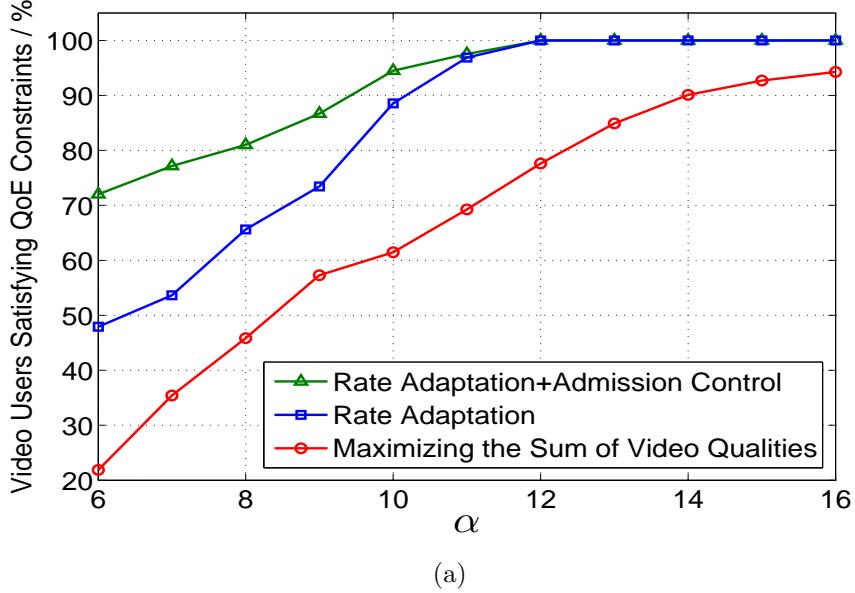
the video users all satisfy the constraints. In comparison, if we adapt the rate vector to maximize $\sum_{u \in \mathcal{U}^{\text{av}}(t)} q_u^{\text{st}}(t)$ in each slot, the OQ constraints is violated by many users (see Fig. 3.4(b)).

Then, I evaluate the performance of our rate adaptation and admission control algorithms under different channel scaling parameters. The percentage of users who satisfy the OQ constraints among all video users that arrive to the network are shown in Fig. 3.5(a). It is seen that The percentage of video users who satisfies the OQ constraints is improved by 17.8% by Algorithm 4 if no admission control is applied. The admission control policy further improves the performance especially when the channel condition is not sufficiently good.

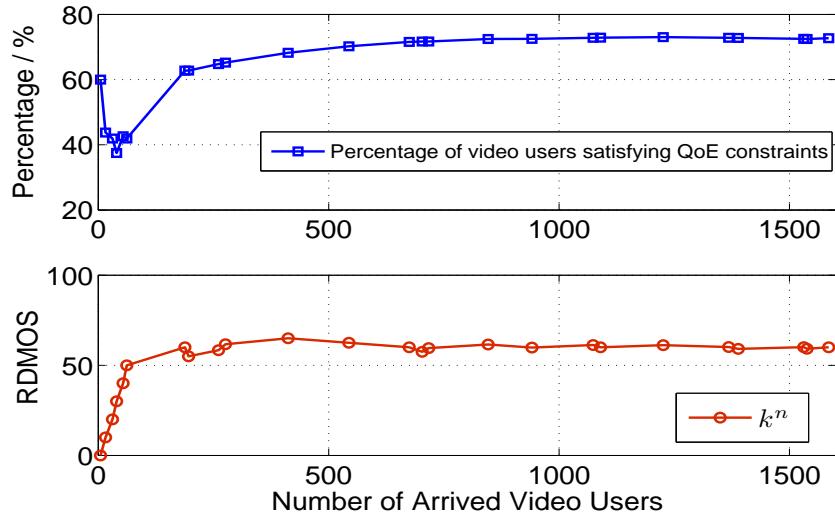
To show the performance of Algorithm 5, in Fig. 3.5(b), I plot the threshold k^n in every iteration of the online threshold optimization algorithm at $\alpha = 6$. Recall that the optimal threshold is $k^* = 60$ (see Fig. 3.3). It is seen in 3.5(b) that k^n converges to k^* after 200 video users arrivals. Since I have assumed that the average arrival rate of video users is 1/20 user/seconds. Thus, 200 video user arrivals cost about 200×20 seconds = 1.1 hours. Since our goal is to optimize the performance of the network in the long run, this convergence speed is acceptable.

3.4 Adaptive Transmission with Quality Expectation

In this section, I extend the rate adaptation algorithm, the admission control algorithm and the threshold optimization algorithm to the case where x_u^* of each user are available. I first explain the extended rate adaptation algorithm



(a)



(b)

Figure 3.5: (a) Simulation results of the proposed admission control policy under different channel scaling parameters. Each data point on the figure is obtained by simulating 2000 video user arrivals. The Y-axis indicates the percentage of users who satisfy the QoE constraints among all video users that arrive to the network. (b) The convergence of Algorithm 6 when the scaling parameter is $\alpha = 6$.

and the admission control strategy. Then, I evaluate their performance via numerical simulations.

3.4.1 Rate Adaptation and Admission Control

In Section 3.2.3, I have defined the finite set $\mathcal{G} = \{g_1, \dots, g_{|\mathcal{G}|}\}$ to represents different x_u^* among video users. In the following, I call the users with $x_u^* = g_j \in \mathcal{G}$ the Type- j users. Each Type- j video user only need to satisfy one OQ constraint, i.e.,

$$F_u^{(2)}(g_j) \leq h_j. \quad (3.27)$$

Thus, I extend the rate adaptation method in Algorithm 4 by maintaining one virtual queue for each user. In particular, for a Type- j user, I define its virtual queue as:

$$\Lambda_u(t) = \begin{cases} 0 & t = t_u^{\text{in}} \\ \max\{\Lambda_u(t-1) + s_u(t), 0\} & t_u^{\text{in}} < t < t_u^{\text{out}}, \end{cases} \quad (3.28)$$

where $s_u(t) := \frac{1}{T_u} (\max\{g_j - q_u^{\text{st}}(t), 0\} - h_j) \mathbb{1}_u(t)$. In each slot, the rate vector $\mathbf{r}(t)$ is adapted by solving:

$$\begin{aligned} & \underset{\mathbf{r}(t)}{\text{minimize}} \quad \sum_{u \in \mathcal{U}^{\text{av}}(t)} \Lambda_u(t-1) s_u(t) \\ & \text{subject to: } \mathbf{r}(t) \in \mathcal{C}(t) \cap \mathcal{R}(t), \end{aligned} \quad (3.29)$$

$$q_u^{\text{st}}(t) = a_u(t) \log(r(t)) + b_u(t), \quad \forall u \in \mathcal{U}^{\text{av}}(t).$$

For admission control, I extend Algorithm 5 by applying different thresholds to different types of users. In particular, for a newly arrived Type- j video

user u^{new} , I initializes the the virtual queue by averaging virtual queues of all existing video users i.e.,

$$v_{u^{\text{new}}}(t-1) \leftarrow \frac{1}{|\mathcal{U}^{\text{av}}(t-1)|} \sum_{u \in \mathcal{U}^{\text{av}}(t-1)} \Lambda_u(t-1). \quad (3.30)$$

Then, the algorithm estimates the video quality q_u^{st} of the new the user by solving

$$\begin{aligned} & \underset{\mathbf{r}}{\text{minimize}} \quad \sum_{u \in \mathcal{U}^{\text{av}}(t) \cup \{u^{\text{new}}\}} \Lambda_u(t-1) s_u(t) \\ & \text{subject to: } \mathbf{r} \in \mathcal{C} \cap \mathcal{R}, \\ & q_u^{\text{st}} = \bar{a} \log(r_u) + \bar{b}_u, \quad \forall u \in \mathcal{U}^{\text{av}}(t) \cup \{u^{\text{new}}\}. \end{aligned} \quad (3.31)$$

Finally, q_u^{st} is compared with a threshold k_j to make the admission decision.

Next, I discuss how to optimize the thresholds.

3.4.2 Threshold Optimization Algorithm

Denote by $\mathbf{k} = (k_1, \dots, k_{|\mathcal{G}|})$ the vector of thresholds for all types of video users. I define $g(\mathbf{k})$ as the probability that a video user satisfies the OQ constraints when the threshold vector is \mathbf{k} . Also, I define $e_j(\mathbf{k})$ as the probability that a Type- j video user's OQ constraint is not satisfied. To find out the optimal threshold vector \mathbf{k} that maximized $g(\mathbf{k})$, I have run simulations under different channel conditions and OQ constraints. I find that a threshold vector $\mathbf{k}^* = (k_1^*, \dots, k_{|\mathcal{G}|}^*)$ maximizes $g(\mathbf{k})$ if and only if:

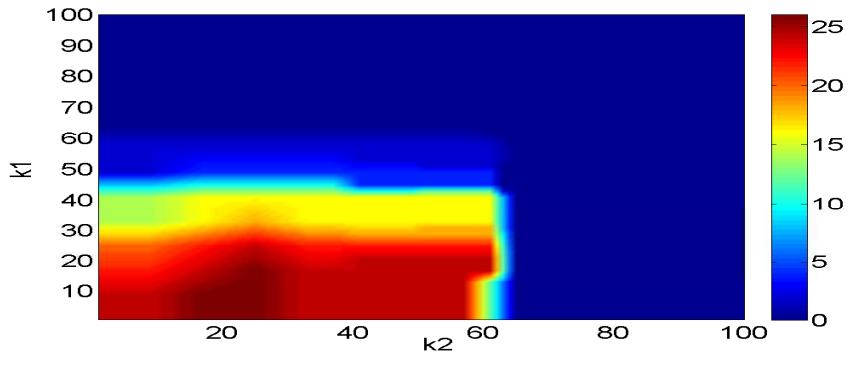
$$\begin{cases} e_j(\mathbf{k}) > 0, & \forall \mathbf{k} \prec \mathbf{k}^* \\ e_j(\mathbf{k}) = 0, & \forall \mathbf{k} \succeq \mathbf{k}^*. \end{cases}, \quad \forall 1 \leq j \leq |\mathcal{G}|. \quad (3.32)$$

Here, the partial order $\mathbf{k} \prec \mathbf{k}^*$ indicates that $\mathbf{k} \neq \mathbf{k}^*$ and $k_i \leq k_i^*, \forall i$ while $\mathbf{k} \succeq \mathbf{k}^*$ indicates that $k_i \geq k_i^*, \forall i$. The condition in (3.32) means that, if \mathbf{k}^* is a optimal threshold vector and we increase all entries of \mathbf{k}^* , the OQ constraints of all the admitted users can still be satisfied. Conversely, if we decrease all the entries of \mathbf{k}^* , the OQ constraints of all types of users will be violated with a non-zero probability. To illustrate this observation, I considered two types of video users who arrive to the network with equal probability and simulate the function $e_1(\mathbf{k})$, $e_2(\mathbf{k})$, and $g_1(\mathbf{k})$ using the same setting as in Section 3.3.4. From Fig. 3.6(c), it is seen that the function $g(\mathbf{k})$ is maximized on a narrow rectangular area where $\mathbf{k}^* = (k_1^*, k_2^*) \in [0, 42] \times [63, 64]$. From Fig. 3.6(a) and Fig. 3.6(b), it can be seen that the points in $[0, 42] \times [63, 64]$ satisfies the condition (3.32).

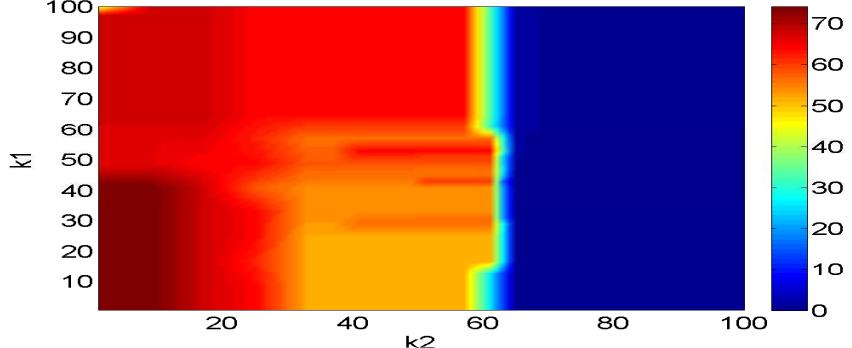
I proposed the iterative Algorithm 7 to find out the threshold vector \mathbf{k}^* satisfying (3.32). Denote by \mathbf{k}^n the threshold vector in the n^{th} iteration, the algorithm observes the 2nd-order eCDFs of L admitted video users and updates the threshold vector by

$$\mathbf{k}^{n+1} = \mathbf{k}^n + \boldsymbol{\epsilon}^n \circ \mathbf{y}^n. \quad (3.33)$$

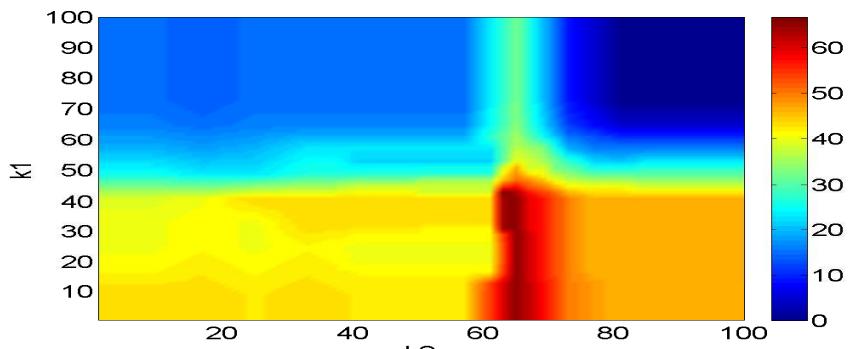
In (3.33), $\mathbf{y}^n = (y_1^n, \dots, y_{|\mathcal{G}|}^n)$ is a $|\mathcal{G}|$ -dimensional vector where $y_j^n \in \{-1, 1\}$ is the updating direction for k_j . The vector $\boldsymbol{\epsilon} = (\epsilon_1^n, \dots, \epsilon_{|\mathcal{G}|}^n)$ is the corresponding updating step-size (see Step 2-9). Among the L video users, if a Type- j video user's OQ constraint is not satisfied, the algorithm sets $y_j^n = 1$. Otherwise, if all the Type- j video users' OQ constraints are satisfied, the algorithm



(a) $e_1(\mathbf{k})$



(b) $e_2(\mathbf{k})$



(c) $g(\mathbf{k})$

Figure 3.6: (a) The probability of admitted Type-1 video users whose OQ constraints are violated. (b) The probability of admitted Type-2 video users whose OQ constraints are violated. (c) The probability of video users whose OQ constraints are satisfied. All results are shown in percentage.

sets $y_j^n = -1$ (see Step 11-15). The step-size ϵ_j^n is given by $\epsilon_j^n = \epsilon_j^n/m_j$, where m_j counts the sign changes in $\{y_j^1, \dots, y_j^n\}$ (see Step 16-19). Next, I show the performance of our rate adaptation algorithm and the admission control strategy via numerical simulations.

Algorithm 7 Online algorithm for optimal threshold searching when x^* is available.

Inputs: $L = 100$, $\mathbf{k}^0 = \mathbf{0}$, $\mathbf{y}^0 = \mathbf{1}$, $\mathbf{m} = \mathbf{1}$, $n = 0$, and $\epsilon^0 = 10$

```

1: while 1 do
2:    $n \leftarrow n + 1$ ;  $\mathbf{w} \leftarrow \mathbf{0}$ ;  $u = 1$ 
3:   while  $\mathbf{w} \neq \mathbf{1}$  and  $u \leq L$  do
4:     for  $j = 1 \rightarrow |\mathcal{G}|$  do
5:       if  $x_u^* = g_j$  and  $F_u^{(2)}(g_j) > h_j$  then
6:          $w_j \leftarrow 1$ 
7:       end if
8:     end for
9:   end while
10:  for  $j = 1 \rightarrow |\mathcal{G}|$  do
11:    if  $w_j \neq 0$  then
12:       $y_j^n \leftarrow 1$ 
13:    else
14:       $y_j^n \leftarrow -1$ 
15:    end if
16:    if  $y_j^n \neq y_j^{n-1}$  then
17:       $m_j \leftarrow m_j + 1$ 
18:    end if
19:     $\epsilon_j^n = \epsilon^0/m_j$ ;
20:  end for
21:  Update threshold with

```

$$\mathbf{k}^{n+1} = \mathbf{k}^n + \boldsymbol{\epsilon}^n \circ \mathbf{y}^n \quad (3.34)$$

22: **end while**

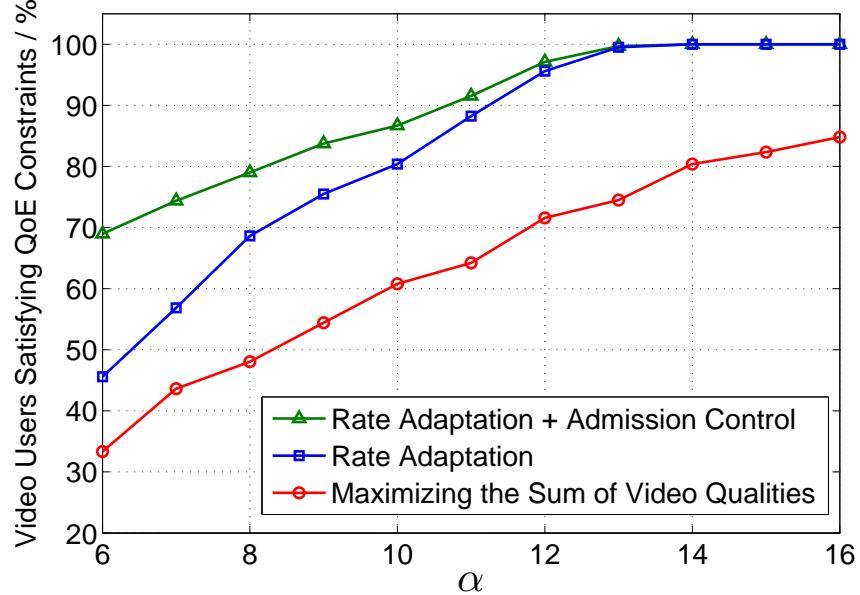
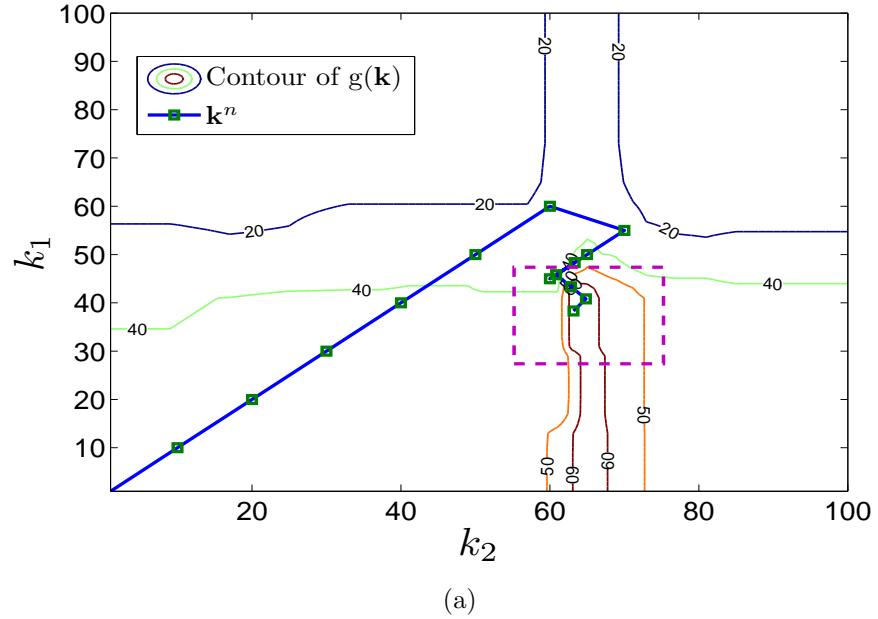


