BALANCING INTERRUPTION FREQUENCY AND BUFFERING PENALTIES IN VBR VIDEO STREAMING

BY

GUANFENG LIANG

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

COPYRIGHT © 2007 BY GUANFENG LIANG.
ALL RIGHTS RESERVED.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-27321-0
Our file Notre référence
ISBN: 978-0-494-27321-0

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To my parents

Balancing Interruption Frequency and Buffering Penalties in VBR Video Streaming

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

by Guanfeng Liang
April 2007

Abstract

We study the optimal streaming of variable bit-rate (VBR) video over a random VBR channel. The goal of a streaming application is to enable the successful decoding of each video object before its displaying deadline is violated. Previous literature has described solutions to estimate the interruption-free probability by assuming either independence in the encoded data process or simplistic channel models. In this thesis, we first present an analytical framework to bound the probability of un-interrupted playback. When a Markov VBR channel model is available, we quantify the distribution of the number of interruptions under the constraint of initial playback delay, receiver buffer size and different recovering schemes. Both the infinite and finite buffer cases are considered. Experimental results with MPEG-4 VBR encoded video validate our analysis. Finally, we show that the proposed analysis provides a theoretical foundation to quantify the tradeoffs between different system parameters.

Acknowledgments

Firstly, I would like to express my sincerest gratitude to my supervisor, Professor Ben Liang, for his guidance, supervision and support throughout the pursuit of my Master of Applied Science degree. Professor Liang motivated me to improve my research work with his insightful suggestions and encouraged me through difficult times with his thoughtful advices. He provided me with an invaluable experience in this exciting research.

To my thesis committee members, Professor H. -Arno Jacobsen, Professor Dimitrios Hatzinakos, and Professor Jorg Liebeherr, thank you for reviewing my thesis. Your valuable feedback and comments allowed me to further enhance my research.

I WOULD like to send my appreciation to the entire colleagues in the WHIMSIC research group. It has been a pleasure working alongside all of you, thank you all for your stimulating discussion and friendly encouragement.

To my friends, thank you for your continuous motivation. I am especially indebted to Phoenix, who has been my emotional support throughout the years.

Lastly, and most importantly, I wish to thank my parents for their love and support throughout my education. To them I dedicate this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The popularity of IP-based video streaming over the Internet is continuously growing, with hundreds of new subscribers registered daily. In addition, existing and emerging wireless systems such as EGPRS, UMTS, CDMA-2000, and WLAN enable IP-based multimedia transmission and reception at any place and time at reasonable and sufficient data rates. Mobile devices of the near future are expected to bring ubiquitous access to streaming multimedia services, such as TV news, music video, and online movie. Streaming multimedia are likely to become major applications in future mobile systems and may indeed be a key factor for their success. However, due to limited resources bandwidth and transmission power, compression efficiency is the main target for wireless video and multimedia applications. This limits the application of error-resilience or scalability features which, in general, suffer from significantly reduced efficiency. In addition, mobile devices are hand-held and still constrained in processing power and storage capacity such that complex receiver algorithms are not applicable [1], [2]. Three major service categories are likely to be integrated in 2.5G and 3G wireless systems: conversational services, packet-switched streaming services (PSS), and multimedia messaging services (MMS).

Whereas conversational services seem to remain a niche application in near-future wireless systems, PSS and MMS are gaining increasing popularity. In this thesis, we focus on the streaming of pre-encoded video to wireless clients, taking into account these system constraints and limited resources.

In general, the transmission rate of the channel varies over time. Video display interruption may occur if data are not delivered on time when the transmission rate does not match the encoded rate. We term the event of playback interruption *playout jitter*, or *jitter*. Clearly, jitter reduces the perceived video quality and is undesirable in video streaming. However, video display interruption may still occur if packets are not delivered on time.

To alleviate the detrimental effect of channel transmission rate variation on video display, a common approach is to delay the playback of a stream through buffering at the receiver, at the beginning of display or after jitters occur. However, jitters can not be eliminated unless the video is fully downloaded before display starts/restarts, which results in unacceptable long initial/jitter recovering delays. For this reason, system designers must trade the reliability of uninterrupted playout against delay when determining the amount of data to buffer.

In this thesis, we quantify the fundamental tradeoffs between the frequency of jitters, jitter-recovery delay, the initial playout delay, and the receiver buffer size, in order to optimize the streaming of variable bit-rate (VBR) encoded videos, with non-linear playback curves [3], over a random VBR channel. Our main contributions include the following:

- When information about the channel is limited, we derive the upper and lower bounds of the jitter-free probability, given the initial playback delay and receiver buffer size, for both infinite buffers and finite buffers.

- When a Markovian channel model is available, we propose an analytical framework to derive the distribution of the number of jitters, for three different jitter-recovery buffering schemes, and for both infinite and finite buffering cases.

- We apply the proposed analysis to optimal streaming over a general wireless network. Numerical analysis results are obtained for wireless systems modeled by a generic Markov channel with Automatic Repeat-reQuest (ARQ) transmission control.

- Experiments using sample MPEG-4 video traces are carried out to validate the proposed analysis and provide new insight into the optimal balancing of delay, buffering, and jitters for optimal multimedia streaming.

The remainder of this thesis is organized as follows. We discuss background information and related work in Chapter 2. In Chapter 3, an analytical framework is presented to bound the jitter probability under constraints of initial delay and receiver buffer size. In Chapter 4, an analytical framework is presented to derive the jitter frequency given different initial delay values and receiver buffer sizes. This framework is applied to study the performance of three different jitter-recovery buffering schemes. Finally, we conclude the thesis in Chapter 5.

Chapter 2

Background and Related Work

In this chapter, we first review some background information about VBR encoded video and wireless video streaming. Then we present a literature survey related to our work.

2.1 Variable Bit Rate Encoded Video

High-quality video requires a large amount of storage space and network bandwidth. Even effective compression techniques, such as MPEG [4] and motion-JPEG [5], still result in video streams with bandwidth requirements in the range of 2-10 megabits/second. 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 viewer. Alternatively, video encoders can generate constant-quality video, resulting in a VBR stream. Constant-quality video typically has higher quality than a constant-bit-rate stream with the same average bandwidth [6], [3]. However, constant-quality video

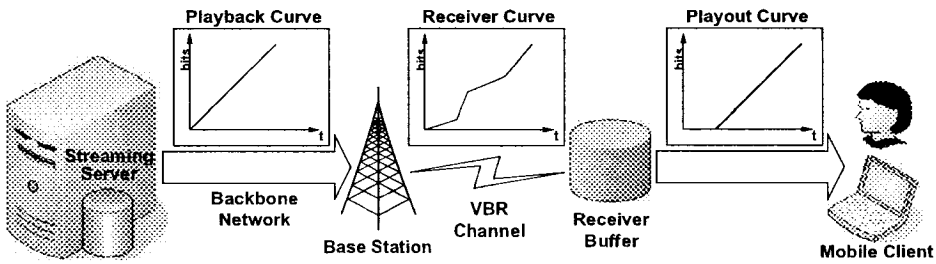


Figure 2.1: A typical wireless media streaming system.

can exhibit signicant burstiness on multiple time scales, due to the natural variations within and between scenes, as well as the frame structure of the encoding algorithm [7], [8], [9], [10], [11].

2.2 Wireless Video Streaming

A general media streaming system is illustrated in Figure 2.1. This system consists of a media streaming *server*, a *transport channel*, and a streaming *client*.

In most cases, content and service providers are reluctant to provide multimedia data, such as pre-encoded video for streaming services, separately for wired and wireless clients. It is expected that a huge amount of content is exclusively stored on a server in the wired Internet accessed by both fixed and wireless clients. We consider general VBR encoded videos in this thesis, with non-linear playback curves [3], because of their superior rate-distortion performance over CBR encoded videos and their growing popularity.

Packet delays and losses are very common in nowadays IP-based packet networks. However, due to predictive video coding, lost IP packets result not only in decoding errors of the current frame, but also in quality degradation of subsequent frames included in the dependency chain. Whereas packet losses and delays in the fixed Internet mainly result from network congestion, wireless transmission packet losses and delays usually originate

from link layer impairment. Therefore, the transport channel is considered to be VBR.

The streaming clients access the video objects stored on the server through a wire or wireless access network. Depending on the channel condition and the chosen jitter recover scheme, a client may experience a number of jitters and some jitter recover buffering delay every time jitter occurs. From a client's point of view, the frequency of jitters experienced should not be too high, and the jitter recover buffering delays should not be too long, either. Unfortunately, these two objectives conflict with each other. To address the balancing of this tradeoff is uniquely important to streaming media systems.

2.3 Related Work

One obvious way to avoid jitter is to start displaying the video only after it is fully downloaded. By doing this, jitter is 100% avoidable, but it also results in the longest initial delay and the maximal buffer size, which is generally unacceptable, especially for small wireless devices. In practice, the system buffers a certain portion of video data at the client before displaying the video, so that transient packet losses and delay do not constantly interrupt the playout of the stream. Intuitively, the more data is buffered, the fewer jitters will occur in the future, but the initial/jitter delay induced by buffering increases, too. For this reason, system designers must trade the reliability of uninterrupted playback against delay and buffer size.

In this context, Sen *et al.* proposed an online smoothing technique for VBR streaming video in [12] by introducing a few seconds of startup delay and a client buffer to compensate for the variation of video encoding rate. Authors of [13] used network calculus analysis to derive an optimal multimedia smoothing scheme. However, both schemes require a dedicated smoothing server or an intermediate smoothing node and only consider

a wired network offering guaranteed bandwidth service. Therefore these schemes are not suitable for error-prone networks such as wireless streaming systems.

V. Varsa and I. Curcio introduced the concept of having two buffers at the receiver - a delay jitter buffer and a decoder buffer [14]. The delay jitter buffer is used to compensate for the delay jitter introduced by the channel and to reduce bit rate variations caused by the VBR behavior of the channel. Streaming video data is first buffered in the delay jitter buffer and then is emitted into the decoder buffer at a constant rate after a 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. Therefore, traditional hypothetical reference decoders (HRD) such as the video buffer verifier (VBV) for MPEG-2 or the H.263 HRD can be applied.

In [15], Stockhammer *et al.* compared the single receiver buffer approach with the aforementioned separate buffer approach and showed that a single receiver buffer always performs at least as good as two separate buffers. They then 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 receiver curve, and only a simple Bernoulli channel is considered.

A Markov chain analysis method was introduced in [16] to examine the tradeoff between buffer underflow probability and latency for Adaptive Media Playout (AMP) video streaming. Applying the two-state Gilbert-Elliott lossy channel model [17, 18], this method represents the streaming system with a Markov chain whose states are described as a combination of the buffer state, the channel state and the playout position.

Representing underflow event as an absorption state, this method gives the underflow probability by solving the Markov chain. However, in order to construct a Markov chain, the transmission time of each frame are assumed to be independent and identically distributed (i.i.d.) random variables, which is usually not true for VBR encoded video streaming where the frames have different sizes.

In [19], D. Towsley *et al.* present the first analytical study of multimedia streaming via TCP. They extended the discrete-time Markov models for VBR TCP traffic in [20] and [21] to incorporate the specific characteristics of constrained (live) and unconstrained (stored) streaming. This TCP traffic model is applied to construct a discrete-time Markov model for streaming system. Using these models, they explored the parameter space (i.e., loss rate, round trip time and timeout value in TCP as well as video playback rate) to provide guidelines as to when direct TCP streaming leads to satisfactory performance. Their results show that direct TCP streaming generally provides good performance when the available network bandwidth, and thus the achievable TCP throughput, is roughly twice the the video bitrate, with only a few seconds of startup delay. However, this work considers only CBR encoded videos and an infinite receiver buffer.

Following the same path of [19], L. Xu and J. Helzer studied the problem of media streaming via TCP-Friendly Rate Control (TFRC) was studied in [22]. Modeling the TFRC traffic by a Markov-Renewal-Modulated Deterministic Process, the authors developed a queueing model for the TFRC client buffer, based on the embedded Markov process of the buffer state immediately after a jitter. This model is then applied to obtain the distribution of the total duration of all rebuffering events experienced by a user. However, this work, as well as [19], considers only CBR encoded videos and an infinite receiver buffer.

Other previous studies on jitter reduction include works on rate-distortion optimized video streaming [23], [24], [25], [26]. However, they require significant modifications in the streaming server, the streaming client, or both.

In this thesis, we quantify the probability of jitter-free streaming and the distribution of the number of jitters during streaming a VBR video over random VBR channels. The proposed analytical techniques can be applied to most general systems. To the best of our knowledge, this work represents the first attempt to analytically quantify the tradeoff between initial playback delay, receiver buffer size, jitter recovering operation, and the frequency of jitters for streaming with arbitrary encoding scheme and random channels.

Chapter 3

Bounds of Probability of Jitter

In this chapter, we provide an analytical framework to bound the jitter-free probability for generic VBR video streaming over random VBR channels. The application of this analysis to a case study on wireless network streaming is also presented.

