UNIVERSITY OF CALIFORNIA, SAN DIEGO

Cross-Layer Bandwidth Allocation and Delay Allocation for Video Transmission over Wireless Channels

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Yushi Shen

Committee in charge:

Professor Pamela C. Cosman, Chair
Professor Laurence B. Milstein, Co-Chair
Professor Serge J. Belongie
Professor William S. Hodgkiss
Professor Paul H. Siegel

2006

Copyright ©

Yushi Shen, 2006

All rights reserved.

The dissertation of Yushi Shen is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

Co-Chair

_____

Chair

University of California, San Diego

2006

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my two advisors, Prof. Pamela Cosman and Prof. Laurence Milstein, for their continual support over the past several years. Their understanding, encouragement and guidance have provided a good basis for the present thesis. And I have benefited tremendously from their broad range of knowledge, acute insight and infectious enthusiasm for research.

While being a great source of valuable ideas and advice, they have always allowed me a high degree of freedom to pursue my research project. Throughout my Ph.D. studies, I also had the opportunity to participate at international conferences and work as a research intern. None of that would have been possible without their support.

In addition to supervising my research, Prof. Cosman has given my invaluable assistance in life, such as helping me improve my writing skills and public speaking skills, and giving me advice on career choices.

The dedication and love that Prof. Milstein shows for research is exceptional. Being a giant in digital communication theory with a large number of Ph.D. students, I am constantly impressed by how he made all of us feel privileged. His approach to research, intuition and creative thinking had a great influence on me.

I would also like to thank my thesis committee members, Prof. Paul Siegel, Prof. William Hodgkiss, and Prof. Serge Belongie, for their time, assistance, and advice. In particular, Prof. Paul Siegel was inspirational as the instructor of *Trellis-Coded Modulation* and *Advance Coding and Modulation in Digital Communications*.

I also thank my fellow researchers in the *Information Coding Laboratory*

(ICL) and *Communication Theory and Systems* (CTS) group, for the friendly and stimulating atmosphere. Special thanks to Athanasios Leontaris, Kartikeya Mehrotra, Sandeep Kanumuri, and Qinghua Zhao, for the many fruitful discussions with them that made this dissertation possible. Also, thanks to Song Cen, Yee Sin Chan, Qi Qu, Jillian Cannons, Ramesh Annavajjala, Vijay Chellappa, Solmaz Alipour, Wei Geng, Mohammed Mohamedali, Ye Yan, and many other friends and co-workers.

Finally, I would like to dedicate this dissertation to my beloved parents, Zhigong Shen and Xiao Liu, my beloved grandparents, Yanlin Liu and Erlan Xiao, and my beloved fiancee, Hongyan Ma. They have provided great support in every aspect of my life, and their importance to me I cannot put into words.

Chapter II of this thesis, in full, is a reprint of the material as it appears in Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Video Coding with Fixed Length Packetization for a Tandem Channel," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 273- 288, Feb. 2006. I was the primary author and the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for Chapter II.

Chapter III of this thesis, in full, is a reprint of the material as it appears in Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Error Resilient Video Communications over CDMA Networks with a Bandwidth Constraint," which will appear in the *IEEE Transactions on Image Processing*, Oct. 2006. I was the primary author and the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which

forms the basis for Chapter III.

Chapter IV of this thesis, in full, is a reprint of the material as it appears in Kartikeya Mehrotra, Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Delay Budget Partitioning in Wireless Multimedia," which is submitted to the *IEEE Transactions on Image Processing* and is currently under review. Both Kartikeya Mehrotra and I were primary authors and contributed equally on this paper, the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for Chapter III.

Chapter VI of this thesis, in full, is a reprint of the material as it appears in Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Weight Distribution of a Class of Binary Linear Block Codes Formed from RCPC Codes," *IEEE Communications Letters*, vol. 9, no. 9, pp. 811-813, Sept. 2005. I was the primary author and the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for Chapter VI.

VITA

1978	Born, Beijing, P. R. China
2001	B.S., Electrical Engineering , Tsinghua University, Beijing, P. R. China
2003	M.S., Electrical Engineering (Communication Theory and Systems), University of California, San Diego
2006	Ph.D., Electrical Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

Y. Shen, P. C. Cosman, and L. B. Milstein, "Video Communications with Optimal Intra/Inter-Mode Switching over Wireless Internet," in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, pp. 1548-1552, Pacific Grove, California, USA, Nov. 9-12, 2003.

Y. Shen, P. C. Cosman, and L. B. Milstein, "Weight Distribution of a Class of Binary Linear Block Codes Formed from RCPC Codes," in *IEEE Communications Letters*, vol. 9, no. 9, pp. 811-813, Sept. 2005.

Y. Shen, P. C. Cosman and L. B. Milstein, "Video Coding for a Time Varying Tandem Channel with Feedback," in *Proceedings of the IEEE Data Compression Conference*, pp. 480, Snowbird, Utah, USA, March 29-31, 2005.

Y. Shen, P. C. Cosman, and L. B. Milstein, "Video Coding with Fixed Length Packetization for a Tandem Channel," in *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 273- 288, Feb. 2006.

Y. Shen, P. C. Cosman, and L. B. Milstein, "Error Resilient Video Communications over CDMA Networks with a Bandwidth Constraint," to appear, *IEEE Transactions on Image Processing*, Oct. 2006.

K. Mehrotra, Y. Shen, P. C. Cosman, and L. B. Milstein, "Delay Budget Partitioning in Wireless Multimedia," submitted to *IEEE Transactions on Image Processing*.

FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in Communication Theory and Systems.
Professors Pamela C. Cosman and Laurence B. Milstein.

