

# VBR VIDEO STREAMING OVER WIRELESS NETWORKS

BY

GUANG JI

A THESIS SUBMITTED IN CONFORMITY WITH THE REQUIREMENTS  
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE,  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,  
AT THE UNIVERSITY OF TORONTO.

COPYRIGHT © 2009 BY GUANG JI.  
ALL RIGHTS RESERVED.

# VBR Video Streaming over Wireless Networks

Master of Applied Science Thesis  
Edward S. Rogers Sr. Dept. of Electrical and Computer Engineering  
University of Toronto

*by Guang Ji*  
*June 2009*

## Abstract

Video streaming applications over wireless networks have turned out to be immensely popular recently. In this thesis, we first study the buffering schemes for the VBR video streaming in heterogeneous wireless networks. An analytical framework is presented to derive the expected number of jitters and average buffering delay. Through experimenting with a wide range of buffering schemes, we quantify the benefit of incorporating user location information in streaming over heterogeneous wireless networks. Second, we consider the delivery of scalable VBR video streams over wireless channels. We propose adaptive rate control algorithms to improve the combined system performance of video frame quality and playout smoothness based on the feedback information of wireless network estimation, buffer content and playback situation. The proposed adaptive rate control algorithms provide significantly improved streaming quality compared with the non-control policy.

# Acknowledgments

First, I would like to express my sincerest gratitude to my supervisor, Professor Ben Liang, for giving me the opportunity to work under his supervision. I have learned a lot from his guidance and support throughout the pursuit of my Master of Applied Science degree. Professor Liang motivated me to think about new ideas and improve my research work with his insightful suggestions. He provided me with an invaluable experience in this exciting research.

I am thankful to the members of my thesis committee, Prof. Elvino Sousa, Prof. Shahrokh Valaee, and Prof. Lacra Pavel for the time spent in reviewing my thesis, and for their helpful feedback and comments on improving its content.

I thank all my current and former colleagues in the WHIMSIC research group for providing an enjoyable and fun working environment. It has been a pleasure working alongside all of you, thank you all for your stimulating discussion and friendly encouragement.

I am very grateful to my parents, for their dedication and the support they have given me during all of my life. To them I dedicate this thesis.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Thesis Outline . . . . .	6
<b>2 Literature Review and Background</b>	<b>7</b>
<b>3 Buffering Schemes of Video Streaming</b>	<b>12</b>
3.1 Introduction . . . . .	13
3.2 System Model . . . . .	16
3.2.1 Network Channel Model . . . . .	16
3.2.2 Receiver Buffer Model . . . . .	18
3.3 Jitter and Delay Analysis . . . . .	19

3.4	Buffering schemes in Heterogeneous Networks . . . . .	25
3.5	Simulation Results . . . . .	27
3.5.1	Simulation Setup . . . . .	27
3.5.2	Model Validation . . . . .	28
3.5.3	Comparison of Buffering Schemes . . . . .	28
3.6	Summary . . . . .	32
<b>4</b>	<b>Stochastic Rate Control for Scalable VBR Video Streaming</b>	<b>33</b>
4.1	Introduction . . . . .	34
4.2	System Model . . . . .	36
4.2.1	Problem Formulation . . . . .	40
4.3	Dynamic Programming Based Algorithm . . . . .	42
4.3.1	Algorithm . . . . .	42
4.3.2	Use of the policy . . . . .	43
4.3.3	Remark . . . . .	44
4.4	Greedy Algorithm . . . . .	44
4.4.1	Problem Formulation . . . . .	44
4.4.2	Algorithm . . . . .	45
4.4.3	Remark . . . . .	48
4.5	Simulation . . . . .	48
4.5.1	Simulation Setup . . . . .	48
4.5.2	Constant Algorithms . . . . .	49
4.5.3	Impact of Control Duration $\Delta T$ and Channel Rate . . . . .	50
4.5.4	Impact of the Loss Weights . . . . .	51
4.6	Summary . . . . .	52

**5 Conclusion**

**54**

**Bibliography**

**56**

# List of Tables

3.1	Notations . . . . .	20
3.2	Video parameters in simulation . . . . .	27
4.1	Table of nomenclature . . . . .	39
4.2	Video parameters in simulation . . . . .	49

# List of Figures

3.1	A typical illustration of video streaming system for mobile users in two-tier wireless networks. The dual-mode handset user is able to switch between 3G cellular network and WLAN. . . . .	15
3.2	Channel state transitions in two-tier wireless network . . . . .	18
3.3	Analysis and simulation results for fixed buffering schemes with different buffer sizes: (a) FBD (b) FPD (c) FPT . . . . .	29
3.4	Comparison of FBD, SBD, and JBD: (a)buffer= $7.2 \times 10^4 bits$ . (b) Infinite buffer. . . . .	30
3.5	Comparison of FPD, SPD, and JPD: (a)buffer= $7.2 \times 10^4 bits$ . (b) Infinite buffer. . . . .	30
3.6	Comparison of FPT, SPT, and JPT: (a)buffer= $7.2 \times 10^4 bits$ . (b) Infinite buffer. . . . .	31
4.1	A typical illustration of video streaming system and the buffer model in the wireless user. . . . .	35
4.2	Illustration the time slots $t_n$ and the control epoch $T_N$ . . . . .	40
4.3	The impact of control time period for DP algorithm for different network settings. $\omega_b$ is chosen as 0.8 . . . . .	50

4.4	Compare the three algorithms with different control time duration. $\omega_b$ is chosen as 0.8 . . . . .	51
4.5	Comparison of the algorithms with different $\omega_b$ . The average channel rate is $2.1 \times 10^4$ bps . . . . .	52
4.6	Comparison of the algorithms with different $\omega_b$ . The average channel rate is $7.6 \times 10^4$ bps . . . . .	53

# Chapter 1

## Introduction

### 1.1 Overview

The rapid growth of wireless communication and networking protocols, such as 802.11 and cellular mobile networks, is expected to bring ubiquitous access to streaming multimedia services, such as online movie, TV news, music video and etc. At the same time, the advances of technology in the areas of video compression permit the deployment of novel video distribution applications which have turned out to be immensely popular in recent years.

Video streaming enables simultaneous delivery and playback of the video, which overcomes the problems associated with file download since users do not have to wait for the *entire* video to be received before viewing it. The basic idea of video streaming is to partition the compressed video source file into parts, transmit them in succession, and decode and playback the video in the receiver. Hence, users are able to watch the videos just after a small delay at the beginning. Also, the storage requirements of the receiver is comparatively low, because only a small portion of the video is stored in the users'

buffer at any point in time.

Generally, there are two types of streaming scenarios based on whether the video is pre-encoded and stored for later viewing, or it is captured and encoded for real-time communication. Video conference, video phone and interactive games are examples of real-time video streaming applications, which have stringent delay requirement. On the other hand, currently in many applications video content is pre-encoded and stored in the multimedia server for later request of viewing, which is also called video-on-demand (VoD), such as YouTube, MSN Video, Google Video, CNN, and a plethora of copycat YouTube sites [1]. In this thesis, we mainly focus on the transmission of on-demand videos over wireless networks.

Multimedia streaming applications have distinctive Quality of Service (QoS) requirements, such as high bandwidth requirement, delay sensitiveness and loss tolerance. We list the challenging QoS issues as follows:

- 1) Bandwidth: Transmission of video sequences typically has a minimum bandwidth requirement in order to achieve acceptable presentation quality. Therefore, supporting the delivery of video over time-varying wireless links could be very unreliable. The challenge then lies in keeping the quality degradation to a level that is hardly noticeable or tolerable while utilizing the wireless resources efficiently.
- 2) Delay: In contrast to data transmission which is usually not subject to strict delay constraints, video streaming requires bounded end-to-end delay. Each video frame needs to arrive at the receiver to be decoded and displayed before its playout deadline. Otherwise, it is useless. If the video packet does not arrive on time, the playout process will have to be temporally paused, which is annoying to human eyes and deteriorates the overall streaming quality. Consequently, video streaming

applications are usually known to be very sensitive to delay.

- 3) Loss: Video streaming technology is tolerant to a certain level of loss, since the visual quality will still be acceptable if the packet loss ratio is kept below a certain threshold. However, loss of packets can potentially make the presentation displeasing to human eyes, especially when some of the key video frames are lost which could make the presentation impossible. Therefore, guaranteeing a low level of packet transmission loss is also important for the video streaming system.

In this thesis, we consider the video streaming process over wireless networks, such as the cellular mobile networks and Wireless Local Area Networks (WLAN). Wireless networks imposes several additional challenges, such as bandwidth limitation, random channel variation, and high error rate [2]. Due to predictive video coding, wireless transmission packet delays and losses may result not only in decoding errors of the current frame, but also in quality degradation of subsequent frames included in the dependency chain.

In order to combat unfavorable channel conditions, resulting from fading, multipath propagation, and scattering of wireless networks, channel coding and automatic repeat request (ARQ) strategies are used to guarantee an error-free packets reception at the expense of delay jitter, which is the variance in end-to-end delay experienced by video frames.

Playout buffering in combination with an initial playback delay is a commonly used technique for compensating for the delay jitter. Buffering allows for a smooth playback of the stream, but it generally induces a playback delay at the client, and thus impacts the general Quality of Service (QoS). On the other hand, if the bandwidth degradation persists, the playout buffer could also underrun and cause video frame freezes.

In a typical wireless video streaming system we considered in this thesis, the video sources are stored in the remote *Media Server*. The video sequences are transmitted through the backbone network to the Access Point (AP) and then sent to the users through a wireless network. The media users' can be computers, cellular phones and other personal digital assistants to be capable of both Internet access and video playback. In order to protect against the influence of the wireless link fluctuation, transmitted video packets are temporarily stored at the user's receiver buffer, which sustains streaming when the network throughput is low.

Variable-bit-rate (VBR) video is considered in this thesis. Many video encoders generate constant-bit-rate (CBR) streams to simplify the allocation of disk, memory, and network resources. However, CBR encoded video ultimately has variable quality, since the encoder is not permitted to increase the output bit rate during periods of action or detail, precisely when degradation in quality would be most noticeable to the viewers. Alternatively, video encoders can generate constant-quality video, resulting in a VBR stream. Compared with the CBR videos, VBR media provides better quality for the same average bit rate by adapting quantization and compression to the time-varying entropy of the media [3].

In this thesis, we study the efficient delivery of VBR videos over wireless networks. The challenges of the problem will be addressed in two main aspects. First, we focus on the video receiver size and research the performances of different buffering schemes for VBR video streaming. Specifically, the video transmission over heterogeneous wireless networks are investigated. Recent trends indicate that wide-area cellular network (CELL), e.g., 3G network, Wireless Local Area Networks (WLANs), e.g., IEEE 802.11, will co-exist to offer seamless wireless multimedia services [4]. Such integration enables

the users to enjoy better streaming performance while exploiting the complementary advantages of different networks. On the one hand, 3G networks provide an expensive universal coverage; on the other hand, WLANs provide ample networking resources for the users at a cheaper cost wherever available. Hence, users will generally enjoy the best of each access technology, and service providers will enjoy better utilization of their resources. Our objective is to discover heterogeneous networking attributes that may influence the streaming performance, in terms of the tradeoff between jitter frequency and buffering delay.

The other focus in this thesis is the adaptive rate control techniques for the video sender based on feedback information from the video receiver. Specifically, the scalable encoded video is considered, which encodes each frame into several layers, which includes one base layer and several enhancement layers. The fundamental problem that we want to address is the dynamic allocation of the available bandwidth to the two layers in order to minimize the impact of client starvation. The objective function is modeled as the weighted sum of video quality and playout continuity degradations. Conservatively, we could allocate all of the available bandwidth to the base layer until the entire base layer has been prefetched. A more aggressive, and optimistic policy is to allocate the bandwidth adaptively according to the the current and past available bandwidth, the current prefetch buffer contents, and the dynamic consumption rates of the videos.

We formulate the problem in a framework of Markov decision processes and propose an adaptive stochastic control policy based on dynamic programming algorithm. In order to decrease the computation complexity, we also develop an online greedy algorithm, which focuses on the current control time period. Through extensive simulations, we find that the proposed adaptive rate control algorithms provide significantly improved

video quality and playout smoothness. Furthermore, when rate control is not used very frequently, the performance of the greedy algorithm nearly matches that of the ideal optimal Dynamic Programming policy.

We present the outline of the thesis in the next section.