3.1 System Model

We consider the same video streaming system as in [15]. It consists of a video streaming server, a VBR transport channel, and a streaming client. Pre-encoded video objects are stored in the server. Each video object is characterize by a *playback curve* $p(t)$. The playback curve describes the total amount of data that have to be received by time t. It is generally assumed that $p(t) = 0$ for $t \leq 0$ and $p(t) = p(L)$ for $t \geq L$, where L is the length of the video in time and $p(L)$ is the size of the video in bits. The playback curve is assumed to be included in the preamble of the video stream and is available to the receiver [15].

When a client requests a video object, after setting up a connection, the server starts

to stream video data to the client through the transport channel. The channel is assumed to be error free, possibly due to an ideal error control mechanism, such as coding or ARQ, but its bit rate may vary over time. This VBR channel is characterized by a random *receiver curve* $R(t)$, which specifies the total amount of data received error-free up to time t at the client side. We also assume $R(t) = 0$ for $t \leq 0$, which is reasonable since no data should be received before the transmission begins. Clearly, both $p(t)$ and $R(t)$ are monotonically increasing.

As we are interested in providing continuous video streaming service, our objective is to minimize the initial playback ("startup") delay $\triangle$ and the corresponding receiver buffer size B, while maintaining jitter-free playback, for a VBR channel. For jitter-free playback, both buffer underflow and overflow at the receiver buffer need to be eliminated. To avoid buffer underflow, the startup delay $\triangle$ has to be chosen such that for any time instant t, at least $p(t - \triangle)$ bits are available at the decoder, i.e.,

$$\{\triangle \in \Re : R(t) \geq p(t - \triangle). \quad \forall \quad t\}. \tag{3.1}$$

Furthermore, to avoid buffer overflow, the buffer size B has to be large enough such that at any time instant t, all received non-decoded data can be stored in the receiver buffer, i.e.,

$$\{B \in \Re : R(t) \leq p(t - \triangle) + B, \quad \forall \quad t\}. \tag{3.2}$$

The right-hand side of (3.1) defines a *delayed playback curve*, and the right-hand side of (3.2) defines a *buffer limit curve*.

If a *deterministic* receiver curve $R(t)$ and the playback curve $p(t)$ are known before transmission at the streaming server, the authors of [15] have proven that the minimum

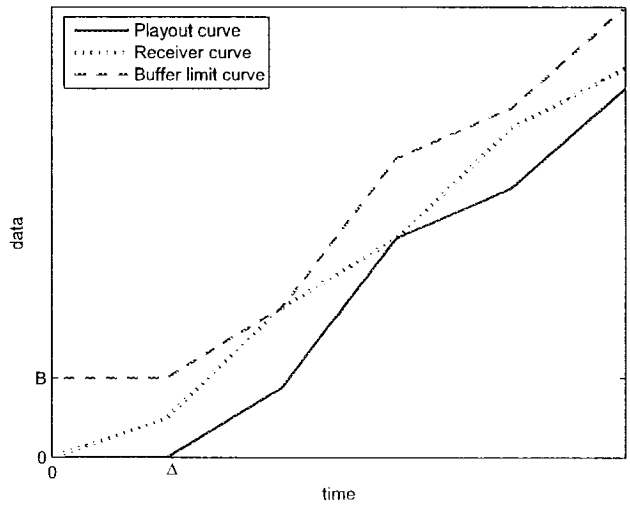


Figure 3.1: Playback, receiver, and buffer-limit curves with their associated startup delay $\triangle$ and buffer size B.

startup delay to avoid buffer underflow should be chosen as

$$\triangle = \max_t \left\{ t - p^{(-1)}(R(t)) \right\} , \tag{3.3}$$

where $p^{(-1)}(x) \triangleq \min\{t : x \leq p(t)\}$ is the so called the pseudoinverse function of the monotonically increasing playback curve $p(t)$. And the corresponding minimum receiver buffer size to avoid buffer overflow is then chosen as

$$B = \max_t \left\{ R(t) - p(t - \triangle) \right\} \tag{3.4}$$

The optimal selection of $\triangle$ and B is illustrated in Figure 3.1.

However, in general, $R(t)$ is random and not known in advance, either at the transmitter or at the receiver. In this work, we focus on the problem of optimal streaming

over VBR channels with a random $R(t)$, described in detail next in Section 3.2.

3.2 Random Receiver Curve

The assumption that $R(t)$ is known in advance at the transmitter or receiver is obviously not realistic for most practical systems. To the best of our knowledge, there is no known method to specify the minimum startup delay without exact knowledge of the receiver curve. However, in general, users may still be satisfied if the playback jitters do not occur too frequently. Therefore, it is reasonable to guarantee a certain performance criterion, by specifying the probability that the transmission of a video is successful without any buffer underflow or buffer overflow, given the constraint of startup delay and receiver buffer size.

To formalize the problem, let us consider a stationary-increment random process $R(t)$ that describes the random receiver curve introduced by the random channel. Obviously, each realization of the random process has the same monotonic properties as the deterministic receiver curve as we defined before. We want to determine the *jitter-free probability*, denoted by π, that a realization of $R(t)$ is entirely within the region between the delayed playback curve, $p(t-\triangle)$, and the buffer limit curve, $p(t-\triangle)+B$, during the playback period of a video, i.e.,

$$\pi \triangleq \Pr\left\{p(t-\triangle) \leq R(t) \leq p(t-\triangle)+B\right\}. \tag{3.5}$$

Generally, π is a difficult quantity to compute exactly, even if we know the stochastic properties of $R(t)$. Next, we will derive a lower bound and an upper bound of π, given some statistical information of $R(t)$.

3.2.1 Infinite Receiver Buffer Case

Let us first look at the case with infinite receiver buffer, i.e., $B = \infty$ or B is sufficiently large for any practical purposes that the buffer never overflows. Assume we know that the bit rate of the VBR channel is upper bounded by b_{max}, i.e., $R'(t) \leq b_{max}$. In the best case, the VBR channel stays at the maximum bit rate b_{max} throughout the transmission. Without loss of generality, we fix the origin at the time instant when the client starts playing out the video. Then the receiver curve in the best case is

$$R_{min}(t) = b_{max}(t + \triangle_{min}) \cdot u(t + \triangle_{min}) \tag{3.6}$$

where $u(t)$ is the unit step function

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases} \tag{3.7}$$

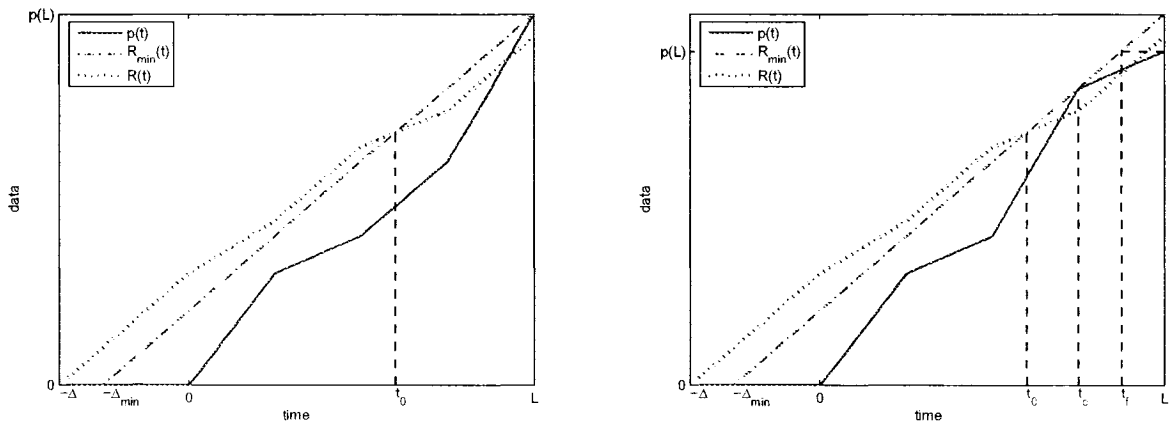
and $\triangle_{min}$ is the minimal achievable startup delay given by (3.1) with $R(t) = b_{max}t \cdot u(t)$, i.e.,

$$\triangle_{min} = \max_t \left\{ t - p^{-1}(b_{max}t \cdot u(t)) \right\}. \tag{3.8}$$

Note that $\triangle_{min}$ can only be achieved in the best case. It is in fact the minimum *supportable* startup delay. By supportable, we mean

$$\Pr\{R(t) \geq p(t) \ \ \forall \ \ t | R(t) = 0 \ \ \forall \ \ t \leq -\triangle\} > 0. \tag{3.9}$$

Therefore, the startup delay should be chosen such that $\triangle \geq \triangle_{min}$. The derivation of π then proceeds in two cases as follows.



(a) Case I: $R_{min}(L) = p(L)$ (b) Case II: $R_{min}(L) > p(L)$, $t_0 < t_c$

Figure 3.2: Examples of infinity receiver buffer.

Case I. $R_{min}(L) = p(L)$

Let $t_0 = \inf\{t : R(t) < R_{min}(t)\}$ be the time instant when $R(t)$ intersects with $R_{min}(t)$. It is easy to see that $R(t) < R_{min}(t)$ for all $t \geq t_0$. Therefore, if $t_0 < L$, we will have

$$R(L) < R_{min}(L) = p(L) \tag{3.10}$$

at the end of the video, as shown in Figure 3.2(a). According to the monotonically increasing property of $p(t)$, jitter is inevitable for this $R(t)$. Then, it is easy to show that, given a supportable startup delay $\triangle$, the probability of continuous playback is

$$\begin{aligned} \pi &= \Pr\{R(t) \geq p(t), \ \forall \ t \leq L\} \\ &= \Pr\{t_0 \geq L\} \\ &= \Pr\{R(t) \geq R_{min}(t), \ \forall \ t \leq L\} \\ &= \Pr\{R(L) \geq R_{min}(L)\}. \end{aligned} \tag{3.11}$$

Case II. $R_{min}(L) > p(L)$

We define the *latest most critical* point t_c of underflowing as

$$t_c = \max\{t : R_{min}(t) = p(t)\}, \tag{3.12}$$

and the *transmission finishing* point t_f in the ideal case with minimal startup delay $\triangle_{min}$ as

$$t_f = \frac{p(L)}{b_{max}} - \triangle_{min}. \tag{3.13}$$

Clearly, if $t_0 \geq t_f$, jitter does not occur; if $t_c \leq t_0 < t_f$, jitter may not occur with positive probability; and if $t_0 < t_c$, jitter occurs. An example when $t_0 < t_c$ is shown in Figure 3.2(b). Hence we can derive a lower bound on π as follows:

$$\begin{aligned}
\pi &= \Pr\{t_0 \geq t_f\} + \Pr\{t_c \leq t_0 < t_f\} \cdot \Pr\{R(t) \geq p(t), \ \forall \ t \leq L \mid t_c \leq t_0 < t_f\} \\
&\geq \Pr\{t_0 \geq t_f\} \\
&= \Pr\{R(t) \geq R_{min}(t), \ \forall \ t \leq t_f\} \\
&= \Pr\{R(t_f) \geq p(L)\} \stackrel{\Delta}{=} \pi_l.
\end{aligned} \tag{3.14}$$

Moreover, noticing that $\frac{\mathrm{d}}{\mathrm{d}t}p(t) < b_{max}$ for some $t \in [t_c, L]$, it is possible that even if a jitter incident occurs, $R(t)$ may increase faster than $p(t)$ and, at the end of the video, achieve $R(L) \geq p(L)$. Hence π is upper bounded by the probability of receiving no less than $p(L)$ bits by the end of the video, i.e.,

$$\pi \leq \pi_u \stackrel{\Delta}{=} \Pr\{R(L) \geq p(L)\}. \tag{3.15}$$

In summary, when a supportable startup delay $\triangle \geq \triangle_{min}$ is chosen, the jitter free

probability of a given playback curve $p(t)$ is bounded by π_l and π_u according to (3.14) and (3.15), respectively. Both bounds are achieved iff $R_{min}(L) = p(L)$.

3.2.2 Finite Receiver Buffer Case

When the receiver buffer is not large enough, overflow may occur. We will derive bounds on the probability that neither underflow nor overflow occurs.

Let t_B denote the *first most critical* point of overflowing:

$$t_B = \min\{\arg\max_t\{R_{min}(t) - p(t)\}\}. \tag{3.16}$$

Obviously, if the buffer size is a design parameter, then B should be chosen such that

$$B \geq R_{min}(t_B) - p(t_B) \triangleq B_{min}. \tag{3.17}$$

Otherwise jitter (either overflow or underflow) is inevitable. Let

$$R_B(t) = b_{max}(t + \triangle_B), \tag{3.18}$$

where

$$\triangle_B = \min_t \left\{ \frac{p(t) + B}{b_{max}} - t \right\}, \tag{3.19}$$

is the *highest* supporting line of the buffer limit curve $p(t) + B$ with slope b_{max}. An example of this finite buffer case is shown in Figure 3.3.