ABSTRACT OF THE DISSERTATION

Cross-Layer Bandwidth Allocation and Delay Allocation for Video Transmission over Wireless Channels

by

Yushi Shen

Doctor of Philosophy in Electrical Engineering

(Communication Theory and Systems)

University of California, San Diego, 2006

Professor Pamela C. Cosman, Chair

Professor Laurence B. Milstein, Co-Chair

This dissertation proposes both coding schemes and system design strategies for video wireless communications. Video coding can benefit greatly from the development of algorithms for optimal intra/inter mode switching within a rate-distortion framework. We propose an algorithm to optimally switch between intra-coding and inter-coding modes for a video coder that operates on a packet-switched network with fixed-length packets. This optimal mode selection algorithm is integrated with an efficient channel encoder. And the framework is further extended to operate on a time-varying tandem wiredline-wireless channel with feedback information from the receiver.

We then discuss a cross-layer optimization scheme for the system design of a wireless video communications system. We present a video transmission scheme with

adaptive bandwidth allocation over wired and wireless CDMA networks. We derive the statistics of the received signal, as well as a theoretical bound on the packet drop rate at the receiver. Based on these results, an algorithm for efficient bandwidth allocation among source coding, channel coding and spreading is derived at the packet level, which incorporates the effects of both the changing channel characteristics and the source content.

Another fundamental tradeoff in the cross-layer design of a communications system that we study is how to optimize the system when there is a delay constraint imposed by the application. The key elements in this tradeoff are the queuing delay in the source encoder output buffer, the delay caused by the interleaver, and the delay caused by channel decoding.

I

Introduction

I.A Basic Concepts

This dissertation provides a unified treatment of video compression, channel coding, modulation and multiple accessing. These basic elements of a wireless video communication system are briefly introduced in this section.

I.A.1 Source Coding

Source coding, also called data compression, is the process of removing redundancy from the original data/information. Compression is important because it helps reduce the consumption of expensive resources, such as channel bandwidth or disk space. However, compression requires processing power, which can also be expensive. The design of data compression schemes therefore involves tradeoffs between various factors, including compression capability, distortion, computational resource requirements, and often other considerations as well.

Source coding is especially important for video transmission, because video is characterized by a huge amount of input data, which inherently contains statistical redundancy (including temporal, spatial, and spectral) and subjective redundancy.

In most current video encoders, such as MPEG-2/4 and H.263/4, there are two basic encoding modes. Inter-coding, or motion compensation, is a basic and efficient approach in video coding. However, it may suffer from potentially severe error propagation, because a single error in a frame may corrupt all subsequent frames if inter-coding is used repeatedly. Intra-coding, by encoding the current video content without reference to any other frame, can stop error propagation. But this mode is usually much more costly in bits than inter-coding. Thus, it is desired to switch between intra and inter coding intelligently, according to both channel conditions and the video content, to achieve the right balance between compression efficiency and robustness. Video codecs that use motion compensation benefit greatly from the development of algorithms for near-optimal intra/inter mode switching within a rate-distortion framework.

I.A.2 Channel Coding

In wireless video communications, the compressed video, which make use of both entropy encoding and predictive coding algorithms, is extremely vulnerable to channel impairments. Real-time video streaming applications are also sensitive to transport delay and delay variation. Therefore, compression and transmission provisions that avoid catastrophic failure caused by lost, delayed, or errant data are imperative.

Channel coding is used in digital communication systems to protect the in-

formation from noise and interference, and thus reduce the bit error rate. It is mostly accomplished by selectively introducing redundant bits into the transmitted information system. The additional bits will allow error correction or error detection, while the cost is either a reduction in data rate or an expansion in bandwidth.

Before 1948, it was generally believed that it was impossible to obtain arbitrarily low error rates when communicating in the presence of noise. This misconception was disproved by Claude Shannon, the father of information theory, in his ground breaking work "A Mathematical Theory of Communication" [1]. Here, he introduced the concept of channel capacity, and proved that as long as the information rate is below the channel's capacity, reliable communication with an arbitrarily small error rate is possible. But the proof did not show how to construct such a good channel code.

Shannon's work spurred a whole new area of study which we now call information theory. To date, there are many different types of channel coding methods that are used in practice. Linear block codes and convolutional codes have been implemented in many practical systems. Recently, turbo codes and low density parity check (LDPC) codes have gained much interest in both academia and industry alike, due to their near-capacity performance.

I.A.3 Multiple Access Techniques

Multiple access techniques are used to enable acceptable communication among users sharing a common channel. Three basic types of multiple access presently used in communication systems are time-division multiple access (TDMA), frequency-division

multiple access (FDMA), and code-division multiple access (CDMA).

Technically, in TDMA, a channel is divided into independent, non-overlapping time slots for each user. In FDMA, a channel is divided into independent, non-overlapping frequency bands, or subchannels, for each user. In CDMA, every user transmits its signal at the same time and in the same frequency band, but with different codes.

In this thesis, we focus on direct-sequence CDMA (DS-CDMA) technology, wherein a data signal (bit) at the transmitter is combined with a higher data rate chip sequence. The multiplication operation in a DS-CDMA transmitter increases the transmission bandwidth based on the number of the chips embedded (called the processing gain).

I.B Motivation

Video communication is characterized by a huge amount of input data, and compression is needed to reduce the number of transmitted bits. This leads to a host of research in lossy video compression that achieves compression by sacrificing signal fidelity, while preserving visual quality. In video coding, progress in the domain of motion compensation is vital to achieve higher compression gains. And it will benefit greatly from the development of algorithms for optimal intra/inter mode switching within a rate-distortion framework.

The well-known “separation theorem”, which states that the performance of the optimum source-channel coding scheme can be achieved by optimizing the source coding and the channel coding separately, does not hold when the realistic constraints,

such as constrained bandwidth, power, delay and complexity, are considered. Thus, joint source-channel coding (JSCC) is investigated to achieve the optimum performance. And the advances achieved in cross-layer design at the lower layers (for example, the physical layer) can be further enhanced by taking into account source characteristics and requirements allowing existing networks to provide optimal time-varying Quality of Service (QoS). This issue is especially important for video/multimedia applications over wireless and heterogeneous channels, as these applications are essentially delay-sensitive and bandwidth-intense.