## **1.2 Thesis Outline**

This thesis is organized as follows. Chapter 2 presents a brief relevant literature survey for the important research issues on video streaming technologies. In Chapter 3, we present an analytical framework for VBR video streaming in a two-tier wireless network with VBR channels and discover the heterogeneous networking attributes that may influence the streaming performance, in term of the tradeoff between jitter frequency and buffering delay. Chapter 4 develops an effective evaluation approach to determine the and evaluate the sending rate for scalable video streaming. Finally, Chapter 5 concludes the thesis.

## Chapter 2

# Literature Review and Background

This chapter gives a brief literature review on the relevant video streaming technologies and existing challenges, which motivate the research work of this thesis. We start with an introduction of several video transmission smoothing techniques. After that, we present some known review on the rate control problems of video streaming.

Streaming eliminates the initial waiting time before video playback starts and the requirement for storing the entire video file as opposed to the download-and-play schemes. But the fast viewing advantage of streaming comes with the price of sensitivity to network transmission errors and throughput fluctuations. In order to protect against the influence of the wireless link fluctuation and maintain a continuous steady flow for smooth playback at the receiver, transmitted video packets are temporarily stored at the receiver buffer. Buffering at the client serves several distinct purposes. First, it allows the client to compensate for short term variations in packet transmission delay, i.e., absorbs delay jitter resulting from network bandwidth variations. Second, it gives the client time to perform packet loss recovery if needed. Third, it allows the client to continue playing back the content during lapses in network bandwidth. And finally, it allows the content

to be coded with variable bit rate, which can dramatically improve overall quality.

When there are not enough data in the buffer to support the video playback consumption, a playback starvation occurs, which is also known as *buffer underflow*. Then video stops playing until sufficient data are gathered. This event of playback interruption is usually termed *playout jitter*, and the time duration for data buffering is termed *buffering delay*. Intuitively, the longer buffering delay is employed, the more packets will be received and the smaller jitter occurrence probability we can anticipate in the future, but at the same time the user viewing experience is correspondingly degraded due to the increased waiting time. By controlling the size of the client buffer over time it is possible for the client to meet the above mentioned user expectations. If the buffer is initially small, it allows a low startup delay. If the buffer never underflows, it allows continuous playback. If the buffer is eventually large, it allows eventual robustness as well as high, nearly constant quality. Thus, client buffer management is a key element affecting the performance of streaming media systems.

Several smoothing techniques deal with network link transfers of stored video streams [5] [6] [7], but they only consider a wired network which offers guaranteed bandwidth service and an intermediate smoothing node or a dedicated smoothing server are required. Hence these schemes are not suitable for error-prone wireless network streaming systems.

Varsa *et al.* [8] proposed a separation between a delay jitter buffer and a decoder buffer for VBR video. The delay jitter buffer is particular designed to compensate for delay jitters and bit rate variations caused by variable bit rate channel. Streamed video data is first buffered in the delay jitter buffer and then emitted into the decoder buffer at a constant rate after an initial delay. By choosing a suitable initial delay, the jittered streaming data is de-jittered by the delay jitter buffer and a virtual CBR channel is

formed at the input of the decoder buffer.

However, in [9], the authors compared the single receiver buffer with the separate buffer, and concluded that the single receiver buffer performs at least as good as the two separate buffers. They described a method to provide a certain Quality of Service (QoS) guarantee, where the initial delay and receiver buffer size are decided according to the upper and lower bounds of the random receiver curve to guarantee a minimum jitter-free probability. However, they did not give a general means to find such bounds of the curve and only consider a simple Bernoulli channel.

Studies in [10] show that the pattern of packet loss can be captured by Markov models. Kalman *et al.* used a Markov chain analysis method in [11] to examine the tradeoff between buffer underflow probability and latency for adaptive playout video streaming. Adaptive media playout allows the streaming client to control the data consumption rate, but can introduce noticeable artifacts in the displayed video.

Xu *et al.* considered the transmission of prerecorded media from a server to a client by using TCP-Friendly Rate Control (TFRC) in [12]. The models focus on the impact of the TFRC rate changes to the probability of rebuffering events and analytically study its impact on media quality. This work does not consider buffering delay as a performance metric and they only consider CBR encoded videos and an infinite receiver buffer.

The authors previously presented in [13] an analytical framework to study the frequency of jitters and buffering delays under the constraint of initial playback delay and receiver buffer size, using a Markov VBR channel model for a homogeneous wireless network. The family of fixed buffering schemes are examined. In Chapter 3, we investigate further into separate and jointly optimal buffering schemes for heterogeneous wireless networks. To the best of our knowledge, this research represents the first attempt to

analyze buffering mechanisms for media streaming over heterogeneous wireless networks.

We consider the rate control problems in Chapter 4. Several rate control techniques dealing with wired network link transfers of stored video streams are based on the TCP Friendly Rate Control (TFRC) [14] [15], which is designed to be fair to TCP flows and involves lower fluctuation than TCP. But TFRC can not distinguish between packet loss due to buffer overflow and that due to bit errors, so these schemes are not suitable for error-prone wireless network streaming systems. A number of efforts have been made to give possible solutions of the rate control for streaming over wireless, such as combining packet inter-arrival times and relative one way delay to differentiate the losses or to use end-to-end statistics to detect congestion. Chen *et al.* propose the use of multiple TFRC connections as an end-to-end rate control solution for wireless video streaming [16].

A large majority of rate control solutions focuses on the source-rate control solutions, which often perform at the frame level or the macroblock level. The authors in [17] introduce a rate control scheme based on *a priori* stochastic models for both source and underlying channel, where a solution based on stochastic dynamic programming is proposed. The video considered in the paper is not layered encoded and the authors mainly explore the rate-distortion model as the performance parameters.

Atzori *et al.* propose a joint source-rate/channel-code control scheme for streaming VBR-encoded video over a wireless channel [18]. The rate control is performed on a cycle basis which characterizes the “good” and “bad” states of the channel. The scheme is designed to maximize the source rate and considers the user’s buffer by guaranteeing an upper bound of starvation probability. They do not use layered encoded video and they use starvation probability as the evaluation metric which is different from our approach.

The work in [19] presents the most closely related system setting compared to our

work. They prove that for an infinitely-long video, the optimal policy takes on a static form. However, the video considered in this paper is constant-bit-rate encoded, and the loss of enhancement layer is simply modeled as a fixed proportion of the base layer loss.

In Chapter 4, we focus on a joint video sender-receiver control scheme for transporting scalable variable-bit-rate (VBR) encoded video over wireless channels. The main objective is to adapt the sending rates of different video layers based on the combined the knowledge of receiver buffer, playback progressing stage, and estimated network condition. To the best of our knowledge, this represents the first attempt to analyze stochastic rate control mechanisms for layered encoded VBR media streaming over wireless networks.

## Chapter 3

# Buffering Schemes of Video

## Streaming

In this chapter, we study the buffer management of the video receiver for on-demand *variable-bit-rate* (VBR) video streaming over heterogeneous wireless networks. With the co-existence of different wireless networks, which exhibit largely different bandwidth and coverage characteristics, much interest has been involved in integrating these networks to support smooth and efficient multimedia services. Wireless clients are able to stream video clips while moving in the wireless networks, such as the cellular network and wireless local area networks (WLAN). We present an analytical framework for the streaming process in a two-tier wireless network with VBR channels, and derive the expected number of jitters and average buffering delay during video playback as measures of system performance. The primary purpose is to research the performance of a wide range of jitter-recovery buffering schemes for the video receiver, based on buffering delay, buffered data, and buffered playback duration.

## 3.1 Introduction

As described in Chapter 2, media streaming applications have distinctive Quality of Service (QoS) requirements, such as delay sensitiveness and loss tolerance. Specifically, each packetized media unit has a presentation deadline at the client, which is determined by the interactivity requirements and buffer limitations. The deadline constraint imposes restrictions on the transmission delay of video packets. Failing to deliver the unit by the deadline causes audio-visual quality degradation in the multimedia application. In addition, the varying wireless environment brings in dramatic fluctuation of network bandwidth which makes the streaming technology even more challenging.

The next-generation wireless communications have been envisioned to be supported by heterogeneous networks using various wireless access technologies. The popular cellular networks and wireless local area networks (WLANs) present perfectly complementary characteristics in terms of service capacity, mobility support, and quality-of-service (QoS) provisioning.

Cellular networks are originally designed to provide high-quality voice service with widearea coverage. Currently, the third generation (3G) augmented with multimedia service support has been commercialized, such as the universal mobile telecommunication system (UMTS) and cdma2000. The UMTS system supports a data rate up to 2 Mbit/s with greater capacity and improved spectrum efficiency. However, the deployment cost remains high due to expensive radio spectrum and implementation complexity. On the other hand, WLANs have also achieved great success and provide higher data rates at a much lower cost. For example, the most popular WLAN standard IEEE 802.11b operates at the license-exempt industrial, scientific, and medical (ISM) frequency band from 2.4 GHz to 2.483 GHz. It extends the physical (PHY) layer of the original 802.11 standard

based on direct sequence spread spectrum (DSSS) and supports a data rate up to 11 Mbit/s. The subsequent revisions 802.11a and 802.11g employ orthogonal frequency-division multiplexing (OFDM) and offer a maximum rate of 54 Mbit/s at the unlicensed 5 GHz and 2.4 GHz bands, respectively. However, designed as a wireless extension to the wired Ethernet, a WLAN can only cover a small geographic area. For instance, an 802.11b access point (AP) can communicate with a mobile within up to 60 m at 11 Mbit/s and up to 100 m at 2 Mbit/s with omnidirectional antennas.

We can see that the two types of networks present complementary strengths in terms of mobility support, data rate, and implementation cost. Cellular/WLAN interworking can provide mobile users with both ubiquitous connectivity and high-rate data service in hot spots. The cellular/WLAN interworking is thus an effective way to promote the evolution of wireless networks.

In this Chapter, we consider the scenario of delivering VBR video over heterogeneous wireless networks. The video is pre-encoded with variable bit rate and stored in a remote *media server* that can be accessed through both tiers of the network, which are labeled “CELL” and “WLAN” for illustration purposes without loss of generality. Mobile end users view the videos streams while roaming in the two-tier network. A typical video streaming system in heterogeneous wireless networks is presented in Figure 4.1.

We initiate an analytical model for the mobile end user’s receiver buffer. Then the expected jitter frequency during the whole streaming session is derived. Furthermore, in order to evaluate the user-perceived streaming media quality, we adopt a cost function combining the jitter numbers and average buffer delay during the entire playback. We first examine the performance of *Fixed Buffering Schemes* which employ the same buffering parameters, such as the fixed buffering delay, fixed buffered playout data, and fixed

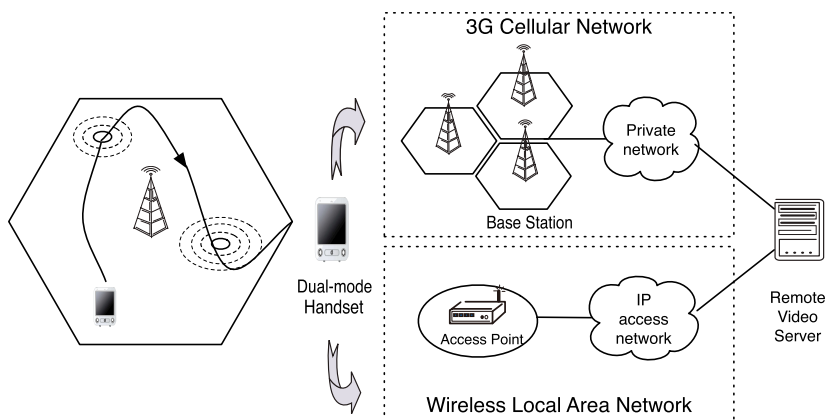


Figure 3.1: A typical illustration of video streaming system for mobile users in two-tier wireless networks. The dual-mode handset user is able to switch between 3G cellular network and WLAN.

playout time. Then we consider *Separate Buffering Schemes* which use different buffering parameters for CELL and WLAN. The parameters are obtained from the analysis of the wireless networks separately. We also study *Jointly Optimal Buffering Schemes*, which select optimal buffering parameters directly from the heterogeneous networks under certain average buffering delay constraints. Through extensive analysis and simulation, we compare these three families of schemes to find appropriate buffering methods for mobile devices with various level of storage memory and computation power.

The rest of this chapter is structured as follows. The system model is presented in Section 4.2. We derive the analysis framework for video streaming process over heterogeneous works in Section 3.3. Section 3.4 presents our comparisons for different buffering schemes. Simulation results and further discussions are provided in Section 4.5. The concluding remarks are given in Section 4.6.