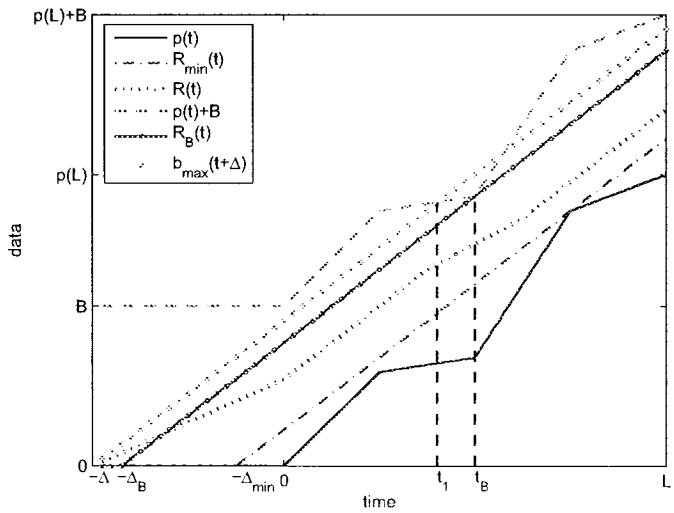


Figure 3.3: An example of finite receiver buffer.

We first consider a lower bound of π. It can be shown that

$$\begin{aligned}
\pi &= \Pr\{p(t) \le R(t) \le p(t) + B, \ \forall \ t \le L\} \\
&= \Pr\{R(t) \ge p(t), \ \forall \ t \le L\} \\
&\quad - \Pr\{R(t) > p(t) + B, \ \exists \ t \le L \wedge R(t) \ge R_{min}(t), \ \forall \ t \le L\} \\
&\ge \Pr\{R(t) \ge p(t), \ \forall \ t \le L\} - \Pr\{R(t) > p(t) + B, \ \exists \ t \le L\} \\
&\ge \Pr\{R(t_f) \ge p(L)\} - \Pr\{R(t) > p(t) + B, \ \exists \ t \le L\}. \qquad (3.20)
\end{aligned}$$

Then, let

$$t_1 = \min\{t : b_{max}(t + \triangle) = p(t) + B\} \qquad (3.21)$$

be the *first possible* point of overflowing when startup delay is $\triangle$. It is easy to see that

if $R(t_1) \leq R_B(t_1)$, overflow does not happen, i.e.

$$\Pr\{R(t_1) \leq R_B(t_1)\} \leq \Pr\{R(t) \leq p(t) + B, \ \forall \ t \leq L\}. \tag{3.22}$$

Hence, we have

$$\Pr\{R(t_1) > R_B(t_1)\} \geq \Pr\{R(t) > p(t) + B, \ \exists \ t \leq L\}. \tag{3.23}$$

Then from (3.20), we conclude that

$$\pi \geq \Pr\{R(t_f) \geq p(L)\} - \Pr\{R(t_1) > R_B(t_1)\}. \tag{3.24}$$

Following the same logic, if $R(0) \leq R_B(0)$, there will be no overflow, then we can also have

$$\pi \geq \Pr\{R(t_f) \geq p(L)\} - \Pr\{R(0) > R_B(0)\}. \tag{3.25}$$

So π can be lower bounded by

$$\pi \ \geq \ \Pr\{R(t_f) \geq p(L)\} - \min\{\Pr\{R(t_1) > R_B(t_1)\}, \Pr\{R(0) > R_B(0)\}\}. \tag{3.26}$$

Notice that when $t_1 > 0$, $R(t_1) > R_B(t_1)$ implies $R(0) > R_B(0)$, i.e.,

$$\Pr\{R(t_1) > R_B(t_1)\} \leq \Pr\{R(0) > R_B(0)\}. \tag{3.27}$$

The inverse is also true when $t_1 < 0$. Then (3.26) can be rewritten as

$$\pi \geq \Pr\{R(t_f) \geq p(L)\} - \Pr\{R(t_1^+) > R_B(t_1^+)\} \triangleq \pi_l. \tag{3.28}$$

Here $(\cdot)^+$ denotes $\max\{0, \cdot\}$, and we have used π_l to denote the lower bound for the jitter-free probability π.

To derive the upper bound, we note that if $R(t_B) > R_B(t_B)$, overflow occurs. Hence we have

$$\Pr\{R(t) > p(t) + B, \ \exists \ t \leq L\} \geq \Pr\{R(t_B) > R_B(t_B)\}. \tag{3.29}$$

Then π is upper bounded by:

$$\begin{aligned}
\pi &= \Pr\{p(t) \leq R(t) \leq p(t) + B, \ \forall \ t \leq L\} \\
&= \Pr\{R(t) \geq p(t), \ \forall \ t \leq L\} \\
&\quad - \Pr\{R(t) \geq p(t), \ \forall \ t \leq L \wedge R(t) > p(t) + B, \ \exists \ t \leq L\} \\
&= \Pr\{R(t) \geq p(t), \ \forall \ t \leq L\} \\
&\quad \cdot (1 - \Pr\{R(t) > p(t) + B, \ \exists \ t \leq L | R(t) \geq p(t), \ \forall \ t \leq L\}) \\
&\leq \Pr\{R(t) \geq p(t), \ \forall \ t \leq L\} \cdot (1 - \Pr\{R(t) > p(t) + B, \ \exists \ t \leq L\}) \\
&\leq \Pr\{R(L) \geq p(L)\} \cdot (1 - \Pr\{R(t_B) > R_B(t_B)\}) \\
&\triangleq \pi_u
\end{aligned} \tag{3.30}$$

The first inequality in (3.30) is due to the fact that the conditional probability of overflow given no underflow occurs is greater than the marginal probability of overflow, and the second inequality is due to a relaxation of t from the range $(0, L]$ to L. Here we use π_u to denote the upper bound for the jitter-free probability π.

3.3 Case Study: Wireless Video Streaming

In a wireless access network, because of the hostile radio channel condition resulting from multipath propagation, fading and scattering, packet delay and losses are common. In this section, we present a case study on applying the theoretical framework in Section 3.2 to evaluate the jitter-free probability of streaming in a generic wireless system with an ARQ transmission protocol.

3.3.1 Modeling Wireless Channels

To properly model the error events in a wireless communication channel, we adopt in this work an extended Gilbert model proposed by Sanneck *et al.* in [27]. It is a generalization of the Gilbert-Elliott model [17, 18], one of the most widely adopted wireless channel representations in the literature. The Gilbert-Elliott model is a two-state Markov model where the channel switches between a "good state" (reception) and a "bad state" (loss). The channel state changes at the beginning of each time slot according to the given transition probabilities. A transmission is successful only if the channel is in the "good state"; otherwise, it fails. The Gilbert-Elliott model captures the temporal dependence in packet-loss processes for communication networks. However, many recent studies have shown that the Gilbert-Elliott model fails to predict performance measures depending on longer-term correlation of errors and hence can lead to lower channel capacity estimates and conservative allocation strategies [28], [29], [30], [31].

There are two categories of extended Gilbert models: those which describe reception run-lengths (RRL) and those which describe loss run-lengths (LRL). In what follows we concentrate on LRL models. As illustrated in Figure 3.4, for a k-th order LRL extended Gilbert model, there are $k+1$ states, $\{S_1, S_2, ..., S_{k+1}\}$. For each state S_i, $i-1$ represents

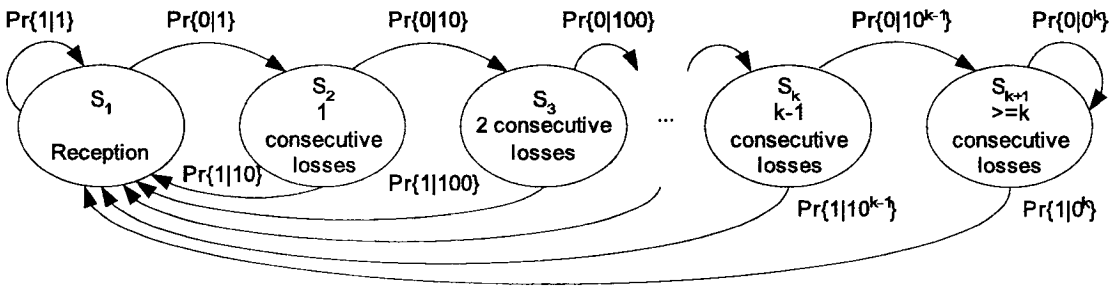


Figure 3.4: A k-th order loss run-length (LRL) extended Gilbert model.

the current loss run-length, except for state S_{k+1}, which represents the case that the current loss run-length is at least k, in which case the system remains in state S_{k+1} with a subsequent loss or returns to S_1 with the first occurrence of reception. In each state, a reception event will bring the system back to state S_1, the "good" state, which means the occurrence of a successful transmission will free the process from dependence upon past history and starts it anew. Hence, letting 0^i represent the occurrence of i successive losses, the LRL extended Gilbert model can be characterized in terms of the following $k+1$ independent parameters:

- $\Pr\{1|10^i\}$: the probability that the next transmission succeeds, given that the current $LRL = i$, $0 \leq i \leq k-1$;
- $\Pr\{1|0^k\}$: the probability that the next transmission succeeds, given that the current $LRL \geq k$;
- $\Pr\{0|10^i\} = 1 - \Pr\{1|10^i\}$: the probability that the next packet is lost, given that the current $LRL = i$, $0 \leq i \leq k-1$; and
- $\Pr\{0|0^k\} = 1 - \Pr\{1|0^k\}$: the probability that the next packet is lost, given that the current $LRL \geq k$.

In addition, we model transmission control with a simple ARQ transmission protocol.[1] We consider a lossless system in which packets may be corrupted but acknowledgements are error free. After sending out a packet, the server waits until an ACK or an NACK arrives from the client. If an ACK is received, it means the packet is correctly received and the server starts transmitting the next packet; if an NACK is received, or if a timeout interval is exceeded, the packet is considered corrupted and the server sends out the same packet again until an ACK is finally received. For simplified numerical analysis, we consider the system to be time-discrete and data are packed into packets of a fixed size. Each time slot is the time between when the server sends out a packet and when the corresponding acknowledgement arrives.

3.3.2 Delay and Buffer Design

We next derive the probability bounds π_l and π_u for the extended Gilbert model. Instead of looking into the probability $\Pr\{R[L] \geq N\}$ directly, let us first consider the random variable X_i, which represents the total number of transmissions that it takes to successfully transmit the i^{th} packet. Notice that except the first packet, every packet starts its transmission following a successful transmission of the previous packet. Hence, if a connection between the server and a client is setup with a successful transmission, then the channel is in the good state at $t = 0$ and $X = \{X_i\}$ is a sequence of i.i.d. random variables. Note that this only simplifies the presentation and does not significantly affect the results.

We denote $P_i = \Pr\{0|10^i\}$, $i = 0, 1, ..., k-1$ and $P_k = \Pr\{0|0^k\}$. According to the

[1]TCP is well known to misbehave in the wireless environment. In this work, we use simple ARQ to illustrate the behavior of a general error recovery protocol.

extended Gilbert model above, we calculate the probability mass function of X

$$P_X[x] = \begin{cases} 1 - P_0, & x = 1 \\ (1 - P_{x-1}) \prod_{i=0}^{x-2} P_i, & x = 2, ..., k \\ (1 - P_k) P_k^{x-(k+1)} \prod_{i=0}^{k-1} P_i, & x > k \end{cases} \tag{3.31}$$

and the moment generating function

$$\begin{aligned} \Phi_X(z) &\triangleq \sum_{k=1}^{\infty} P_X[k] z^{-k} \\ &= (1 - P_0) z^{-1} + \sum_{x=2}^{k} \left[(1 - P_{x-1}) z^{-x} \prod_{i=0}^{x-2} P_i \right] \\ &\quad + (1 - P_k) \prod_{i=0}^{k-1} P_i \sum_{x=k+1}^{\infty} P_k^{x-(k+1)} z^{-x} \\ &= (1 - P_0) z^{-1} + \sum_{x=2}^{k} \left[(1 - P_{x-1}) z^{-x} \prod_{i=0}^{x-2} P_i \right] \\ &\quad + (1 - P_k) \prod_{i=0}^{k-1} P_i \frac{z^{-(k+1)}}{1 - P_k z^{-1}}, \quad |z| > P_k \end{aligned} \tag{3.32}$$

Let $Y_N = \sum_{i=1}^{N} X_i$ be the total number of transmissions until the N^{th} packet is successfully transmitted. Then the moment generating function of Y_N is

$$\Phi_Y(z) = \Phi_X^N(z). \tag{3.33}$$

And the distribution function of Y_N is

$$\begin{aligned} F_{Y_N}[y] &= \Pr\{Y_N \leq y\} \\ &= \sum_k P_{Y_N}[k]u[y-k] \\ &= P_{Y_N}[y] * u[y]. \end{aligned} \tag{3.34}$$

Here $u[n] = 1(n \geq 0)$ is the discrete unit step function and its z-transform is $(1 - z^{-1})^{-1}$, $|z| > 1$. According to the convolution property of z-transform, we have

$$\begin{aligned} \mathcal{Z}\{F_{Y_N}[y]\} &= \mathcal{Z}\{P_{Y_N}[y] * u[y]\} \\ &= \Phi_{Y_N}(z)\frac{1}{1-z^{-1}} \\ &= \frac{\Phi_X^N(z)}{1-z^{-1}}, \quad |z| > 1 . \end{aligned} \tag{3.35}$$

Obviously, the probability that the client has received at least N packets by time L equals to the probability that the transmission of the N^{th} packet is completed no later than $L + \triangle$, i.e., when $L + \triangle \geq N$, we have

$$\begin{aligned} \Pr\{R(L) \geq N\} &= \Pr\{Y_N \leq L + \triangle\} \\ &= F_{Y_N}[L + \triangle] \\ &= \mathcal{Z}^{-1}\left\{\frac{\Phi_X^N(z)}{1-z^{-1}}, \ |z| > 1\right\} \\ &= \frac{1}{2\pi j}\oint_C \frac{\Phi_X^N(z)}{1-z^{-1}} z^{L+\triangle-1} dz \\ &= \sum_{\xi \in \sigma} \operatorname*{Res}_{z=\xi}\left[\frac{\Phi_X^N(z)}{1-z^{-1}} z^{L+\triangle-1}\right], \end{aligned} \tag{3.36}$$

where C is a counterclockwise closed path encircling the unit circle, σ denotes the set of

poles of $\Phi_X^N(z)z^{L+\triangle-1}/(1-z^{-1})$ within C, and $\text{Res}_{z=\xi}$ denotes the residue at pole $z=\xi$. Then the lower bound π_l and upper bound π_u can be obtained by plugging (3.36) into (3.14), (3.15), (3.28) and (3.30). These analytical results are validated by experimenting with VBR video traces shown in the next section.