I.C Thesis Outline

In Chapter II, we propose an algorithm to optimally switch between intra-coding and inter-coding modes for a video coder that operates on a packet-switched network with fixed-length packets. Different re-synchronization schemes are considered and compared. This optimal mode selection algorithm is integrated with an efficient channel encoder. The system performance is both analyzed and simulated, where the channel is assumed to have bit errors (due to noise and fading on the wireless portion of the channel) and packet erasures (due to congestion on the wired portion). The framework is further extended to operate on a time-varying wireless Internet channel with feedback information from the receiver. Both instantaneous feedback and delayed feedback are evaluated, and an improved method of refined distortion estimation for encoding is presented and evaluated.

In Chapter III, we investigate the cross-layer design problem for a wireless

video system with a constant bandwidth. We present a video transmission scheme with adaptive bandwidth allocation for wireless CDMA networks. We derive the statistics of the received signal and a theoretical bound on the packet drop rate at the receiver. Based on these results, an algorithm for efficient bandwidth allocation among source coding, channel coding and spreading is derived at the packet level, which incorporates the effects of both the changing channel characteristics and the source content. Detailed simulations are done to evaluate the performance of the systems. The sensitivity of the system to estimation error is also analyzed.

In Chapter IV, we study another fundamental tradeoff in the cross-layer design of a wireless video communications system, that is to optimize the system when there is a delay constraint imposed by the applications, i.e., the issue of delay allocation. The key elements in this tradeoff are the queuing delay in the source encoder output buffer, the delay caused by the interleaver, and the delay caused by channel decoding. We formulate the delay partitioning problem mathematically, and end up with a relationship among these three main delay components. In particular, we study how the tradeoff will be affected by the motion of the video content, the rate of variation of the channel, the delay budget and the channel bit rate.

In Chapter V, we enumerate and discuss all our contributions in this dissertation for each individual chapter, and discuss potential future work. We note that partial conclusions are also given at the end of each individual chapter.

II

Video Coding with Fixed Length Packetization for a Tandem Channel

II.A Introduction

Packet video is becoming a significant portion of traffic over wireless and wireline networks. However, network congestion and wireless channel errors can yield tremendous packet loss and thus degrade the video quality. The transmitted bitstream should be organized to minimize the possible corruption and error propagation.

We are interested in using fixed-length packets over tandem channels, whereby we mean a channel that has both wireline and wireless links, and so experiences both packet erasures due to congestion on the wireline component, and bit errors due to noise and fading on the wireless component of the link. Video communications over tandem channels has been addressed in references such as [2, 3, 4, 5].

It is desired to switch between intra and inter coding intelligently according

to channel conditions and video content, to achieve the right balance between compression efficiency and robustness. The ROPE algorithm for inter/intra mode selection was proposed in [6]; it used variable length packets and was designed for a packet erasure channel whose loss rate is fixed and known. Our work uses distortion estimation and mode switching in the style of the ROPE algorithm, but for more complex channels, so significant modifications are needed.

This chapter is organized as follows. In Section II.B, we derive a modified ROPE algorithm for fixed-length packets with two different re-synchronization approaches. Both analysis and simulation results suggest that the performance of fixed-length packets is worse than that of variable-length packets. We also compare different re-synchronization approaches. In Section II.C, we study video coding over a constant tandem channel with both bit errors and packet erasures. By means of a well-designed concatenated channel coder, the tandem channel can dynamically be treated as a simple erasure channel by the source encoder, thus the modified ROPE algorithm can be used. In Section II.D, we extend our framework to the scenario where the channel has time-correlated variation, and a feedback channel is used to tell the encoder about the channel status. The performance is evaluated with both instantaneous and delayed feedback information. Conclusions are drawn in Section IV.E.

II.B Optimal Mode Switching with Fixed-length Packets

In video compression, typically each frame is segmented into macroblocks (MBs) of size 16 by 16 pixels. One horizontal row or slice of MBs is called a Group of

Blocks (GOB). The encoding mode and the quantization step are selected for each MB individually in DCT-based video encoders such as MPEG-2 and MPEG-4. In a packetized transmission system, the compressed bit stream is then sent by either variable-length or fixed-length packets.

For variable-length packets, each GOB can be carried in a separate packet; a short packet header says which GOB is in the packet. One packet loss entails loss of the whole GOB, without affecting decoding of other packets (GOBs). The loss rate of a pixel equals the packet erasure rate.

For fixed-length packets, packet boundaries are rarely GOB or MB boundaries. Thus, when one packet is lost, the decoder will be unable to interpret the start of the next one. We refer to this as loss of synchronization. As packet loss causes bits in the next (and perhaps subsequent) packets to be lost, the loss rate of pixels exceeds the packet erasure rate due to loss of synchronization. We propose two methods to efficiently re-synchronize: re-synchronization once per GOB, and once per packet.

In this section, we lay the groundwork for the tandem wireline/wireless channel to be presented in Section II.C. For ease of combining source coding with channel coding, our scheme will employ fixed-length packets. Since most previous work in this general area has been done with variable-length packets, and since, as will become obvious by the end of this section, fixed-length packets do not perform as well as do variable-length packets, we use this section to describe in detail the fixed-length packet system, and to compare its performance to that of a variable-packet scheme.

More specifically, in this section we concentrate on the performance of a sys-

tem employing fixed-length packets over an erasure channel, where the erasure rate is constant and known by the encoder. This model will be used to represent the wireline component of the tandem channel. In Section II.C, we will add the wireless component, and this latter component will be modelled as the concatenation of an inner RCPC coder and an outer error detection code. Thus, it too will function as a packet erasure channel to the source encoder.

II.B.1 Encoding with Re-synchronization per GOB

This method inserts re-synchronizing bits at the beginning of each GOB. Video in QCIF format contains 9×11 MBs, so there are 9 GOBs per frame. With a frame rate of 30 frames per second, and bit rate of 450 kbps, each GOB occupies about 450k / (30 $\times$ 9) bits on the average, or 1667 bits. Therefore, for packet sizes in the range of 400 to 800 bits, usually the bits corresponding to one GOB will be split into several packets.