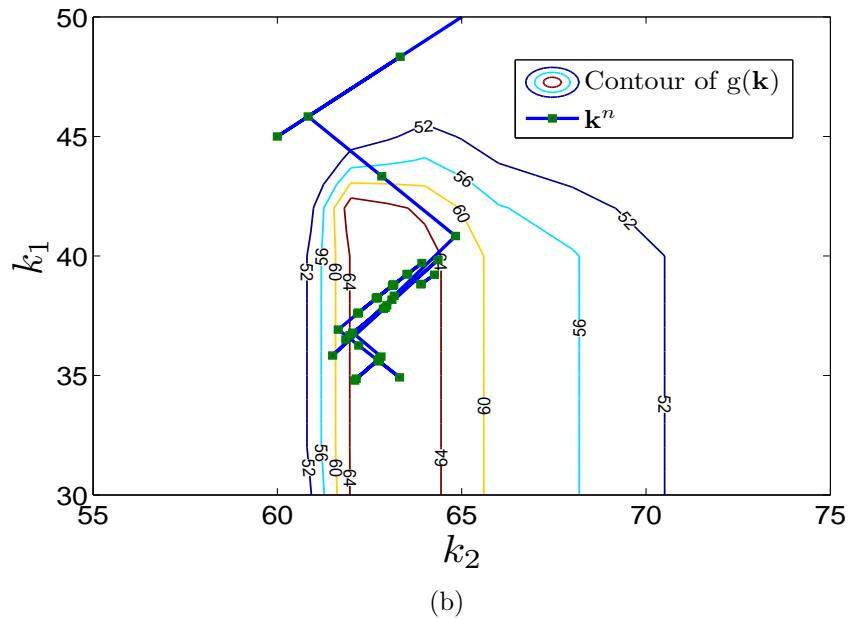
Figure 3.7: Simulation results of the proposed admission control policy under different channel scaling parameters.

3.4.3 Simulation Results

In our simulations, I assume that there are two types of video users. Both types of video users arrives as a Poisson process with the arrival rate of $1/40$ users/second. I assume the Type-1 users have $x_u^* = 40$ while the Type-2 users have $x_u^* = 60$. Furthermore, I set $h_1 = h_2 = 1$. In Fig. 3.7, I plot the percentage of video users whose video quality satisfy the QoE constraint (3.8). It can be seen that, for all tested channel scaling parameters, our rate adaptation algorithm outperforms the benchmark algorithm. The percentage of video users who satisfies the QoE constraints is improved by 19.3% even if the admission control algorithm is not applied. It is also seen that our admission



(a)



(b)

Figure 3.8: The updated threshold vector \mathbf{k}^n 's of Algorithm 7 in the first 50 iterations are shown in (a). The dashed box is shown in (b) to illustrate more details. The contours of $g(\mathbf{k})$ is also shown on the figure for reference.

control strategy can further improve the percentage of video users satisfying the QoE constraints.

In Fig. 3.8, I plot threshold vectors \mathbf{k}^n in Algorithm 7 when the channel scaling parameter is fixed as $\alpha = 6$. It is seen that the threshold vector converges quickly to the area where $g(\mathbf{k})$ is maximized.

3.5 Summary

In this paper, we consider a new QoE metric that is based on the second-order cumulative distribution function of the time-varying video qualities. Using on this metric, we propose a rate adaptation algorithm to maximize the percentage of video users who satisfy the QoE constraints on their cumulative distribution functions. Furthermore, we devise a threshold-based admission control strategy that blocks the video users whose QoE constraints can hardly be satisfied.

Chapter 4

Adaptive Transmission of Stored Scalable Videos

In Chapter 2 and Chapter 3, I have investigated adaptive transmission algorithms for bit-switching videos. This chapter focuses on SVC-based adaptive video transmission algorithms. I propose two adaptive transmission algorithms that seek to optimize the STSQ of scalably coded videos transmitted over wireless channels. The first adaptive transmission algorithm is derived from a Markov Decision Process (MDP) formulation. I model the dynamics of the channel as a Markov chain and reduce the problem of adaptive video transmission to a tractable Markov decision problem over a finite state space. Based on the MDP formulation, a near-optimal transmission policy is computed that maximizes an objective proxy of STSQ, the time-average *Multi-Scale Structural SIMilarity* (MS-SSIM) index. Using insights taken from the development of the optimal MDP-based transmission policy, the second proposed algorithm is an online method that only requires easily measurable knowledge of the channel dynamics, and is thus viable in practice. Simulation results show that the performance of both algorithms is close to a performance upper bound.

4.1 Introduction

As discussed in Section 1.2, an SVC video encoder produces a layered video stream that contains a base layer and several enhancement layers. If the throughput is low, the transmitter can choose to transmit the base layer only, which provides a moderate, but acceptable, degree of visual quality at the receiver. If the channel conditions improve, the transmitter can transmit one, or more, enhancement layers to further improve the visual quality. Conceptually, SVC provides a means to adapt the data rate for wireless video transmission. The wireless transmitter can adapt the data rate by selectively scheduling video data associated with various layers for transmission rather than transcoding the video sequence into a different rate.

Designing scalable video scheduling algorithms for wireless channels is a complex task. The scheduling policy depends, not only on the channel conditions, but also, on the receiver buffer state. For example, if the receiver has successfully buffered base layer data over many frames, the scheduler could choose to transmit some enhancement layer data to improve the video quality even if the throughput is low. At any time, the scheduling decision will determine the receiver buffer state which, in turn, affects the future scheduling decisions. Therefore, adaptive video data scheduling is a sequential decision problem. The most natural way to address such problems is to model the dynamics of the channel as a finite state Markov chain and to employ a Markov decision process (MDP)-based formulation to study scheduling methods. Directly determining an optimal scheduling policy using an MDP formulation

is not possible, however, because the system state space is infinitely large (see Section. 4.3.1). Moreover, in a practical wireless network, a model for the dynamics of the channel states is not typically available, which limits the applicability of this approach.

The objective of this chapter is to leverage the MDP framework to develop practical scheduling algorithms and optimize the receiver STSQ for scalable video transmission over wireless channels. First, I propose a tractable MDP formulation based on a reasonable approximation of the state space. Near optimal scheduling policies can be derived from this MDP formulation. Second, I propose a scheduling algorithm that substantially simplifies the MDP-based scheduling policy as it requires only limited information regarding the channel state dynamics. Third, I prove an upper bound on the achievable STSQ of all possible scheduling algorithms. Finally, I provide simulation results that show, under different channel conditions, the performance of the proposed scheduling algorithms is indeed very close to the upper bound.

Adaptive video data scheduling is an important topic of research [18, 26, 27, 31, 33, 66, 90]. In [66], adaptive video transmission over a packet erasure channel was studied by modeling the buffer state as a controlled Markov chain. In [27], an MDP-based scheduling algorithm was proposed for video transmission over packet loss networks. This work was further extended for wireless video streaming in [18], where the wireless channel was modeled as a binary symmetric channel. This channel model can only be justified for fast fading channels, where the coherence time is much less than the delay constraint. In

that case, interleaving can be applied without violating the delay constraint, and the channel will appear as an i.i.d channel. For slow fading channels such as those considered here, the bit error rate cannot be modeled as a constant. In [90], [31], and [33], reinforcement learning frameworks were proposed for wireless video transmission. Their proposed algorithms were based on MDP using a discounted-reward maximization formulation. The transmitter learns the characteristics of the channel and the video sequence during the transmission process. The scheduling policy is updated according to the learned characteristics. In our previous work [26], an infinite-horizon average-reward maximization MDP formulation was proposed. The channel and source characteristics, unlike in this dissertation, were assumed to be known.

The rest of the chapter is organized as follows. The assumptions I make about the video codec and the rate-quality model are described in Section 4.2. In Section 4.3, the MDP formulation and the performance upper bound are proposed. A near-optimal on-line scheduling algorithm is introduced and validated by simulations in Section 4.4.

4.2 System Model

In this section, I describe the wireless video system to be considered. Then, I present our video codec configuration and introduce the rate-distortion model.

4.2.1 System Overview

I consider a time-slotted system that transmits scalable videos over a slow fading wireless channel. The video sequence is encoded with a quality-scalable video encoder and is stored in a video server. The video server transmits video data to a mobile user via a wireless transmitter. In each slot, the server sends some video data upon request of a scheduler at the wireless transmitter. This data is packetized at the wireless transmitter for physical layer transmission. The scheduler operates according to a policy that maps the channel and receiver buffer state to the scheduling action (see Fig. 4.1).

I assume that the link between the video server and the wireless transmitter is not the bottleneck for transmission to the mobile. Thus, from the perspective of the wireless transmitter, the whole video sequence is available for transmission. I also assume that the physical layer channel state information is available at the transmitter and that the modulation and coding scheme (MCS) is determined by a given physical layer link-adaptation policy.

4.2.2 Video Codec Configuration

I assume that the video sequence is encoded by an H.264/SVC-compatible scalable video encoder. The duration of each frame ΔT is called a frame slot. The video frames are uniformly partitioned into Groups of Pictures (GOPs). Every GOP has F^{GOP} frames. The first frame in a GOP is an I frame while the other frames are P frames. Every frame is encoded into L layers. The first layer is the base layer; the other layers are enhancement layers. Every

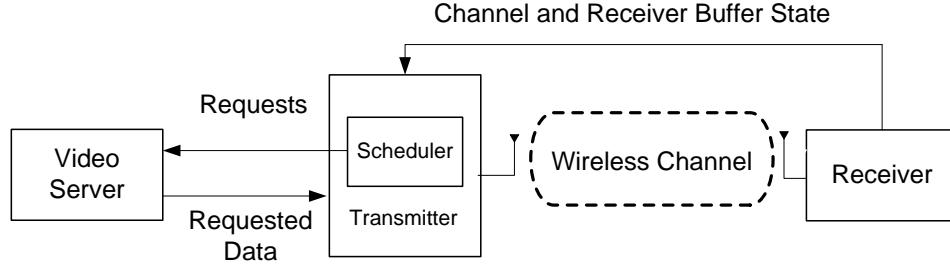


Figure 4.1: Dynamic scheduling system for wireless video transmission. The scheduler is located at the transmitter. Scheduling decisions are based on the channel and receiver buffer state fed back by the receiver.

enhancement layer of a frame is predictively encoded using the lower layers of the frame. The base layer of a P frame is predictively encoded using the base layer of its preceding frame. The base layer of an I frame is encoded independently (see Fig. 4.2).

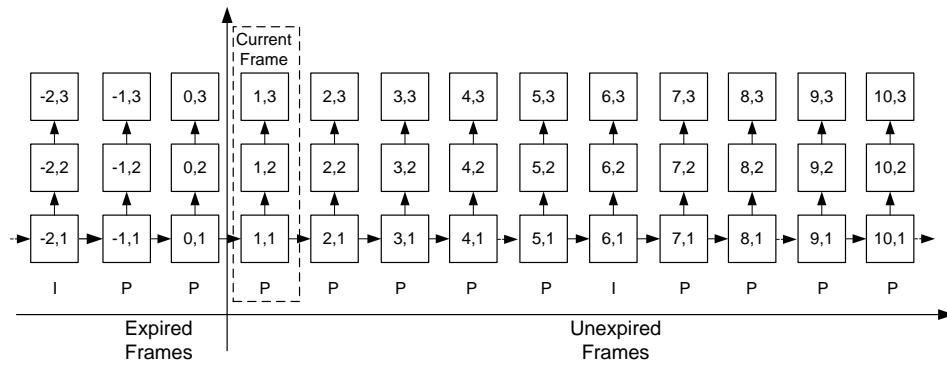


Figure 4.2: Encoder prediction structure when $L = 3$. The prediction order is indicated by arrows. The data unit index (f, ℓ) is also shown on each data unit.

Each frame has a playout deadline at the receiver. In the following, frames whose deadlines have expired are called expired frames, otherwise they

are said to be active frames. The first active frame is called the “current frame”. At any point in time, frames are indexed relative to the current frame as shown in Fig. 4.2. The video data in the ℓ^{th} layer of the f^{th} frame is called the $(f, \ell)^{\text{th}}$ **video data unit**.

I adopt the prediction structure in Fig. 4.2 rather than the “Hierarchical B” structure because no structural delay is introduced [71]. In the “Hierarchical B” prediction structure, the encoding order differs from the display order, thus the transmission of a frame must be delayed until all necessary predictors are received. Because the enhancement layers are used to predict other frames in the “Hierarchical B” structure, dropped enhancement layers can give rise to error propagation and unpredictable visual quality degradation. At the possible cost of lower compression efficiency, the prediction structure that we use eliminates error propagation arising from *enhancement* layer losses, since there is no inter-frame prediction among enhancement layers.

4.2.3 Rate-Quality Model

Let z_f be the amount of received data for the f^{th} frame. The rate-quality function $q_f(z_f)$ captures the quality of the frame when it is decoded. Let $\omega_{(f,\ell)}$ be the amount of data in the $(f, \ell)^{\text{th}}$ data unit and $q_{(f,\ell)}$ be the visual quality *increment* if the ℓ^{th} layer is correctly received, given all its predictors have also been received. As illustrated in Fig. 4.3, since a data unit can be decoded only when all its associated data has been received, $q_f(z_f)$ is a piecewise constant and right-continuous function with jumps at $z_f = \sum_{\ell=1}^m \omega_{(f,\ell)}$, $m = 1, 2, \dots, L$.

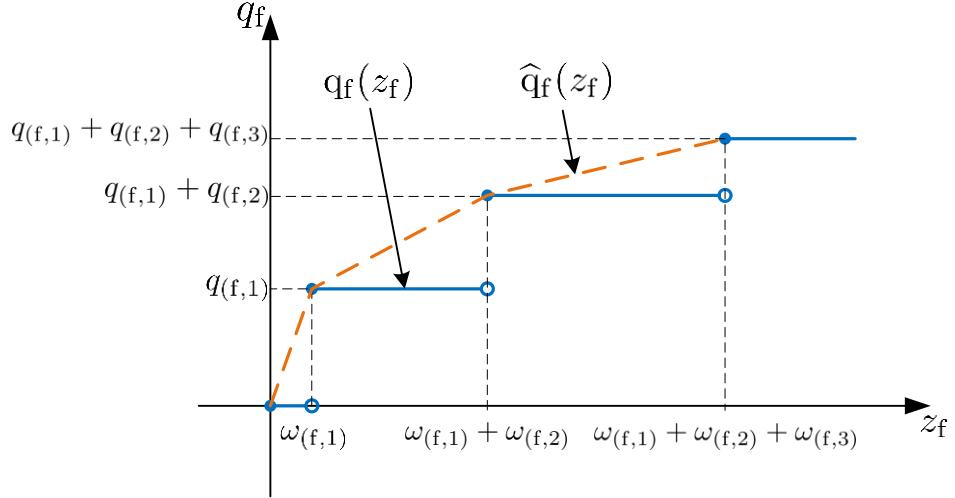


Figure 4.3: An illustration of the rate-quality function $q_f(z_f)$ for the f^{th} frame. The rate-quality function $q_f(z_f)$ is piecewise constant and right-continuous (solid). Its concave envelope $\hat{q}_f(z_f)$ is also shown (dotted).

Thus $q_{(f,\ell)}$ and $\omega_{(f,\ell)}$ characterize $q_f(z_f)$.

In a real video sequence, for a given layer ℓ , the rate-quality characteristics $q_{(f,\ell)}$ and $\omega_{(f,\ell)}$ vary across frames. In this chapter, I approximate $q_f(z_f)$ using a simple model. Let n be the number of frames in a video sequence. Since for each layer, $q_{(f,\ell)}$ is almost the same for all frames, I use $q^\ell = 1/n \sum_{f=1}^n q_{(f,\ell)}$ as an estimate for visual quality *increment* if the ℓ^{th} layer is correctly received. I also assume that $\omega_{(f,\ell)}$ is almost the same for I frames and P frames, respectively. Let ω_ℓ^I and ω_ℓ^P be the average values of $\omega_{(f,\ell)}$ across the video for I frames and P frames, respectively. Our rate-quality models $q^I(z_f)$ and $q^P(z_f)$ for I frames and P frames are respectively constructed as piecewise constant functions with jumps at $z_f = \sum_{\ell=1}^m \omega_\ell^I$ and $z_f = \sum_{\ell=1}^m \omega_\ell^P$,

$m = 1, 2, \dots, L$.

Conventional image quality measures such as the PSNR reflect absolute signal fidelity but without accounting for perceptual visual quality. Recently, a variety of models that accurately predict STSQ have been proposed [20, 54, 65, 85, 86]. In our formulation, I adopt the MS-SSIM index as the visual quality measure [85], since it has been shown to correlate well with perceptual visual quality and it is of reasonable computational complexity[74].

The MS-SSIM index of a video sequence ranges from 0 to 1. The larger the index, the better the quality. In our rate-quality model, the marginal quality increment q^ℓ is measured using the MS-SSIM index. Larger values of q^ℓ mean a larger marginal improvement can be achieved by transmitting the ℓ^{th} layer data units.