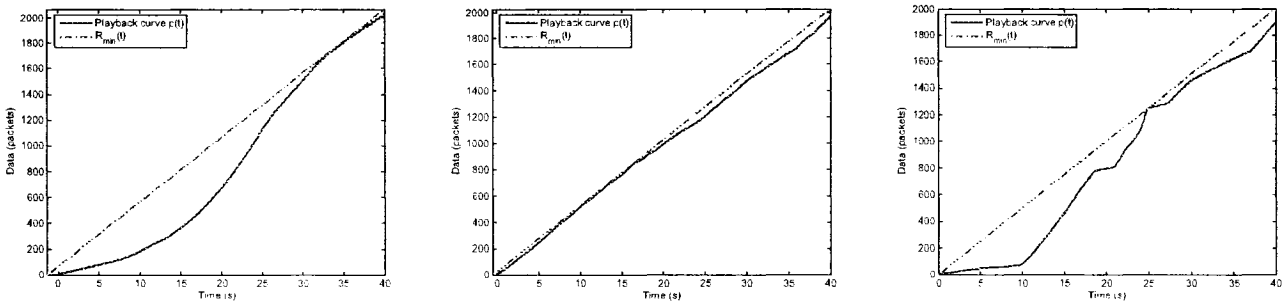
3.4 Experimental and Numerical Results

In this section, we present experimental results to validate the proposed analytical framework. We investigate the effects of the initial playback delay and the receiver buffer size on jitter-free streaming and discuss the optimal balance between playback delay and buffer size.

3.4.1 Experimental Setup

We experiment on wireless streaming using MPEG-4 VBR video traces provided by [32]. Figure 3.5 shows the playback curves $R(t)$ for the first 40 seconds of three video sequences and their corresponding $R_{min}(t)$ values. Some main statistics of these video clips, pertinent to the computation of jitter-free probability bounds, are listed in Table 3.1. These sequences were encoded at a constant frame rate of 25 frames/s in the Quarter Common Intermediate Format (QCIF) resolution (176×144). We have chosen the QCIF format because we are particularly interested in wireless networking systems where handheld wireless devices typically have a screen size that corresponds to the QCIF video format.

For numerical analysis, we adopt a three-state extended Gilbert model (e.g., $k=2$), with state transition probabilities $\Pr\{0|1\}=0.005$, $\Pr\{0|10\}=0.9$ and $\Pr\{0|00\}=0.95$.



(a) Alpin Ski (b) Formula 1 (c) Jurassic

Figure 3.5: VBR video playback curves and the corresponding $R_{min}(t)$'s.

Video name	Parameters of the video stream				
	Mean rate (bits/s)	Peak rate (bits/s)	$\triangle_{min}$ (s)	B_{min} (bits)	t_B (s)
Alpin Ski	1.8247e+05	3.0248e+06	1.26	1.638e+06	15.36
Formula 1	1.7754e+05	2.1428e+06	0.5	3.826e+05	36
Jurassic	1.7100e+05	3.1668e+06	0	1.530e+06	10.08

Table 3.1: Parameters of different video streams

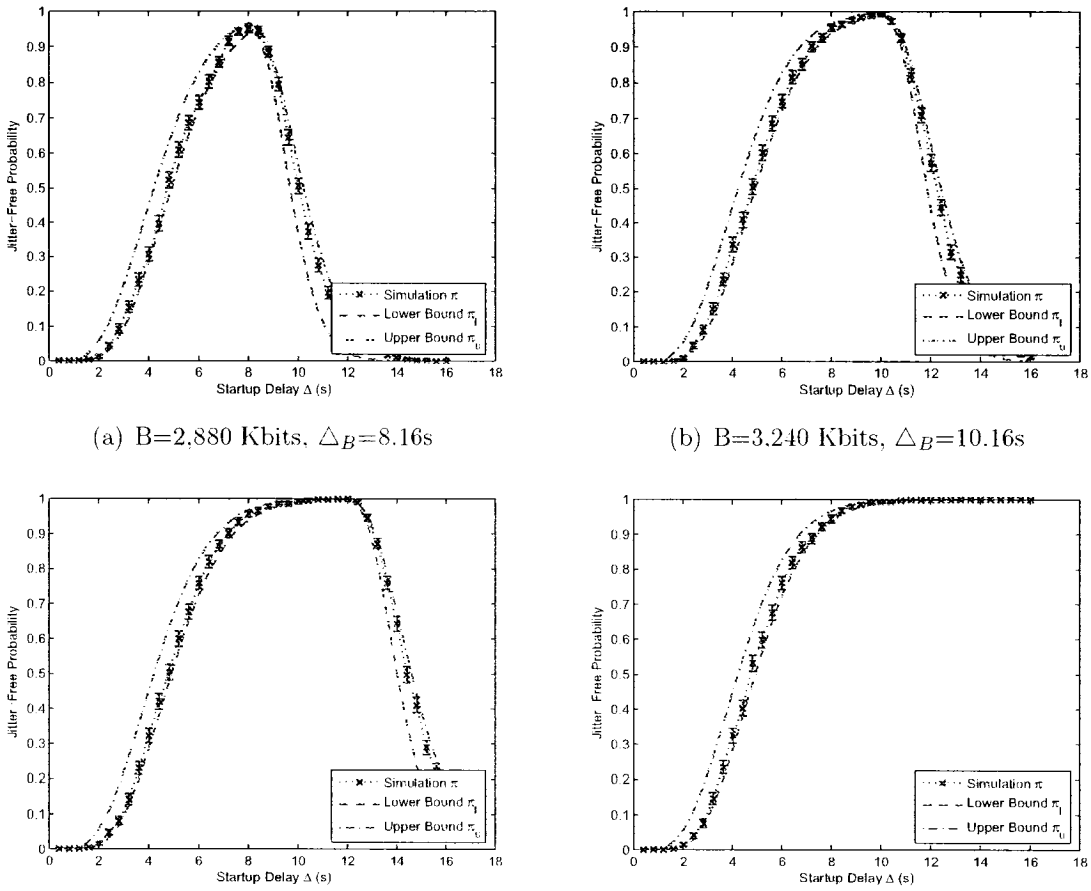
We set the packet size to be 3600 bits and the time interval between two consecutive transmissions as 20 ms. This results in a maximum transmission bit rate of 180,000 bit/s. Clearly the numerical results scale correspondingly to networks with different transmission rates. Corresponding to the settings in numerical analysis, we perform simulation in Matlab. For different startup delay values and buffer sizes, we simulate the transmission and playback of each video sequence over 2000 realizations of the random VBR channel and measure the jitter free probability. Furthermore, we compute the upper and lower bounds of the jitter free probability based on the proposed analysis and compare them with the simulation results.

3.4.2 Effect of Initial Playback Delay

In Figure 3.6 we plot the jitter-free probability vs. startup delay value for video "Alpin Ski" with different fixed buffer sizes, including a case with infinite buffer size. Also indicated are the corresponding values of $\triangle_B$, the highest supporting line of the buffer limit curve $p(t) + B$ as defined in (3.19). We observe that the derived upper bounds and lower bounds provide close estimation of the actual jitter free probability, especially within desired parameter ranges, where the startup delay is small to moderate and the jitter-free probability is high. Similar results are observed for different playback curves and omitted to reduce redundancy.

We note that, when $\triangle \leq \triangle_B$, the finite buffer case and the infinite buffer case are nearly the same, and the jitter free probability increases as $\triangle$ increases. This is because when $\triangle \leq \triangle_B$, as we can infer from Figure 3.3, $R(t)$ will always stay below $R_B(t)$, so there will not be any overflow, and jitter is only induced by underflow. A larger $\triangle$ reduces the probability of underflow. However, after $\triangle$ surpasses $\triangle_B$, the jitter free probability first increases, but by only a very small amount, and then decreases as $\triangle$ increases. This is because when $\triangle > \triangle_B$, overflow occurs. And since the buffer size is fixed, a longer startup delay will result in accumulating too much data and hence increasing the probability of overflow. When $\triangle$ becomes larger, overflow starts to dominate and the jitter free probability goes to zero.

The practical implication of the above observation is clear. The initial playback delay is a delicate parameter in the optimal design of multimedia streaming. Increasing the initial playback delay can drastically improve the jitter-free probability, but only up to a point determined by the playback curve and the receiver buffer size. Beyond that point, increasing the initial playback delay can in fact significantly increase playback jittering.



(a) B=2,880 Kbits, $\triangle_B$=8.16s

(b) B=3,240 Kbits, $\triangle_B$=10.16s

(c) B=3,600 Kbits, $\triangle_B$=12.16s

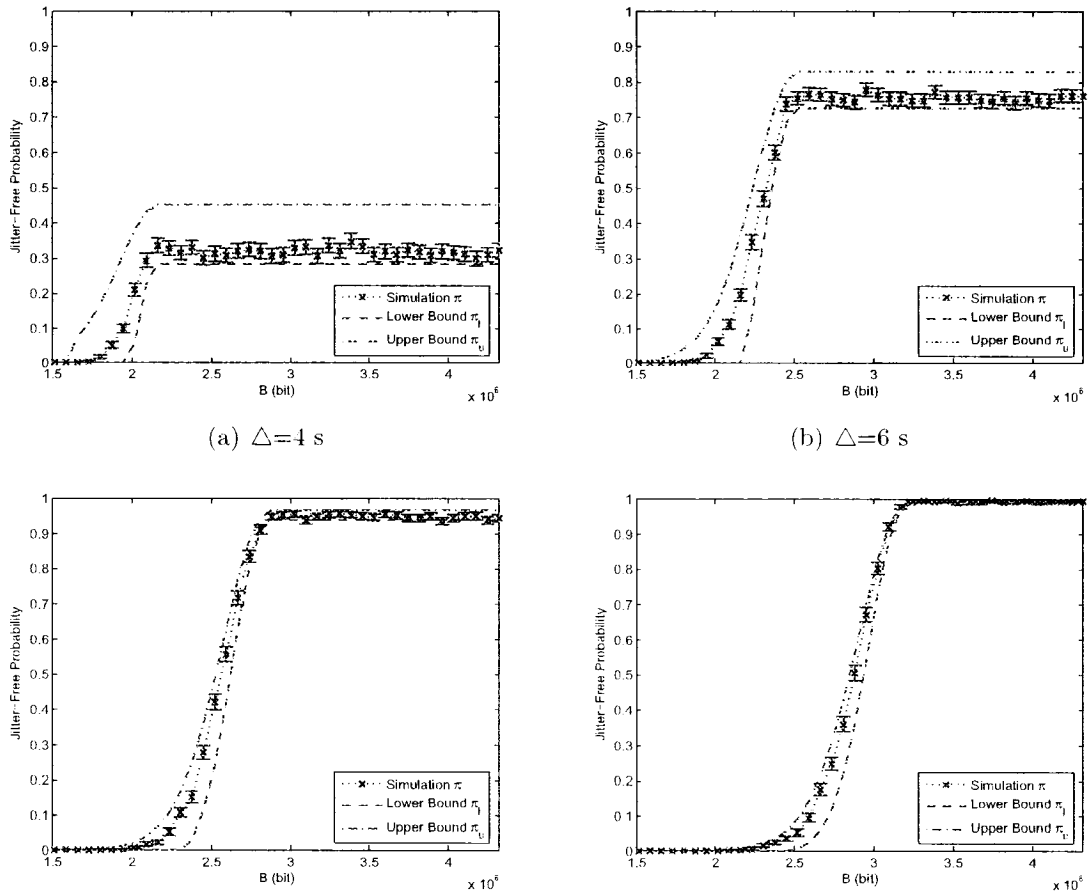
(d) B=∞

Figure 3.6: Jitter free probability vs. startup delay with different receiver buffer sizes for video "Alpin Ski". With 95% confidence intervals for 2000 runs.