We use the first bit of each packet to tell whether there is a new GOB in this packet. If there is, the next 9 or 10 bits (depending on the packet length) indicate the new GOB's starting location. The frame/GOB number follows. In this case, an MB will not be reconstructable at the decoder if either the packet containing this MB is lost, or any of the former MBs in the same GOB are lost. If any of the former MBs are lost, the decoder will lose synchronization until the next re-sync information is received, thus the remaining MBs of the current GOB will be unreconstructable even if the decoder receives the following packets. It is possible, although unlikely, for the compressed bit stream of one MB to extend over several packets. For simplicity, we assume the decoder

loses the whole MB if any one of these packets is lost.

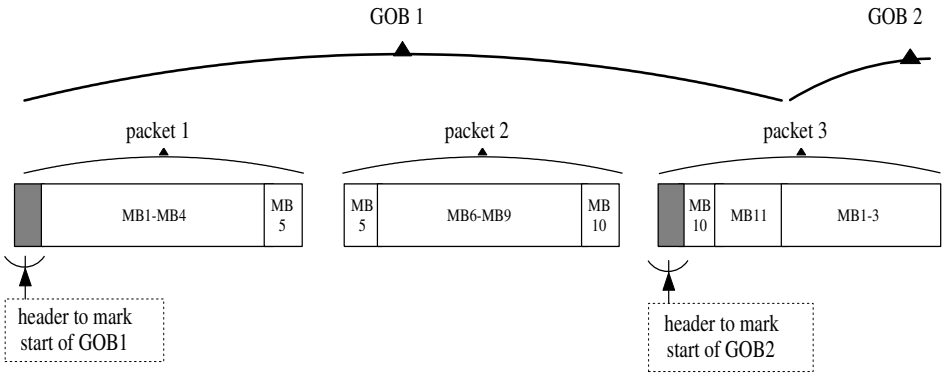


Figure II.1: An Example of Re-sync per GOB.

We count the packet number from the first packet of each GOB. Assume the current MB extends to packet m of this GOB. The probability that this MB can be reconstructed at the decoder is the probability that all m packets of this GOB are received by the decoder. This equals $(1-p)^m$, where p is the packet erasure rate. If $P_{\bar{R}}$ denotes the probability that an MB cannot be reconstructed at the decoder, we have $P_{\bar{R}} = P_{\bar{R}}(m) = 1-(1-p)^m$. For example, in Fig. II.1, for GOB1, $m = 1$ for MB1 to MB4, $m = 2$ for MB5 to MB9, and $m = 3$ for MB10 and MB11. For MB10 we have $P_{\bar{R}} = 1-(1-p)^3$.

When an MB is lost, the decoder uses a temporal concealment method. The three nearest MBs above the lost MB are denoted A, B, C from left to right. Their motion vectors (MVs) define the substitute motion vector (SMV), where the SMV indicates which MB in the previous frame will be used for concealment. We assume, if any of A, B, and C were intra-coded, that its MV=$(0,0)$. First, if MB A is lost, then so are B and C, and we set SMV=$(0,0)$. If the decoder knows A, but not B and C, we set the SMV equal to the MV of A. If both A and B survive, but not C, we set the SMV equal

to the MV of B. Lastly, if the decoder has all of A, B and C, we set the SMV equal to their median MV. When the current MB belongs to the top GOB of this frame, we set SMV=$(0, 0)$, and if the lost MB is on the side of the frame, we use the MV of the MB directly above.

We are ready to derive the expected decoder distortion per pixel for this case. Using the notation from [6], f_n denotes original frame n, which is compressed and reconstructed at the encoder as $\hat{f}_n$ (only quantization error is considered). The (possibly error-concealed) reconstruction at the receiver is denoted by $\tilde{f}_n$ (including quantization error, error propagation, packet loss and concealment distortion). The encoder does not know $\tilde{f}_n$, and treats it as a random variable.

Let f_n^i denote the original value of pixel i in frame n, and let $\hat{f}_n^i$ denote its encoder reconstruction. The reconstructed value at the decoder, possibly after error concealment, is denoted by $\tilde{f}_n^i$. The expected distortion for pixel i is

$$d_n^i = E\{(f_n^i - \tilde{f}_n^i)^2\} = (f_n^i)^2 - 2f_n^i E\{\tilde{f}_n^i\} + E\{(\tilde{f}_n^i)^2\} \tag{II.1}$$

Calculation of d_n^i requires the first and second moments of the random variable of the estimated image sequence $\tilde{f}_n^i$. To compute these, recursion functions are developed in [6], in which it is necessary to separate out the cases of intra- and inter-coded MBs. Here, since we use a modified pixel loss rate and a modified concealment method for fixed-length packets, the recursion formulas must be modified.

For each MB and for each mode selection and quantization step, we determine the packet number m for the current MB and $P_{\bar{R}} = 1 - (1 - p)^m$. A, B, C are the three

nearest MBs above this MB from left to right. We define some probabilities as follows: $P_{\bar{A}} = Pr(A\,lost)$, and $P_A = Pr(A\,received) = 1 - P_{\bar{A}}$, where "lost" means not reconstructable at the decoder and "received" means reconstructable. We also define $P_{\bar{B}|A} = Pr(B\ lost \mid A\ received)$, and $P_{\bar{C}|AB} = Pr(C\ lost \mid A\ received\ and\ B\ received)$. Similarly, $P_{A\bar{B}} = Pr(A\,received\,and\,B\,lost) = P_A P_{\bar{B}|A}$, $P_{ABC} = P_A(1 - P_{\bar{B}|A})(1 - P_{\bar{C}|AB})$, and $P_{AB\bar{C}} = P_A(1 - P_{\bar{B}|A})P_{\bar{C}|AB}$. We obtain:

$$P_{\bar{A}} = 1 - (1-p)^{m_A} \tag{II.2}$$

$$P_{\bar{B}|A} = 1 - (1-p)^{l_B} \tag{II.3}$$

$$P_{\bar{C}|AB} = P_{\bar{C}|B} = 1 - (1-p)^{l_C} \tag{II.4}$$

where m_A is the number of packets that A spans from the beginning of its GOB, l_B is the number of packets that B spans beyond the end of the packet with A, and l_C is the number of packets that C spans beyond the end of the packet with B. Note that since we assume p is known at the encoder, the probabilities required in (II.2) to (II.4) will be computed and stored at the time the MBs are encoded.