## 3.2 System Model

### 3.2.1 Network Channel Model

We consider the video streaming over a two-tier network, using CELL-WLAN integration as an example. In general, CELL provides universal coverage, with WLAN forming several hotspots. The mobile clients use the dual-mode handset which enables the network access switching between CELL and WLAN when necessary. We assume the mobile users will automatically switch to WLAN service when traversing into the overlapping of the two networks, in order to obtain a potentially higher data throughput.

The streaming video process is considered to be time-discrete with equal time slots. In each time slot, multiple video packets are sent to the mobile user. The base stations of CELL and access points in WLAN forward video sequences to the roaming users independently. Clearly, the number of packets transmitted per time slot in WLAN is much larger than that in CELL.

We assume the transport channel to be error free, possibly due to an ideal error control mechanism or concealment scheme, but the network transmission rate may change over time. Noting the delay-sensitivity of the video streaming technology, the fluctuations in transmission rate may possibly lead to late packet arrivals and significant playback interruptions. In each of CELL and WLAN, we can model the network transmission channel as a discrete-time Markov variable-bit-rate channel [10]. Following the common assumption of exponential network residence times in CELL and WLAN, the transitions between these two sub-networks are memoryless. Hence, we can characterize the overall channel status over time by a Markov chain  $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ , where  $\mathcal{S}$  is the set of possible channel states,  $\mathcal{T}$  is the transition probability matrix of the channel states, and  $\mathcal{R}$  is

the set of possible transmission rates associated with the state. We define  $P_{ij}$ , where  $i, j \in \{c, w\}$  as the probability that the user will be in subnetwork  $j$  in the next time slot given she is in subnetwork  $i$  in the current time slot. For example,  $P_{wc}$  is the transition probability from WLAN to CELL. Then the sojourn time or the average residence time [20] (measured in time slots ) in cellular network can be achieved as:

$$T^c = \sum_{t=1}^{\infty} t P_{cc}^{t-1} P_{cw} = \frac{1}{P_{cw}} \quad (3.1)$$

$$T^w = \sum_{t=1}^{\infty} t P_{ww}^{t-1} P_{wc} = \frac{1}{P_{wc}} \quad (3.2)$$

The following is an example on how to combine the channels states in CELL and WLAN. To characterize the error events in the wireless communication channel, a simple and widely used model is the *Gilbert-Elliot* model with states  $\Omega \in \{good, bad\}$  [21]. The network state can be transmitted from good to bad by losing one packet, or from bad to good by receiving one packet. Sanneck and Carle further proposed in [22] an extended Gilbert model for the wireless channels, which is able to provide better prediction of performance measures depending on longer-term correlation of errors. We start with CELL constructed as an  $M$ -state extended Gilbert model while WLAN an  $N$ -state extended Gilbert model (Figure 3.2). Thus, the channel states set becomes  $\mathcal{S} = \{S_1, S_1, S_2, \dots, S_{M+N}\}$ , where  $S_1$  and  $S_{M+1}$  are the *good* or *reception* states for CELL and WLAN respectively. Other states are the *bad* or *loss* states for the two subnetworks with different loss patterns. For example,  $S_2$  represents two consecutive packet losses. In combining the two extended Gilbert models, we add subnetwork transitions and adjust correspondingly the transition probabilities in the original models. Figure 3.2 shows an

example of our channel model. Note that in this model, we further restrict the transitions between subnetworks, so that only the *reception* states can be the destination of such transitions. We emphasize that this is adopted only as a common-sense assumption. The general analytical model presented in Section 3.3 is applicable to all transition patterns.

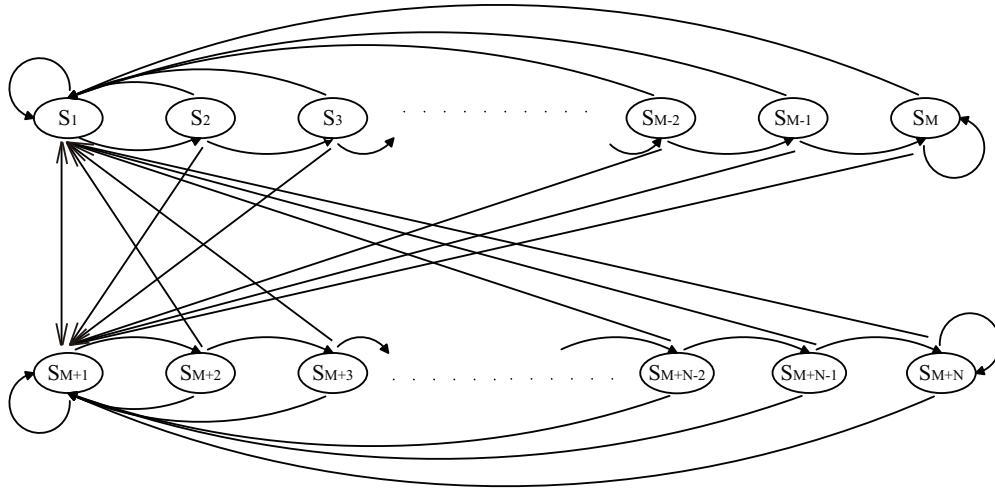


Figure 3.2: Channel state transitions in two-tier wireless network

### 3.2.2 Receiver Buffer Model

The video receiver of the mobile terminal consists of a playout buffer and a playout scheduler. The playout buffer is used to temporarily store the incoming video packets. We denote the total number of video packets as  $L$  and the duration of the video as  $T$ . Let  $p(t)$  be the playback schedule which describes the total amount of packets which *should* be received at time  $t$ . Thus we have  $p(T) = L$ . Denote  $r(t)$  as the entire number of packets which are successfully arrived at the receiver at  $t$ . Then, if  $r(t) < p(t)$ , i.e., *buffer underflow*, a jitter occurs and further buffering is required. Playout is assumed only after enough packets are aggregated, which is termed *jitter recovery*. Furthermore, if the buffer size is finite, there may be instances that the incoming packets numbers exceeds

the buffer limit. Then these packets will be lost due to *buffer overflow*. In this case, we assume that playout scheduler will send control signals to the video server requesting re-transmission of the loss packets in the next time slot.

The playout scheduler is responsible of managing the buffering schemes. We consider three types of buffering schemes, based on the *buffering delay* (BD), the buffered *playout data* (PD), and the buffered *playout time* (PT) [13]. One common setting of the playout scheduler is to use a fixed BD, PD, or PT after each jitter. However, in heterogeneous networks, a fixed scheme would give no consideration of the user mobility or the present network conditions. For example, when the mobile user is with WLAN which is able to provide high bandwidth and data rate, it is quite possible to buffer the same amount of required data in less time than that in cellular network. Considering the delay sensitivity for streaming and the limited buffer size, fixed buffer delay could bring unnecessary data loss and delay in WLAN. Conversely, it will be insufficient to resume the playback in cellular network if we set the buffering delay based on the WLAN channel conditions.

We then go one step further to ask the question that how to select appropriate buffering delay for video streaming in such a two-tier wireless network? Would it be possible to bring superior streaming performance with the consideration of the client mobility and different channel conditions? Before we answer these questions, we first develop the analytical model for the video streaming in the next section and then investigate into appropriate buffering schemes for video streaming in a two-tier wireless network.

### 3.3 Jitter and Delay Analysis

In this section, we derive the calculation of the expected jitters' number and average buffering delay during the streaming process. We tabulate the notations used in the rest

of the chapter in Table 3.1.

Notation	Description
$\Delta$	Initial delay
$R^c$	Maximum transmission rate in cellular network
$R^w$	Maximum transmission rate in WLAN network
$P_{ij}$	Transition probability from network $i$ to network $j$
$L$	Total number of packets of the video sequence
$M$	Number of states in cellular network channel
$N$	Number of states in WLAN channel
$J_n$	Index of the first packet whose deadline is violated after the $n - 1$ th jitter
$X_i$	Channel states for heterogeneous networks, where $i \in [1, M + N]$
$D^c$	Buffering delay in cellular
$D^w$	Buffering delay in WLAN
$\bar{D}$	Average buffering delay

Table 3.1: Notations

We index the incoming video packets with  $i$ , where  $i$  is an integer ranging from 1 to the total number of packets of the video source  $L$ . Let  $J_n$  denote the time index of the video when the  $n$ th jitter occurs, and  $X_n$  the channel state when the  $n$ th jitter occurs. We use  $R^c$  and  $R^w$  to denote the maximum numbers of packets transmitted per time slot in CELL and WLAN, respectively. For the extend Gilbert model, they simply are the transmission rates in the good states.

We define  $p_k^{(n)}(i) = Pr\{J_n = i, X_n = S_k\}$  as the probability that the  $n$ th jitter occurs at packet  $i$  with channel state  $S_k$ . Then, the expected number of jitters  $E\{J\}$  during the

whole streaming process can be expressed as [20]

$$\begin{aligned}
E\{J\} &= \sum_{n=1}^{\infty} \sum_{i=1}^L \left( \sum_{k=1}^{M+N} p_k^{(n)}(i) \right) \\
&= \underbrace{\sum_{n=1}^{\infty} \sum_{i=1}^L \sum_{k=1}^M p_k^{(n)}(i)}_{\text{Number of jitters in CELL}} + \underbrace{\sum_{n=1}^{\infty} \sum_{i=1}^L \sum_{k=M+1}^{M+N} p_k^{(n)}(i)}_{\text{Number of jitters in WLAN}}. \tag{3.3}
\end{aligned}$$

In order to obtain  $p_k^{(n)}(i)$ , we specify  $Q_{k,l}(i, j)$ , the probability that the  $(n+1)$ th jitter takes place at packet  $j$  with channel state  $S_l$ , given that the  $n$ th jitter occurs at packet  $i$  with channel state  $S_k$ . Therefore, we have

$$Q_{k,l}(i, j) = Pr\{J_{n+1} = j, X_{n+1} = S_l | J_n = i, X_n = S_k\}. \tag{3.4}$$

Applying the total probability theorem [20], we have

$$p_l^{(n+1)}(j) = \sum_{i=1}^j \sum_{k=1}^{M+N} Q_{k,l}(i, j) p_k^{(n)}(i). \tag{3.5}$$

In this way, with the first jitter probability  $p_k^{(1)}(i)$ , and the next jitter probabilities  $Q_{k,l}(i, j)$ , we are able to obtain the entire statistics of  $p_k^{(n)}(i)$  to calculate the expected number of jitters  $E\{J\}$ .

We model the streaming system states with two tuples  $(g, s)$ , where  $g$  denotes the total number of received packets till current time and  $s \in \mathcal{S}$  specifies the channel state. We define  $P_{l,k,r}$  as the transition probability from state  $S_l$  to state  $S_k$  with  $r$  packets



$$U_t = \begin{bmatrix} 0_{(M+N)p(t) \times (M+N)p(t)} & 0 & 0 & 0 \\ 0 & I_{(M+N)(B-1) \times (M+N)(B-1)} & 0 & 0 \\ 0 & 0 & I' & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and  $I' = \underbrace{[I_{(M+N) \times (M+N)} \cdots I_{(M+N) \times (M+N)}]}_{R^{w+1}}^T$ .

Then we have

$$\pi_t = \pi_0 \left( \prod_{s=-\Delta}^{t-1} \Psi U_s \right) \Psi. \quad (3.7)$$

(1) *Buffering Delay.* After each jitter, the stream stops and data is buffered for a certain buffering delay. We denote the buffering delay by  $D^c$  for CELL and  $D^w$  for WLAN. In order to find  $Q_{k,l}(i, j)$ , we imagine the video starts playing out from the jitter occurring time with an empty buffer. Denote this virtual initial state distribution, after a jitter occurs at  $t_j$  and the channel state is  $S_l$ , by  $\pi_{j,l} = [0 \cdots 0 \ 1 \ 0 \cdots 0]$ , where 1 is the  $((M+N)(j-1) + l)$ th element. Then, the state probability distribution at time  $t$  of having no jitter by  $t-1$  is given by

$$\begin{cases} \pi_t^c = \pi_{j,l} (\prod_{s=-D^c}^{t-1} \Psi U_s) \Psi, & \text{if } l \in [1, M] \\ \pi_t^w = \pi_{j,l} (\prod_{s=-D^w}^{t-1} \Psi U_s) \Psi, & \text{if } l \in [M+1, M+N]. \end{cases}$$

Finally, the  $Q_{l,k}(j, i)$  is obtained by

$$Q_{l,k}(j, i) = \begin{cases} \pi_t^c [(M+N)(i-1) + k], & \text{if } j \in [1, M] \\ \pi_t^w [(M+N)(i-1) + k], & \text{if } j \in [M+1, M+N]. \end{cases}$$



until the amount of buffered data can sustain a certain amount of playout time. We denote the buffered playout time by  $T^c$  for CELL and  $T^w$  for WLAN. The process to find  $Q_{l,k}(j, i)$  is similar to that in the PD scheme, except the number of packets to buffer is  $p(d_j + T^c) - (j - 1)$  for jitters happening in CELL, and the number of packets to buffer is  $p(d_j + T^w) - (j - 1)$  for jitters happening in WLAN.