In this regard, the proposed analysis framework provides a means to accurately estimate the optimal value of the initial playback delay.

3.4.3 Effect of Receiver Buffer Size

Figure 3.7 shows the effect of receiver buffer size on the jitter-free probability for the video "Alpin Ski" given different startup delay values. It can be seen in this figure that



(a) △=4 s (b) △=6 s

(c) △=8 s (d) △=10 s

Figure 3.7: Jitter free probability vs. buffer size with different startup delays for video "Alpin Ski". With 95% confidence intervals for 2000 runs

the jitter-free probability π increases quite fast as the buffer size B increases at first and then after B surpasses a certain value, π is approximately constant. Thus, increasing the receiver buffer will not necessarily result in drastically improved performance.

Furthermore, we note that for a given initial delay $\triangle$, there exists a maximum buffer size

$$B_{max}(\triangle) = \max_t \{\min\{b_{max}(t+\triangle), p(L)\} - p(t)\} \tag{3.37}$$

such that for any $B \geq B_{max}(\triangle)$, π is constant:

$$\pi = \Pr\{R(t) \geq p(t) \ \ \forall \ \ t \leq L\}. \tag{3.38}$$

The idea is that since $R(t) \leq b_{max}(t + \triangle)$ for all t, if $R_B(t) \geq b_{max}(t + \triangle)$, overflow will not occur and only underflow contributes to jitter, which is equivalent to the infinite buffer case.

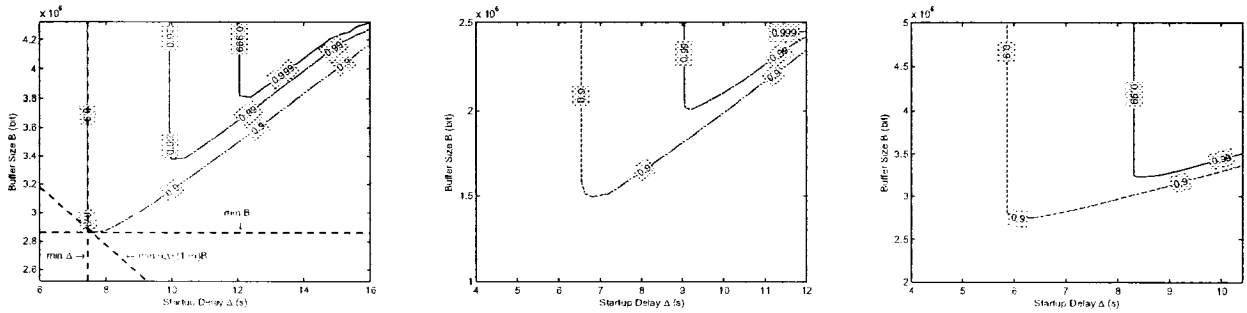
Clearly, in terms of resource allocation, hardware implementation, and energy consumption cost, a smaller buffer size is preferred in practice. Therefore, the proposed analytical framework, allowing numerical results such as those presented in Figure 3.7, can provide a means for a first step toward quantifying the most sensible buffer sizes for optimal system design, taking into consideration different system cost factors.

3.4.4 Optimal Delay-Buffer Tradeoff

Finally, we consider the optimal tradeoff between the initial playback delay $\triangle$ and the receiver buffer size B in jitter-free streaming. Figure 3.8 shows the contour maps of the startup delay and the buffer size that achieve different jitter free probabilities, for the three video clips described above.

We observe that the right branch of a contour curves have a slope roughly equal to b_{max}, the maximum transmission rate of the channel. This corresponds to the fact that as $\triangle$ increases, regardless the boundary effect, $B_{max}(\triangle)$ increase at rate b_{max}. This confirms our previous observation that, in order to achieve $\pi \rightarrow 1$ for a given delay $\triangle$, we should have $B \rightarrow B_{max}(\triangle)$.

Finally, this figure provides a convenient means to obtain an optimal operating point for the system that balances the tradeoff between $\triangle$ and B, given a certain required



(a) Alpin Ski (b) Formula 1 (c) Jurassic

Figure 3.8: Contour maps of buffer size vs. startup delay, with jitter-free probabilities labelled on the curves.

jitter-free probability π. If we define a cost function as a weighted sum of the two

$$C = \alpha\triangle + (1-\alpha)B. \tag{3.39}$$

where $\alpha \in [0,1]$, then to minimize this cost function, we simply find the tangent line of the corresponding contour curve with slope $\frac{\alpha}{\alpha-1}$. The point of contact between the tangent line and the contour curve given a value of π defines the optimal operating point for the system. An illustration of such a procedure is shown in Figure 3.8.

Chapter 4

Frequency of Jitters

In the previous chapter, we study the probability of having no jitter during a streaming session. However, in practical video streaming systems, when a jitter occurs, display of the video pauses, and after a certain amount of data, determined by the chosen jitter recover buffering scheme, is buffered and the display of the video is resumed. Depending on the channel condition and the chosen jitter recover scheme, a client may experience a number of jitters with some jitter recover buffering delay every time jitter occurs. In this chapter, we will study the characteristics of the number of jitters and the jitter buffering delay.

4.1 Network Model

In this chapter, we consider the same video streaming system as in Chapter 3. Here, we consider a generic class of Markov channel models characterized by $(\mathfrak{S}, \mathbf{A}, \mathfrak{R})$, where $\mathfrak{S} = \{S_1, ..., S_K\}$ is the set of channel states, $\mathbf{A}$ is the transition probability matrix, and $\mathfrak{R} = \{r_1, ..., r_K\}$ is the set of transmission rates associated the states. More precisely, r_i

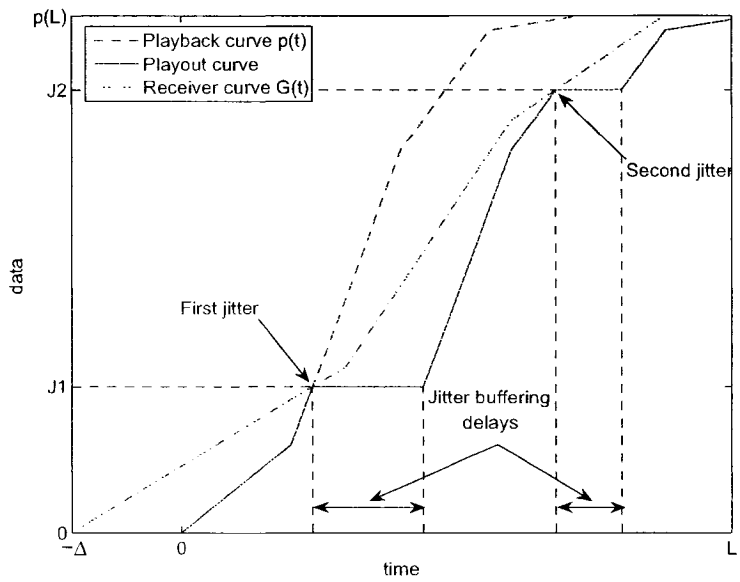


Figure 4.1: An example of jitters.

represents the number of packets that can be transmitted in one time slot. Obviously, the extended Gilbert model belongs to this class of Markov channel models. A k-th order LRL extended Gilbert model can be represented in this form by simply setting $K = k + 1$, $r_1 = 1$ and $r_i = 0$, $i = 2, 3, ..., k + 1$. Studies have also shown that TCP [20] and TFRC [22] can be well modeled by a Markov chain, respectively. Moreover, most Markovian channel models which have been studied in literature also belong to this class, which implies that the analysis technique we present in this work can easily be adapted to accommodate different situations.

A streaming example with unlimited receiver buffer is illustrated in Figure 4.1. Without loss of generality, we assume the video starts playing out at time 0 and the transmission begins at time $-\triangle$, where $\triangle$ represents the *initial delay*. In the figure, $G(t)$ represents the amount of data received at the client by time t. When $G(t) < p(t)$, jitter occurs. Then the application stops playing out the video and data accumulates in the

receiver buffer, until the jitter recover buffering scheme decides to resume displaying.

We study three jitter recover buffering schemes in this chapter:

- Fixed Jitter Buffering Delay (FBD): after a jitter, buffer for a fixed amount of time D_{jit}, and then resume display;
- Fixed Buffered Playout Data (FPD): after a jitter, buffer for a fixed number of packets B_{jit}, and then resume display; and
- Fixed Buffered Playout Time (FPT): after a jitter, buffer the number of packets that corresponds to a fixed playout duration T_{jit}.

Depending on the channel condition and the jitter-recovery buffering scheme, a client may experience a number of jitters and some jitter buffering delay. Next, we will present our analytical framework and study the tradeoffs among different system parameters.

4.2 Analysis Framework of VBR Video Streaming

In this section, we propose a recursive analytical framework to study the effects of initial delay, receiver buffer size, and the parameters of the three aforementioned jitter recover buffering schemes. We are particularly interested in computing $E\{N\}$, the expected number of jitters in a given streaming session. The key notation introduced in this section is summarized in Table 4.1 for easy reference.

We define the position of a jitter by the packet index where it occurs. Thus, by "a jitter occurs at packet i", we mean that the ith packet is the first packet whose playout deadline d_i is violated after the start of display or the previous jitter. Denote J_n as the position of the nth jitter, and X_n as the channel state when the nth jitter occurs. Also,

Notation	Definition
$\triangle$	Initial delay
B	Receiver buffer size
t	Distance in time in the video with respect to the beginning
$p(t)$	The minimum amount of data that has to be received by t
d_i	Arrival deadline of the ith packet, $p^{-1}(i)$
J_n	Index of the first packet whose deadline is violated after $n-1$ jitters
X_n	State of the VBR channel when the nth jitter occurs
L	Length of a video
D_{jit}	Fixed jitter buffering delay in FBD
B_{jit}	Fixed jitter buffered playout data in FPD
T_{jit}	Fixed jitter buffered playout time in FPT
N	Total number of jitters during streaming of a video
$\overline{\pi}[i]$	The ith element of vector $\overline{\pi}$

Table 4.1: Table of nomenclature

let $P_k^{(n)}(i)$ be the probability that the nth jitter occurs at packet i in channel state S_k, i.e.

$$P_k^{(n)}(i) = \Pr\{J_n = i, X_n = S_k\}. \ k = 1, ..., K. \tag{4.1}$$

Now, let's first assume we have a means to compute the conditional probability that the $(n+1)$th jitter occurs at packet i in channel state S_k, given it is after the nth jitter that occurred at packet j in channel state S_l. We denote it as

$$Q_{l,k}(j,i) = \Pr\{J_{n+1} = i, X_{n+1} = S_k | J_n = j, X_n = S_l\}. \tag{4.2}$$

Note that $\sum_i \sum_k Q_{l,k}(j,i)$ does not necessarily equal to one, because there may be no more jitters after the one at packet j. Moreover, because of the Markovian behavior

of the channel and the schemes being considered only make their decision based on the current state, $Q_{l,k}(j,i)$ only depends on the positions and the channel states, but not n.

By direct application of the Total Probability Theorem [33], the probability that the $(n+1)$th jitter occurs at packet i equals to the sum of probabilities that the nth jitter occurs at packets with $j \leq i$ and the $(n+1)$th jitter occurs at packet i. We have

$$\begin{aligned} P_k^{(n+1)}(i) &= \sum_{j=1}^{i} \sum_{l=1}^{K} \Pr\{J_n = j, X_n = S_l, J_{n+1} = i, X_{n+1} = S_k\} \\ &= \sum_{j=1}^{i} \sum_{l=1}^{K} Q_{l,k}(j,i) P_l^{(n)}(j). \end{aligned} \tag{4.3}$$

Then the probability that there are at least n jitters during the playout of a video is

$$\begin{aligned} \Pr\{N \geq n\} &= \sum_{i=1}^{p(L)} \Pr\{J_n = i\} \\ &= \sum_{i=1}^{p(L)} \sum_{k=1}^{K} P_k^{(n)}(i). \end{aligned} \tag{4.4}$$

Since $N \geq 0$, the expected number of jitters can be obtained by

$$\begin{aligned} E\{N\} &= \sum_{n=1}^{\infty} \Pr\{N \geq n\} \\ &= \sum_{n=1}^{\infty} \sum_{i=1}^{p(L)} \sum_{k=1}^{K} P_k^{(n)}(i). \end{aligned} \tag{4.5}$$

If $P_k^{(1)}(i)$ and $Q_{l,k}(j,i)$ are known, from equation 4.2, we can recursively compute $P_k^{(n)}(i)$, for any n, i and k, and then the expected number of jitters can be obtained from equation 4.2. Next, we present the derivation of these probabilities for different jitter

recover buffering schemes and for both the infinite buffer and finite buffer cases.