4.2.4 Streaming Setup

I focus on scheduling for a slow fading channel. By slow fading, I mean that the coherence time of the channel is less than the duration of a GOP and larger than the duration of a frame. Assuming the mobile users are moving in a 1.5m/s walking speed and the carrier frequency is 2GHz, the Doppler spread is about 10Hz. The coherence time is about 100ms. A typical GOP duration is about 1 second and a frame slot is about 30ms. Hence, for pedestrian video users, wireless channels are slow fading.

As the channel state is stable during each frame slot, the scheduling decision is made on a frame-by-frame basis. At the beginning of each frame

Table 4.1: Frequently Used Notations in Chapter 4.

Notations	Descriptions
F^{GOP}	number of frames in a GOP.
L	number of layers.
z_t	the received data for the frame played out at t .
$\omega_\ell^P, \omega_\ell^I$	the data in the ℓ^{th} layer of a P, I frame.
q^ℓ	the quality increment when the ℓ^{th} layer is received.
$q^I(z_t), q^P(z_t)$	the rate-quality model for I frames and P frames.
$\widehat{q}^I(z_t), \widehat{q}^P(z_t)$	the concave envelopes of $q^I(z_t)$ and $q^P(z_t)$.
X_t, Y_t	the transmission bitrate and the packet error rate at t .
x_t, y_t	the realizations of X_t, Y_t .
R_t	the channel throughput, i.e., $R_t = X_t(1 - Y_t)$.
r_t	the realization of R_t
r^{avg}	the average channel throughput.
C_t, V_t and S_t	channel state, buffer state, system state at t .
$\mathbf{c}_t, \mathbf{v}_t$ and \mathbf{s}_t	the realizations of C_t, V_t and S_t .

slot, a frame is played out, and video data units are scheduled for transmission.

The scheduling action is defined as a set of ordered video data units

$$\mathcal{D} = \{(f_1, \ell_1), (f_2, \ell_2), \dots, (f_{|\mathcal{D}|}, \ell_{|\mathcal{D}|})\}. \quad (4.1)$$

When scheduling action \mathcal{D} is taken, the associated data units are transmitted sequentially. Each scheduled data unit is packetized into physical layer packets and each packet is repeatedly transmitted, i.e., if errors occur, until acknowledged.

The frequently used notation is summarized in Table 4.1.

4.3 Markov Decision Process-Based Model

In this section, I propose an MDP-based model to determine the near-optimal scheduling policy. To that end I describe the scheduler’s state space and the policies to be considered. I then show how to reduce the scheduling problem to a finite-state Markov decision problem using reasonable approximations. To validate the optimality of the MDP-based scheduling policies, I develop a performance upper bound at the end of this section.

4.3.1 Scheduling Policy and State Space

Considering all possible scheduling actions makes defining the scheduling policy and representing the buffer state unmanageably complex. On one hand, to capture the buffer state, the frame index and the layer index of each received data unit need to be recorded. If we assume an infinite playback buffer, the number of received data units is not bounded. So we cannot represent all possible buffer states using a finite-dimensional space. On the other hand, not all possible scheduling policies need to be considered. For example, video data units should not be transmitted before their predictors. If their predictors are not received before their playout deadlines, these units are undecodable and useless. Thus we need only consider those scheduling strategies that are not dominated and have potential to achieve good performance.

Specifically, I consider scheduling policies under the following assumptions:

Assumption 3. The scheduler always schedules a data unit for transmission

after its predictors.

Assumption 4. The amount of video data scheduled on a slot exceeds the amount of data which can be transmitted in the slot.

Assumption 5. Let \mathcal{W} denote the set of data units associated with the first W active frames. I assume the scheduler first sends the video data in \mathcal{W} . Then, if all the data in \mathcal{W} has been received, the policy greedily schedules as many enhancement layers as possible, i.e., starts transmitting the next frame only when all the layers of preceding frame have been received.

Assumption 6. The scheduler never schedules enhancement layer data units for future P frames if those for earlier P frames in the same GOP have not been sent.

Assumption 3 ensures that the transmission order is compatible with the prediction order given in Section 4.2.2, since a data unit can be decoded only when its predictors are received. Assumption 4 ensures the transmitter will not be idle during a slot. Assumption 5 stems from the fact that, when many frames are buffered at the receiver, the scheduler can transmit more enhancement layers because there is sufficient time before the frames are played out. As will be discussed in Section 4.3.3, this assumption helps to simplify the policy optimization problem. It should be noted that policies under Assumption 5 are different from the sliding window policies defined in [27]. Indeed, our scheduling policy allows the transmitter to transmit data units outside the window. With Assumption 6 in effect, at any time and for all the P frames

in a GOP, the scheduler does not sacrifice the quality of the frames that will be displayed sooner for the frames to be displayed later by transmitting more enhancement layer data for the latter. Because the optimization objective is the time-average MS-SSIM index, the rate-quality function of each P frame is assumed to be the same and thus their qualities are equally important. Transmitting more enhancement layer data for later frames does not help to improve the time-average MS-SSIM value.

Note that, although the P frames within a GOP are equally important in terms of contribution to the time-average MS-SSIM index, the I and P frames in different GOPs are not. For example, when the channel throughput is very low, it may be beneficial to sacrifice P frames in the current GOP to transmit the base layer for an I frame in the next GOP, because an I frame contains much more data than a P frame and the loss of an I frame would cause severe decoding failures throughout the next GOP. To differentiate the importance of current and future GOPs, I partition the data units of the active frames into three sets: \mathcal{I} , \mathcal{I}^{pre} and $\mathcal{I}^{\text{post}}$. The set \mathcal{I} contains the data units of the first active I frame, \mathcal{I}^{pre} contains data units preceding the first active I frame, and $\mathcal{I}^{\text{post}}$ contains the remaining active data units (see Fig. 4.4).

I define the overall buffer state space \mathcal{V} via three sets \mathcal{V}^I , \mathcal{V}^{pre} and $\mathcal{V}^{\text{post}}$, where $\mathcal{V} = \mathcal{V}^I \times \mathcal{V}^{\text{pre}} \times \mathcal{V}^{\text{post}}$.

\mathcal{V}^I : The state of \mathcal{I} is defined as $\mathbf{v}^I = (f^I, b^I)$, where $f^I \in \{1, \dots, F^{\text{GOP}}\}$ is the number of frames until the first active I frame and b^I is the number of

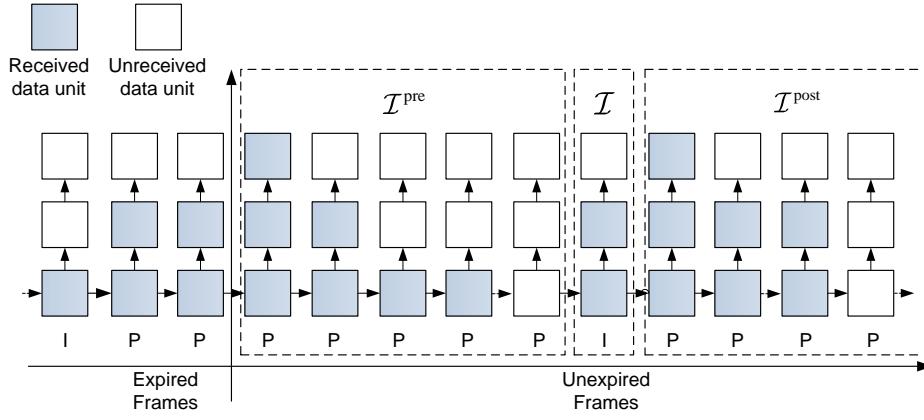


Figure 4.4: An illustration of the receiver buffer state when $F^{\text{GOP}} = 8$, $L = 3$. $\mathbf{v}_t^{\text{pre}} = (4, 2, 1)$, $\mathbf{v}_t^I = (6, 2)$, $\mathbf{v}_t^{\text{post}} = (3, 3, 1)$.

the received data units of \mathcal{I} , thus $\mathcal{V}^I = \{1, \dots, F^{\text{GOP}}\} \times \{1, \dots, L\}$.

\mathcal{V}^{pre} : When Assumption 6 is enforced, the number of data units received in \mathcal{I}^{pre} must be non-increasing in the frame index. Hence, we only need record the total number of received data units for each layer. I define the buffer state space for \mathcal{I}^{pre} as a L -dimensional vector $\mathbf{v}^{\text{pre}} = (b_1^{\text{pre}}, b_2^{\text{pre}}, \dots, b_L^{\text{pre}})$, where b_ℓ^{pre} is the number of the received data units in the ℓ^{th} layer for \mathcal{I}^{pre} , thus $\mathcal{V}^{\text{pre}} = \{0, 1, \dots, F^{\text{GOP}} - 1\}^L$.

$\mathcal{V}^{\text{post}}$: As with \mathcal{V}^{pre} , I define the buffer state space of $\mathcal{I}^{\text{post}}$ as a L -dimensional vector $\mathbf{v}^{\text{post}} = (b_1^{\text{post}}, b_2^{\text{post}}, \dots, b_L^{\text{post}})$, where b_ℓ^{post} is the number of the received data units in the ℓ^{th} layer of $\mathcal{I}^{\text{post}}$. Because the receiver buffer size is assumed to be large, i.e., essentially infinite, b_ℓ^{post} is unbounded. Thus $\mathcal{V}^{\text{post}} = \mathbb{N}^L$, where $\mathbb{N} = \{0, 1, \dots, \infty\}$.

In [89] and [84], it is shown that a first-order finite state Markov chain

(FSMC) can be used to describe the first-order channel state transition probabilities for Rayleigh fading channels. First-order FSMC models have also been validated in [43] and [79] by wireless channel measurements in urban areas. In our MDP-based model, I employ a first-order FSMC to describe the dynamics of the channel state.

At the physical layer, the transmission bitrate x is determined by the modulation and coding scheme (MCS) and the packet error rate y is determined by both the channel state and the MCS¹. I assume the chosen MCS is a function of the channel state under a given link adaptation mechanism. For example, the physical layer can always choose the MSC which maximizes channel throughput $x(1 - y)$. Thus, there is a one-to-one mapping from channel state to the tuple (x, y) . I define the channel state as $\mathbf{c} = (x, y)$. Due to the Markov property of the channel state, channel state can also be modeled by an FSMC. The channel state space is $\mathcal{C} = \{\mathbf{c}^1, \dots, \mathbf{c}^{|\mathcal{C}|}\}$, where $\mathbf{c}^i = (x^i, y^i)$ is the i^{th} channel state. The state transition matrix \mathbf{P}^c is a $|\mathcal{C}| \times |\mathcal{C}|$ matrix with entry $\mathbf{P}_{i,j}^c = \mathbb{P}(\mathbf{c}^j | \mathbf{c}^i)$ being the transition probability from state \mathbf{c}^i to \mathbf{c}^j .

The system state space \mathcal{S} is defined as the product of the channel state space \mathcal{C} and the buffer state space \mathcal{V} . For each state $\mathbf{s} \in \mathcal{S}$, I define a feasible control set \mathcal{D}_s that contains all the scheduling actions (see Equation (4.1)) complying with all the four assumptions. The state \mathbf{s} contains all the information about the receiver buffer and the channel. The transmitter must decide

¹Here, bitrate x is the number of bits transmitted in a time slot ΔT , i.e., the transmission rate normalized with slot duration ΔT .

which action in \mathcal{D}_s to take in order to maximize the time-average MS-SSIM index value. I define the scheduling policy $\mu(\cdot)$ as the mapping from the system state s to an action in \mathcal{D}_s . In the following sections, I show how to optimize the scheduling policy $\mu(\cdot)$.

4.3.2 Optimization Objective

Since the channel condition is modeled as a random process, I denote by $(C_t, V_t, S_t)_{t \in \mathbb{N}}$ the random processes modeling channel, buffer and system state, respectively. Accordingly, I denote by $(c_t, v_t, s_t)_{t \in \mathbb{N}}$ their realizations. At the beginning of each time slot t , the first frame in the window is played out and the MS-SSIM index is

$$q(S_t) = \sum_{\ell=1}^L q^\ell \times \mathbb{1}_\ell(S_t), \quad (4.2)$$

where $\mathbb{1}_\ell(S_t)$ is the indicator that the ℓ^{th} layer of the displayed frame is available in state S_t . The quantity q^ℓ is the marginal STSQ improvement if, in addition to layers $1, 2, \dots, \ell - 1$, layer ℓ is available (see Section 4.2.3). Our aim is to find an optimal policy $\mu^*(\cdot)$ that maximizes the time-average MS-SSIM index, i.e.,

$$J_\mu = \lim_{n \rightarrow \infty} \mathbb{E}_\mu \left\{ \frac{1}{n} \sum_{t=0}^{n-1} q(S_t) \right\}. \quad (4.3)$$

4.3.3 Finite State Problem Formulation

Since the state space $\mathcal{V}^{\text{post}}$ is infinite, the state space \mathcal{S} is also infinite. Optimizing the scheduling policy over this infinite-state space is intractable. With Assumption 5, the scheduling policy is actually fixed when all the data in win-

dow \mathcal{W} is received. We only need to determine the optimal scheduling policy for states where some of the video data in the window has not been received, which is a finite state set. The system state, however, still evolves in the infinite state space \mathcal{S} . In the following, I show how to simplify this infinite state space problem to a finite-state problem.

I define the set of states where some of the video data in \mathcal{W} has not been received as follows:

$$\mathcal{S}_W = \{\mathbf{s} | \mathbf{s} \in \mathcal{S}, \mathcal{O}(\mathbf{s}) \subset \mathcal{W}\}, \quad (4.4)$$

where $\mathcal{O}(\mathbf{s})$ is the set of buffered active video data units when the state is \mathbf{s} . I define another subset of \mathcal{S} as follows

$$\mathcal{S}_{\overline{W}} = \{\mathbf{s} | \mathbf{s} \in \mathcal{S}, \mathcal{W} \subseteq \mathcal{O}(\mathbf{s})\}. \quad (4.5)$$

For all the states in $\mathcal{S}_{\overline{W}}$, all the video data units in \mathcal{W} has been received. Note that, under Assumption 5, the video scheduler focuses on transmitting data in \mathcal{W} until $\mathcal{W} \subseteq \mathcal{O}(\mathbf{s})$. Thus \mathcal{S}_W and $\mathcal{S}_{\overline{W}}$ form a partition of state space \mathcal{S} . In other words, we have $\mathcal{S}_W \cup \mathcal{S}_{\overline{W}} = \mathcal{S}$ and $\mathcal{S}_W \cap \mathcal{S}_{\overline{W}} = \emptyset$.

Given a policy $\mu(\cdot)$, the system state evolves as a Markov chain in set $\mathcal{S}_W \cup \mathcal{S}_{\overline{W}}$. Because the transmission rate is finite, the number of states in $\mathcal{S}_{\overline{W}}$ which can be reached from \mathcal{S}_W in one step is also finite. I formally define this set of states as follows

$$\mathcal{S}_\Delta = \{\mathbf{s} | \mathbf{s} \in \mathcal{S}_{\overline{W}}, \exists \mathbf{s}' \in \mathcal{S}_W, s.t., \mathbb{P}_\mu(\mathbf{s}|\mathbf{s}') > 0\}, \quad (4.6)$$

where $\mathbb{P}_\mu(\mathbf{s}|\mathbf{s}')$ is the state transition probability under policy μ (for the expression for $\mathbb{P}_\mu(\mathbf{s}|\mathbf{s}')$, see Appendix E). Thus to move from \mathcal{S}_W into the set $\mathcal{S}_{\overline{W}}$, the system state first hits a state in \mathcal{S}_Δ and then stays in $\mathcal{S}_{\overline{W}}$ for some time. During this period, the decoded STSQ is always $\sum_{\ell=1}^L q^\ell$, because all the layers in \mathcal{W} are available. The evolution of the system when it moves into set $\mathcal{S}_{\overline{W}}$ affect the performance of the system. In general, the longer it stays in $\mathcal{S}_{\overline{W}}$, the better the performance is. Although the scheduling policy in $\mathcal{S}_{\overline{W}}$ is fixed as described in Assumption 5, the policy in \mathcal{S}_W determines how frequently the system state will hit $\mathcal{S}_{\overline{W}}$ and thus critically impacts the system performance.

In the following, I denote the system under a given policy μ as system Π_μ . Let $t_\mu(\mathbf{s})$ be the expected time spent by Π_μ in $\mathcal{S}_{\overline{W}}$ after it enters $\mathcal{S}_{\overline{W}}$ at state $\mathbf{s} \in \mathcal{S}_\Delta$. Let $\tilde{\mathbb{P}}_\mu(\mathbf{s}'|\mathbf{s})$ denote the probability that Π_μ jumps back to \mathcal{S}_W at state $\mathbf{s}' \in \mathcal{S}_W$ after it enters $\mathcal{S}_{\overline{W}}$ at state \mathbf{s} . To find the optimal policy, I define a finite-state system $\tilde{\Pi}_\mu$ as follows

Definition 4.3.1. A system $\tilde{\Pi}_\mu$ is called the simplified system of the original system Π_μ if it has the following dynamics:

1. The system is a semi-Markov process over state space $\tilde{\mathcal{S}} = \mathcal{S}_W \cup \mathcal{S}_\Delta$. In any state $\mathbf{s} \in \tilde{\mathcal{S}}$, the visual quality is $q(\mathbf{s})$ as in (4.2). In any state in \mathcal{S}_W , the system evolves according to the policy μ . The system state transition probability is $\mathbb{P}_\mu(\cdot|\cdot)$.
2. When the system jumps to a state $\mathbf{s} \in \mathcal{S}_\Delta$, it spends $t_\mu(\mathbf{s})$ slots in \mathbf{s} with STSQ $\sum_{\ell=1}^L q^\ell$ for each slot. The system then transitions to a state

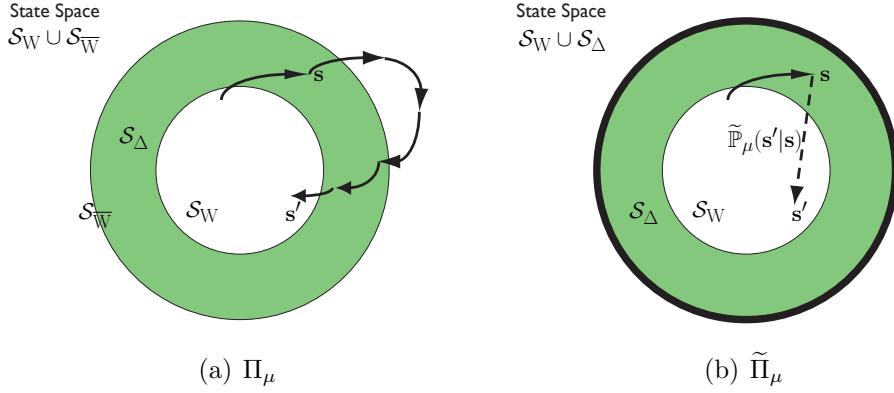


Figure 4.5: The dynamics of the system Π_μ and the corresponding simplified system $\tilde{\Pi}_\mu$.