### 3.4 Buffering schemes in Heterogeneous Networks

Ideally, video frames should be displayed continuously with each successive frame displayed immediately after its predecessor. However, due to the unstable network situations, continuous playout is not always possible, especially for streaming over wireless networks. Clearly, there exists a tradeoff between the jitter occurrences and the average buffering delay after each jitter. A superior buffering scheme should strike a balance between the two factors that leads to an overall optimized user satisfactory. Thus, we introduce a cost function  $C$  as the weighted sum of the expected number of jitters  $E\{J\}$  and the average jitter-recovery buffering delay  $\bar{D}$ :

$$C = (1 - \alpha)\bar{D} + \alpha E\{J\}. \quad (3.9)$$

where  $\alpha$  is the weight parameter ranging from 0 to 1, indicating the video viewer's preference.

(1) *Fixed Buffering Schemes.* In the fixed buffering schemes, the same buffering parameter value is used in both subnetworks. For each  $\alpha$ , the parameter that minimizes (3.9) is chosen. Thus, we have Fixed Buffering Delay (FBD), where  $D^c = D^w$ , Fixed Buffered Playout Data (FPD), where  $B^c = B^w$ , and Fixed Buffered Playout Time (FPT),

where  $T^c = T^w$ .

The fixed buffering schemes can be easily implemented, as they do not have to take into consideration the network conditions or the mobile user location. However, it cannot provide optimal performance the heterogeneous wireless networks. It is possible that designed buffering amount is too large for users in WLAN or too small for users in CELL.

(2) *Separate Buffering Schemes.* In the separate buffering schemes, we first find, independently for each type of subnetwork, the optimal buffering parameter value that minimizes (3.9). We then use them in the two-tier network. Thus, we have Separate Buffering Delay (SBD), Separate Buffered Playout Data (SPD), and Separate Buffered Playout Time (SPT).

The separate buffering schemes consider the different subnetworks separately. Surprisingly, our numerical results in Section 4.5 show that the performances of these schemes generally do not improve over the fixed buffering schemes.

(3) *Jointly Optimal Buffering Schemes.* In the jointly optimal buffering schemes, we find the optimal pair of buffering parameter values:

$$\begin{aligned} \text{Minimize} \quad & C = (1 - \alpha)\bar{D} + \alpha E\{J\} \\ \text{subject to} \quad & \alpha \in [0, 1], \bar{D} \in \mathcal{D} \end{aligned} \tag{3.10}$$

Thus, we consider the combination of  $(D^c, D^w)$  for Jointly Optimal Buffering Delay (JBD),  $(B^c, B^w)$  for Jointly Optimal Buffered Playout Data (JPD), and  $(T^c, T^w)$  for Jointly Optimal Buffered Playout Time (JPT).

The jointly optimal buffering schemes provide improved performance over fixed and

separate buffering schemes. However, they involve much higher computational complexity. In the following section, we validate our analytical results and evaluate the proposed buffering schemes via detailed simulations.

## 3.5 Simulation Results

### 3.5.1 Simulation Setup

We use the “Alpin ski” MPEG-4 variable-bit-rate video trace provided by [23]. The video sequences were encoded at a constant frame rate of 25 frames/s in the Quarter Common Intermediate Format (QCIF) resolution. Table 4.2 summarizes the main parameters of the video trace in the simulation. The packet size is set to 1800 bytes and the transmission time slot duration is 80 ms.

Parameter	Value
Sequence Length	89998 frames
Video Size	$7.2e + 06$ bytes
Format	QCIF ( $176 \times 144$ pixels)
Video Run Time	$1.6e + 06$ msec
Mean Bit Rate	$1.9e + 05$ bps
Peak Bit Rate	$1.8e + 06$ bps

Table 3.2: Video parameters in simulation

We assume that the mobile user is initially located in the cellular network. We use a two-state Gilbert model for each of the wireless networks, so that for the two-tier system, we have totally four states:  $S_1$ ,  $S_2$ ,  $S_3$ , and  $S_4$ , where  $S_1$  and  $S_3$  are the good states in CELL and WLAN respectively, and  $S_2$  and  $S_4$  are the corresponding bad states. For CELL, the transition probabilities from “good” to “bad” and reverse are 0.2 and 0.5,

respectively, before alterations due to user movement. For WLAN, they are 0.05 and 0.4. The subnetwork transition probabilities are  $P_{cw} = 0.005$  and  $P_{wc} = 0.01$ . The data rate in  $S(1)$  is 180 kbps, i.e. 1 packets per unit time slot, while data rate in  $S(3)$  is 1.8 Mbps, i.e. 10 packets per time slot. Hence, the average rate in CELL is 90 kbps, and that in WLAN is about 1.2 Mbps.

We simulate the transmission and playback for the target sequence in Matlab for over 500 realizations of the random VBR channel and obtain the average jitter numbers and average buffering delay. The initial delay  $\Delta$  is set to 0.4 seconds for all cases.

### 3.5.2 Model Validation

We compare the analytical and simulation results of the expected number of jitters for different average buffering delay values. Fig. 3.3 shows the comparison for fixed buffering schemes with different buffer sizes. We observe a good match between the simulation and analysis results. Moreover, as expected, the mean number of jitters decreases as the buffering delay increases. Note that the variations in the analysis curve of Fig. 3.3(c) is due to the VBR nature of the video. The comparison for other schemes are similar and is omitted to reduce redundancy.

### 3.5.3 Comparison of Buffering Schemes

We compare the fixed, separate, and jointly optimal buffering schemes. Fig. 3.4, Fig. 3.5, and Fig. 3.6 show the results for BD, PD, and PT, respectively. In each case, both finite and infinite buffers are studied.

All three figures show an approximately convex shape for the cost function over  $\alpha$ . Recall that a small  $\alpha$  favors the cost of jitters over the cost of buffering delay. This

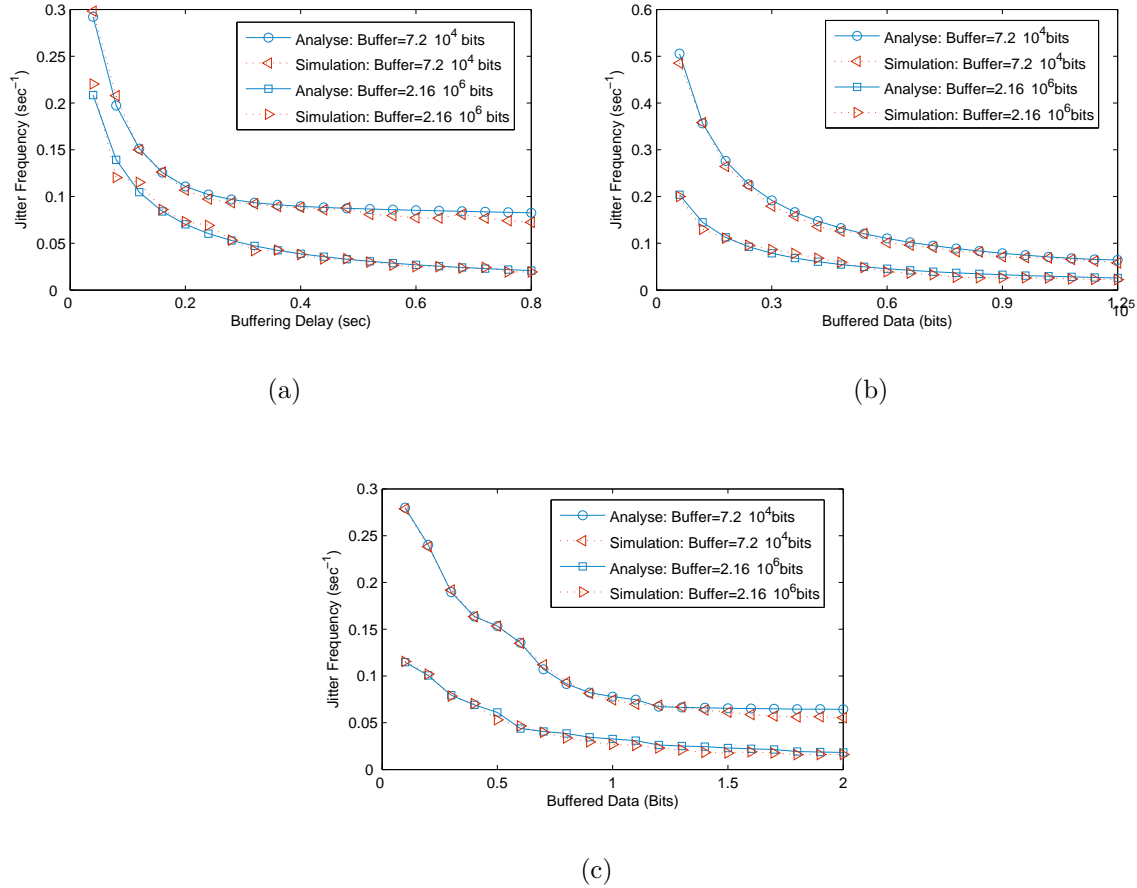


Figure 3.3: Analysis and simulation results for fixed buffering schemes with different buffer sizes: (a) FBD (b) FPD (c) FPT

suggests that in general it is easier to reduce only one of either the number of jitters or the buffering delay, and harder to strike a balance between the two. Indeed, if we allow the buffer delay to become very large to maximize the amount of data buffered, we can significantly reduce jitters.

These figures also show that the jointly optimal buffering schemes indeed perform the best. They also show an interesting phenomenon. Even though the parameters chosen for the separate buffering schemes are individually optimal in each subnetwork, their

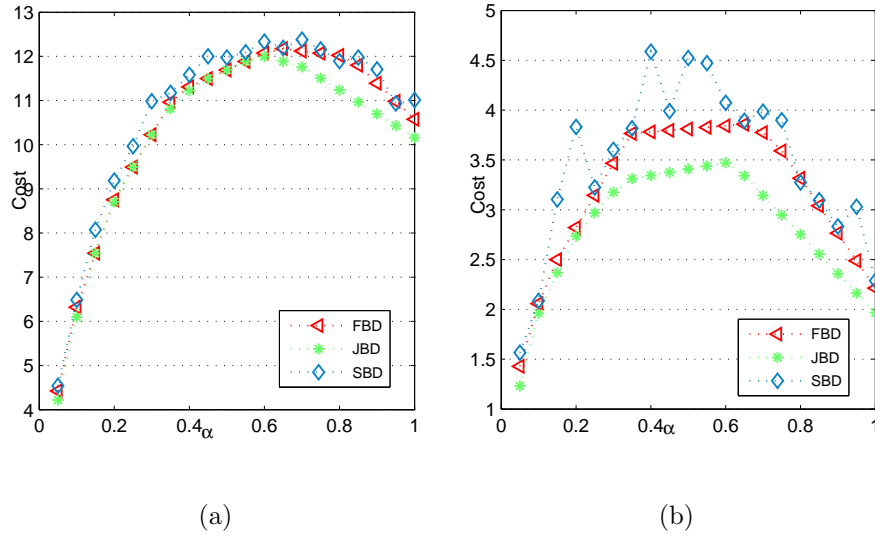


Figure 3.4: Comparison of FBD, SBD, and JBD: (a)  $\text{buffer} = 7.2 \times 10^4 \text{ bits}$ . (b) Infinite buffer.