Let $k1$, $k2$ and $k3$ correspond to the pixels in the previous frame that are used to conceal pixel i, using the MV of A, B and C respectively, and let $k4$ correspond to the pixel for concealment using the median of the MVs of these three MBs. For an intra-coded MB, $\tilde{f}_n^i = \hat{f}_n^i$ with probability $1 - P_{\bar{R}}$. If the current packet is lost, and if A is also lost (with probability $P_{\bar{A}}$), so are B and C, then $\tilde{f}_n^i = \hat{f}_{n-1}^i$ because the SMV is set to $(0,0)$. Given A is received (with probability $1 - P_{\bar{A}}$), if B is lost and so is C, then

$\tilde{f}_n^i = \hat{f}_{n-1}^{k1}$; if B is received but C is lost, then $\tilde{f}_n^i = \hat{f}_{n-1}^{k2}$; lastly, if both B and C are received, $\tilde{f}_n^i = \hat{f}_{n-1}^{k4}$. Thus, the two moments for a pixel in an intra-coded MB are given by

$$E\{\tilde{f}_n^i\} = (1 - P_{\bar{R}})\hat{f}_n^i + P_{\bar{R}} \Big(P_{\bar{A}} E\{\tilde{f}_{n-1}^i\} + P_{A\bar{B}} E\{\tilde{f}_{n-1}^{k1}\} + P_{AB\bar{C}} E\{\tilde{f}_{n-1}^{k2}\} + P_{ABC} E\{\tilde{f}_{n-1}^{k4}\} \Big) \quad \text{(II.5)}$$

$$E\{(\tilde{f}_n^i)^2\} = (1 - P_{\bar{R}})(\hat{f}_n^i)^2 + P_{\bar{R}} \Big(P_{\bar{A}} E\{(\tilde{f}_{n-1}^i)^2\} + P_{A\bar{B}} E\{(\tilde{f}_{n-1}^{k1})^2\} + P_{AB\bar{C}} E\{(\tilde{f}_{n-1}^{k2})^2\} + P_{ABC} E\{(\tilde{f}_{n-1}^{k4})^2\} \Big) \quad \text{(II.6)}$$

For an inter-coded MB, assume the true MV of current pixel i is predicted from pixel j in the previous frame. Thus, the encoder prediction of this pixel is $\hat{f}_{n-1}^j$. The prediction error, e_n^i, is compressed and the quantized residue is $\hat{e}_n^i$. So the encoder reconstruction is: $\hat{f}_n^i = \hat{f}_{n-1}^j + \hat{e}_n^i$. The encoder transmits $\hat{e}_n^i$ and the MV. If received, the decoder knows $\hat{e}_n^i$ and the MV, but must use its own reconstruction of pixel j in the previous frame, $\tilde{f}_{n-1}^j$, which may differ from the encoder value $\hat{f}_{n-1}^j$. Thus, the decoder reconstruction of pixel i is given by: $\tilde{f}_n^i = \tilde{f}_{n-1}^j + \hat{e}_n^i$. The moments of $\tilde{f}_n^i$ for a pixel in an inter-coded MB are given by

$$E\{\tilde{f}_n^i\} = (1 - P_{\bar{R}}) \Big(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\} \Big) + P_{\bar{R}} \Big(P_{\bar{A}} E\{\tilde{f}_{n-1}^i\} + P_{A\bar{B}} E\{\tilde{f}_{n-1}^{k1}\} + P_{AB\bar{C}} E\{\tilde{f}_{n-1}^{k2}\} + P_{ABC} E\{\tilde{f}_{n-1}^{k4}\} \Big) \quad \text{(II.7)}$$

$$E\{(\tilde{f}_n^i)^2\} = (1 - P_{\bar{R}}) \Big((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} + E\{(\tilde{f}_{n-1}^j)^2\} \Big) + P_{\bar{R}} \Big(P_{\bar{A}} E\{(\tilde{f}_{n-1}^i)^2\} + P_{A\bar{B}} E\{(\tilde{f}_{n-1}^{k1})^2\}$$

$$+P_{AB\bar{C}}\ E\{(\tilde{f}_{n-1}^{k2})^2\} + P_{ABC}E\{(\tilde{f}_{n-1}^{k4})^2\}\Big) \qquad \text{(II.8)}$$

Lastly, since the first frame must be intra-coded, and we also assume the first frame is not lost, the initial conditions of the recursion are given as: $E\{\tilde{f}_1^i\} = \hat{f}_1^i$ and $E\{(\tilde{f}_1^i)^2\} = (\hat{f}_1^i)^2$. These recursions are performed at the encoder to calculate the expected distortion at the decoder. The encoder uses this to optimally choose the coding mode for each MB.

II.B.2 Encoding with Re-synchronization per Packet

For re-sync per packet, we insert a header at the front of each packet, telling the location (within the packet) of the beginning of the first MB and its frame/GOB/MB number. All zero location bits are used in the very unlikely case that a packet does not contain the beginning of any MB. A typical illustration is given in Fig. II.2. Now, an MB can be reconstructed at the decoder if and only if all packets that contain this MB are received. So we count the number m of packets that include this MB. The probability that an MB cannot be reconstructed at the decoder is $P_{\bar{R}} = P_{\bar{R}}(m) = 1 - (1-p)^m$. Because usually the compressed bit stream corresponding to one MB is much smaller than the fixed packet length, m usually equals 1 or 2. For example, in Fig. II.2, for GOB1, $m = 1$ for all MBs except MB5 and MB10 for which $m = 2$.