$s' \in \mathcal{S}_W$ with probability $\tilde{\mathbb{P}}_\mu(s'|s)$ (see Fig. 4.5).

It should be noted that $\tilde{\Pi}_\mu$ is not coupled with the original system Π_μ . It just shares some properties with the original system. The following theorem relates the visual quality under $\tilde{\Pi}_\mu$ and that of Π_μ .

Theorem 4.3.1. *If the jump chain of the original system Π_μ is positive recurrent, then the time-average MS-SSIM index of Π_μ is the same as the simplified system $\tilde{\Pi}_\mu$.*

Proof Sketch. If the jump chain is positive recurrent, the jump from \mathcal{S}_W to \mathcal{S}_Δ can partition the Markov process into i.i.d segments. We only need to optimize the policy μ to maximize the average quality in each segment. Every segment consists of two consecutive subsegments. During the first subsegment, $s_t \in \mathcal{S}_{\overline{W}}$. In the other subsegment, $s_t \in \mathcal{S}_W$. Because every state in $\mathcal{S}_{\overline{W}}$ has the same visual quality $\sum_{\ell=1}^L q^\ell$, we can abstract the first subsegment as a single state

with transition probability $\tilde{\mathbb{P}}_\mu(\cdot|\cdot)$. This simplified system provides the same average quality as the original system. For a detailed proof, see the technical report [1]. \square

Remark The positive recurrent condition for the jump chain means that the average throughput of the channel is neither too large nor too small relative to the average data rate of the video. If the average throughput of the channel is very large, the receiver buffer can always buffer enough frames and dynamic scheduling is unnecessary. If the average channel throughput is too small, the channel cannot support the video stream and dynamic scheduling cannot help either.

As indicated by Theorem 4.3.1, given any policy μ , the visual quality of Π_μ is the same as $\tilde{\Pi}_\mu$. Thus, we can optimize our policy with respect to $\tilde{\Pi}_\mu$, which has a finite-state space, and a standard policy optimization algorithm can be applied.

Before I can apply an MDP algorithm to optimize the policy, we need to compute $t_\mu(\mathbf{s})$ and $\tilde{\mathbb{P}}_\mu(\mathbf{s}'|\mathbf{s})$ for every state $\mathbf{s} \in \mathcal{S}_\Delta$ and $\mathbf{s}' \in \mathcal{S}_W$. Both $t_\mu(\mathbf{s})$ and $\tilde{\mathbb{P}}_\mu(\mathbf{s}'|\mathbf{s})$ only involve dynamics of the system in $\mathcal{S}_{\overline{W}}$. Details on how to compute $t_\mu(\mathbf{s})$ and $\tilde{\mathbb{P}}_\mu(\mathbf{s}'|\mathbf{s})$ are found in Appendix F.

4.3.4 Policy Optimization

Given $t_\mu(\cdot)$ and $\tilde{\mathbb{P}}_\mu(\cdot|\cdot)$, the optimal policy for an MDP can be determined for the simplified system $\tilde{\Pi}_\mu$, which is also the optimal policy of Π_μ . Let \mathbf{s}^{ini} be

any state in $\tilde{\mathcal{S}} = \mathcal{S}_W \cup \mathcal{S}_\Delta$. The hitting time to state s^{ini} can partition the process into i.i.d cycles. Optimizing the policy $\mu(\cdot)$ in the cycles maximizes the time-average MS-SSIM index of the system. Similar to the derivation in [14, p. 441], this is equivalent to an average-reward maximization problem with stage-reward $g(s) - \eta(s)\lambda$, where λ is the expected average-reward of each cycle and

$$g(s) = \begin{cases} q(s) & : s \in \mathcal{S}_W \\ t_\mu(s) \sum_{\ell=1}^L q^\ell & : s \in \mathcal{S}_\Delta, \end{cases}$$

$$\eta(s) = \begin{cases} 1 & : s \in \mathcal{S}_W \\ t_\mu(s) & : s \in \mathcal{S}_\Delta, \end{cases}$$

where $q(s)$ is defined in (4.2). Let us denote by $h(s)$ the average reward-to-go in each cycle when the system starts at state s . Then we have the following Bellman's equation array:

$$h(s) = g(s) - \eta(s)\lambda + \sum_{s' \in S_W \cup \mathcal{S}_\Delta} \mathbb{P}_\mu(s'|s)h(s'), \quad (4.7)$$

where $h(s^{ini}) = 0$. To find the optimal policy, the standard value iteration algorithm can be applied [14, p. 430].

On the one hand, the assumptions on scheduling policy result in the finite state MDP-based formulation. On the other hand, the assumptions may render the derived scheduling policy sub-optimal. To verify the performance of the scheduling policy derived from the MDP formulation is actually close to optimal, I prove a performance upper bound in the next section.

4.3.5 Performance Upper Bound

In the rest of the chapter, let $R_t = X_t(1 - Y_t)$ be the throughput of the channel at time t , where X_t is the transmission bitrate and Y_t is the packet error rate as defined in Section. 4.3.1. Since the channel condition is modeled as a random process, X_t , Y_t and R_t are random processes and I will denote by x_t , y_t and r_t their realizations. As discussed in Section. 4.2.3, $q^I(z_t)$ and $q^P(z_t)$ are the rate quality models of I frames and P frames, respectively. Let $\mathbb{1}_t^I$ be the indicator that the t^{th} frame is an I frame. The time-average MS-SSIM of the transmitted video can be written as $\frac{1}{n} \sum_{t=1}^n [q^I(z_t) \mathbb{1}_t^I + q^P(z_t)(1 - \mathbb{1}_t^I)]$, where n is the number of frames in the video sequence. An upper bound on the performance of any scheduler is given by the following offline optimization problem:

$$\begin{aligned} & \underset{z_{1:n}}{\text{maximize}} \quad \frac{1}{n} \sum_{t=1}^n [q^I(z_t) \mathbb{1}_t^I + q^P(z_t)(1 - \mathbb{1}_t^I)] \\ & \text{s.t.} \quad \frac{1}{t} \sum_{i=1}^t z_i \leq \frac{1}{t} \sum_{i=1}^t r_i, \quad \forall t \in \{1, 2, \dots, n\}, \end{aligned} \tag{4.8}$$

where the constraint $\frac{1}{t} \sum_{i=1}^t z_i \leq \frac{1}{t} \sum_{i=1}^t r_i$ guarantees that the received data for the frames displayed before time t does not exceed the cumulative throughput prior to time t . We can further relax the constraints in (4.8) by only keeping the last one, i.e., when $t = n$. The relaxed optimization problem is

then given by

$$\begin{aligned} \underset{z_{1:n}}{\text{maximize}} \quad & \frac{1}{n} \sum_{t=1}^n [q^I(z_t) \mathbb{1}_t^I + q^P(z_t)(1 - \mathbb{1}_t^I)] \\ \text{s.t.} \quad & \frac{1}{n} \sum_{t=1}^n z_t \leq \frac{1}{n} \sum_{t=1}^n r_t. \end{aligned} \tag{4.9}$$

Let $\widehat{q}^I(z_t)$ and $\widehat{q}^P(z_t)$ be the concave envelope of $q^I(z_t)$ and $q^P(z_t)$ respectively (see Fig. 4.3). Since, $q^I(z_t)$ and $q^P(z_t)$ are upper bounded by $\widehat{q}^I(z_t)$ and $\widehat{q}^P(z_t)$, we can bound problem (4.9) by:

$$\begin{aligned} \underset{z_{1:n}}{\text{maximize}} \quad & \frac{1}{n} \sum_{t=1}^n [\widehat{q}^I(z_t) \mathbb{1}_t^I + \widehat{q}^P(z_t)(1 - \mathbb{1}_t^I)] \\ \text{s.t.} \quad & \frac{1}{n} \sum_{t=1}^n z_t \leq \frac{1}{n} \sum_{t=1}^n r_t. \end{aligned} \tag{4.10}$$

Let $n^I = \sum_{t=1}^n \mathbb{1}_t^I$ denote the number of I frames and $n^P = \sum_{t=1}^n (1 - \mathbb{1}_t^I)$ denote the number of P frames. Since the functions $\widehat{q}^I(z_t)$ and $\widehat{q}^P(z_t)$ are concave, by Jensen's inequality, we have

$$\frac{1}{n^I} \sum_{t=1}^n \widehat{q}^I(z_t) \mathbb{1}_t^I \leq \widehat{q}^I \left(\frac{1}{n^I} \sum_{t=1}^n z_t \mathbb{1}_t^I \right)$$

and

$$\frac{1}{n^P} \sum_{t=1}^n \widehat{q}^P(z_t)(1 - \mathbb{1}_t^I) \leq \widehat{q}^P \left(\frac{1}{n^P} \sum_{t=1}^n z_t(1 - \mathbb{1}_t^I) \right).$$

Problem (4.10) can then be bounded by:

$$\begin{aligned} \underset{z_{1:n}}{\text{maximize}} \quad & \frac{n^I}{n} \widehat{q}^I \left(\frac{1}{n^I} \sum_{t=1}^n z_t \mathbb{1}_t^I \right) + \frac{n^P}{n} \widehat{q}^P \left(\frac{1}{n^P} \sum_{t=1}^n z_t(1 - \mathbb{1}_t^I) \right) \\ \text{s.t.} \quad & \frac{n^I}{n} \left(\frac{1}{n^I} \sum_{t=1}^n z_t \mathbb{1}_t^I \right) + \frac{n^P}{n} \left(\frac{1}{n^P} \sum_{t=1}^n z_t(1 - \mathbb{1}_t^I) \right) \leq \frac{1}{n} \sum_{t=1}^n r_t. \end{aligned} \tag{4.11}$$

If the video is reasonably long, e.g. several minutes, the frame number n will be very large. If we let $n \rightarrow \infty$ and assume the channel throughput r_t is ergodic, $\frac{1}{n} \sum_{t=1}^n r_t$ will converge to the ergodic capacity $r^{\text{avg}} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n r_t$. Furthermore, since F^{GOP} is the length of the GOP, $\frac{1}{F^{\text{GOP}}}$ and $1 - \frac{1}{F^{\text{GOP}}}$ are proportion of I and P frames in the video sequences. Thus we have $\frac{n^I}{n} \rightarrow \frac{1}{F^{\text{GOP}}}$ and $\frac{n^P}{n} \rightarrow 1 - \frac{1}{F^{\text{GOP}}}$. Similarly, for stationary policies², the limits $z^I = \lim_{n \rightarrow \infty} \frac{1}{n^I} \sum_{t=1}^n z_t \mathbb{1}_t^I$ and $z^P = \lim_{n \rightarrow \infty} \frac{1}{n^P} \sum_{t=1}^n z_t (1 - \mathbb{1}_t^I)$ exist. We have $\lim_{n \rightarrow \infty} \left[\frac{n^I}{n} \left(\frac{1}{n^I} \sum_{t=1}^n z_t \mathbb{1}_t^I \right) + \frac{n^P}{n} \left(\frac{1}{n^P} \sum_{t=1}^n z_t (1 - \mathbb{1}_t^I) \right) \right] = \frac{1}{F^{\text{GOP}}} z^I + (1 - \frac{1}{F^{\text{GOP}}}) z^P$. Thus, we have shown the following theorem:

Theorem 4.3.2. *For ergodic wireless throughput and stationary adaptive scheduling policies the following optimization gives an upper bound on performance:*

$$\begin{aligned} & \underset{z_I, z_P}{\text{maximize}} \quad \frac{1}{F^{\text{GOP}}} \widehat{q}^I(z^I) + \left(1 - \frac{1}{F^{\text{GOP}}}\right) \widehat{q}^P(z^P) \\ & \text{s.t.} \quad \frac{1}{F^{\text{GOP}}} z^I + \left(1 - \frac{1}{F^{\text{GOP}}}\right) z^P \leq r^{\text{avg}}. \end{aligned} \quad (4.12)$$

Since the rate-quality functions $\widehat{q}^I(\cdot)$ and $\widehat{q}^P(\cdot)$ are assumed to be concave, the above optimization problem is convex and easily solved. In Section 4.3.6, this upper bound will be employed as a benchmark to evaluate the performance of our MDP-based scheduling policy.

²A policy is called stationary if it is a function of state s and the function is invariant with respect to time t .

4.3.6 Performance Evaluation

In this section, I evaluate the performance of the policy obtained from our MDP-based formulation. Parallel to [90] and [32], I employ the FSMC channel model proposed in [89] to model the dynamics of Rayleigh fading channels. The SNR at the receiver is partitioned into 4 regions using the algorithm proposed in [89]. In our simulations, I set average SNR to $\Lambda^{\text{avg}} = 10\text{dB}$. The MDP-based scheduling algorithm was evaluated on test sequences “foreman”, “bus”, “flower”, “mobile” and “Paris” [2]. These video sequences were encoded using H.264/SVC reference software JSVM [68] with 3 layers. For each sequence, 200 transmissions were sent over the simulated channel. A startup delay constraint was fixed to 100ms, i.e., video playback began 3 frames after the transmission began. To conceal errors, every lost frame was reconstructed by copying the preceding frame. For more details about the FSMC channel model and encoding parameter of the video sequences, see G.

The performance of the MDP-based scheduling algorithm was tested over the simulated Markov channel models with different Doppler frequencies ($f^d = 5\text{Hz}$ and 3Hz , respectively). The simulation results are summarized in Table 4.2 and Table 4.3. The time-averaged MS-SSIM value is converted to Difference Mean Opinion Score (DMOS) using the following mapping

$$q^{\text{dmos}} = 13.3442 \log(1 - q^{\text{ssim}}) + 3.6226(1 - q^{\text{ssim}}) + 77.0117, \quad (4.13)$$

where q^{ssim} denotes the time averaged quality measured in MS-SSIM and q^{dmos} is the corresponding DMOS value. Equation (4.13) is obtained by logistic

Table 4.2: The performance of the near-optimal policy in SSIM-predicted DMOS. $f^d = 5$.

	Paris	mobile	flower	bus	foreman
MDP-based Policy	33.5361	46.5898	39.8506	47.3337	36.8433
Upper bound (4.12)	33.4279	44.7976	38.0841	46.8378	36.3160

Table 4.3: The performance of the near-optimal policy in SSIM-predicted DMOS. $f^d = 3$.

	Paris	mobile	flower	bus	foreman
MDP-based Policy	33.8628	44.9776	42.8653	48.2840	36.6403
Upper bound (4.12)	33.4279	44.7941	38.0925	46.9338	36.3536

regression using the MS-SSIM indices and MOS values of the images in the LIVE database [3]. DMOS ranges from 0 to 100. Value 0 means perfect visual quality and value 100 means bad visual quality. Roughly speaking, value 50 means fair quality. It can be seen from Table 4.2 and Table 4.3 that the DMOS value of the MDP-based scheduling policy is worse than the performance bound by at most 4, which is visually insignificant. Given that the bound given by Theorem 4.3.2 is an upper bound (i.e. a lower bound of DMOS value), the MDP-based scheduling policy is indeed near-optimal.

4.4 Near-optimal Heuristic On-line Scheduling Algorithm

Although the MDP-based formulation makes it possible to compute a good scheduling policy using value iteration algorithm, off-line computation of such policies requires *a priori* knowledge of the channel dynamics. This motivates

us to design a simple on-line scheduling policy that delivers similar performance as the MDP-based policy that only requires little *a priori* knowledge about the channel dynamics.

A good online video scheduling algorithm should explicitly take advantage of the channel dynamics and schedule data from different quality layers as a function of the receiver buffer state. There are three fundamental questions in designing such a scheduler: 1) How should one incorporate limited knowledge of channel dynamics in adaptive scheduling? 2) How should one determine the number of enhancement layers to schedule? 3) How should one allocate appropriate transmission rate among \mathcal{I}^{pre} , \mathcal{I} and $\mathcal{I}^{\text{post}}$ (see definition in 4.3.1)? In the following, I will show how to address these fundamental problems by reasonably simplifying the MDP-based scheduling algorithm.

4.4.1 Channel Model Simplification

In a practical wireless communication environment, accurate channel dynamics models such as the state transition probability \mathbf{P}^c are not generally available. Some basic characteristics for the channel dynamics can, however, be easily used. At any slot t , the instantaneous channel throughput $r_t = x_t(1 - y_t)$ can be derived using receiver channel state information feedback. The ergodic channel throughput r^{avg} can be estimated by averaging r_t over time. Furthermore, the temporal correlation coefficient $\rho = \frac{\text{cov}(R_t, R_{t+1})}{\sigma(R_t)\sigma(R_{t+1})}$ can also be estimated from r_t . Further it is reasonable to assume the channel throughput R_t will typically regress to the mean r^{avg} . This inspires us to use

a simple autoregressive model to capture the dynamics of the channel. A first order autoregressive model (AR(1)) for R_t is given as,

$$R_t - \phi R_{t-1} = c + N_t, \quad (4.14)$$

where N_t is an i.i.d random variable with zero mean value. From (4.14), parameter c and ϕ can be estimated as $\phi = \rho$ and $c = r^{\text{avg}}(1 - \rho)$ [61]. Thus, we have

$$R_t - \rho R_{t-1} = r^{\text{avg}}(1 - \rho) + N_t. \quad (4.15)$$

Using this autoregressive model, the amount of data that will be delivered in the next τ slots by the channel can be estimated as

$$d(r_t) = \mathbb{E} \left[\sum_{j=0}^{\tau-1} R_{t+j} \middle| R_t = r_t \right] = \sum_{j=0}^{\tau-1} [r_t \rho^j + r^{\text{avg}}(1 - \rho^j)]. \quad (4.16)$$

To obtain an accurate estimate in the near future, I set the length of the window τ into the future that will be considered to be the relaxation time³ of the channel, i.e. , $\tau = \lceil -(\ln \rho)^{-1} \rceil$. In the following, I use this to determine which quality layers to schedule.

4.4.2 Layer Selection

Given the current channel state, receiver buffer state, and estimated available capacity for a window τ into the future, the goal is to determine which layers to schedule. I will focus on determining the number of enhancement layers

³The relaxation time is defined as the temporal distance at which the temporal correlation coefficient is reduced to $\frac{1}{e}$

which should be scheduled. I denote by $L^{sch}(\mathbf{s}_t)$ the number of layers to be scheduled if the state is \mathbf{s}_t . Once $L^{sch}(\mathbf{s}_t)$ is determined, the online scheduling algorithm only schedules data units from the first $L^{sch}(\mathbf{s}_t)$ layers.