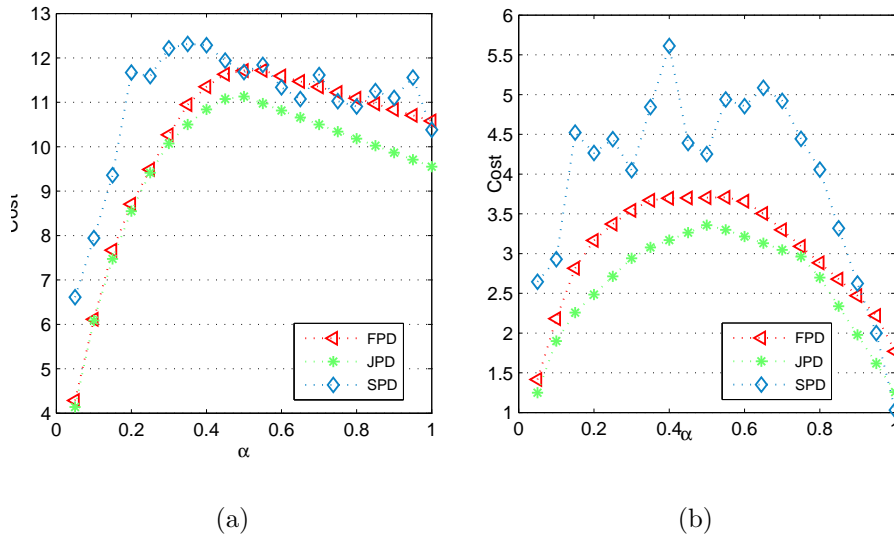


Figure 3.5: Comparison of FPD, SPD, and JPD: (a)  $\text{buffer} = 7.2 \times 10^4 \text{ bits}$ . (b) Infinite buffer.

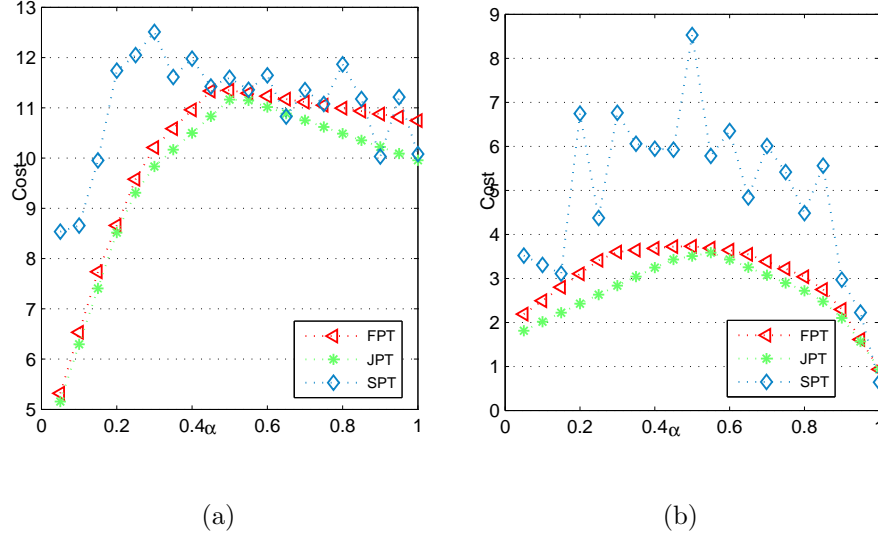


Figure 3.6: Comparison of FPT, SPT, and JPT: (a) buffer =  $7.2 \times 10^4$  bits. (b) Infinite buffer.

application to the two-tier network actually degrades the system performance, often to a degree worse than the fixed buffering schemes. This suggests that there exists strong correlation between streaming performance and the user mobility dynamics between the subnetworks, so that the subnetworks should not be considered separately in optimizing the performance of streaming in a heterogeneous network.

Comparing the three figures, we see that, when there is no limit on the buffer size and the buffering parameters are optimally chosen, JBD, JPD, and JPT all give similar performance. Furthermore, in this case, FBD, FPD, and FPT all give similar degradation from the optimal performance. However, if buffer size is limited, and the fixed buffering scheme is used, then FPT outperforms FPD and FBD. This result is unique to heterogeneous networks and is in contrast to [13], which shows that in homogeneous networks FPT, FPD, and FBD perform similarly in terms of jitter frequency and buffering delay. It suggests that buffering based on playout time can be more adaptive to the streaming

client's movement between the subnetworks. Hence, mobile devices with limited storage memory and insufficient computation power to produce network-aware optimal buffering parameters should adopt FPT in heterogeneous wireless networks.

### 3.6 Summary

We have studied a wide variety of buffering schemes for VBR video streaming over heterogeneous networks, including fixed, separate, and jointly optimal schemes. These schemes can be based on buffering delay, buffered playout data, or buffered playout time. We model the video transmission process for mobile clients roaming within the network using a two-tier Markov variable-bit-rate channel model and analyze the jitter and delay characteristics of such systems. Our analytical and simulation results suggest that the streaming performance can be significantly improved by utilizing the location information of a mobile client, but separate optimization within the subnetworks is unsuitable. Furthermore, we show that buffering based on playout time is more appropriate for simple mobile devices that have limited storage memory and use a constant buffering parameter regardless of location.

## Chapter 4

# Stochastic Rate Control for Scalable VBR Video Streaming

In this chapter, we consider the transmission of scalable VBR video over wireless network. Scalable encoded video can be used to improve the system performance by adapting the sending rate for different video frame layers to the varying network and playout situations. At the sender side, the dilemma is the following: we want to transmit all media units to provide the video at its original quality; but during the periods when the bandwidth is scarce, we may choose to transmit the most important units, i.e., the base layer packets, and skip the less important ones, i.e., the enhancement layer packets. Therefore, the purpose is to adaptively control the sending rates for the base layers and the enhancement layers based on the estimation of future bandwidth, the receiver buffer's feedback information and the video playback stage. We analyze the problem as a stochastic decision process and propose a stochastic dynamic programming (DP) algorithm to provide optimal rate control. Furthermore, we explore the performance of an online greedy algorithm in order to decrease the computation time and complexity.

## 4.1 Introduction

Advances in video coding technology and standardization are enabling an increasing number of video applications. The scalable Video Coding (SVC) [24] standard as an extension of H.264/AVC [25] allows efficient, standard-based scalability of temporal, spatial, and quality resolution of a decoded video signal through adaptation of the bit stream. The bitrate of a full quality SVC video can be reduced in three dimensions. The first dimension is spatial scalability, where a video with lower resolution picture frames can be extracted. The temporal resolution, i.e. frame rate, of a scalable video may be reduced by simply discarding certain frames. The SVC standard enables temporal scalability by hierarchical B (bi-directionally predicted) and P (uni-directionally predicted) pictures. The third dimension is the quality or SNR scalability where the frame rate and resolution is preserved, however, the bitrate is controlled by adjusting the transform coefficient quantization levels.

The scalable video encodes each frame into several layers, which includes one base layer and several enhancement layers. Without the existence of base layer, the corresponding video frame can not be decoded, which is one of the most important traits for the video streaming system. On the other hand, the enhancement layers are used to supplement the base layer to improve the quality of the video pictures. Therefore, such scalability of a video stream allows for media bit rate as well as for device capability adaptation without the need of transcoding or re-encoding [26]. We may protect the important part of the scalable media (the base layer) and give less protection to the enhancement layer in order to overcome the most typical deteriorated network situations. Intuitively, when the transmission rate is low, we can drop some of the enhancement layers and transmit more base layer packets in order to guarantee playout continuity,

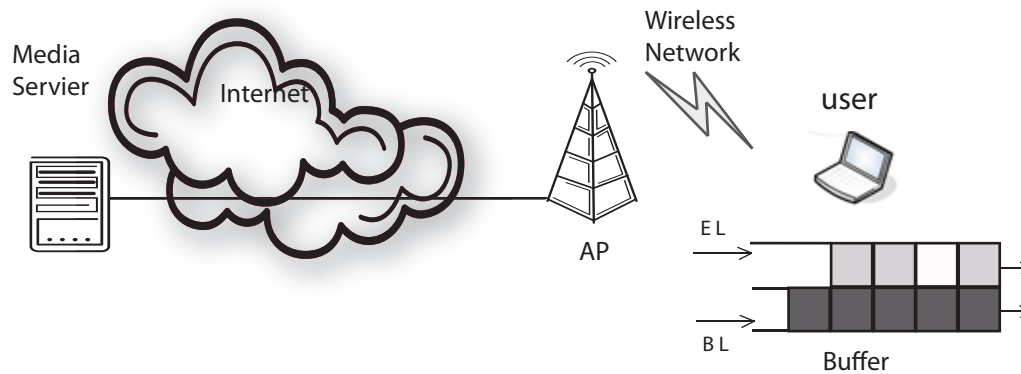


Figure 4.1: A typical illustration of video streaming system and the buffer model in the wireless user.

while with a the high bandwidth, we can increase the enhancement layers ratio so as to enhance the video quality. Therefore, properly control of the transmission rates for both of the layers provides a way for the media streaming system to adapt to the vacillating wireless network conditions.

In this Chapter, we consider the transmission of on-demand scalable *variable-bit-rate* (VBR) video over wireless networks. We initiate an analytical model for the end user's receiver buffer. Based on the knowledge of the current buffered data and the estimation of the future network condition, the receiver can send signals to the sender through a feedback channel to control the sending rate. Furthermore, in order to evaluate the user-perceived streaming media quality, we adopt a cost function combining the weighted sum of the base layer loss and the enhancement layer loss. Particularly, the base layer loss models the continuity of the video transmission and the enhancement layer loss characterizes the video quality degradation. Dynamic programming based algorithm is applied to achieve the optimal policy for the choices of the sending rates. We also propose a greedy based algorithm that takes less time of execution but at the expenses of performance declination.

The rest of the this chapter is organized as follows. The system model is presented and the problem is formulated in Section 4.2. We derive the dynamic programming policy in Section 4.3. Section 4.4 presents the greedy policy. Simulation results and further discuss are provided in Section 4.5. We conclude the chapter in Section 4.6.

## 4.2 System Model

As described in Section 4.1, the video is stored in the remote server and transmitted to wireless users through AP. We consider the transmission of VBR video, so the encoding rates of the base and enhancement layers are varying from time to time, denoted as  $C_b(t)$  and  $C_e(t)$  respectively. We model the video streaming process as time-discrete with equal time slots. Specifically, time is divided into slots  $[t_k, t_{k+1})$ , where  $k \in [0, n-1]$  and  $n$  is the total number of time slots for the whole streaming process. Also, we have  $t_0 = 0$  as the beginning of the streaming. Suppose before the video starts playing out, an *initial delay* is employed and at time  $t_0$  there are  $\Delta_b$  Base Layer (BL) packets and  $\Delta_e$  Enhancement Layer (EL) packets stored in the receiver buffer already by then. Thereafter, multiple video packets are sent to the user in each time slot.

We assume the transport channel to be error free, possibly due to an ideal error control mechanism, such as coding or ARQ, so that video packets' losses may only occur by missing the playout deadline. In such cases, tackling fluctuating application throughput becomes more important than recovering errors. So the feature we are interested in capturing is the time-varying nature of the wireless channel, whether it is in 802.11, 802.16, or cellular. Let  $R(t)$  be the wireless network transmission rate at time  $t$ , which is also the maximum rate our server streams the video at  $t$ .  $R(t)$  depends on the channel state at time  $t$  and  $R(t)$  is from the set of  $R_i$ . At any point in time, the sender devotes

a certain percentage  $\pi_b$  of the wireless network bandwidth to transmit the BL packets, and  $\pi_e$  for EL packets. Apparently,  $\pi_b + \pi_e = 1$ , but note that  $\pi_b$  and  $\pi_e$  can vary over time.

On the user side, the video receiver consists of a BL playout buffer and an EL playout buffer, which is also shown in Figure 4.1. The playout buffers are used to temporarily store the incoming video packets. Noting the delay-sensitivity of the video streaming technology, the fluctuations in transmission rate may possibly lead to late packet arrivals, packet losses, significant playback interruptions and video quality degradation. In one time slot, if the required base layer packets do not arrive by the playback schedule time, we can not decode the corresponding frame and we denote the missing amount of BL as  $L_b$ . Encountering such occurrence of interruptions, we assume that certain concealment technology [27] is incorporated to guarantee continuous playback. The error-concealment techniques are motivated by the insensitivity of human perception to high frequency components. They recover the lost information by making use of some *a priori* knowledge about the video signals, primarily the temporal and spatial smoothness property. Typically, the receiver-based schemes perform loss-concealment actions by repeating the last received packet, or by pattern matching using small segments of samples immediately before or after lost packets, or by performing waveform substitution based on previously received frames on each subband of linear prediction (LP) residues [28]. These strategies only work well when losses are infrequent and when packet sizes are small. Due to the high probability of losses in the wireless network, these schemes are not very promising. Thus, the users' viewing experience could be largely degraded even if we apply the concealment.