The concealment method also needs to be modified. Denote the three nearest MBs above the current decoding MB as A, B and C, from left to right. This time, loss of A does not necessarily mean loss of B or C. With re-sync per packet, it is possible that A and C are received but B is lost, although this is very unlikely because it means B

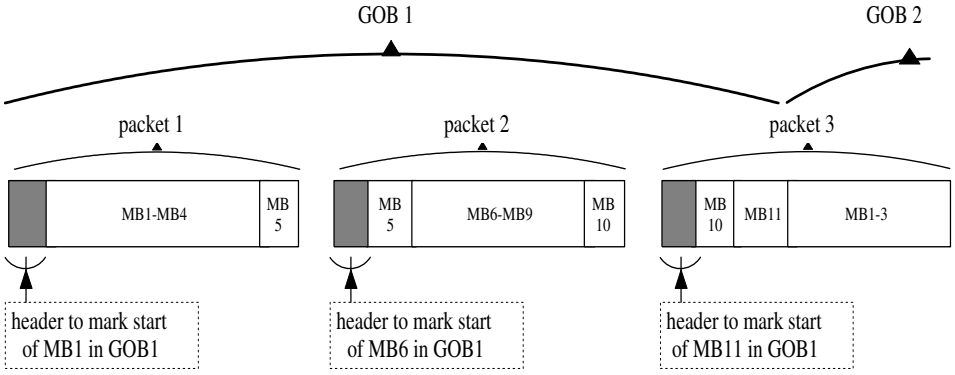


Figure II.2: An Example of Re-sync per Packet.

occupies more than one packet. For this situation, if only one of A or C is inter-coded, we set the SMV equal to the MV of the inter-coded one; if both are inter-coded, we use the MV with smaller value. Let $k5$ denote the pixel used for concealment under this situation. We summarize all the situations, the pixels used to conceal, and the corresponding probabilities, in Table II.1. For example, the first line means A, B and C are all lost, we use pixel i in the previous frame for the concealment (i.e., SMV=(0,0)), and the probability corresponding to this situation is $P_{\bar{A}\bar{B}\bar{C}}$. Also, a modified treatment is needed for special cases when the MBs are on the boundaries of a frame.

Table II.1: The Concealment Method for Different Situations

Situation	Pixel	Corresponding Probability
$\bar{A}\bar{B}\bar{C}$	i	$P_{\bar{A}\bar{B}\bar{C}} = P_{\bar{A}} P_{\bar{B}\|\bar{A}} P_{\bar{C}\|\bar{A}\bar{B}}$
$\bar{A}\bar{B}C$	$k3$	$P_{\bar{A}\bar{B}C} = P_{\bar{A}} P_{\bar{B}\|\bar{A}} (1 - P_{\bar{C}\|\bar{A}\bar{B}})$
$\bar{A}BC or \bar{A}B\bar{C}$	$k2$	$P_{\bar{A}B} = P_{\bar{A}} (1 - P_{\bar{B}\|\bar{A}})$
$AB\bar{C}$	$k2$	$P_{AB\bar{C}} = (1 - P_{\bar{A}})(1 - P_{\bar{B}\|A}) P_{\bar{C}\|AB}$
ABC	$k4$	$P_{ABC} = (1 - P_{\bar{A}})(1 - P_{\bar{B}\|A})(1 - P_{\bar{C}\|AB})$
$A\bar{B}\bar{C}$	$k1$	$P_{A\bar{B}\bar{C}} = (1 - P_{\bar{A}}) P_{\bar{B}\|A} P_{\bar{C}\|A\bar{B}}$
$A\bar{B}C$	$k5$	$P_{A\bar{B}C} = (1 - P_{\bar{A}}) P_{\bar{B}\|A} (1 - P_{\bar{C}\|A\bar{B}})$

Equations (II.2), (II.3) and (II.4) are still valid to compute $P_{\bar{A}}$, $P_{\bar{B}|A}$ and $P_{\bar{C}|AB}$, respectively, for re-sync per packet, except that here m_A means the number of packets that include A. The parameters l_B and l_C have the same definitions as before, e.g., l_B is the number of packets that B spans beyond the end of the packet with A. If d is the event that the packet shared by A and B is received at the decoder, $\bar{d}$ means this packet is lost. As illustrated in Fig. II.3, $P_{\bar{d}} = 0$ if and only if the end of A happens to be the boundary of a packet, and thus the packet shared by A and B does not exist (this situation is very unlikely), otherwise $P_{\bar{d}} = p$. Also $P_{\bar{A}\bar{d}} = P_{\bar{d}}P_{\bar{A}|\bar{d}} = P_{\bar{d}} \times 1 = P_{\bar{d}}$. Then, we can compute $P_{\bar{B}|\bar{A}}$ as follows:

$$\begin{aligned} P_{\bar{B}|\bar{A}} &= P_{\bar{d}|\bar{A}}P_{\bar{B}|\bar{A}\bar{d}} + P_{d|\bar{A}}P_{\bar{B}|\bar{A}d} \\ &= P_{\bar{d}|\bar{A}} \times 1 + (1 - P_{\bar{d}|\bar{A}})(1 - (1-p)^{l_B}) \\ &= \frac{P_{\bar{d}}}{P_{\bar{A}}} + (1 - \frac{P_{\bar{d}}}{P_{\bar{A}}})(1 - (1-p)^{l_B}) \end{aligned} \tag{II.9}$$

Similarly, to compute $P_{\bar{C}|\bar{A}\bar{B}}$ and $P_{\bar{C}|A\bar{B}}$, we define the event e that the packet shared by B and C is received at the decoder, and $P_{\bar{e}} = p$ except if the end of B happens to be the boundary of a packet, in which case $P_{\bar{e}} = 0$. Then,

$$\begin{aligned} P_{\bar{A}\bar{B}\bar{e}} &= P_{\bar{A}\bar{e}}P_{\bar{B}|\bar{A}\bar{e}} \;=\; P_{\bar{A}\bar{e}} \;=\; P_{\bar{e}}P_{\bar{A}|\bar{e}} \\ &= \begin{cases} P_{\bar{e}}\ , & A \text{ shares the same packet with } C \\ P_{\bar{A}}P_{\bar{e}}\ , & \text{No common packet for } A \text{ and } C \end{cases} \\ P_{A\bar{B}\bar{e}} &= P_{A\bar{e}} = P_{\bar{e}}P_{A|\bar{e}} \end{aligned} \tag{II.10}$$

$$= \begin{cases} 0\ , & A \text{ shares the same packet with } C \\ P_A P_{\bar{e}}\ , & \text{No common packet for } A \text{ and } C \end{cases} \tag{II.11}$$