4.3 Infinite Buffer Case

We first consider the case with unlimited receiver buffer, or the buffer size is large enough such that it will not be fulled during streaming.

4.3.1 First Jitter Distribution $P_k^{(1)}(i)$

The first jitter distribution $P_k^{(1)}(i)$ is common to all jitter-recovery buffering schemes, and hence we provide its derivation first. Denote

$$F_{l,k,r} = \Pr\{S_l \rightarrow S_k\} e(r_k, r), \ r \leq R. \tag{4.6}$$

as the probability that, given the channel state in the previous step is S_l, the current channel state is S_k, and r packets are successfully transmitted. Here $e(r_k, r)$ equals to 1 if $r_k = r$ and 0 otherwise, and R is the maximum number of packets that can be transmitted in one time slot. We can express the state of the system with the tuple (g, s), where g specifies the number of received packets, and $s \in S$ specifies the channel state. Then we can construct a Markov chain of the system with the following transition

probability matrix

$$\mathbf{\Phi} = \begin{array}{c} 0 \\ 1 \\ \vdots \\ p(L)-1 \\ p(L) \\ \vdots \end{array} \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 & & \cdots & 0 \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 \\ 0 & & \cdots & 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R \\ \vdots & \vdots & & & & \ddots & & \end{bmatrix}, \tag{4.7}$$

where

$$\mathbf{A}_r = \begin{array}{c} \\ (g, S_1) \\ \vdots \\ (g, S_K) \end{array} \begin{array}{c} \begin{array}{ccc} (g+r, S_1) & \cdots & (g+r, S_K) \end{array} \\ \begin{bmatrix} F_{1,1,r} & \cdots & F_{1,K,r} \\ \vdots & \ddots & \vdots \\ F_{K,1,r} & \cdots & F_{K,K,r} \end{bmatrix} \end{array}. \tag{4.8}$$

With this transition matrix, we can obtain the distribution of the number of packets that have been received by time t simply from $\overline{\pi}_{init}\mathbf{\Phi}^{\triangle+t}$, where $\overline{\pi}_{init}$ is the initial state distribution at $-\triangle$. However, what we are interested in is the probability of the system arriving at a state without any jitter by time t. Moreover, since the video is VBR encoded, the consumption speed of data is predetermined but varies over time. Hence the standard homogeneous Markov chain approach [33] can not be applied in this non-homogeneous case.

Instead, we propose the following. At any time t, states with $g < p(t)$ should not be considered for the computation of the state distribution of the next time slot because these states have already violated the playout deadline by time t. Figure 4.2 illustrates

the idea: the contribution of arrows starting from the shaded states should be eliminated. This can be done by setting $\overline{\pi}_t[Kg+l]$, the $(Kg+l)$th element of $\overline{\pi}$ corresponding the state (g, S_l), to 0 for $g < p(t)$, $l = 1, ..., K$, before using it to compute the state distribution of the next time slot. It is equivalent to modifying $\mathbf{\Phi}$ into $\mathbf{\Phi U}_t$, with

$$\mathbf{U}_t = \begin{bmatrix} \mathbf{0}_{Kp(t)\times Kp(t)} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \tag{4.9}$$

in each time slot. Then the probability of arriving at a state without having jitter by $d_i - 1$ is given by

$$\overline{\pi}_{d_i-1} = \overline{\pi}_{init}(\prod_{t=-\triangle}^{d_i-1} \mathbf{\Phi U}_t). \tag{4.10}$$

And the value of the $(K(i-1)+k)$th $(k \leq K)$ element of

$$\overline{\pi}'_{d_i} = \overline{\pi}_{d_i-1}\mathbf{\Phi} = \overline{\pi}_{init}(\prod_{t=-\triangle}^{d_i-1} \mathbf{\Phi U}_t)\mathbf{\Phi} \tag{4.11}$$

is the state distribution by d_i, given that there has been no jitter by $d_i - 1$. And the $(K(i-1)+k)$th element of it represents the event that the client has only received $i-1$ packets in channel state S_k by d_i, which is $P_k^{(1)}(i)$, i.e.,

$$P_k^{(1)}(i) = \overline{\pi}'_{d_i}[K(i-1)+k]. \tag{4.12}$$

4.3.2 Next Jitter Probability $Q_{l,k}(j,i)$

Next, we will study the distributions of the position of the subsequent jitter for the aforementioned jitter recovery buffering schemes.

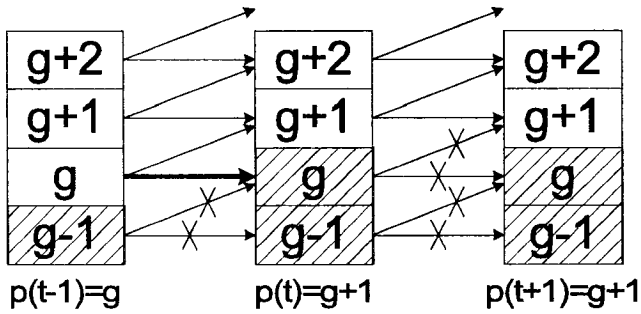


Figure 4.2: Arrows indicate the transition of states. The shaded blocks represent the states that have violated the playout deadline. The large arrow indicates the transition that introduces a jitter at packet g at time t.

Fixed Jitter Buffering Delay

In the FBD scheme, every time a jitter occurs, the application stops displaying and buffers data for a fixed jitter buffering delay D_{jit}. The procedure to find $Q_{l,k}(j,i)$ is similar to that of finding the first jitter probability in section 4.3.1. We only need to imagine we start playing out the video from packet j with a empty buffer. Then $(j-1, S_l)$ is the virtual "initial" state, and D_{jit} is the virtual "initial" delay. Denote the virtual initial state distribution after a jitter occurs at (j, S_l) as

$$\overline{\pi}_{j,l} = [0 \ \cdots \ 0 \quad \underset{\text{the } (K(j-1)+l) \text{ th element}}{1} \quad 0 \ \cdots \ 0]. \tag{4.13}$$

Then similar to equation (4.11), the state probability distribution at time d_i of having no jitter by $d_i - 1$ is given by

$$\overline{\pi}'_{d_i} = \overline{\pi}_{j,l}\Big(\prod_{t=d_j-D}^{d_i-1} \mathbf{\Phi U}_t\Big)\mathbf{\Phi}. \tag{4.14}$$

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

$$Q_{l,k}(j,i) = \overline{\pi}'_{d_i}[K(i-1)+k]. \tag{4.15}$$

Fixed Buffered Playout Data

In the FPD scheme, after a jitter happens, the application stops playing out the video and buffers data until the number of packets in the buffer reaches B_{jit}. If a jitter occurs at the jth packet, it will restart playing out the video once the $(j+B_{jit}-1)$th packet arrives. To find $Q_{l,k}(j,i)$, we first have to find the probability distribution of restarting playout at state (g,S_h), where $g \in [j+B_{jit}-1, j+B_{jit}+R-2]$ represents the last packet received when display resumes. To find these probabilities, we construct a Markov chain of the receiver buffer during this buffering stage with transition probability matrix

$$\mathbf{\Psi} = \begin{array}{c} \\ j-1 \\ j \\ \vdots \\ j+B_{jit}-2 \\ j+B_{jit}-1 \\ \vdots \\ j+B_{jit}+R-2 \end{array} \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 & \cdots & 0 \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 & \cdots \\ \vdots & \ddots & \ddots & & & & \vdots \\ 0 & \cdots & 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R \\ 0 & \cdots & & 0 & \mathbf{I} & 0 & \cdots \\ \vdots & & & \cdots & \ddots & \ddots & 0 \\ 0 & & & & \cdots & 0 & \mathbf{I} \end{bmatrix}. \tag{4.16}$$

States with $g \in [j+B_{jit}-1, j+B_{jit}+R-2]$ are modeled as absorption states, because once the system arrives at any one of these states, it exits the jitter recover buffering stage, and resumes playing out the video. Then the distribution of resuming display among these states can be obtained by solving the absorption probabilities of these states, starting

from state $(j-1, S_l)$ [33].

Now we can compute $Q_{l,k}(j,i)$ for the FPD scheme. Let $\overline{\pi}_{j,l}$ be the state distribution of exiting the buffering stage after a jitter at (j, S_l), with each element of $Kg+h$, for $g = j + B_{jit} - 1, ..., j + B_{jit} + R - 2$ and $h \leq K$, equal to the absorption probability of the corresponding state (g, S_h). Remember, after exiting the buffering stage, the system restarts playing out the video from packet j. Then we can compute $Q_{l,k}(j,i)$ from

$$\overline{\pi}'_{d_i} = \overline{\pi}_{j,l}(\prod_{t=d_j}^{d_i-1} \mathbf{\Phi}\mathbf{U}_t)\mathbf{\Phi}, \tag{4.17}$$

and

$$Q_{l,k}(j,i) = \overline{\pi}'_{j,l}[K(i-1)+k], \ i \geq j + B_{jit}. \tag{4.18}$$

Moreover, in order to compare FPD with the FBD scheme, we want to quantify the jitter recover buffering delay of FPD. Since the delay is not constant, we will look into the expected buffering delay. The expected buffering delay after a jitter at (j, S_l) can be obtained by solving the mean time to absorption of the above Markov chain Ψ [33]. Let m_k be the mean time to absorption into any one of the absorption states, starting from state S_k. Then the expected buffering delay of the nth jitter is

$$E\{D_n\} = \sum_{i}^{p(L)} \sum_{k=1}^{K} P_k^{(n)}(i) m_k. \tag{4.19}$$

Here we define $D_n = 0$ if the nth jitter does not exist. And the expected total jitter buffering delay experienced by the client becomes

$$E\{D_{total}\} = \sum_{n=1}^{\infty} E\{D_n\} = \sum_{n=1}^{\infty} \sum_{i=1}^{p(L)} \sum_{k=1}^{K} P_k^{(n)}(i) m_k. \tag{4.20}$$

Then the average buffering delay per jitter can be given by

$$\frac{E\{D_{total}\}}{E\{N\}} = \frac{\sum_{n=1}^{\infty}\sum_{i=1}^{p(L)}\sum_{k=1}^{K} P_k^{(n)}(i)m_k}{\sum_{n=1}^{\infty}\sum_{i=1}^{p(L)}\sum_{k=1}^{K} P_k^{(n)}(i)}. \tag{4.21}$$

Fixed Buffered Playout Time

The FPT scheme is similar to the FPD scheme, except that, rather than buffering a fixed amount of data before resuming display, the FPT scheme buffers data until the playout duration of these data reaches a fixed value T_{jit}. One advantage of this scheme is that, when display resumes, it is guaranteed that the next jitter is at least T_{jit} away in the future. This scheme is used in Windows Media Player and Real Player.

The process to find $Q_{l,k}(j,i)$ is similar to the one in the FPD scheme. The difference is that, for different values of the jitter position j, the number of packets to buffer is $p(d_j + T_{jit}) - (j-1)$, which is not a fixed number as it is in the FPD scheme. Then for each jitter position j, we have a Markov chain characterized by the transition probability matrix

$$\mathbf{\Psi}_j = \begin{array}{c} \\ j-1 \\ j \\ \vdots \\ p(d_j+T_{jit})-1 \\ p(d_j+T_{jit}) \\ \vdots \\ p(d_j+T_{jit})+R-1 \end{array} \begin{bmatrix} & & & & & & \\ \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 & \cdots & 0 \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R & 0 & \cdots \\ \vdots & \ddots & \ddots & & & & \vdots \\ 0 & \cdots & 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_R \\ 0 & \cdots & & 0 & \mathbf{I} & 0 & \cdots \\ \vdots & & & \cdots & \ddots & \ddots & 0 \\ 0 & & & & \cdots & 0 & \mathbf{I} \end{bmatrix}. \tag{4.22}$$

Then we can compute the state distribution of exiting the buffering stage after a jitter

at (j, S_l), $\overline{\pi}_{j,l}$, by computing the absorption probabilities of $\mathbf{\Psi}_j$ starting from state (j, S_l). Similar to (4.17) and (4.18), we have

$$\overline{\pi}'_{d_i} = \overline{\pi}_{j,l}(\prod_{t=d_j}^{d_i-1} \mathbf{\Phi}\mathbf{U}_t)\mathbf{\Phi}, \tag{4.23}$$

and

$$Q_{l,k}(j,i) = \overline{\pi}'_{j,l}[K(i-1)+k]. \tag{4.24}$$

Further, for different $\mathbf{\Psi}_j$, the mean time to absorption is different. We use $m_k(i)$ to denote the mean time to absorption of $\mathbf{\Psi}_i$, starting from $(i-1, S_k)$. Then similar to the FPD scheme, the average buffering delay per jitter is given by

$$\frac{E\{D_{total}\}}{E\{N\}} = \frac{\sum_{n=1}^{\infty}\sum_{i=1}^{p(L)}\sum_{k=1}^{K} P_k^{(n)}(i)m_k(i)}{\sum_{n=1}^{\infty}\sum_{i=1}^{p(L)}\sum_{k=1}^{K} P_k^{(n)}(i)}. \tag{4.25}$$

4.4 Finite Buffer Case

In the previous sections, we have studied the case where the receiver buffer size is infinite, or large enough, such that the amount of data received is not limited from above. Next, we study the case where the receiver buffer size is limited.