The loss amount of the enhancement layer is denoted as  $L_e$ . The layered structure of

the scalable video stream and different priorities between the layers provide two sources of enhancement layer loss: one is due to the late arrival of EL packets  $L_e^1$ ; the other suffers from the missing of the corresponding BL packets in the same frame  $L_e^2$ . As mentioned in Section 4.1, the base layer provides a basic level of quality and can be decoded independently of enhancement layers. Hence, without the corresponding base layer, the received enhancement layers can not be used for decoding and thereby added to the loss.

Apparently, there exists a tradeoff between the video continuity and video quality when we decide the distribution of the network resources for BL and EL transmissions. A superior rate control scheme should strike a balance between these two factors to achieve optimized overall user satisfaction. Toward this goal, we introduce a cost function  $C$  as the weighted sum of  $L_b$  and  $L_e$ :

$$C = \omega_b L_b + \omega_e L_e = \omega_b L_b + \omega_e L_e^1 + \omega_e L_e^2. \quad (4.1)$$

where  $\omega_b$  and  $\omega_e$  are the weights decided by the video viewer's preference and ranging from  $[0, 1]$ . In this cost function,  $L_b$  models the continuity degradation of the streaming process since without the presence of the base layer packets, the corresponding video frame can not be played. On the other hand,  $L_e$  captures the reduction of video frame quality, which could be caused by both the loss of the enhancement layer packets and the corresponding base layer packets.

We denote  $B_b(t_k)$  and  $B_e(t_k)$  as the buffered BL and EL data in the receiver buffers at  $t_k$ . We also define  $D_b(t_k)$  and  $D_e^1(t_k)$  as the “virtual” buffered data at  $t_k$ , which is calculated as the remained BL/EL data from the previous time slot  $[t_{k-1}, t_k)$  plus the newly incoming data minus the consumed data during  $[t_k, t_{k+1})$ :

Notation	Definition
$\Delta_b, \Delta_e$	Buffered BL and EL data during the startup delay.
$B_b(t_k), B_e(t_k)$	The BL and EL packets in the buffers at $t_k$ .
$n$	Length of a video in time slots.
$N$	The total number of control time periods.
$\Delta T$	The control period duration.
$t_k$	Index of the basic time units for transmission and playback. $k \in [0, n - 1]$ .
$T_k$	Index of the rate controls. $k \in [0, N - 1]$ .
$C_b(t), C_e(t)$	The minimum amount of data that has to be received by $t$ for BL and EL.
$R(t)$	Wireless network rate at time $t$ .
$K$	Number of states of the channel.
$L_b, L_e$	The BL and EL video packets losses.
$\omega_b, \omega_e$	The weight of the BL and EL losses.
$\pi_b, \pi_e$	The percentages of network bandwidth assigned to BL and EL transmissions. $\pi_b + \pi_e = 1$ .

Table 4.1: Table of nomenclature

$$D_b(t_k) = \underbrace{B_b(t_{k-1})}_{\text{Remained BL}} + \underbrace{\int_{t_{k-1}}^{t_k} R(t)\pi_b dt}_{\text{New BL}} - \underbrace{\int_{t_{k-1}}^{t_k} C_b(t) dt}_{\text{To be consumed BL}} . \quad (4.2)$$

$$D_e^1(t_k) = B_e(t_{k-1}) + \int_{t_{k-1}}^{t_k} R(t)\pi_e dt - \int_{t_{k-1}}^{t_k} C_e(t) dt.$$

It is obvious that  $D_b(t_k)$  and  $D_e^1(t_k)$  can be positive or negative. If they are positive, it means that there is no video packet loss and we have stored the packets for future use in advance. Otherwise, if they are negative, it means there exists packet loss. We then express the loss as the absolute value of  $D_b(t_k)$  and  $D_e^1(t_k)$ .

We tabulate the notations used in the rest of the chapter in Table 4.1.

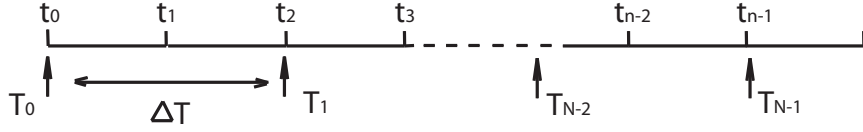


Figure 4.2: Illustration the time slots  $t_n$  and the control epoch  $T_N$ .

### 4.2.1 Problem Formulation

The receiver is aware of the video consumption information. In order to adapt to the varying network transmission rate, we adopt a rate control policy, in which the receiver periodically sends feedback signals to the sender to adjust the value of  $(\pi_b, \pi_e)$  based on the combined knowledge of buffer level, consumption rates, and the estimated future network condition. The goal is to achieve the minimized total cost defined in (4.1). We denote  $N$  as the total number of rate control intervals, and  $T_0, T_1, \dots, T_{N-1}$  as the epoches of control executions, where  $T_0 = t_0$ . The time duration between two consecutive control is fixed:  $\Delta T = \lceil T/N \rceil$ <sup>1</sup>. Figure 4.2 illustrates the relationship between the control period and the basic transmission time units.

Based on the above analysis, we formulate the problem as follows:

---

<sup>1</sup> $\lceil X \rceil$  returns the smallest integer greater than or equal to  $X$ .

Minimize  $\sum_{k=0}^{n-1} \{\omega_b L_b(t_k) + \omega_e L_e(t_k)\}$

$$\text{where } L_b(t_k) = \begin{cases} |D_b(t_k)|, & \text{if } D_b(t_k) < 0 \\ 0, & \text{otherwise} \end{cases}$$

$$L_e(t_k) = \begin{cases} |D_e^1(t_k)|, & \text{if } D_e^1(t_k) < 0 \text{ and all of the } \textit{corresponding} \\ & \text{base layer packets have been received} \\ & \text{successfully.} \\ \Gamma(|L_b(t_k)|), & \text{if the } \textit{corresponding} \text{ base layer packets} \\ & \text{are lost.} \\ 0, & \text{if } D_b(t_k) \geq 0 \text{ and } D_e(t_k) \geq 0 \end{cases} \quad (4.3)$$

$$\pi_b(t_k) + \pi_e(t_k) = 1, 0 \leq k \leq n - 1$$

where  $\Gamma(X)$  calculates the traffic amount of enhancement layers corresponding to the base layers  $X$ , which is determined specifically by the scalable coding scheme and the video source itself. In the optimization problem,  $\pi_b$  and  $\pi_e$  are the control parameters.

At each of the control epochs, we adjust the value of  $\pi_b$  and  $\pi_e$  for the next period of  $\Delta T$ . Denote the control action as  $a = (\pi_b, \pi_e)$ . Then the transmission policy is  $A = (a(0), a(1), \dots, a(N - 1))$ , where  $a(t)$  is the control decision made at epoch  $t$ . Our objective is to find the optimal policy  $A$  which minimizes the total loss.

### 4.3 Dynamic Programming Based Algorithm

In this section, we study the problem in a dynamic programming (DP) framework [29]. We define the state of the system as  $s = (B_b, B_e)$ , where  $B_b$  and  $B_e$  are the remained BL and EL video packet numbers in the buffer at the current time stamp. Let  $\Lambda$  be the set of control actions  $a$ , which contains all of the possible combinations of  $\pi_b$  and  $\pi_e$ .

We denote  $q_{s,s'}(a; T_k)$  as the transition probability from the state  $s : (B_b(T_k) = i, B_e(T_k) = j)$  at  $T_k$  to the next state  $s' : (B_b(T_{k+1}) = h, B_e(T_{k+1}) = k)$  at  $T_{k+1}$  while taking the action of  $a$ . The value of  $q_{s,s'}(a; T_k)$  not only depends on the states and the action, but on the time  $T_k$  as well, because the video is VBR encoded and the consumption speed of data varies over time. Consequently, the resource distribution will change over time in order to adapt to the video consuming requirement. Also, we know that the transition matrix is non-homogeneous and different from time to time.

Let  $l_s(a; T_k)$  denote the loss from time  $T_k$  to  $T_{k+1}$  with the state  $s = (B_b(T_k), B_e(T_k))$  at  $T_k$  and employing the action of  $a$ .  $J_s(T_k)$  is defined as the optimal cost to go, i.e., the minimized total loss starting at  $T_k$  from current state  $s$  to the end of the video stream assuming that optimal control is used at every control epoch. Then the system becomes a finite-horizon controlled Markov decision process [30] and the optimal control policy  $A$  can be obtained from the following optimality equation:

$$J_s(T_k) = \min_{a \in \Lambda} \{l_s(a; T_k) + \sum_{s'} q_{s,s'}(a; T_k) J_{s'}(T_{k+1})\}. \quad (4.4)$$

#### 4.3.1 Algorithm

The system starts at  $T_0 = 0$  with the state  $s_0 = (\Delta_b, \Delta_e)$ , where  $\Delta_b$  and  $\Delta_e$  are the buffered data during the startup delay period. Then the system evolves according to

the actions generated from Equation (4.4), while  $T_k$  is the control epochs with  $k$  ranging from 0 to  $N - 1$ . The Markov decision problem computation is based on using backward induction to recursively evaluate expected costs. We present the algorithm that solves optimality equation (4.4) in Algorithm 1.

---

**Algorithm 1** Find the optimal policy:  $A = (a(0), a(1), \dots, a(N - 1))$

---

**Require:**  $N, \Lambda, C_b(t), C_e(t)$

```

1:  $k = N - 1$ 
2: for all states  $s$  do
3:    $J_s(T_{N-1}) = \min_{a \in \Lambda} l_s(a; T_{N-1})$ 
4: end for
5: while  $k \geq 0$  do
6:    $k \leftarrow k - 1$ 
7:    $t \leftarrow T_k$ 
8:   for all states  $s$  do
9:      $J_s(T_k) = \min_{a \in \Lambda} \{l_s(a; T_k) + Q\}$ ,
10:    where  $Q = \sum_{s'} q_{s,s'}(a; T_k) J_{s'}(T_{k+1})$ 
11:   end for
12:   set  $a_s(T_k) = \arg_{a \in \Lambda} J_s(T_k)$ 
13: end while

```

---

### 4.3.2 Use of the policy

Solving the dynamic programming formulation involves recursively computing and filling up two tables, in a bottom-up way: one table stores the optimal controls and another stores the resulting cost  $J_s(T_k)$  for every system state  $s$  at every control epoch. The optimal policy should be computed and stored in a table at the receiver. At each of the control epoch, the receiver collects the joint information of current buffer state, video progress, and network condition. A lookup is next performed in the table to find the optimal control for current state of the system. Then the receiver sends signals back to the sender providing the updated values of  $\pi_b$  and  $\pi_e$ .

### 4.3.3 Remark

The DP based algorithm is able to find the optimal control policy for the Markov decision process at an expected level. The algorithm uses recursive computation which depends on the sizes of the buffer state space and possible control sets. Based on the magnitude and granularity of the above variables, the computation complexity can be quite high. In the following section, we propose a greedy based algorithm which only focuses on minimizing the total loss during the current control period.

## 4.4 Greedy Algorithm

### 4.4.1 Problem Formulation

In this section, we propose the online greedy based algorithm, which is executed at each control epoch. The goal of the Dynamic Programming Based Algorithm described in last section is the video packet loss optimization over the whole streaming process. But the focus of the online greedy algorithm is only on the next time period  $[T_k, T_{k+1})$  and the goal is to minimize the total loss in  $[T_k, T_{k+1})$ , which is denoted as  $L(T_k)$ :

$$L(T_k) = \omega_b L_b(T_k) + \omega_e L_e(T_k) \quad (4.5)$$

$$= \sum_{t_k=T_k}^{T_k+\Delta T} \{\omega_b L_b(t_k) + \omega_e L_e(t_k)\} \quad (4.6)$$

To simplify the expression, we denote  $\int_{T_k}^{T_{k+1}} R(t)dt$  as  $\bar{R}(T_k)$ ,  $\int_{T_k}^{T_{k+1}} C_b(t)dt$  as  $\bar{C}_b(T_k)$ , and  $\int_{T_k}^{T_{k+1}} C_e(t)dt$  as  $\bar{C}_e(T_k)$ . Then, we have:

$$D_b(T_{k+1}) = \bar{R}(T_k)\pi_b + \underbrace{D_b(T_k) - \bar{C}_b(T_k)}_{N_b(T_k)} \quad (4.7)$$

$$D_e^1(T_{k+1}) = \bar{R}(T_k)\pi_e + \underbrace{D_e(T_k) - \bar{C}_e(T_k)}_{N_e(T_k)} \quad (4.8)$$

#### 4.4.2 Algorithm

$N_b(T_k)$  and  $N_e(T_k)$  presents whether the remained data in the buffer are sufficient for future playout. We then provide the greed algorithm based on different situations shown as follows: (we omit  $T_k$  in  $N_b(T_k)$ ,  $N_e(T_k)$ , and  $\bar{R}(T_k)$ .)