The layer selection scheme for our proposed on-line algorithm is motivated by that of the MDP-based policy. Using $d(r_t)$ defined in (4.16), we can estimate the amount of data which can be delivered in the next τ slots. Let $\Gamma(\ell, \mathbf{s}_t)$ be the amount of data which is not currently available at the playback buffer at time t , and belongs to the first ℓ layers of the next τ frames. The quantities $d(r_t)$ and $\Gamma(\ell, \mathbf{s}_t)$ summarize the channel and buffer states for the next τ slots. Note that $\Gamma(\ell - 1, \mathbf{s}_t) \leq d(r_t) < \Gamma(\ell, \mathbf{s}_t)$ means that we can probably transmit all the data up to the ℓ^{th} layer in the next τ slots. Intuitively, we can simply choose $L^{sch}(\mathbf{s}_t) = \ell - 1$ when $\Gamma(\ell - 1, \mathbf{s}_t) \leq d(r_t) < \Gamma(\ell, \mathbf{s}_t)$. As discussed next, this layer selection scheme can be motivated by the near-optimal scheduling policies computed for the MDP-based model.

Note that $r_t = x_t(1 - y_t)$ is determined by state \mathbf{s}_t , thus $d(r_t)$ can also be written as function of \mathbf{s}_t , i.e., $d(\mathbf{s}_t)$. Suppose I partition the state space into subsets $\mathcal{P}^\ell = \{\mathbf{s} \in \mathcal{S} : \Gamma(\ell - 1, \mathbf{s}) \leq d(\mathbf{s}) < \Gamma(\ell, \mathbf{s})\}, \ell \in \{1, \dots, L + 1\}$ and calculate the fraction of states in \mathcal{P}^ℓ where the MDP-based policy only schedules the first $\ell - 1$ layers⁴. As shown in Fig. 4.6, for 70% of the states of \mathcal{P}^1 and \mathcal{P}^2 , the MDP-based policy only schedules the first layer. For about 65% of the states of \mathcal{P}^3 , the MDP-based policy only schedules the first 2 layers.

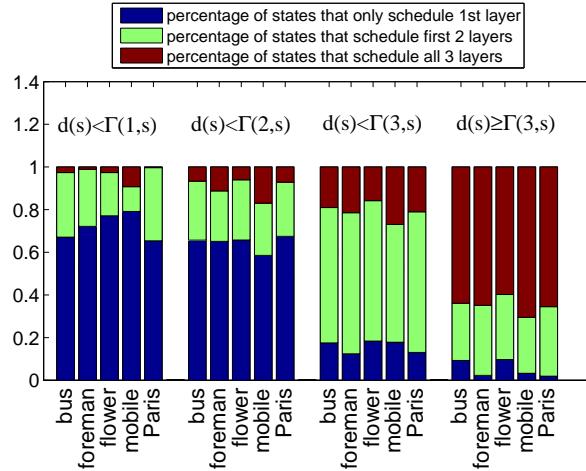
⁴I define $\Gamma(L + 1, \mathbf{s}_t) = +\infty$

Finally the MDP-based policy will schedule all the layers on 65% of the states in \mathcal{P}^4 . These observations justify our intuition regarding layer selection. In our proposed on-line scheduling algorithm, I will simply choose $L^{sch}(\mathbf{s}_t) = \ell - 1$ if $\Gamma(\ell - 1, \mathbf{s}_t) \leq d(r_t) < \Gamma(\ell, \mathbf{s}_t)$. In other words, our heuristic algorithm determines $L^{sch}(\mathbf{s}_t)$ by roughly estimating the number of layers which can be transmitted.

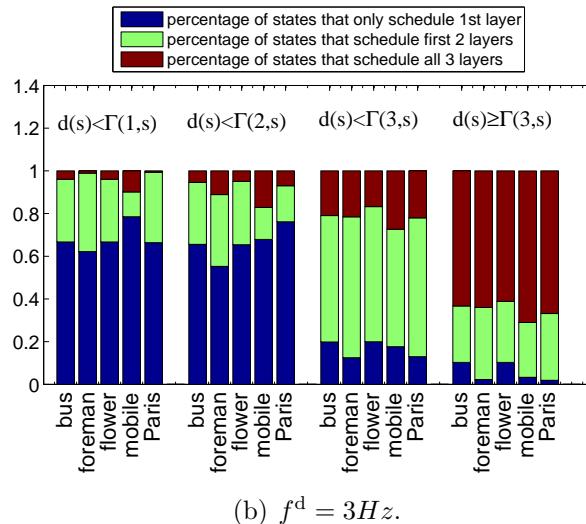
4.4.3 Resource Allocation Among GOPs

In each transmission slot, r_t bits of video data are delivered to the receiver. In the following, I refer to r_t as the budget for slot t . Once $L^{sch}(\mathbf{s}_t)$ is determined, we still need to determine how to allocate this budget among \mathcal{J}^{pre} , \mathcal{I} and \mathcal{J}^{post} . Sometimes it is necessary to transmit data associated with next I frame before the data units in the current GOP. For example, when the next I frame is approaching its display deadline and its base layer has not yet been received, if we focus on transmitting the frames in the current GOP sequentially, this increases the risk that the next I frame can not be decoded before its deadline. This in turn would cause severe decoding failures throughout the next GOP.

I denote by $\Psi^{pre}(\ell, \mathbf{s}_t)$ the amount of unreceived data in the first ℓ^{th} layer of \mathcal{J}^{pre} at state \mathbf{s}_t . I denote by $\Psi^I(\ell, \mathbf{s}_t)$ the amount of unreceived data in the first ℓ^{th} layer of \mathcal{I} at state \mathbf{s}_t . I propose the following heuristic for allocating the bit budget between \mathcal{J}^{pre} and \mathcal{I} . In each transmission slot, the scheduling algorithm allocates up to $\Omega_t = \frac{\Psi^I(L^{sch}(\mathbf{s}_t), \mathbf{s}_t)}{\Psi^{pre}(L^{sch}(\mathbf{s}_t), \mathbf{s}_t) + \Psi^I(L^{sch}(\mathbf{s}_t), \mathbf{s}_t)}$ of the transmission bit budget to \mathcal{I} . In other words, the number of bits allocated to



(a) $f^d = 5Hz$.



(b) $f^d = 3Hz$.

Figure 4.6: Given different relationship between $d(s)$ and $\Gamma(\ell, s)$, the proportions of states corresponding to different $L^{sch}(s)$ are shown in different colors. Results are obtained under Rayleigh fading channels with different Doppler shifts (5Hz in (a) and 3Hz in (b)) and are calculated on 5 different video sequences (“bus”, “foreman”, “flower”, “mobile”, “Paris”).

\mathcal{J} is $\min(\Omega_t \times r_t, \Psi^I(L^{sch}(\mathbf{s}_t), \mathbf{s}_t))$.

Here Ω_t gives the relative importance of the next I frame and current GOP. If $\Psi^I(L^{sch}(\mathbf{s}_t), \mathbf{s}_t) = 0$, then $\Omega_t = 0\%$. It is not necessary to transmit any data for the next I frame. If $\Psi^{pre}(L^{sch}(\mathbf{s}_t), \mathbf{s}_t) = 0$, then $\Omega_t = 100\%$. I only focus on transmitting the future GOPs.

The online scheduling algorithm is summarized in Algorithm 8.

Algorithm 8 On-line adaptive scheduling algorithm

Inputs: \mathbf{s}_t , r^{avg} , r_t and ρ

```

1:  $\tau = \lceil -(\ln \rho)^{-1} \rceil$ 
2: loop  $t$ 
3:    $d(r_t) \leftarrow \sum_{j=0}^{\tau-1} [r_t \rho^j + r^{avg} (1 - \rho^j)]$                                  $\triangleright$  Channel estimation
4:   for  $\ell = 1 \rightarrow L$  do                                                  $\triangleright$  Determine  $L^{sch}(\mathbf{s}_t)$ 
5:     Compute  $\Gamma(\ell, \mathbf{s}_t)$ 
6:     if  $d(r_t) < \Gamma(\ell, \mathbf{s}_t)$  then
7:       break
8:     end if
9:   end for
10:  if  $\ell=1$  then
11:     $L^{sch}(\mathbf{s}_t) \leftarrow 1$ 
12:  else
13:     $L^{sch}(\mathbf{s}_t) \leftarrow \ell - 1$ 
14:  end if
15:  Compute  $\Psi^{pre}(L^{sch}, \mathbf{s}_t)$  and  $\Psi^I(L^{sch}, \mathbf{s}_t)$ 
16:   $\Omega_t \leftarrow \frac{\Psi^I(L^{sch}, \mathbf{s}_t)}{\Psi^{pre}(L^{sch}, \mathbf{s}_t) + \Psi^I(L^{sch}, \mathbf{s}_t)}$ 
17:  Schedule  $\min(\Omega_t \times r_t, \Psi^I(L^{sch}, \mathbf{s}_t))$  bits from  $\mathcal{J}$ .       $\triangleright$  Scheduling data
18:  Schedule  $r_t - \min(\Omega_t \times r_t, \Psi^I(L^{sch}, \mathbf{s}_t))$  bits from  $\mathcal{J}^{pre}$  and  $\mathcal{J}^{post}$ .
19: end loop

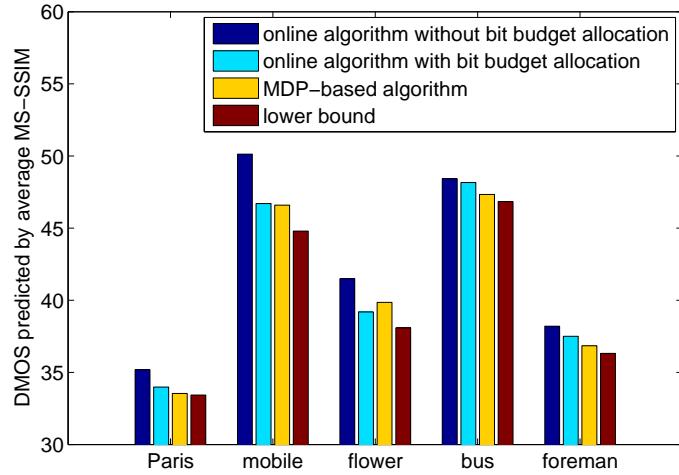
```

4.4.4 Performance Evaluation

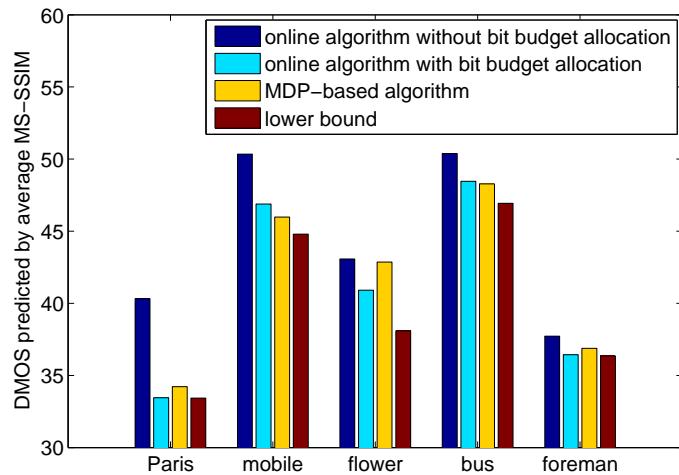
The performance of the on-line scheduling algorithm was tested over the simulated Markov channel models with different Doppler frequencies ($f^d = 5\text{Hz}$ and 3Hz , respectively). This setting is the same as the simulation setting in Section 4.3.6. The results are summarized in Fig. 4.7. As can be seen, the performance of the proposed online-scheduling algorithm is almost as good as the MDP-based scheduling algorithm. Moreover the online scheduling algorithm’s performance is close to the bound given by Theorem 4.3.2. I conclude it is a near-optimal scheduling algorithm.

According to our MDP model in Section 4.3, the MDP-based optimal scheduling policy is supposed to be optimal among all considered scheduling policies. For the test video sequence “flower”, however, the online-scheduling algorithm even outperforms the MDP-based policy. That is due to the fact that the MDP-based scheduling policy is derived based on the rate quality model in Section 4.2.3 which assumes average rate quality characteristics of all frames. The online algorithm schedules data units according to $\Gamma(\ell, \mathbf{s}_t)$ using actual size of data units rather than the rate distortion model. Therefore, the online algorithm tends to estimate the buffer states more accurately, thus may result in better performance.

I have also tested the performance of the online algorithm without bit budget allocation between current and future GOPs. As can be seen, the performance is worse than the MDP-based scheduling policy and the performance bound.



(a) $f^d = 5Hz$.



(b) $f^d = 3Hz$.

Figure 4.7: Performance comparison of different scheduling algorithms. The STSQ is measured in DMOS which is predicted by MS-SSIM using equation (4.13).

4.5 Summary

In this chapter, I have developed an adaptive scheduling algorithm for efficient scalable video transmission in wireless channels. By modeling the wireless channel as a Markov chain, an MDP model is proposed in which policies that maximize the STSQ predicted by MS-SSIM index can be computed. By simplifying the scheduling algorithm obtained from the MDP formulation, I propose an online scheduling algorithm which only requires limited knowledge of channel dynamics. Simulation results demonstrate the near-optimality of the proposed online scheduling policy versus a proposed bound on performance.

Chapter 5

Conclusions

This chapter summarizes the main results of this dissertation in Section 5.1 and discusses future research directions in Section 5.2.

5.1 Summary of the Dissertation

This dissertation focuses on designing adaptive video transmission systems that optimize the QoE of the video viewers in wireless networks. The three major contributions of the dissertation are summarized as follows:

In Chapter 2, I propose a model to predict the TVSQ of rate-switching videos transmitted over wireless networks. The model is trained and validated on a new database of quality-varying videos that simulate the true rate-adaptive videos commonly encountered in rate-switching systems. The linear correlation coefficient between the predicted and measured TVSQ is 0.88. The outage rate of the predicted TVSQ is 8.06%. Based on the analysis of the TVSQ prediction model, I draw three important conclusions. First, the behavioral response of viewers to quality variation is more sensitive in the low quality region than in the high quality region. Second, the current TVSQ can affect the TVSQ in the subsequent 16 seconds. Third, the mapping from

STSQ to TVSQ is not only monotone but also concave. Using these properties, I devise an adaptive video transmission algorithm that maximizes the average TVSQ of video users sharing a wireless down-link. The proposed algorithm has provable optimality. Numerical simulation results show that, compared with the STSQ-optimized counterpart, the proposed algorithm reduces the consumption of network resources by 12.5%.

In Chapter 3, I propose a novel OQ metric, which is based on video users' quality expectations and the second-order cumulative distribution function of served video qualities. The proposed metric achieves a linear correlation of 0.84 with the measured OQ. Using the OQ metric, I develop adaptive transmission algorithms to maximize the number of video users who satisfy certain OQ constraints. Two types of OQ constraints are investigated. In particular, if users' video quality expectations are not known, I consider the same constraints on the second-order cumulative distribution functions of all video users. If users' video quality expectations are known, I consider differentiated constraints on each user. For each type of OQ constraints, I devise a virtual queue-based rate-switching algorithm. Furthermore, I devise a threshold-based admission control strategy to block the video users whose OQ constraints cannot be satisfied. To determine the optimal threshold, an online optimization method is proposed to automatically adjust the threshold to its optimal value. Simulation results shows that, compared with the conventional average-STSQ maximizing algorithms, the proposed algorithm reduces the consumption of network resources by 40%.

In Chapter 4, I develop an adaptive scheduling algorithm for efficient scalable video transmission in wireless channels. By modeling the wireless channel as a Markov chain, an MDP model is proposed in which policies that maximize the STSQ predicted by the MS-SSIM index can be computed. By simplifying the scheduling policies obtained from the MDP formulation, I propose an online scheduling algorithm that requirers only limited knowledge of channel dynamics. Simulation results demonstrate the near optimality of the proposed online scheduling algorithm versus a proposed upper bound on performance.

5.2 Future work

Adaptive video transmission systems have been investigated extensively since the last decade. But designing QoE-optimized adaptation algorithms is still far from its maturity. In this section, I enumerate several possible directions for future research.

Predicting the difference of QoE Most existing adaptive transmission algorithms are designed to optimize the QoE metrics such as PSNR, MS-SSIM or the second-order CDF [19, 24, 25, 35, 36, 44, 49, 91]. These QoE metrics, however, are designed to predict the mean opinion scores (MOSs) obtained from the subjective studies [3, 23, 40]. In most subjective studies reported so far, the subjects only report their quality judgments with respect to each test video. Their opinions on the perceptual differences between the test videos are not studied. Hence, the obtained MOSs cannot provide insight into the difference

in QoE. In particular, a larger difference in MOSs may not imply a larger difference in QoEs. This is illustrated in Fig. 5.1. Because a large MOS indicates a better QoE, QoE is an increasing function of MOS. The function, however, may be nonlinear and a larger difference in MOS may correspond to a smaller difference in QoE.

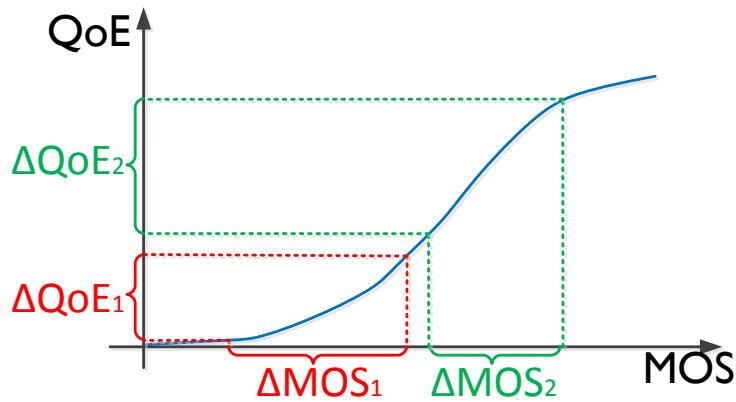


Figure 5.1: The relationship between MOS and QoE. Because a higher MOS implies a better QoE, QoE is an increasing function of MOS. But a larger difference in MOS (see ΔMOS_1 and ΔMOS_2) does not imply a larger difference in QoE (see ΔQoE_1 and ΔQoE_2).