Suppose the receiver buffer size is B. At any video position t, we can have at most received $p(t)+B$ packets. We consider systems in which the client continuously updates its buffer fullness to the server, the server stops sending data packets when it knows the receiver buffer is full, and the server restarts sending data packets once the buffer starts to empty again[1]. Figure 4.3 illustrates the state transitions in this case. The transitions to any state with $g \geq p(t)+B$ should be merged into the states with $g = p(t)+B-1$.

[1]This mechanism is vital, as jitters due to buffer overflow would occur without it.

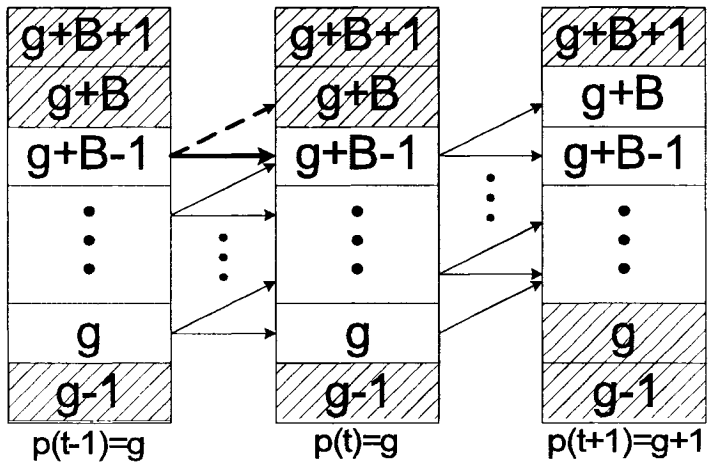


Figure 4.3: Solid arrows indicate the transition of states that should be considered. The shaded blocks indicate the states that have violated the playout deadline or that have surpassed the receiver buffer limit. The dotted arrow represents the state transitions that should be merged into the state transitions represented by the large solid arrow.

This can be done by setting

$$\overline{\pi}_t[K(p(t)+B-1)+l] = \sum_{g \geq p(t)+B-1} \overline{\pi}_t[Kg+l], \tag{4.26}$$

and $\overline{\pi}_t[Kg+l] = 0$ for $g \geq p(t)+B$ and $l \leq K$, while computing the state distribution.

And this operation can be formulated in matrix multiplications similar to (4.10) as:

$$\overline{\pi}_{d_i-1} = \overline{\pi}_{init}(\prod_{t=-\triangle}^{d_i-1} \mathbf{\Phi U}'_t), \tag{4.27}$$

where

$$\mathbf{U}'_t = \begin{bmatrix} \mathbf{0}_{Kp(t)\times Kp(t)} & 0 & \overset{p(t)+B-1}{0} & 0 \\ 0 & \mathbf{I}_{K(B-1)\times K(B-1)} & 0 & 0 \\ 0 & 0 & \mathbf{I}' & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.28}$$

and

$$\mathbf{I}' = [\underbrace{\mathbf{I}_{K\times K} \cdots \mathbf{I}_{K\times K}}_{R+1}]^T. \tag{4.29}$$

$\mathbf{I}'$ here is a $K(R+1)\times K$ matrix. By doing this, any state with $g \geq p(t)+B$ is merged into a state with $g = p(t)+B-1$ according to its channel state. Then (4.11) can be rewritten as

$$\overline{\pi}'_{d_i} = \overline{\pi}_{init}(\prod_{t=-\triangle}^{d_i-1} \mathbf{\Phi}\mathbf{U}'_t)\mathbf{\Phi}. \tag{4.30}$$

The process is similar while computing $Q_{l,k}(j,i)$. We only need to replace $\mathbf{U}_t$ with $\mathbf{U}'_t$ in (4.17) and (4.23), and in (4.14) for $t \geq d_j$. In (4.14) when $t < d_j$, we should set

$$\mathbf{U}'_t = \begin{bmatrix} \mathbf{0}_{Kj\times Kj} & 0 & \overset{j+B-1}{0} & 0 \\ 0 & \mathbf{I}_{K(B-1)\times K(B-1)} & 0 & 0 \\ 0 & 0 & \mathbf{I}' & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.31}$$

Video name	Parameters of the video stream	
	Mean bit rate (per sec)	Peak bit rate (per sec)
Alpin Ski	1.8247e+05	3.0248e+06
Formula 1	1.7754e+05	2.1428e+06
Jurassic	1.7100e+05	3.1668e+06

Table 4.2: Parameters of different video streams

because within the buffering delay, the buffer can contain up to the $(j + B - 1)$th packet.

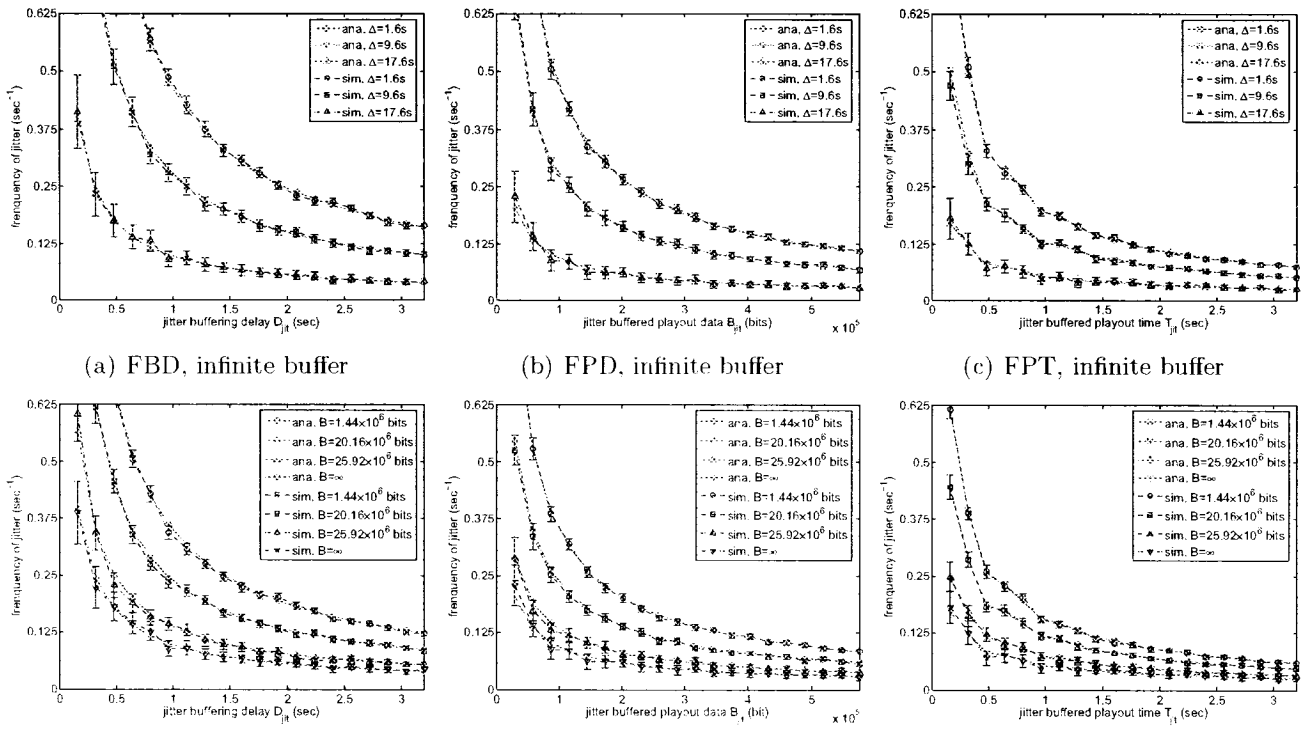
4.5 Experiment and Numerical Results

In this section, we apply the proposed analytical framework to study how different jitter recovering schemes and the choice of parameters affect the jitter performance in VBR video streaming. The analysis results are validated by comparison with simulation results. We further investigate the optimal balance between jitters and buffering delay.

4.5.1 Experimental Setup

We experiment on wireless streaming using the same MPEG-4 VBR video traces we use in chapter 3. To emulate a realistic wireless channel, we adopt a three-state extended Gilbert model shown in figure 3.4 as the VBR channel model (i.e., $k = 2$), with transition probabilities $\Pr\{0|1\} = 0.1$, $\Pr\{0|10\} = 0.5$ and $\Pr\{0|00\} = 0.8$. Then the transition matrixes are constructed by

$$A_0 = \begin{bmatrix} 0 & 0.1 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0.8 \end{bmatrix}, \text{ and } A_1 = \begin{bmatrix} 0.9 & 0 & 0 \\ 0.5 & 0 & 0 \\ 0.2 & 0 & 0 \end{bmatrix}.$$



(a) FBD, infinite buffer (b) FPD, infinite buffer (c) FPT, infinite buffer

(d) FBD, finite buffer, $\triangle = 17.6s$ (e) FPD, finite buffer, $\triangle = 17.6s$ (f) FPT, finite buffer, $\triangle = 17.6s$

Figure 4.4: Frequency of jitters vs. $D_{jit}/B_{jit}/T_{jit}$ for FBD/FPD/FPT scheme for video "Alpin Ski".

We set the packet size to 1800 bytes and the time interval between two consecutive transmission to 80 ms. This results in a maximum transmission bit rate of 180,000 bit/s. For different initial delay values, receiver buffer sizes and jitter recover scheme parameters, we simulate in Matlab the transmission and playback of each video sequence over 200 realizations of the random VBR channel and measure the number of jitters and the buffering delay of each jitter. Furthermore, we compute the expected number of jitters and expected buffering delay based on the proposed analysis and compare them with the simulation results.

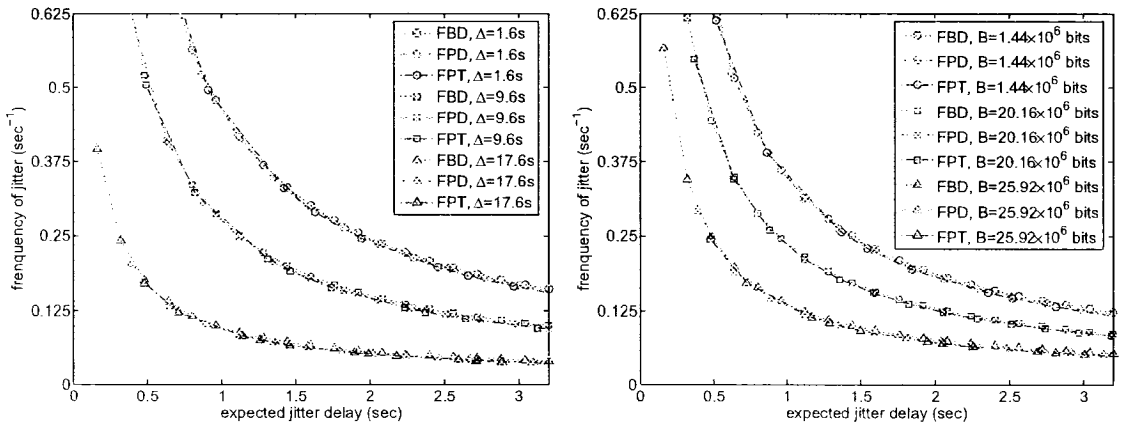
4.5.2 Numerical Validation

In Figure 4.4, we plot the frequency of jitters vs. jitter recovery parameters for video "Alpin Ski" in different initial delay and receiver buffer size settings. Figure 4.4(a) and 4.4(d) are based on the fixed jitter buffering delay FBD scheme as in 4.3.2. Figure 4.4(b) and 4.4(e) are based on the fixed jitter buffered playout data FPD scheme as in 4.3.2. Figure 4.4(c) and 4.4(f) are based on the FPT scheme as in 4.3.2. The 95% confidence intervals for the simulations are also shown in these figures. We observe good match between the derived expected number of jitters and simulation. Similar results are observed for different playback curves and omitted to reduce redundancy.

It can be seen in Figure 4.4(a) that, for all initial delay and buffer size settings, we observe a deminishing gain by increasing D_{jit} on the performance: the number of jitters decreases dramatically as D_{jit} increases at the beginning and then slowly after D_{jit} surpasses a certain value. We also observe that for a fixed buffer size, a larger initial delay results in a smaller number of jitters and a more dramatic decrease as D_{jit} increases. On the other hand, for a fixed initial delay, a larger buffer size also results in fewer jitters and a sharper drop.