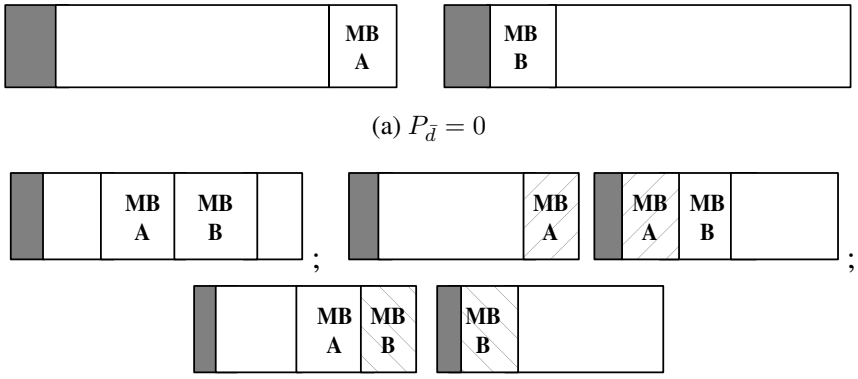
At last, we have the following conditional probability:

$$\begin{aligned} P_{\bar{C}|\bar{A}\bar{B}} &= P_{\bar{e}|\bar{A}\bar{B}} P_{\bar{C}|\bar{A}\bar{B}\bar{e}} + P_{e|\bar{A}\bar{B}} P_{\bar{C}|\bar{A}\bar{B}e} \\ &= \frac{P_{\bar{A}\bar{B}\bar{e}}}{P_{\bar{A}\bar{B}}} \times 1 + (1 - \frac{P_{\bar{A}\bar{B}\bar{e}}}{P_{\bar{A}\bar{B}}})(1 - (1-p)^{l_C}) \\ &= \begin{cases} \frac{P_{\bar{e}}}{P_{\bar{A}} P_{\bar{B}|\bar{A}}} + (1 - \frac{P_{\bar{e}}}{P_{\bar{A}} P_{\bar{B}|\bar{A}}})(1 - (1-p)^{l_C})\ , \\ \qquad A \text{ shares the same packet with } C \\ \frac{P_{\bar{e}}}{P_{\bar{B}|\bar{A}}} + (1 - \frac{P_{\bar{e}}}{P_{\bar{B}|\bar{A}}})(1 - (1-p)^{l_C})\ , \\ \qquad \text{No common packet for } A \text{ and } C \end{cases} \end{aligned} \tag{II.12}$$

and we can calculate $P_{\bar{C}|A\bar{B}}$ in a similar fashion. With these, we compute the probability terms in Table II.1.

The expected distortion for pixel i is given by (III.14). For each MB and for each mode selection and quantization step, we first calculate the loss probability $P_{\bar{R}} = 1 - (1-p)^m$, where m is the number of packets that contain this MB. Then, for an intra-coded MB, $\tilde{f}_n^i = \hat{f}_n^i$ with probability $1 - P_{\bar{R}}$, corresponding to correct receipt of the MB. The recommended concealment method is used if the current MB is lost. The two moments for a pixel in an intra-coded MB are given by

$$E\{\tilde{f}_n^i\} = (1 - P_{\bar{R}})\hat{f}_n^i + P_{\bar{R}} \Big(P_{\bar{A}\bar{B}\bar{C}} E\{\tilde{f}_{n-1}^i\} + P_{\bar{A}\bar{B}C} E\{\tilde{f}_{n-1}^{k3}\} + (P_{\bar{A}B} +$$



(a) $P_{\bar{d}} = 0$

(b) $P_{\bar{d}} = p$

Figure II.3: Illustration of event d and the corresponding probability $P_{\bar{d}}$. (a) The end of A happens to be the boundary of a packet, so $P_{\bar{d}} = 0$; (b) Three situations in which A and B share (at least partly share) a same packet, so $P_{\bar{d}} = p$.

$$P_{AB\bar{C}})E\{\tilde{f}_{n-1}^{k2}\} + P_{ABC}E\{\tilde{f}_{n-1}^{k4}\} + P_{A\bar{B}\bar{C}}E\{\tilde{f}_{n-1}^{k1}\} + P_{A\bar{B}C}E\{\tilde{f}_{n-1}^{k5}\}\Big) \quad \text{(II.13)}$$

$$E\{(\tilde{f}_n^i)^2\} = (1-P_{\bar{R}})(\hat{f}_n^i)^2 + P_{\bar{R}}\Big(P_{\bar{A}\bar{B}\bar{C}}E\{(\tilde{f}_{n-1}^i)^2\} + P_{\bar{A}\bar{B}C}E\{(\tilde{f}_{n-1}^{k3})^2\}$$
$$+(P_{\bar{A}B} + P_{AB\bar{C}})E\{(\tilde{f}_{n-1}^{k2})^2\} + P_{ABC}E\{(\tilde{f}_{n-1}^{k4})^2\} + P_{A\bar{B}\bar{C}}$$
$$E\{(\tilde{f}_{n-1}^{k1})^2\} + P_{A\bar{B}C}E\{(\tilde{f}_{n-1}^{k5})^2\}\Big) \quad \text{(II.14)}$$