- 1) Calculate current  $N_b$  and  $N_e$ .
- 2) Estimate the total incoming packets amount  $\bar{R}$  based on the previous wireless network statistics.
- 3) Discussion of different situations:

(3.1) If  $N_b \geq 0$  and  $N_e \geq 0$ , the remained packets alone are already enough to sustain the future playout, so  $L(T_k) = 0$ .

$$\pi_b = \text{GetPib} (N_b, N_e, \bar{R}, T_k).$$

(3.2) If  $N_b \geq 0$  and  $N_e < 0$ , which means  $L_b(T_k) = 0$ ,  $L_e(T_k) \geq 0$ .

Then if  $\bar{R} + N_e < 0$ , we know that the estimated future incoming resource will be not enough to transmit all of the needed enhancement layer data, then  $L_e(T_k) > 0$ , so we choose  $\pi_e = 1$ ;

If  $\bar{R} + N_e \geq 0$ , we will receive enough packets for both the BL and EL, then we distribute the  $(\pi_b, \pi_e)$  based on the future consumption rate ratio.  $\pi_b = \text{GetPib}(N_b, N_e, \bar{R}, T_k)$ .

(3.3) If  $N_b < 0$  and  $N_e \geq 0$ ,  $L_e(T_k) \geq 0$  (because of EL's dependence on BL),  $L_b(T_k) \geq 0$ .

Then if  $\bar{R} + N_b < 0$ , we will definitely choose  $\pi_b = 1$ ;

if  $\bar{R} + N_b \geq 0$ ,  $\pi_b = \text{GetPib}(N_b, N_e, \bar{R}, T_k)$ .

(3.4) If  $N_b < 0$  and  $N_e < 0$ , the remained data can not sustain the future playout without new data.

(3.4.a) If  $\bar{R}(t) + N_b + N_e \geq 0$ ,  $L(T_k) = 0$ . It requires:

$$-\frac{N_b}{\bar{R}(t)} \leq \pi_b \leq 1 + \frac{N_e}{\bar{R}(t)}. \quad (4.9)$$

(3.4.b) If  $\bar{R}(t) + N_b + N_e < 0$  but  $\bar{R}(t) + N_b > 0$ ,  $L_e(T_k) > 0$  and  $L_b(T_k) < 0$ . We choose  $\pi_b = -N_b/\bar{R}(t)$ .

If  $\bar{R}(t) + N_b + N_e < 0$  and  $\bar{R}(t) + N_b < 0$ ,  $L_e(T_k) > 0$  and  $L_b(T_k) > 0$ . We choose  $\pi_b = 1$ .

In the above algorithm, function  $\text{GetPib}(N_b, N_e, \bar{R}, T_k)$  is used to get the value of  $\pi_b$  such that the ratio between  $\bar{R}\pi_b + N_b$  and  $\bar{R}\pi_e + N_e$  is the same as that between the average BL and EL consumption rates. We present the algorithm that find the optimal  $(\pi_b, \pi_e)$  to minimize  $L(T_k)$  in Algorithm 2.

---

**Algorithm 2** Greedy Based Algorithm at each control epoch  $T_k$ :

---

**Require:**  $N$ ,  $(C_b(T_k), C_e(T_k))$ , Current time  $T_k$ ,

**if**  $N_b(T_k) \geq 0$  and  $N_e(T_k) \geq 0$  **then**

$\pi_b = \text{GetRatio}(N_b(T_k), N_e(T_k), \overline{R}(T_k), T_k)$

**end if**

**if**  $N_b(T_k) \geq 0$  and  $N_e(T_k) < 0$  **then**

**if**  $\overline{R}(t) + N_e < 0$  **then**

$\pi_b = 0$

**else**

$\pi_b = \text{GetRatio}(N_b(T_k), N_e(T_k), \overline{R}(T_k), T_k)$

**end if**

**end if**

**if**  $N_b < 0$  and  $N_e \geq 0$  **then**

**if**  $\overline{R}(t) + N_b < 0$  **then**

$\pi_b = 1$

**else**

$\pi_b = \text{GetRatio}(N_b(T_k), N_e(T_k), \overline{R}(T_k), T_k)$

**end if**

**end if**

**if**  $N_b < 0$  and  $N_e < 0$  **then**

**if**  $\overline{R}(t) + N_b + N_e \geq 0$  **then**

$\pi_b = \text{GetRatio}(N_b(T_k), N_e(T_k), \overline{R}(T_k), T_k)$

**else**

**if**  $\overline{R}(t) + N_b > 0$  **then**

$\pi_b = -N_b/\overline{R}(t)$

**else**

$\pi_b = 1$

**end if**

**end if**

**end if**

---

Function **GetRatio**: outputs the  $\pi_b$  such that the resulted ratio of BL and EL is the same as that of the average encoding rate of BL and EL.

---

### 4.4.3 Remark

The greedy algorithm can be easily implemented and consume short computation time. However, it may not provide the overall optimal performance as the DP based algorithm since it is only focused on the next future control period. We can anticipate that as the control period expands, the performance of the greedy based algorithm will be closer to that of the DP based algorithm. But it also depends on the accuracy of the wireless network bandwidth estimation.

## 4.5 Simulation

In this section, we show results from extensive simulations by which we evaluate our proposed rate-control algorithms and analyze the impact of system parameters, such as the wireless channel statistics, the weights for BL and EL losses, and the control interval length.

### 4.5.1 Simulation Setup

We use the MPEG-4 variable-bit-rate video trace provided by [23]. The video sequences are encoded in the Common Image Format (CIF) resolution. The video frames are encoded into two layers: base layer and enhancement layer, which exhibit temporally scalable. Table 4.2 summarizes the main parameters of the video trace in the simulation. The packet size is set to be 2100 bytes and the transmission time slot duration is 800 ms.

For the wireless channel model, we use  $R$  as the set of possible channel rates and  $Pr$  as the corresponding probability set for different rates. At each time step, the system will randomly choose one rate from  $R$  based on  $Pr$ . We apply 3 sets of wireless

Parameter	Value
Sequence Length	120 frames
Video Size	$6.1e + 05$ bytes
Format	CIF ( $352 \times 288$ pixels)
Video Run Time	$9.6e + 05$ msec
Mean Bit Rate	$7.5e + 04$ bps
Peak Bit Rate	$1.5e + 05$ bps

Table 4.2: Video parameters in simulation

channel parameters, in which  $R_1 = [0, 2, 4, 6]$  packets per unit time slot with  $Pr_1 = [0.45, 0.45, 0.05, 0.05]$ ;  $R_2 = [0, 2, 4, 6]$  packets per unit time slot with  $Pr_2 = [0.1, 0.2, 0.5, 0.2]$ ;  $R_3 = [0, 2, 6, 10]$  packets per unit time slot with  $Pr_3 = [0.3, 0.17, 0.3, 0.5]$ . The ratios between the average channel rate and the average video consumption rate are 0.3, 1.0, and 2.0. Discrete action set:  $a = [0, 0.1, 0.2, \dots, 0.9, 1]$  is used for  $\pi_b$  and  $\pi_e$ . The initial packets in the buffer  $\Delta_b$  and  $\Delta_e$  are set to be  $4.2e + 03$  bytes and  $2.1e + 03$  bytes. We simulation the transmission and playback for the target video sequence in Matlab for over 500 realizations of the random VBR channel and obtain the results by implementing different algorithms.

### 4.5.2 Constant Algorithms

Before presenting the performances of the proposed DP algorithm and greedy algorithm, we introduce the “non-control” algorithm for the purpose of comparison with the adaptive rate control algorithms. In the constant algorithm the values of  $\pi_b$  and  $\pi_e$  are not changing during the whole streaming process. We set  $\pi_b = 0.4$  and  $\pi_e = 0.6$ . The ratio between them are approximately the same as the ratio between the BL and EL average consumption rates.

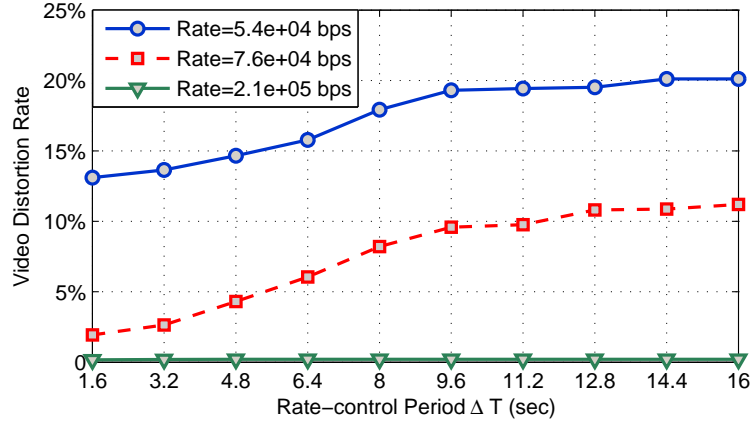


Figure 4.3: The impact of control time period for DP algorithm for different network settings.  $\omega_b$  is chosen as 0.8

### 4.5.3 Impact of Control Duration $\Delta T$ and Channel Rate

The impact of control time length  $\Delta T$  and the network settings for the dynamic programming based algorithm is shown in Figure 4.3. We define the evaluation metric as the video distortion rate, which is the weighted sum of packet losses divided by the weighted sum of total video packets. We observe that when the average channel rate is lower than or approximately equal to the average video consumption rate, the distortion rate decreases as  $\Delta T$  becomes smaller. In other words, to receive improved video streaming quality we should increase the frequency of rate controls in DP algorithm. But with a high channel rate, the video distortion rate is below 1%. This suggests that when we have plenty network resource, the buffer starvation probability is quite low and shortening the control period will not make a big difference on the overall system performance.

Furthermore, we compare the performances of the DP, greedy and constant algorithms with various control time durations and channel rates. The results are plotted in Figure 4.4. From Figure 4.4(a), we see that, the DP algorithm and the greedy algorithm largely improve the system’s performance compared to the “non-control” algorithm when the

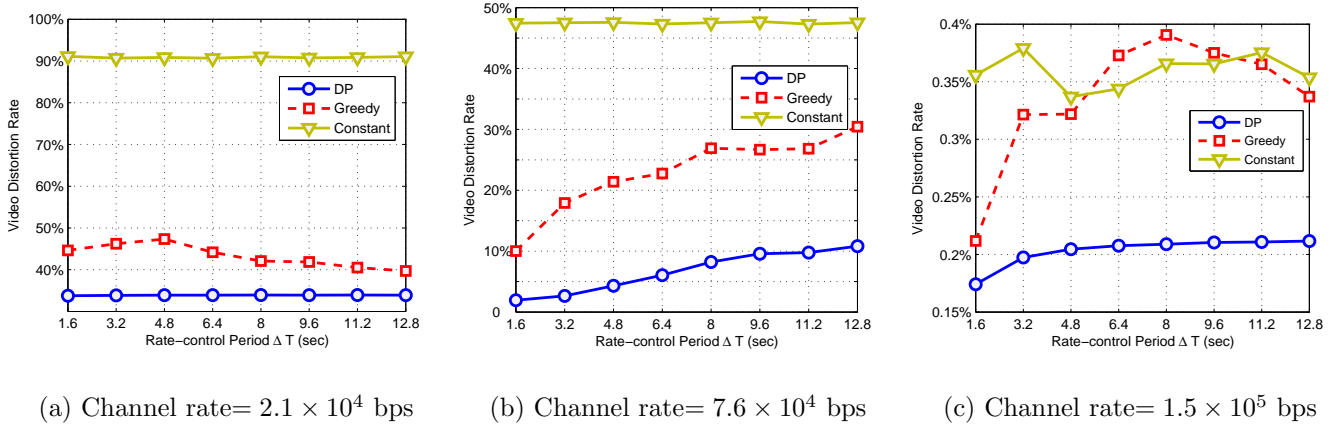


Figure 4.4: Compare the three algorithms with different control time duration.  $\omega_b$  is chosen as 0.8

network bandwidth is insufficient. Moreover, as the  $\Delta T$  increases, the performance of greedy algorithm almost matches that of DP algorithm. When the average channel rate and the video consumption rate is equal as shown in Figure 4.4(b), both the distortion rates of DP and greedy algorithms are increasing when  $\Delta T$  is becoming larger. But they are still outperforming the constant algorithm. From Figure 4.4(c), since the network condition is very good, the distortion rates from all of the algorithms are below 1%, which suggests that there is no need to implement the rate control algorithms in this circumstance.