The practical implication of this observation is clear. The jitter buffering delay is a delicate parameter in the optimal design of multimedia streaming. Increasing the jitter buffering delay can drastically reduce the number of jitters, but only up to a certain level. Beyond that, the improvement is negligible, and the long jitter buffer delay may actually harm the prospected quality of the streaming.

Similar observations can be found in other jitter-recovery buffering schemes.



(a) Infinite buffer with different initial delays (b) Finite buffers with fixed initial delay $\triangle = 17.6s$

Figure 4.5: Comparison of three jitter recovery buffering schemes. Jitter vs. average jitter recover buffering delay.

4.5.3 Comparison of Jitter-Recovery Buffering Schemes

In Figure 4.5 we plot the jitter frequency vs. expected buffering delay of the three aforementioned jitter recovery schemes, with different fixed initial delays and receiver buffer sizes. For the FBD scheme, the expected delay is just D_{jit}. For the FPD and FPT schemes, the expected delays are given by equation (4.21) and (4.25). It is interesting to note that these schemes have very close performance in terms of jitter frequency.

However, as shown in Figure 4.6, the FBD scheme has zero delay variance because the delay is fixed, while the other two schemes have large delay variances. Moreover, the FPT scheme has a much larger variance than the FPD scheme. It is because in the FPT scheme, the amount of data that has to be buffered varies drastically when jitter occurs at different position in the video, and hence results in more fluctuations, in comparison with the FPD scheme, which always buffers for a fixed amount of data. In another words, while jitters can not occur close to each other in the FPT scheme, it is more likely to

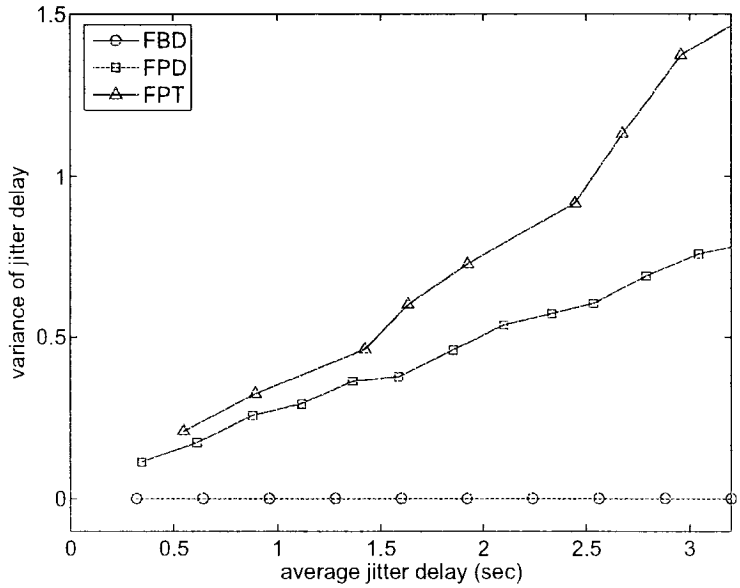
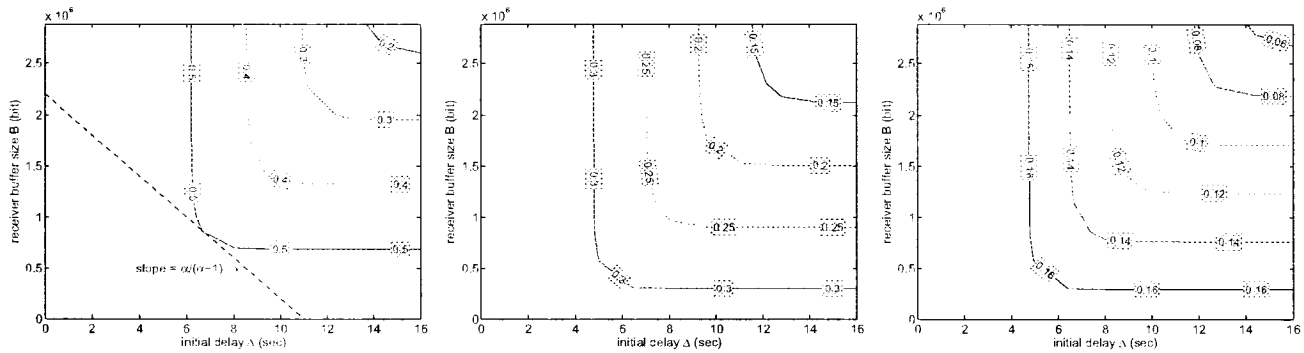


Figure 4.6: Variance of jitter recovery buffering delay.



(a) $D_{jit} = 0.8s$ (b) $D_{jit} = 1.6s$ (c) $D_{jit} = 3.2s$

Figure 4.7: Contour maps of buffer size vs. initial delay, with the frequency of jitters labeled on the curves.

have unacceptably long buffering delay compared with the other two schemes.

4.5.4 Optimal Delay-Buffer Tradeoff

Finally, we consider the optimal tradeoff between the initial playback delay $\triangle$ and the receiver buffer size B in video streaming. In most cases, long buffering delay in the middle of a streaming session is less acceptable for a streaming client. Therefore, a client may impose stiff constraints on the jitter recovery buffering delay and may wish to tradeoff the initial delay and the receiver buffer size for a lower frequency of jitters. Figure 4.7 shows contour maps for the initial delay and the buffer size that achieve different frequencies of jitters, for different fixed jitter recover buffering delays.

We observe that the right branch of a contour curve is roughly horizontal. This suggests that when the buffer size is fixed, increasing the initial delay can only decrease the frequency of jitters to a certain level. When $\triangle$ is large enough, the buffer is always filled up during the initial delay, and the display of the video always begins with a full buffer. Further increasing the initial delay will not change this situation, resulting in no improvement of performance. Similarly, the left branch of a contour curve is roughly vertical. The rational behind this is that, when B is large enough, the receiver buffer will never be completely filled throughout a streaming session, which is equivalent to the case with infinite buffer. Finally, these figures provide a convenient means to obtain an optimal operating point for the system that balances the tradeoff between $\triangle$ and B, given a certain required frequency of jitters and jitter recovery buffering delay. If we define a cost function as a weighted sum of the two

$$C = \alpha\triangle + (1-\alpha)B, \tag{4.32}$$

where $\alpha \in [0, 1]$, then to minimize this cost function, we simply find the tangent line

of the corresponding contour curve with slope $\frac{\alpha}{\alpha-1}$. The point of contact between the tangent line and the contour curve, given any value of frequency of jitter, defines the optimal operating point for the system. An illustration of such a procedure is shown in Figure 4.7(a).

Chapter 5

Concluding Remarks

In this thesis, we have considered the problem of providing QoS to VBR encoded video streaming service over random VBR channels. Partial of our research have been published in QShine 2006 [34] and INFOCOM 2007 [35]. We have shown that, when some statistical characteristics of the channel, such as the maximum bandwidth and the channel state transition probability, are available, a certain level of QoS can be guaranteed by selecting appropriate startup delay, buffer sizes, and jitter recover schemes.

We first present an analytical framework that only requires knowledge of the playback curve, the maximum channel bit rate, and general statistics of the channel. The probability bounds for buffer underflow and overflow have been derived for both the infinite buffer and finite buffer cases. We demonstrate the application of this analytical framework to a wireless access network with a generic extended Gilbert model and ARQ error control. Numerical and experimental results using MPEG-4 encoded VBR video traces validate our findings. The computed bounds allow precise estimation of the effect of startup delay and receiver buffer size, especially within system parameter ranges that lead to acceptable jitter free probabilities.

We then present an analytical framework that only requires knowledge of the playback curve and a generic Markov VBR channel model. 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. The proposed numerical analysis allow precise estimation of the effect of the choice of jitter recovery schemes, initial playout delay and receiver buffer size.

To practical streaming system designers, the proposed analysis techniques provide convenient frameworks to optimize the tradeoffs between the various system parameters for optimal VBR multimedia streaming over random VBR channels.

Bibliography

[1] M. Etoh, and T. Yoshimura, "Advances in Wireless Video Delivery," *Proc. of IEEE*, vol. 93, no. 1, pp. 111–122, Jan 2005.

[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 Communitions Networks*, Sept 2002.

[3] T. V. Lakshman, A. Ortega, and A.R. Reibman, "VBR Video: Trade-offs and Potentials," *Proc. of IEEE*, vol. 86, no. 1, pp. 952- 973, May 1998.

[4] D. L. Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, pp. 46–58, Apr 1991.

[5] G. K. Wallace, "The JPEG Still Picture Transmission Standard," *Communications of the ACM*, vol. 34, pp. 30–44, Apr 1991.

[6] I. Dalgic and F. A. Tobagi, "Performance Evaluation of ATM Networks Carrying Constant and Variable Bit-Rate Video Traffic," *IEEE Journal on Selected Areas in Communications*, vol. 15, Aug 1997.

[7] E. P. Rathgeb, "Policing of Realistic VBR Video Traffic in an ATM Network," *International Journal of Digital and Analog Communication Systems*, vol. 6, pp. 213–226, Oct 1993.

[8] S. S. Lam, S. Chow, and D. K. Yau, "An Algorithm for Lossless Smoothing of MPEG Video," in *Proc. ACM SIGCOMM*, Aug 1994, pp. 281–293.

[9] A. R. Reibman and A. W. Berger, "Traffic Descriptors for VBR Video Teleconferencing over ATM Networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 329–339, Jun 1995.

[10] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic," *IEEE/ACM Transcactions on Networking*, vol. 5, pp. 741–755, Dec 1997.

[11] O. Rose, "Statistical Properties of MPEG Video Traffic and their Impact on Traffic Modeling in ATM Systems," in *Proc. of Conference on Local Computer Networks*, Oct 1995, pp. 397–406.

[12] 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.

[13] P. Thiran, J.-Y. Le Boudec, and F Worm,, "Network Calculus Applied to Optimal Multimedia Smoothing," in *Proc. IEEE Infocom*, Apr 2001.

[14] 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.

[15] 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.

[16] 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.

[17] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, Oct 1960.

[18] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, 1963.

[19] B. Wang, J. F. Kurose, P. Shenoy, and D. F. Towsley, "Multimedia Streaming via TCP: An Analytic Performance Study," in *Proc. of the 12th annual ACM international conference*, Oct 2004.

[20] J. Padhye, V. Firoiu, and D. Towsley, "A Stochastic Model of TCP Reno Congestion Avoidance and Control," *Tech. Rep. 99-02, Department of Computer Science, University of Massachusetts, Amherst*, 1999.

[21] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, "On the Autocorrelation Structure of TCP Traffic," *Computer Networks Journal Special Issue on Advances in Modeling and Engineering of Long-Range Dependent Traffic*, 2002.

[22] 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 2006*, Apr 2006.

[23] P. A. Chou and Z. Miao, "Rate-Distortion Optimized Streaming of Packetized Media," *Microsoft Research Tech. Rep. MSR-TR-2001-35*, Feb 2001.

[24] J. Chakareski, P. A. Chou, and B. Girod, "Rate-Distortion Optimized Streaming from the Edge of the Network," in *Proc. IEEE Fifth Workshop on Multimedia Signal Processing*, Dec 2002.

[25] M. Kalman and B. Girod, "Rate-Distortion Optimized Video Streaming with Multiple Deadlines for Low Latency Applications," in *Proc. of Packet Video Workshop*, Dec 2004.

[26] SH. eferoglu, Y. Altunbasak, O.Gurbuz, and O. Ercetin, "Rate Distortion Optimized Joint ARQ-FEC Scheme for Real-Time Wireless Multimedia," in *Proc. of IEEE ICC*, May 2005.

[27] 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*, Jan 2000.

[28] A. Kpke, A. Willig, and H. Karl, "Chaotic maps as parsimonious bit error models of wireless channels," in *Proc. IEEE INFOCOM*, March 2003.

[29] A. Willig, "A New Class of Packet- and Bit-Level Models for Wireless Channels," in *Proc. 13th IEEE Int. Symp. Personal Indoor and Mobile Radio Communication (PIMRC)*, 2000.

[30] H. S. Wang and N. Moayeri, "Finite State Markov Channel - A Useful Model for Radio Communications Channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, 1995.

[31] M. Hassan, M. M. Krunz, and I. Matta, "Markov-Based Channel Characterization for Tractable Performance Analysis in Wireless Packet Networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 821–831, 2004.

[32] F.H.P. Fitzek and M. Reisslein, "Mpeg-4 and h.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, November 2001.

[33] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, Fourth edition, 2002.

[34] Guanfeng Liang and Ben Liang, "Jitter Free Probability Bounds for Video Streaming over Random VBR Channel," in *Proc. the Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, Aug 2006.

[35] Guanfeng Liang and Ben Liang, "Balancing Interruption Frequency and Buffering Penalties in VBR Video Streaming," in *Proc. of IEEE INFOCOM*, May 2007.