The incapability of MOSs to interpret the difference of QoE is especially critical for wireless video transmission. In wireless networks, video users share the limited network resources (such as a resource block in LTE [9]). Most existing resource allocation algorithms are designed to maximize the sum of the MOSs. But these MOS-optimized resource allocation algorithms may not maximize the sum of the QoEs. For example, let's suppose that there are network resources that can either be allocated to user 1 or user 2. If the

resources are allocated to user 1, its MOS increases by ΔMOS_1 and its QoE increases by ΔQoE_1 . Otherwise, if the resources are allocated to user 2, its MOS increases by ΔMOS_2 and its QoE increases by ΔQoE_2 . If $\Delta\text{MOS}_1 > \Delta\text{MOS}_2$, it seems that allocating the resource to user 1 is more beneficial as it maximizes the sum of the MOSs. But, if the mapping from MOS to QoE is a nonlinear mapping (as shown in Fig. 5.1), it is possible that $\Delta\text{QoE}_1 < \Delta\text{QoE}_2$. Then allocating the resources to user 1 may not maximize the sum of QoEs.

To develop QoE metrics that can reflect the differences in QoE, a more sophisticated design of subjective studies is necessary. For example, in every round of the subjective tests, the subjects could be asked to first view two pairs of videos. Each video pair consists of two distorted versions of the same pristine video. Then, the subjects are asked to judge which pair of videos has larger difference in QoE. Based on the feedback of the subjects, a new QoE metric can be obtained to predict the differences in QoE. Although such a subjective study has not been conducted on videos, a similar study on images has been reported in [21]. Using the maximum likelihood difference scaling (MLDS) method [51], a new quality metric was derived to predict the QoE differences of images. Future research should extend the subjective study in [21] to video QoE assessment. A major limitation of [21] is that the two image pairs involved in the subjective study correspond to the same pristine image. Thus, the proposed QoE metric can only be applied to the case of image broadcasting in which all users view the same image. In a wireless network, different users may watch different images. In the future, the subjective study needs to be

generalized to compare video pairs corresponding to different pristine videos.

Modeling the QoE in natural viewing environments In Chapter 3, we find that the viewers' QoE depends on their quality expectations. The video quality expectation, in turn, depends on the viewing environment. For example, the viewers tend to expect higher quality in a quiet bedroom than in a noisy train station. Thus, viewers' QoE is environment-dependent. The subjective studies reported so far are all conducted in controlled environments [3, 23, 40]. Every subject views the videos on the same device and in the same place. Although controlling the viewing environment is helpful for other researchers to reproduce the subjective study, the measured QoEs cannot reflect the impact of the viewing environment. In a practical video transmission system, the users watch videos in different environments. Thus, developing QoE metrics that could reflect the impact of the viewing environment is important.

One method of modeling the relationship between the viewing environment and the QoE is to perform subjective studies in each type of environment. Based on the subjective studies, a series of QoE models can be built for each environment. An alternative method is to exploit the huge amount of user information recorded by video service providers such as YouTube and Netflix. These records contain important features related to the viewing environments, the served video quality, and the QoE of viewers. For example, the viewing place can be traced by the IP address. The second-order CDF of the video qualities can be derived from the rate of the video streams. The QoE of a video can be deduced from the time spent by the viewers in watching it. Future re-

search could aim at developing a generic QoE model like the one illustrated in Fig. 5.2. The model maps the environment features (e.g., IP address) and the video-quality features (e.g., CDF) to the QoE features (e.g., viewing time).

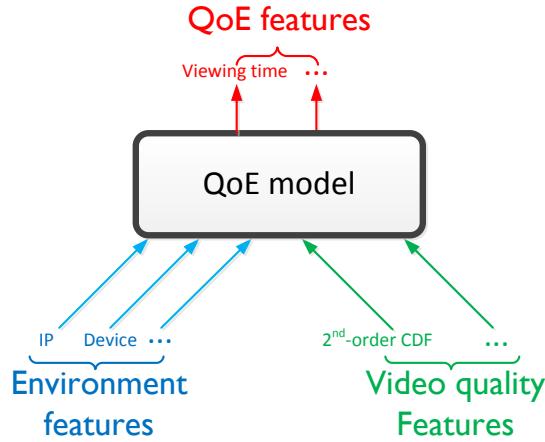


Figure 5.2: A generic QoE model for natural viewing environment. It maps the environment features and the video-quality features to the QoE features.

Adaptive video streaming with transmission delay As illustrated in Fig. 5.3, in an adaptive video transmission system, the video source is compressed into multiple representations or layers, which correspond to different video source rates. The video packets of these representations or layers are first sent to and buffered at the base station. The buffered packets are then forwarded by the base station to the video users. The video users feed back the wireless channel condition, the receiver buffer state, and the decoded video quality to a controller located at the base station. Based on the information fed back by the users, the controller fetches the video packets from the video server and sends the buffered video packets to the receiver. In this disser-

tation and most prior work, it is assumed that the transmission delay from the video server to the base station is negligible [19, 24, 25, 35, 36, 44, 49, 91]. Under this assumption, the base station could fetch any video packet from the video server in a very short time, as if all the video packets have been buffered at the base station. Thus, there is no need to design a separate algorithm to manage the video packets buffered at the base station, and the design of the controller is thus significantly simplified. Since the content delivery network is ubiquitous and the channel connecting the video server and base station has large throughput, the assumption is valid in most cases. For example, the round-trip time from a computer located at The University of Texas at Austin to YouTube is less than 1ms. For pedestrian users, the wireless channel condition varies in every tens of milliseconds [29]. Thus, the 1ms delay is negligible.

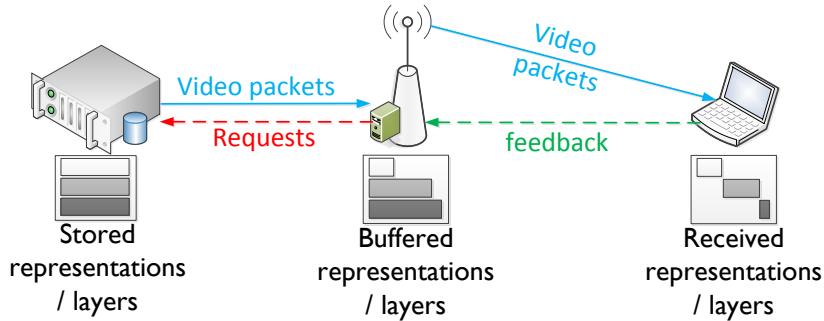


Figure 5.3: An adaptive video transmission system. Video packets are fetched from the server and buffered by the base station. Then, the packets are forwarded to the users.

If the server is far from the base station, the delay may not be ignored. For example, the round-trip time from The University of Texas at Austin to

Youku [5], a Chinese video hosting service, is 250ms. This delay is larger than the time scale at which the wireless channel condition varies. In this case, the buffer state of the base station affects the viewers' QoE because the base station can only forward the video packets that have been buffered. Future work should develop a control algorithm that properly manages the buffer at the base station such that the users' QoE is optimized.

QoE-optimized rate adaptation over HTTP In recent years, HTTP-based adaptive video streaming techniques have been widely used. [8, 53, 55, 62]. Companies such as Apple, Microsoft, and Adobe have developed HTTP-based video streaming protocols [8, 53, 62]. In these HTTP-based video streaming protocols, the videos are encoded into multiple representations. At any moment, the user can dynamically download a segment from an appropriate representation such that the downloading bitrate is adapted to the channel capacity. The rate-adaptation mechanism of HTTP-based streaming is similar to that of rate-switching systems. But the QoE-optimized rate adaptation algorithms proposed in this dissertation cannot be applied to HTTP-based streaming directly. In particular, in HTTP-based streaming, each user determines its video source rates independently. On the contrary, the QoE-optimized rate adaptation algorithms proposed in the dissertation are operated by a centralized controller at the base station.

To optimize the QoE of the videos streamed over HTTP, future research should develop distributed rate adaptation algorithms. For example, the primal-dual decomposition techniques (see [60] and the references therein)

can be employed to decentralize the QoE-optimized algorithm proposed in this dissertation. The decentralized algorithm is illustrated in Fig 5.4. The base station sends a resource price to each user that indicates the difficulty in maximizing the QoE of the user with available network resources. Each user then determines its video source rate by solving an optimization problem that only involves its video source rate and its resource price. Thus, the users can determine their video source rate independently.

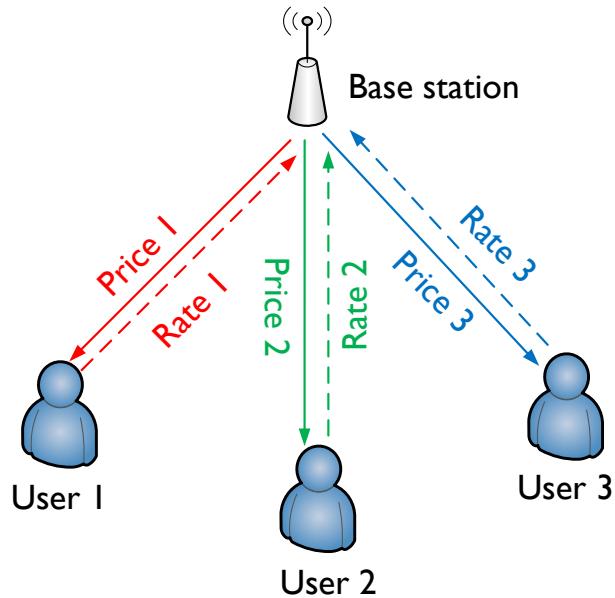


Figure 5.4: The decentralized rate adaptation algorithm. The base station signals the price of network resources to each user. Each user then determines its rate by solving an optimization problem independently.

Appendices

Appendix A

Instructions for Subjective Study

You are taking part in a study to assess the quality of videos. You will be shown a video at the center of the monitor and there will be a rating bar at the bottom, which can be controlled by a mouse on the table. You are to provide feedback on how satisfied you are with your viewing experience up to and including the current moment, i.e., by moving the rating bar in real time based on your satisfaction. The extreme right on the bar is "excellent" and the extreme left is "bad". There is no right or wrong answer.

At the end of the video, you are to provide feedback on how satisfied you are with the overall quality of the video by positioning the cursor on a rating bar shown at the bottom of the screen. The extreme right on the bar is "excellent" and the extreme left is "bad". There is no right or wrong answer.

Appendix B

Gradient Calculation for Model Identification

For the parameter γ , we have $\nabla_\gamma q^{tv}[t] = \left(\frac{\partial q^{tv}[t]}{\partial \gamma_1}, \frac{\partial q^{tv}[t]}{\partial \gamma_2}, \frac{\partial q^{tv}[t]}{\partial \gamma_3}, \frac{\partial q^{tv}[t]}{\partial \gamma_4} \right)^\top$, where

$$\begin{aligned} \frac{\partial q^{tv}[t]}{\partial \gamma_1} &= \frac{\gamma_4 v[t] \exp(-(\gamma_1 v[t] + \gamma_2))}{(1 + \exp(-(\gamma_1 v[t] + \gamma_2)))^2}, \\ \frac{\partial q^{tv}[t]}{\partial \gamma_2} &= \frac{\gamma_4 \exp(-(\gamma_1 v[t] + \gamma_2))}{(1 + \exp(-(\beta_1 v[t] + \beta_2)))^2}, \\ \frac{\partial q^{tv}[t]}{\partial \gamma_3} &= 1, \\ \frac{\partial q^{tv}[t]}{\partial \gamma_4} &= \frac{1}{1 + \exp(-(\gamma_1 v[t] + \gamma_2))}. \end{aligned} \quad (\text{B.1})$$

For the parameter \mathbf{b} , \mathbf{f} and $\boldsymbol{\beta}$, we have

$$\nabla_{\boldsymbol{\xi}} q^{tv}[t] = \frac{\partial q^{tv}[t]}{\partial v[t]} \nabla_{\boldsymbol{\xi}} v[t] = \frac{\gamma_1 \gamma_4 \exp(-(\gamma_1 v[t] + \gamma_2))}{(1 + \exp(-(\gamma_1 v[t] + \gamma_2)))^2} \nabla_{\boldsymbol{\xi}} v[t], \quad (\text{B.2})$$

where $\boldsymbol{\xi}$ can be \mathbf{b} , \mathbf{f} or $\boldsymbol{\beta}$. Thus we only need to compute $\nabla_{\boldsymbol{\xi}} v[t]$. For \mathbf{b} and \mathbf{f} , we have

$$\begin{aligned} \nabla_{\mathbf{b}} v[t] &= (u[t])_{t-1:t-r} + \sum_{d=1}^r f_d \nabla_{\mathbf{b}} v[t-d] \\ \nabla_{\mathbf{f}} v[t] &= (v[t])_{t-1:t-r} + \sum_{d=1}^r f_d \nabla_{\mathbf{f}} v[t-d]. \end{aligned} \quad (\text{B.3})$$

For $\boldsymbol{\beta}$, we have

$$\nabla_{\boldsymbol{\beta}} v[t] = \sum_{d=0}^r b_d \nabla_{\boldsymbol{\beta}} u[t-d] + \sum_{d=1}^r f_d \nabla_{\boldsymbol{\beta}} v[t-d], \quad (\text{B.4})$$

where $\nabla_{\beta} u[t] = \left(\frac{\partial u[t]}{\partial \beta_1}, \frac{\partial u[t]}{\partial \beta_2}, \frac{\partial u[t]}{\partial \beta_3}, \frac{\partial u[t]}{\partial \beta_4} \right)^T$ can be computed similarly as (B.1).

It may be seen from (B.3) and (B.4) that $\nabla_b v[t]$, $\nabla_f v[t]$ and $\nabla_{\beta} v[t]$ can be recursively computed. The stability of the recursions can be ensured by the following lemma.

Lemma B.0.1 (Stability of recursive gradient calculation). *If the roots of polynomial $1 - \sum_d f_d z^{-d}$ are confined within the unit circle of the complex plane, the recursive gradient calculation is stable.*

Proof. In (B.3) and (B.4), the gradients of $\nabla_b v[t]$, $\nabla_f v[t]$ and $\nabla_{\beta} v[t]$ are actually the outputs of IIR filters, where the denominator of the transfer function is $1 - \sum_d f_d z^{-d}$. Thus the roots of $1 - \sum_d f_d z^{-d}$ determines the stability of the recursive calculation process and the lemma is proved. \square

To calculate the gradient using (2.19), we also need to know the initial values of $(\hat{q}(t, \theta))_{1:r}$ to calculate $\left(\frac{\partial \hat{q}(t, \theta)}{\partial \theta_i} \right)_{1:r}$. For the purpose of model training, we simply set $(\hat{q}(t, \theta))_{1:r} = (q^{tv}(t, \theta))_{1:r}$. Thus, we have $\frac{\partial \hat{q}(t, \theta)}{\partial \theta_i} = \frac{\partial q^{tv}[t]}{\partial \theta_i} = 0, \forall t \leq r$.

Appendix C

Proof for Theorem 3.2.1

Proof. Since $\{x - q^{st}(t), 0\}$ is an convex function of x , $F_m^{(2)}(x)$ is a linear combination of $\{x - q^{st}(t), 0\}$ and is thus also convex. Without loss of generality, we assume the function $h(x)$ is also convex. Otherwise, we can simply replace $h(x)$ with another function whose epigraph is the convex hull of $h(x)$'s epigraph. Let $x^i < x^j$, where $i, j \in \mathcal{I}$. If (3.7) is satisfied, we have $F_m^{(2)}(x^i) \leq h(x^i)$ and $F_m^{(2)}(x^j) \leq h(x^j)$. For any $\lambda \in [0, 1]$ and $x = \lambda x^i + (1 - \lambda)x^j$, we have $F_m^{(2)}(x) = F_m^{(2)}(\lambda x^i + (1 - \lambda)x^j) \leq \lambda F_m^{(2)}(x^i) + (1 - \lambda)F_m^{(2)}(x^j) \leq \lambda h(x^i) + (1 - \lambda)h(x^j) = \bar{h}(x)$. Because $[0, 100]$ is a compact set, the convexity of $h(x)$ implies its continuity. Therefore, $h(x)$ can be approximated by piece-wise linear functions to arbitrary accuracy. \square

Appendix D

Proof of Theorem 3.3.1

Proof. Note that Algorithm 6 can be viewed as a stochastic approximation algorithm [39] with an associated mean ordinary differential equation (ODE)

$$\frac{dk(t)}{dt} = \mathbb{E}[\mathbf{Y}(k(t))], \quad (\text{D.1})$$

where $\mathbf{Y}(k)$ is a random variable that denotes the updating direction when the threshold is k , we have

$$\mathbb{E}[\mathbf{Y}(k(t))] = 1 - 2p^L(k(t)). \quad (\text{D.2})$$

According to Assumption 2, there exists a unique k' such that $p^L(k') = 1/2$ and $\mathbb{E}[\mathbf{Y}(k')] = 0$. By the monotonicity of $p^L(k)$, we have $\mathbb{E}[\mathbf{Y}(k)] > 0$, $\forall k < k'$ and $\mathbb{E}[\mathbf{Y}(k)] < 0$, $\forall k > k'$. If we define a function $V(k) = 1/2(k - k')^2$, then

$$\begin{aligned} \frac{dV(k(t))}{dt} &= (k(t) - k') \frac{dk(t)}{dt} \\ &= (k(t) - k') (1 - 2p^L(k(t))) \\ &= -(k(t) - k') (2p^L(k(t)) - 2p^L(k')) \\ &= -2(k(t) - k') (p^L(k(t)) - p^L(k')) \end{aligned}$$

For $\forall k < k'$, we have $p^L(k) - p^L(k') \leq M(k - k')$. For $\forall k > k'$, $p^L(k) - p^L(k') \geq M(k - k')$. In sum, we have

$$\frac{dV(k(t))}{dt} \leq -2M(k(t) - k')^2.$$

By Theorem 5.4.1 in [39, p.145], we have $\lim_{n \rightarrow \infty} k^n = k'$. Next, we prove that $k' \in [k^* - \delta, k^*)$.