#### 4.5.4 Impact of the Loss Weights

We compare the algorithms with different  $(\omega_b, \omega_e)$  in Figure 4.5 and Figure 4.6. Since in real practice the weight of BL losses is always larger than that of the EL losses, we set the values of  $\omega_b$  from 0.6 to 1. We observe that as the  $\omega_b$  increases, the video distortion rates of the DP and greedy algorithms are decreasing accordingly, which implied that

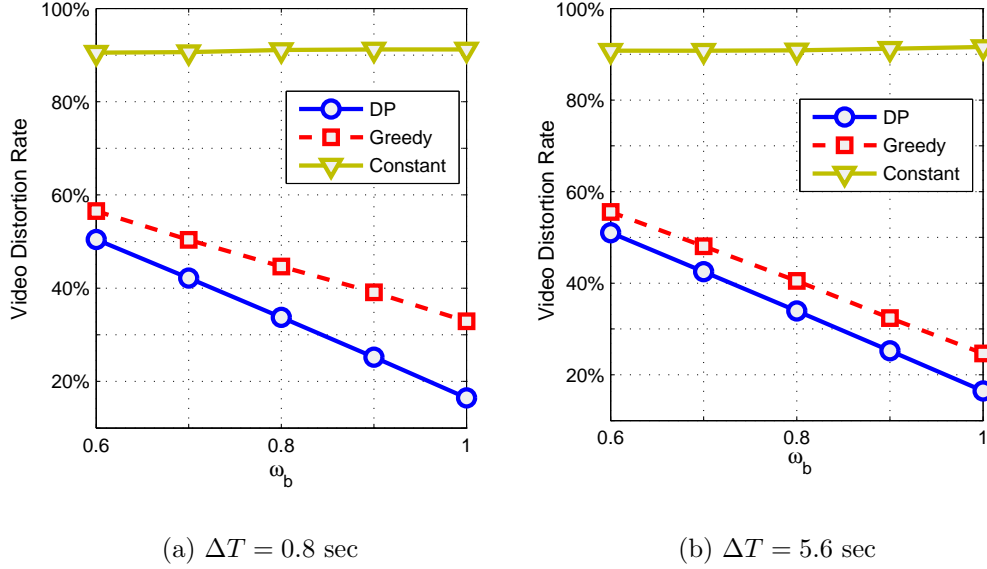


Figure 4.5: Comparison of the algorithms with different  $\omega_b$ . The average channel rate is  $2.1 \times 10^4$  bps

the proposed algorithms are favoring the BL packets. The larger weight we put on the BL losses, the better system performance we will achieve by using the DP and greedy algorithms.

## 4.6 Summary

We have studied the problem of stochastic rate control policies for VBR layer encoded video streaming over wireless networks. We formulate the problem as a Markov Decision Process and solve it in a dynamic programming framework. We combine the video receive's information of current buffer situation, video consumption rates, and the estimation of wireless network condition to obtain the optimal control policy. Then the sender will adjust the proportion of network resource for sending base layer and enhancement

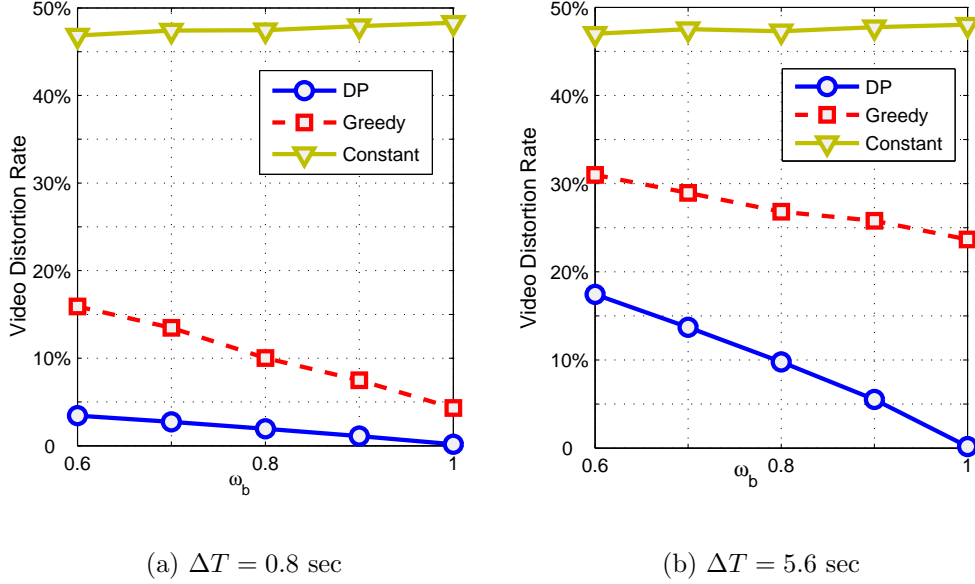


Figure 4.6: Comparison of the algorithms with different  $\omega_b$ . The average channel rate is  $7.6 \times 10^4$  bps

packet based on the feedback signals. The DP based algorithm provides the optimal policy but could be inefficient when the magnitude and granularity of the system parameters largely increase. Then we propose a greedy based algorithm which is targeting to minimize the loss only in current control period. Simulation results show that when the average network rate is smaller than or equal to the average video consumption rate, the DP and the greedy algorithm provide largely improved system performance compared to the “non-control” algorithm.

# Chapter 5

## Conclusion

Video streaming is one of the most challenging services to offer because of the high and consistent bandwidth requirements of the digital video bitstreams. In this thesis, we have considered the problem of providing QoS to VBR encoded video streaming service over random VBR channels. We have shown that, for VBR video streaming over heterogeneous networks, a certain level of QoS can be guaranteed by utilizing the location information of a mobile client. Furthermore, through using scalable encoded video and appropriate rate control, the video transmission efficiency could be largely improved.

We first present an analytical framework for variable-bit-rate (VBR) video streaming in a two-tier wireless network with VBR channels, and derive the expected number of jitters and average buffering delay during video playback as measures of system performance. Our objective is to discover heterogeneous networking attributes that may influence the streaming performance, in terms of the tradeoff between jitter frequency and buffering delay. The frequency of jitters and the expected jitter recovery buffering delay have been derived for both the infinite buffer and finite buffer cases. Numerical

and experimental results using MPEG-4 encoded VBR video traces validate our findings. Through experimenting with a wide variety of buffering schemes, including fixed, separate, and jointly optimal schemes based on buffering delay, buffered playout data, or buffered playout time, we find that the streaming performance can be significantly improved by incorporating user location information, and separate optimization within the subnetworks is unsuitable. Furthermore, we show that buffering based on playout time is more appropriate for simple mobile devices that has limited storage memory and uses a constant buffering parameter regardless of location.

We secondly present a framework of the VBR layer encoded video streaming over wireless networks. We formulate the problem as a Markov Decision Process and solve it in a dynamic programming framework. The video receive's information of current buffer situation, video consumption rates, and the estimation of wireless network condition are combined to achieve the optimal control actions. We then propose a greedy based algorithm which is targeting to minimize the loss only in current control period. Simulation results using MPEG-4 variable-bit-rate video trace with temporal scalability show that when some statistical characteristics of the channel are available, adaptively selecting appropriate sending rates for the base layer and enhancement layer will provide largely improved system performance compared to the "non-control" algorithm. We also show that when the average network rate is smaller than or equal to the average video consumption rate, the DP and the greedy algorithm. To practical streaming system designers, the proposed analysis techniques and control algorithms provide convenient frameworks to optimize the tradeoffs between the various system parameters for optimal VBR multimedia streaming over random VBR channels.

# Bibliography

- [1] C. Huang, J. Li, and K. W. Ross, “Can internet video-on-demand be profitable?”, in *SIGCOMM’07*, Kyoto, Japan, Aug 2007, vol. 27.
- [2] C. Chiasserini, M. Meo, D. Tarfanelli, and D. Visconti, “A study of video services in a wireless environment”, in *The 4th IEEE Conference on Mobile and Wireless Communications Networks*, Sep. 2002.
- [3] T.V. Lakshman, A. Ortega, and A.R. Reibman, “Variable bit rate (VBR) video: Tradeoffs and potentials”, *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.
- [4] D. Axiotis, T. Al-Gizawi, E. Protonotarios, F. Lazarakis, C. Papadias, and P. Philippopoulos, “Services in interworking 3G and WLAN environments”, *IEEE Wireless Commun. Mag.*, vol. 11, no. 5, pp. 14–20, Oct. 2004.
- [5] S. Sen, J. L. Rexford, J. K. Dey, J. F. Kurose, and D. F. Towsley, “Online smoothing of variable-bit-rate streaming video”, *IEEE Trans. Multimedia*, vol. 2, no. 1, pp. 37–48, Mar. 2000.

- [6] W. E. Naylar and L. Kleinrock, “Stream traffic communication in packet switched networks: Destination buffering considerations”, *IEEE Trans. Communication*, vol. 30, no. 12, pp. 2527–2534, Dec. 1982.
- [7] P. Thiran, J. Y. Le Boudec, and F. Worm, “Network calculus applied to optimal multimedia smoothing”, in *Proc. of IEEE INFOCOM*, 2001.
- [8] V. Varsa and I. Curcio, “Transparent end-to-end packet switched streaming service (pss); rtp usage model (release 5)”, *3GPP TR 26.937 V1.4.0*, 2003.
- [9] T. Stockhammer, H. Jenkac, and G. Kuhn, “Streaming video over variable bit-rate wireless channels”, *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 268–277, Apr. 2002.
- [10] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, “Measurement and modelling of the temporal dependence in packet loss”, in *Proc. of IEEE INFOCOM*, 1999.
- [11] M. Kalman, E. Steinbach, and B. Girod, “Adaptive media playout for low-delay video streaming over error-prone channels”, *IEEE Trans. Circuits and System for Video Technology*, vol. 14, no. 6, pp. 841–851, Jun. 2004.
- [12] L. Xu and J. Helzer, “Media streaming via tfrc: An analytical study of the impact of tfrc on user-perceived media quality”, in *Proc. of IEEE INFOCOM*, Apr. 2006.
- [13] G. Liang and B. Liang, “Balancing interruption frequency and buffering penalties in vbr video streaming”, in *Proc. of IEEE INFOCOM*, May 2007.
- [14] S. Flyod, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications”, *Proc. ACM SIGCOMM*, pp. 43–56, Aug. 2000.

- [15] W. Tan and A. Zakhor, “Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol”, *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 172–186, Jun. 1999.
- [16] M. Chen and A. Zakhor, “Rate control for streaming over wireless”, in *Proc. IEEE INFOCOM*, Apr. 2004.
- [17] J. Cabrera, A. Ortega, and J. Ronda, “Stochastic rate-control of video coders for wireless channels”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 496–510, Jun. 2002.
- [18] L. Atzori, M. Krunz, and M. Hassan, “Cycle-based rate control for one-way and interactive video communications over wireless channels”, *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 176–184, Jan. 2007.
- [19] D. Saporilla and K.W. Ross, “Optimal streaming of layered video”, in *Proc. IEEE INFOCOM*, Mar. 2000.
- [20] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, fourth edition, 2002.
- [21] E. N. Gilbert, “Capacity of a burst-noise channel”, *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, Oct. 1960.
- [22] H. Sanneck, G. Carle, and R. Koodli, “A framework model for packet loss metrics based on loss run length”, in *Proceedings of SPIE/ACM SIGMM Multimedia Computing and Networking Conference*, Nashville, USA, Jan. 2000.
- [23] P. Seeling, F. H. P. Fitzek, and M. Reisslein, *Video Traces for Network Performance Evaluation*, Springer, Nov. 2006.

- [24] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of h.264/avc”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
- [25] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h.264/avc video coding standard”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [26] T. Schierl, T. Stockhammer, and T. Wiegand, “Mobile video transmission using scalable video coding”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1204–1217, Sep. 2007.
- [27] Y. Wang and Q.F. Zhu, “Error control and concealment for video communication: A review”, *Proceedings of the IEEE*, vol. 86, pp. 974–997, May 1998.
- [28] B. W. Wah, X. Su, and D. Lin, “A survey of error-concealment schemes for real-time audio and video transmissions over the internet”, in *IEEE International Symposium on Multimedia Software Engineering*, Dec. 2000.
- [29] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1 & 2, Athena Scienrific, 2 edition, 2000.
- [30] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, New York, 1994.