We define a binary random variable S_u such that $S_u = 1$ if video user u satisfies the QoE constraints $\{F^{(2)}(x_i; q_u) \leq h_i, \forall x_i \in \mathcal{I}\}$. Otherwise, we define $S_u = 0$. Denote by $\mathcal{U}^L = \{u_1, \dots, u_L\}$ the indices of the L admitted video users in an iteration of Algorithm 5. Let π_k be the joint distribution of the variables $\{S_u, \forall u \in \mathcal{U}^L\}$ when the admission threshold is k . Then, we have

$$\begin{aligned} p^L(k) &= \mathbb{P}^{\pi_k}(S_u = 1, \forall u \in \mathcal{U}^L) \\ &= \mathbb{P}^{\pi_k}(S_{u_1} = 1) \mathbb{P}^{\pi_k}(S_{u_\ell} = 1, 2 \leq \ell \leq L | S_{u_1} = 1). \end{aligned} \tag{D.3}$$

Since the users are competing with each other for network resources, if the QoE constraints of a video user are satisfied, the probability of satisfying other users' QoE constraints is reduced. Thus,

$$\begin{aligned} &\mathbb{P}^{\pi_k}(S_{u_\ell} = 1, \forall 2 \leq \ell \leq L | S_{u_1} = 1) \\ &\leq \mathbb{P}^{\pi_k}(S_{u_\ell} = 1, \forall 2 \leq \ell \leq L). \end{aligned} \tag{D.4}$$

Substitute (D.4) into (D.3) yields

$$\begin{aligned} p^L(k) &\leq \mathbb{P}^{\pi_k}(S_{u_1} = 1) \mathbb{P}^{\pi_k}(S_{u_\ell} = 1, \forall 2 \leq \ell \leq L) \\ &\leq \mathbb{P}^{\pi_k}(S_{u_1} = 1) \mathbb{P}^{\pi_k}(S_{u_2} = 1) \mathbb{P}^{\pi_k}(S_{u_\ell} = 1, \forall 3 \leq \ell \leq L) \\ &\leq \prod_{\ell=1}^L \mathbb{P}^{\pi_k}(S_{u_\ell} = 1) \\ &= (1 - e(k))^L. \end{aligned} \tag{D.5}$$

Since $L \geq \frac{-\log 2}{\log(1-e(k^*-\delta))}$, it follows that $p^L(k^* - \delta) \leq (1 - e(k^* - \delta))^L \leq 1/2$.

Because of the monotonicity of $p^L(k)$, we know that $k^* - \delta \leq k'$. Moreover, because $p^L(k') = 1/2 > 0$, we have $k' < k^*$. \square

Appendix E

Transition Probability

Notations: Let $\mathbf{1}$ is the unit vector of all-ones and $\mathbf{0}$ is the zero vector. $\max\{\mathbf{a}, \mathbf{b}\}$ and $\min\{\mathbf{a}, \mathbf{b}\}$ are the componentwise maximum and minimum of vector \mathbf{a} and \mathbf{b} , respectively. $\mathbb{1}(\cdot)$ is the indicator function.

Let $\mathbf{s}_t = (\mathbf{c}_t, \mathbf{v}_t)$ and $\mathcal{U}_{\mathbf{s}_t}$ be the system state and the corresponding feasible control set at slot t , where $\mathbf{c}_t = (x_t, y_t)$ and $\mathbf{v}_t = (\mathbf{v}_t^{\text{pre}}, \mathbf{v}_t^{\text{I}}, \mathbf{v}_t^{\text{post}})$. At the beginning of each slot, one frame is decoded and played out. We let $\mathbf{v}_t^+ = (\mathbf{v}_t^{\text{pre}+}, \mathbf{v}_t^{\text{I}+}, \mathbf{v}_t^{\text{post}+})$ denote the buffer state right after the first frame is displayed. If $f_t^{\text{I}} = 1$, i.e., the decoded frame is an I frame, then the frame set \mathcal{J} becomes the next I frame, i.e., the $F^{\text{GOP}^{\text{th}}}$ frame. Hence, the buffer state is

$$\mathbf{v}_t^{\text{I}+} = \left(F^{\text{GOP}}, \sum_{\ell=1}^L \mathbb{1}(b_{\ell}^{\text{post}} \geq F^{\text{GOP}}) \right), \quad (\text{E.1})$$

where $\sum_{\ell=1}^L \mathbb{1}(b_{\ell}^{\text{post}} \geq F^{\text{GOP}})$ is the number of received layers in the next I frame. Meanwhile, \mathcal{J}^{pre} becomes the first $F^{\text{GOP}} - 1$ frames and $\mathcal{J}^{\text{post}}$ contains the frames whose index is larger than F^{GOP} . Thus, we have

$$\mathbf{v}_t^{\text{pre}+} = \min \left\{ \mathbf{v}_t^{\text{post}}, (F^{\text{GOP}} - 1)\mathbf{1} \right\} \quad (\text{E.2})$$

and

$$\mathbf{v}_t^{\text{post}+} = \max \left\{ \mathbf{v}_t^{\text{post}} - F^{\text{GOP}}\mathbf{1}, \mathbf{0} \right\}. \quad (\text{E.3})$$

If the decoded frame is not an I frame, the frame set $\mathcal{J}^{\text{post}}$ will not be affected and the buffer state $\mathbf{v}_t^{\text{post}}$ does not change. $\mathbf{v}_t^{\text{pre}}$ becomes

$$\mathbf{v}_t^{\text{pre+}} = \max \{ \mathbf{v}_t^{\text{pre}} - \mathbf{1}, \mathbf{0} \}. \quad (\text{E.4})$$

Summarizing (E.1), (E.2), (E.3) and (E.4), we have

$$\begin{aligned} \mathbf{v}_t^{\text{pre+}} &= \begin{cases} \min \{ \mathbf{v}_t^{\text{post}}, (F^{\text{GOP}} - 1)\mathbf{1} \} & \text{if } f_t^I = 1, \\ \max \{ \mathbf{v}_t^{\text{pre}} - \mathbf{1}, \mathbf{0} \} & \text{if } f_t^I \neq 1, \end{cases} \\ \mathbf{v}_t^{I+} &= \begin{cases} \left(F^{\text{GOP}}, \sum_{\ell=1}^L \mathbb{1}(b_\ell^{\text{post}} \geq F^{\text{GOP}}) \right) & \text{if } f_t^I = 1, \\ (f_t^I - 1, \ell_t^I) & \text{if } f_t^I \neq 1, \end{cases} \end{aligned}$$

and

$$\mathbf{v}_t^{\text{post+}} = \begin{cases} \max \{ \mathbf{v}_t^{\text{post}} - F^{\text{GOP}}\mathbf{1}, \mathbf{0} \} & \text{if } f_t^I = 1, \\ \mathbf{v}_t^{\text{post}} & \text{if } f_t^I \neq 1. \end{cases}$$

After the first frame is displayed, the transmitter begins to sequentially transmit the collection of video data units indicated by the action $\mathcal{U}_t = \mu(\mathbf{s}_t) = \{(f_1, \ell_1), \dots, (f_{|\mathcal{U}_t|}, \ell_{|\mathcal{U}_t|})\}$. Let $\Delta\mathcal{U}_t = \{(f_1, \ell_1), \dots, (f_{n_t}, \ell_{n_t})\}$ denote the completely received data units by the end of the slot, where n_t is the number of received data units. Among the data units in $\Delta\mathcal{U}_t$, let $\Delta\mathbf{v}_t^{\text{pre}} = (\Delta b_1^{\text{pre}}, \Delta b_2^{\text{pre}}, \dots, \Delta b_L^{\text{pre}})$ be the number of newly received data units for each layer in frame set \mathcal{J}^{pre} . Similarly, we denote $\Delta\mathbf{v}_t^{\text{post}} = (\Delta b_1^{\text{post}}, \Delta b_2^{\text{post}}, \dots, \Delta b_L^{\text{post}})$ as the number of newly received data units for each layer in frame set $\mathcal{J}^{\text{post}}$ and $\Delta\ell^I$ as the number of received data units for \mathcal{J} . At the beginning of the $(t + 1)^{\text{th}}$ slot, we have the following state transition

relationship

$$\mathbf{v}_{t+1}^{\text{pre}} = \mathbf{v}_t^{\text{pre+}} + \Delta \mathbf{v}_t^{\text{pre}}, \quad (\text{E.5})$$

$$\mathbf{v}_{t+1}^{\text{I}} = (f_t^{\text{I+}}, \ell_t^{\text{I+}} + \Delta \ell^{\text{I}}), \quad (\text{E.6})$$

$$\mathbf{v}_{t+1}^{\text{post}} = \mathbf{v}_t^{\text{post+}} + \Delta \mathbf{v}_t^{\text{post}}. \quad (\text{E.7})$$

The amount of video data in $\Delta \mathcal{U}_t$, denoted by $\Phi(\mathbf{v}_t, \Delta \mathcal{U}_t)$, can be estimated according to buffer state \mathbf{v}_t^{I} and the rate-quality model introduced in Section 4.2.3. Specifically, for each data unit in $\Delta \mathcal{U}_t$, we first determine whether it belongs to an I frame or a P frame according to \mathbf{v}_t^{I} and then estimate the amount of data by the rate-quality model. The set $\Delta \mathcal{U}_t$ records the completely transmitted data units up to $(f_{n_t}, \ell_{n_t})^{\text{th}}$ data unit. However, data unit $(f_{n_t+1}, \ell_{n_t+1})$ is only partially received. Denoting the amount of data in unit $(f_{n_t+1}, \ell_{n_t+1})$ by $\tilde{\Phi}(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t)$, the amount of received data is at least $\Phi(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t)$ and at most $\Phi(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t) + \tilde{\Phi}(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t)$. Assuming the physical layer packet length is L^{PHY} , there is $N = \lceil \frac{x_t}{L^{\text{PHY}}} \rceil$ packet transmissions during a time slot. The number of successfully transmitted packets is at least $N_l = \lceil \frac{\Phi(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t)}{L^{\text{PHY}}} \rceil$ and is less than $N_h = \lceil \frac{\Phi(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t) + \tilde{\Phi}(\mathbf{v}_t^{\text{I}}, \Delta \mathcal{U}_t)}{L^{\text{PHY}}} \rceil$. As assumed in Section 4.2.4, the channel state is constant over each slot. Thus, the packet losses are independent within each slot. The number of successful packet transmissions in a slot is distributed binomially. Hence, the state transition probability from $\mathbf{s}_t = (\mathbf{c}_t, \mathbf{v}_t)$ to $\mathbf{s}_{t+1} = (\mathbf{c}_{t+1}, \mathbf{v}_{t+1})$ is

$$\mathbb{P}_\mu(\mathbf{s}_{t+1} | \mathbf{s}_t) = \left[\sum_{n_t=N_l}^{N_h-1} \binom{N}{n_t} y_t^{N-n_t} (1-y_t)^{n_t} \right] \mathbb{P}(\mathbf{c}_{t+1} | \mathbf{c}_t), \quad (\text{E.8})$$

where the first multiplicative term is the transition probability of the receiver buffer state from \mathbf{v}_t to \mathbf{v}_{t+1} and the second term is the transition probability of the channel state from \mathbf{c}_t to \mathbf{c}_{t+1} .

Appendix F

Computation of t_μ and $\tilde{\mathbb{P}}_\mu$

Let $v_t^{\overline{W}}$ be the number of buffered packets outside the window. Noted that, when the system moves in $\mathcal{S}_{\overline{W}}$, the system always schedules as many enhancement layer data units as possible. Hence, $v_t^{\overline{W}}$ and f_t^I contain all the information about the buffer state \mathbf{v}_t . We can further simplify the state representation $(\mathbf{c}_t, \mathbf{v}_t)$ to $(\mathbf{c}_t, v_t^{\overline{W}}, f_t^I)$ when $(\mathbf{c}_t, \mathbf{v}_t) \in \mathcal{S}_{\overline{W}}$. All the states in $\mathcal{S}_{\overline{W}}$ correspond to some states with $v_t^{\overline{W}} \geq 0$. All the states in \mathcal{S}_W correspond to some states with $v_t^{\overline{W}} < 0$.

When the system evolves in $\mathcal{S}_{\overline{W}}$, at the beginning of a slot t , the state $v_t^{\overline{W}}$ first decreases by $\Delta v_{\text{dec}}^{\overline{W}}(\mathbf{s}_t)$ when the current frame is displayed. Then, the transmitter schedules as many enhancement layer data units as possible. At the end of the slot, $v_t^{\overline{W}}$ increases by $\Delta v_{\text{inc}}^{\overline{W}}(\mathbf{s}_t)$. Because the quantity $\Delta v^{\overline{W}}(\mathbf{s}_t) = \Delta v_{\text{inc}}^{\overline{W}}(\mathbf{s}_t) - \Delta v_{\text{dec}}^{\overline{W}}(\mathbf{s}_t)$ only depends on state \mathbf{s}_t , the state $v_t^{\overline{W}}$ varies like a random walk but with Markovian step-size $\Delta v^{\overline{W}}(\mathbf{s}_t)$. Now we need to compute, given the starting state \mathbf{s}_t with $v_t^{\overline{W}} \geq 0$, how long it takes to jump to a state where $v_t^{\overline{W}} < 0$. Let $\Delta v_{\max}^{\overline{W}} = \max_{\mathbf{s} \in \mathcal{S}_{\overline{W}}} \{|\Delta v^{\overline{W}}(\mathbf{s})|\}$ be the maximum step size. We define the k^{th} level set as

$$\mathcal{S}_k = \mathcal{C} \times \{k\Delta v_{\max}^{\overline{W}} + 1, \dots, (k+1)\Delta v_{\max}^{\overline{W}}\} \times \{1, \dots, F^{\text{GOP}}\}, \quad (\text{F.1})$$

where $k \geq -1$. After the system moves to $\mathcal{S}_{\overline{W}}$, the system state transits in level set \mathcal{S}_k with $k \geq 0$ and then jumps to a state in \mathcal{S}_{-1} . If we concatenate all these level set as $\mathcal{S}_{\mathcal{K}} = \cup_{k=-1}^{\infty} \mathcal{S}_k$, then the state transition matrix is a banded infinite matrix of the following form

$$\begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & & & \\ \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & & \\ & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \\ & & \mathbf{A}_0 & \mathbf{A}_1 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}.$$

The blocks \mathbf{B}_0 , \mathbf{B}_1 , \mathbf{A}_0 , \mathbf{A}_1 and \mathbf{A}_2 are all square matrices of dimension $d = |\mathcal{C}| \Delta v_{\max}^{\overline{W}} F^{\text{GOP}}$. When the system state jumps to \mathcal{S}_{Δ} , the state lies in level set \mathcal{S}_0 . Now we need to compute how long the system takes to reach \mathcal{S}_{-1} for the first time. This problem of continuous time quasi-birth-death processes was essentially solved by Neuts in the early 1980s [58]. The following derivation follows similarly the one in Neuts' book but for the discrete time case.

Let $G^{(\nu)}(k, x)_{j,j'}$ be the probability that, starting from the j^{th} state in level set \mathcal{S}_n , the system state hits level set $\mathcal{S}_{(n-\nu)}$ for the first time after x movements in which there are k left movements. The hitting point is the j' th state in $\mathcal{S}_{(n-\nu)}$. Let $\mathbf{G}^{(\nu)}(k, x)$ be the matrix with $G^{(\nu)}(k, x)_{j,j'}$ as the (j, j') th entry. Applying the Z transform to this distribution we have

$$\hat{\mathbf{G}}^{(\nu)}(z, s) = \sum_{k=0}^{\infty} z^k \left(\sum_{x=0}^{\infty} \mathbf{G}^{(\nu)}(k, x) s^x \right). \quad (\text{F.2})$$

Denoting $\hat{\mathbf{G}}^{(1)}(z, s)$ by $\hat{\mathbf{G}}(z, s)$, it can be proved that $\hat{\mathbf{G}}^{(\nu)}(z, s) = (\hat{\mathbf{G}}(z, s))^{\nu}$ [58]. Conditioning on the state visited in the first state transition, we have

$$\hat{\mathbf{G}}(z, s) = \mathbf{A}_0 z s + \mathbf{A}_1 \hat{\mathbf{G}}(z, s) s + \mathbf{A}_2 \hat{\mathbf{G}}^2(z, s) s. \quad (\text{F.3})$$

Now, define $\mathbf{G} = \hat{\mathbf{G}}(1, 1)$, $\mathbf{C}_0 = (\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{A}_0$, $\mathbf{C}_1 = (\mathbf{I} - \mathbf{A}_1)^{-1}$ and $\mathbf{C}_2 = (\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{A}_2$. By simple matrix manipulation of (F.3), we have

$$\mathbf{G} = \mathbf{C}_0 + \mathbf{C}_2\mathbf{G}^2. \quad (\text{F.4})$$

We can compute \mathbf{G} by successive substitutions starting with the zero matrix. Because $\mathbf{G} = \hat{\mathbf{G}}(z, s)|_{z=1, s=1}$, its entry $\mathbf{G}_{j,j'}$ is actually the conditional probability that, given the initial state $j \in \mathcal{S}_0$, the system will move to \mathcal{S}_{-1} for the first time at the $(j')^{\text{th}}$ state. Hence, we can find $\tilde{\mathbb{P}}_\mu(\cdot|\cdot)$ from \mathbf{G} .

Let $\mathbf{M} = \frac{d\hat{\mathbf{G}}(z, s)}{ds}|_{z=1, s=1}$. The $(j, j')^{\text{th}}$ entry $\mathbf{M}_{j,j'} = \sum_{k=0}^{\infty} \sum_{x=0}^{\infty} (x\mathbf{G}_{j,j'}(k, x))$ is the conditional expectation of the time that, given the initial state $j \in \mathcal{S}_0$, the system takes to hit \mathcal{S}_{-1} for the first time at the $j^{\text{'th}}$ state. Differentiating both sides of (F.3) with respect to s , then setting $s = 1$, $z = 1$, we have

$$\mathbf{M} = \mathbf{A}_0 + \mathbf{A}_1\mathbf{M} + \mathbf{A}_1\mathbf{G} + \mathbf{A}_2\mathbf{G}^2 + \mathbf{A}_2(\mathbf{GM} + \mathbf{MG}). \quad (\text{F.5})$$

Using the definition of \mathbf{C}_1 and \mathbf{C}_2 , equation (F.5) can be simplified as

$$\mathbf{M} = \mathbf{C}_1\mathbf{G} + \mathbf{C}_2(\mathbf{GM} + \mathbf{MG}). \quad (\text{F.6})$$

We can compute \mathbf{M} by successive substitutions starting with the zero matrix.

We define a vector

$$\mathbf{t} = \mathbf{M}\mathbf{1}. \quad (\text{F.7})$$

The j^{th} entry $\mathbf{t}_j = \sum_{j'=0}^d \sum_{k=0}^{\infty} \sum_{x=0}^{\infty} (x\mathbf{G}_{j,j'}(k, x))$ is the conditional expectation of the time that the system takes to go back to \mathcal{S}_{-1} given the initial state $j \in \mathcal{S}_0$, i.e., $t_\mu(j)$. Therefore, we can compute $t_\mu(\cdot)$ and $\tilde{\mathbb{P}}_\mu(\cdot|\cdot)$ using (F.4) and (F.7).

Appendix G

Simulation Settings in Chapter 4

We employ the FSMC channel model proposed in [89] to model the dynamics of Rayleigh fading channels. The SNR at the receiver is partitioned into $|\mathcal{C}|$ regions using the algorithm proposed in [89]. Let Λ_i be the partition thresholds, where $\Lambda_0 = -\infty$ and $\Lambda_{|\mathcal{C}|} = \infty$. Let $\tilde{\Lambda}_k$ be the representative SNR in the k^{th} region. For Rayleigh fading channels, we have

$$\tilde{\Lambda}_k = \frac{\int_{\Lambda_{k-1}}^{\Lambda_k} \lambda p(\lambda) d\lambda}{\int_{\Lambda_{k-1}}^{\Lambda_k} p(\lambda) d\lambda}, \quad (\text{G.1})$$

where $p(\lambda) = \frac{1}{\Lambda^{\text{avg}}} \exp(-\frac{\lambda}{\Lambda^{\text{avg}}})$ is the probability distribution function of the received instantaneous SNR of Rayleigh fading channels with average SNR Λ^{avg} . According to [89], the state transition probability \mathbf{P}^c is computed as

$$\mathbf{P}_{i,j}^c = \begin{cases} \frac{\mathcal{K}(\Lambda_j)\Delta T}{\pi_i} & \text{if } j = i + 1, \\ \frac{\mathcal{K}(\Lambda_i)\Delta T}{\pi_i} & \text{if } j = i - 1, \\ 1 - \frac{\mathcal{K}(\Lambda_j)\Delta T}{\pi_i} - \frac{\mathcal{K}(\Lambda_i)\Delta T}{\pi_i} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

where $\pi_i = \int_{\Lambda_{i-1}}^{\Lambda_i} p(\lambda) d\lambda$. $\mathcal{K}(\Lambda_i) = \sqrt{\frac{2\pi\Lambda_i}{\Lambda^{\text{avg}}}} f^d \exp(-\frac{\Lambda_i}{\Lambda^{\text{avg}}})$ is the level crossing rate of threshold Λ_i where f^d is the Doppler frequency. The coherence time is estimated via $t_{\text{cor}} = 0.423/f^d$. In our simulations, we set $|\mathcal{C}| = 4$ and $\Lambda^{\text{avg}} = 10\text{dB}$.

We assume that BPSK, QPSK and 8PSK are used for modulation. The symbol error rate p_k^s in the k^{th} SNR region is $p_k^s = 2Q(\sqrt{2\tilde{\Lambda}_k} \sin \frac{\pi}{2^M})$, where $M = 1, 2, 3$ for BPSK, QPSK and 8PSK, respectively. Each packet contains 2048 symbols. Thus, the packet length $L^{\text{PHY}} = 2048 \times M$, where $M = 1, 2$ and 3 for BPSK, QPSK and 8PSK, respectively. The transmission time for each packet is $\Delta t = 1.5\text{ms}$. The transmission data rate is given by $x_k = \frac{\Delta T}{\Delta t} L^{\text{PHY}}$. The packet error rate is given by $y_k = 1 - (1 - p_k^s)^{2048}$. The modulation scheme for k^{th} channel states is chosen such that the throughput $x_k(1 - y_k)$ is maximized.

The proposed dynamic scheduling algorithm was evaluated on the test sequences “foreman”, “bus”, “flower”, “mobile” and “Paris” [2]. These video sequences were encoded using H.264/SVC reference software JSVM [68] with 3 layers. The GOP length was fixed at $F^{\text{GOP}} = 16$. The encoding parameters and rate-quality model parameters are listed in Table G.1. The parameters r_ℓ^I and r_ℓ^P are measured in megabits and q^ℓ is measured in MS-SSIM index values. The quantization parameters (QP) were chosen such that the the data rate of the base layer is lower than the average channel throughput. The Lagrangian multipliers for motion estimation and mode decision were set as $QP - 2$. We employ this configuration to make sure that the channel is at least good enough to support the base layer. Otherwise, any scheduling policy cannot provide acceptable visual quality.

Table G.1: The encoding parameters and rate-quality model parameters of the tested sequences.

sequences	Layer 1 (base layer)				Layer 2				Layer 3			
	QP	ω_1^I	ω_1^P	q^1	QP	ω_2^I	ω_2^P	q^2	QP	ω_3^I	ω_3^P	q^3
foreman	31	0.0612	0.0149	0.9362	27	0.0638	0.0245	0.0222	26	0.0228	0.0290	0.0045
bus	39	0.0565	0.0140	0.8408	35	0.0581	0.0203	0.0644	33	0.0329	0.0297	0.0277
flower	40	0.0846	0.0130	0.9117	36	0.075	0.0225	0.0400	35	0.028	0.0268	0.008
mobile	40	0.0972	0.0133	0.8839	37	0.0782	0.0222	0.0408	36	0.0309	0.0284	0.0121
Paris	33	0.1121	0.0146	0.9487	28	0.1014	0.0209	0.0241	27	0.0387	0.0206	0.0041

Bibliography

- [1] Technical Report [On line]. Available: <https://webspace.utexas.edu/cc39488/pdf/report.pdf>.
- [2] Test sequences [On line]. Available: <http://trace.eas.asu.edu/yuv/>.
- [3] LIVE image quality assessment database [On line]. Available: <http://live.ece.utexas.edu/research/quality/subjective.htm>.
- [4] Netflix. <https://signup.netflix.com/>, 2013.
- [5] Youku. <http://www.youku.com/>, 2013.
- [6] Youtube. <https://www.youtube.com/>, 2013.
- [7] 3GPP. Further advancements for E-UTRA physical layer aspects. TR 36.814, 3rd Generation Partnership Project (3GPP), 2010.
- [8] Adobe Systems. HTTP dynamics streaming. <http://www.adobe.com/products/hds-dynamic-streaming.html>, Mar. 2012.
- [9] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall. Lte: the evolution of mobile broadband. *IEEE Communications Magazine*, 47(4):44–51, 2009.

- [10] M. Barkowsky, B. Eskofier, R. Bitto, J. Bialkowski, and A. Kaup. Perceptually motivated spatial and temporal integration of pixel based video quality measures. In *Welcome to Mobile Content Quality of Experience*, pages 1 – 7, Mar. 2007.
- [11] A. Barron, J. Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743 –2760, Oct. 1998.
- [12] F. Bellard and M. Niedermayer. FFmpeg. <http://ffmpeg.org/>, 2012.
- [13] Dimitri Bertsekas and Robert Gallager. *Data networks (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
- [14] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3rd edition, 2005.
- [15] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [16] S. Brandt. *Statistical and Computational Methods in Data Analysis*. North Holland, 1983.
- [17] Center for Perceptual Systems, The University of Texas at Austin. The XGL Toolbox. <http://svi.cps.utexas.edu/software.shtml>, 2012.
- [18] J. Chakareski, P. A. Chou, and B. Aazhang. Computing rate-distortion optimized policies for streaming media to wireless clients. In *Proceedings of Data Compression Conference*, pages 53–62, 2002.

- [19] J. Chakareski and P. Frossard. Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources. *IEEE Transactions on Multimedia*, 8(2):207–218, 2006.
- [20] D. M. Chandler and S. S. Hemami. VSNR: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Trans. Image Process.*, 16(9):2284–2298, Sept. 2007.
- [21] C. Charrier, K. Knoblauch, L.T. Maloney, A.C. Bovik, and A.K. Moorthy. Optimizing multiscale ssim for compression via mlds. *Image Processing, IEEE Transactions on*, 21(12):4682–4694, 2012.
- [22] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath Jr., and A. C. Bovik. A dynamic system model of time-varying subjective quality of video streams over HTTP. In *38th International Conference on Acoustics, Speech, and Signal Processing*, May 2013.
- [23] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath Jr., and A. C. Bovik. A model of the time-varying subjective quality of HTTP video streams. *submitted to IEEE Transactions on Image Processing*, 2013.
- [24] C. Chen, X. Q. Zhu, G. de Veciana, A. C. Bovik, and R. W. Heath Jr. Rate adaptation and admission control for video transmission with subjective quality constraints. *in preparation*, 2013.

- [25] Chao Chen, R.W. Heath Jr., A.C. Bovik, and G. de Veciana. A markov decision model for adaptive scheduling of stored scalable videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(6):1081–1095, 2013.
- [26] Chao Chen, Robert W. Heath Jr., Alan C. Bovik, and Gustavo de Veciana. Adaptive policies for real-time video transmission: a Markov decision process framework. In *18th IEEE International Conference on Image Processing*, Sept. 2011.
- [27] P. A. Chou and Zhourong Miao. Rate-distortion optimized streaming of packetized media. *IEEE Trans. Multimedia*, 8(2):390–404, Apr. 2006.
- [28] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017 Feb. 2013.
- [29] Pramod Viswanath David Tse. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
- [30] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the impact of video quality on user engagement. In *Proceedings of the ACM SIGCOMM 2011 conference*, pages 362–373, 2011.
- [31] F. Fu and M. van der Schaar. A new systematic framework for autonomous cross-layer optimization. *IEEE Trans. Veh. Technol.*, 58(4):1887–1903, May. 2009.

- [32] Fangwen Fu and M. van der Schaar. A systematic framework for dynamically optimizing multi-user wireless video transmission. *IEEE J. Sel. Areas Commun.*, 28(3):308–320, Apr. 2010.
- [33] Fangwen Fu and M. van der Schaar. Structural solutions for dynamic scheduling in wireless multimedia transmission. *IEEE Trans. Circuits Syst. Video Technol.*, 22(5):727 –739, may 2012.
- [34] Xiangdong He and Haruhiko Asada. A new method for identifying orders of input-output models for nonlinear dynamic systems. In *American Control Conference, 1993*, pages 2520 –2523, June 1993.
- [35] H. Hu, X. Zhu, Y. Wang, R. Pan, J. Zhu, and F. Bonomi. Proxy-based multi-stream scalable video adaptation over wireless networks using subjective quality and rate models. *IEEE Transactions on Multimedia*, PP(99):1–1, 2013.
- [36] Jianwei Huang, Zhu Li, Mung Chiang, and A.K. Katsaggelos. Joint source adaptation and resource allocation for multi-user wireless video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(5):582–595, 2008.
- [37] ITU-R Recommendation BT.500-13. Methodology for the subjective assessment of the quality of television pictures. http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-13-201201-I!!PDF-E.pdf, Jan. 2012.

- [38] V. Joseph and G. de Veciana. Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. In *2012 Proceedings IEEE INFOCOM*, pages 567–575, 2012.
- [39] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Press, 2003.
- [40] Laboratory for Image and Video Engineering, The University of Texas at Austin. LIVE Video Quality Database. http://live.ece.utexas.edu/research/quality/live_video.html, 2012.
- [41] P. Le Callet, C. Viard-Gaudin, and D. Barba. A convolutional neural network approach for objective video quality assessment. *IEEE Transactions on Neural Networks*, 17(5):1316 –1327, sept. 2006.
- [42] Weiping Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Trans. Circuits Syst. Video Technol.*, 11(3):301–317, Mar. 2001.
- [43] Hsin-Piao Lin and Ming-Jian Tseng. Two-layer multistate Markov model for modeling a 1.8 GHz narrow-band wireless propagation channel in urban Taipei city. *IEEE Trans. Veh. Technol.*, 54(2):435–446, Mar. 2005.
- [44] K.C. Lin, Wei-Liang Shen, Chih-Cheng Hsu, and Cheng-Fu Chou. Quality-differentiated video multicast in multirate wireless networks. *IEEE Transactions on Mobile Computing*, 12(1):21–34, 2013.

- [45] H. Liu, N. Klomp, and I. Heynderickx. A no-reference metric for perceived ringing artifacts in images. *IEEE Trans. Circuits Syst. Video Technol.*, 20(4):529–539, 2010.
- [46] Lennart Ljung. *System identification: theory for the user*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.
- [47] D.J. Love, R.W. Heath Jr., V.K.N. Lau, D. Gesbert, B.D. Rao, and M. Andrews. An overview of limited feedback in wireless communication systems. *IEEE J. Sel. Areas Commun.*, 26(8):1341 –1365, Oct. 2008.
- [48] D.J. Love, R.W. Heath Jr., W. Santipach, and M.L. Honig. What is the value of limited feedback for MIMO channels? *IEEE Communications Magazine*, 42(10):54 – 59, Oct. 2004.
- [49] C.E. Luna, Y. Eisenberg, R. Berry, T.N. Pappas, and A.K. Katsaggelos. Joint source coding and data rate adaptation for energy efficient wireless video streaming. *IEEE Journal on Selected Areas in Communications*, 21(10):1710–1720, 2003.
- [50] Lin Ma, Weisi Lin, Chenwei Deng, and King Ngi Ngan. Image retargeting quality assessment: A study of subjective scores and objective metrics. 6(6):626–639, 2012.
- [51] L. T. Maloney and J. N. Yang. Maximum likelihood difference scaling. *Journal of Vision*, (8):573–585, 2003.

- [52] M. A. Masry and S. S. Hemami. A metric for the continuous quality evaluation of video with severe distortions. *Journal of Signal Processing: Image Communications*, 19(2):133 – 146, Feb. 2004.
- [53] Microsoft Corporation. IIS smooth streaming technical overview. <http://www.microsoft.com/en-us/download/default.aspx>, Sept. 2009.
- [54] A. K. Moorthy and A. C. Bovik. Visual importance pooling for image quality assessment. *IEEE J. Sel. Topics Signal Process.*, 3(2):193–201, April 2009.
- [55] MPEG Requirements Group. *ISO/IEC FCD 23001-6 Part 6: Dynamics adaptive streaming over HTTP (DASH)*. http://mpeg.chiariglione.org/working_documents/mpeg-b/dash/dash-dis.zip, Jan. 2011.
- [56] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
- [57] J. A. Nelder. The fitting of a generalization of the logistic curve. *Biometrics*, 17(1):pp. 89–110, 1961.
- [58] M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithm Approach*. The Johns Hopkins University Press, 1981.
- [59] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba. Considering temporal variations of spatial visual distortions in video quality assessment. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):253–265, 2009.

- [60] D.P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451, 2006.
- [61] S. M. Pandit and S. M. Wu. *Time-Series and System Analysis with Application*. J. Wiley & Sons, 1983.
- [62] R. Pantos and E. W. May. HTTP live streaming. *IETF Internet Draft, work in progress*, Mar. 2011.
- [63] Jincheol Park, K. Seshadrinathan, Sanghoon Lee, and A.C. Bovik. Video quality pooling adaptive to perceptual distortion severity. *IEEE Transactions on Image Processing*, 22(2):610–620, 2013.
- [64] D. E. Pearson. Viewer response to time-varying video quality. *Proc. SPIE*, 3299(1), 1998.
- [65] M. H. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Trans. Broadcast.*, 50(3):312–322, Sept. 2004.
- [66] M Podolsky, S. McCanne, and M. Vetterli. Soft ARQ for layered streaming media. *Technical Report, Computer Science Division, University of California, Berkeley*, UCB/CSD-98-1024, 1998.
- [67] Wei Pu, Zixuan Zou, and Chang Wen Chen. Video adaptation proxy for wireless dynamic adaptive streaming over HTTP. In *19th International Packet Video Workshop (PV)*, pages 65–70, 2012.

- [68] J. Reichel, S. Schwarz, and M Wien. Joint scalable video model 11 (JSVM 11). *Joint Video Team, Doc. JVT-X202*, Jul. 2007.
- [69] F. Ribeiro, D. Florencio, and V. Nascimento. Crowdsourcing subjective image quality evaluation. In *Proc. of 18th IEEE International Conference on Image Processing (ICIP)*, pages 3097–3100, 2011.
- [70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.
- [71] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9):1103–1120, Sept. 2007.
- [72] K. Seshadrinathan and A. C. Bovik. Motion-tuned spatio-temporal quality assessment of natural videos. *IEEE Transactions on Image Processing*, 19(2):335 – 350, Feb. 2010.
- [73] K. Seshadrinathan and A. C. Bovik. Temporal hysteresis model of time-varying subjective video quality. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, May. 2011.
- [74] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. Study of subjective and objective quality assessment of video. *IEEE Trans. Image Process.*, 19(6):1427 –1441, Jun. 2010.
- [75] S. Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, 2(3), January 2008.

- [76] H.R. Sheikh, M.F. Sabir, and A.C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Process.*, 15(11):3440–3451, 2006.
- [77] R. Soundararajan and A. C. Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, to appear, 2012.
- [78] Alexander L Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.
- [79] Tao Su, Hao Ling, and W. J. Vogel. Markov modeling of slow fading in wireless mobile channels at 1.9 GHz. *IEEE Trans. Antennas Propag.*, 46(6):947–948, Jun. 1998.
- [80] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.
- [81] K. T. Tan, M. Ghanbari, and D. E. Pearson. An objective measurement tool for MPEG video quality. *Signal Processing*, 70:279 – 294, July 1998.
- [82] A. M. van Dijk, J. B. Martens, and A. B. Watson. Quality assessment of coded images using numerical category scaling. In *Proc. SPIE*, volume 2451, Feb. 1995.

- [83] P. V. Vu, C. T. Vu, and D. M. Chandler. A spatiotemporal most-apparent-distortion model for video quality assessment. In *18th IEEE International Conference on Image Processing (ICIP)*, pages 2505–2508, Sept. 2011.
- [84] Hong Shen Wang and Pao-Chi Chang. On verifying the first-order Markovian assumption for a Rayleigh fading channel model. *IEEE Trans. Veh. Technol.*, 45(2):353–357, May. 1996.
- [85] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398–1402, Nov. 2003.
- [86] Zhou Wang and Qiang Li. Video quality assessment using a statistical model of human visual speed perception. *Journal of the Optical Society of America*, A 24, B61-B69, Jul. 2007.
- [87] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.
- [88] Fuzheng Yang, Shuai Wan, Qingpeng Xie, and Hong Ren Wu. No-reference quality assessment for networked video via primary analysis of bit stream. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1544–1554, 2010.

- [89] Qinqing Zhang and S. A. Kassam. Finite-state Markov model for Rayleigh fading channels. *IEEE Trans. Commun.*, 47(11):1688–1692, Nov. 1999.
- [90] Yu Zhang, Fangwen Fu, and M. van der Schaar. On-line learning and optimization for wireless video transmission. *IEEE Trans. Signal Process.*, 58(6):3108–3124, Jun. 2010.
- [91] Xiaoqing Zhu and B. Girod. Distributed media-aware rate allocation for wireless video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1462–1474, 2010.
- [92] M. Zink, O. Künzel, J. Schmitt, and R Steinmetz. Subjective impression of variations in layer encoded videos. In *11th International workshop on quality of service*, pages 137–154. Springer, 2003.