Similarly, for an inter-coded MB, assume the true MV of current pixel i is predicted from pixel j in the previous frame. The first and second moments of $\tilde{f}_n^i$ for a pixel in an inter-coded MB are given by

$$E\{\tilde{f}_n^i\} = (1-P_{\bar{R}})\Big(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\Big) + P_{\bar{R}}\Big(P_{\bar{A}\bar{B}\bar{C}}E\{\tilde{f}_{n-1}^i\} + P_{\bar{A}\bar{B}C}$$
$$E\{\tilde{f}_{n-1}^{k3}\} + (P_{\bar{A}B} + P_{AB\bar{C}})E\{\tilde{f}_{n-1}^{k2}\} + P_{ABC}E\{\tilde{f}_{n-1}^{k4}\} + P_{A\bar{B}\bar{C}}$$
$$E\{\tilde{f}_{n-1}^{k1}\} + P_{A\bar{B}C}E\{\tilde{f}_{n-1}^{k5}\}\Big) \quad \text{(II.15)}$$

$$E\{(\tilde{f}_n^i)^2\} = (1-P_{\bar{R}})\Big((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} + E\{(\tilde{f}_{n-1}^j)^2\}\Big) + P_{\bar{R}}$$

$$\Big(P_{\bar{A}\bar{B}\bar{C}}E\{(\tilde{f}_{n-1}^{i})^2\} + P_{\bar{A}\bar{B}C}E\{(\tilde{f}_{n-1}^{k3})^2\} + (P_{\bar{A}B} + P_{AB\bar{C}})E\{(\tilde{f}_{n-1}^{k2})^2\}$$
$$+P_{ABC}\ E\{(\tilde{f}_{n-1}^{k4})^2\} + P_{A\bar{B}\bar{C}}E\{(\tilde{f}_{n-1}^{k1})^2\} + P_{A\bar{B}C}E\{(\tilde{f}_{n-1}^{k5})^2\}\Big) \quad \text{(II.16)}$$

II.B.3 Rate-Distortion Framework

We take into account the expected distortion due to both compression and transmission errors for optimal mode switching. The distortion is computed recursively by the formulas given above for the two possible re-synchronization schemes separately. We incorporate this overall expected distortion within the rate-distortion framework at the encoder, to optimally switch between intra- and inter-coding on a macroblock basis. The goal is to minimize the total distortion D subject to a bit rate constraint R.

This problem is an unconstrained Lagrangian minimization, where the algorithm minimizes the total cost $J = D + \lambda R$. Individual MB contributions to this cost are additive, so it can be minimized on a macroblock basis [7]. Therefore, the encoding mode and the quantization parameter (QP) for each MB are chosen by minimizing

$$\min_{(mode,QP)} J_{MB} = \min_{(mode,QP)} (D_{MB} + \lambda R_{MB}) \quad \text{(II.17)}$$

where the distortion D_{MB} is the sum of the distortion contributions of the individual pixels (d_n^i's), and d_n^i is calculated by (III.14), where the first and second moments of $\tilde{f}_n^i$ are given by (5) to (II.8) for re-sync per GOB, and by (13) to (II.16) for re-sync per packet.

Rate control is achieved by modifying λ. As in ROPE [6], we update λ per

frame via

$$\lambda_{n+1} = \lambda_n \left(1 + \alpha(\sum_{i=1}^{n} R_i \; - \; nR_S^*) \right) \tag{II.18}$$

where R_S^* is the target encoding bit rate, $\alpha = \frac{1}{5R_S^*}$, and λ_0 is set to be 70.

The coding mode and QP are chosen to minimize the Lagrangian cost. For each choice of mode and QP, the encoder computes the number of bits needed for the current MB, the reconstruction failure probability $P_{\bar{R}}$, the individual pixel distortions, and D_{MB}. The algorithm chooses the mode/step size such that D_{MB} and R_{MB} minimize J. Since QP ranges from 1 to 31, and the mode has two choices (Intra or Inter), this algorithm optimizes over 62 potential combinations.

As to the complexity of this approach, a computational burden is incurred in computing the probabilities corresponding to the different concealment scenarios and the two moments of $\tilde{f}$ for each mode choice for each pixel. For re-sync per GOB, for each pixel, the algorithm typically needs about 8 addition/multiplication operations to calculate $P_{\bar{A}}$, $P_{\bar{B}|A}$ and $P_{\bar{C}|AB}$, and about 32 addition/multiplication operations to calculate the two moments in (5) to (II.8). (Note that the identical concealment for both intra and inter coding reduces the complexity.) For re-sync per packet, for each pixel, the algorithm typically requires about 36 addition/multiplication operations to create Table II.1, and about 42 addition/multiplication operations to calculate the two moments in (13) to (II.16). This complexity is comparable to that of the original ROPE algorithm, which needs about 27 operations to calculate the two moments for each pixel [6]. Also, note that all the complexity mentioned above is incurred only at the encoder.

II.B.4 Performance Analysis and Simulation Results

We anticipate that fixed-length packets will perform worse than variable GOB-length packets. Three kinds of penalties explain this performance downgrade. *Rate penalty* comes from sending re-sync information. Re-synchronization per packet involves more re-sync bits than re-sync per GOB. For a shorter fixed packet length, re-sync bits are sent more often. *Division penalty* arises because usually bits of one GOB extend over several fixed-length packets. For example, suppose GOB1 is encoded into packets a and b, and suppose packet and MB boundaries coincide. Similarly GOB2 is encoded into packets c and d. Under the same packet erasure rate, losing one variable-length packet which contains an entire GOB, is equivalent to losing two fixed-length packets. However, losing two fixed-length packets means losing more than one GOB on the average because of sync loss. For example, if packets a and c are lost and we re-sync once per GOB, both GOBs will be entirely lost. A smaller fixed packet length entails a more severe division penalty. If we re-sync once per packet, this penalty will still exist, but will be smaller. *Boundary penalty* occurs whenever the boundary of a lost fixed-length packet is not exactly the boundary of an MB (or GOB). Suppose packet b contains a few bits of GOB2; losing packet b causes the loss of half of GOB1 and the entire GOB2 if we re-sync per GOB. It causes the loss of half of GOB1 and the first MB of GOB2 if we re-sync per packet. Losing two such packets at different points in the stream causes the loss of two GOB halves plus two additional MBs.

Thus the performance with fixed-length packets should be worse than that with variable-length packets. Re-sync per packet has higher rate penalty but much smaller

division and boundary penalties, so it should yield a better performance than re-sync per GOB. Note that we assume Internet congestion causes an equal loss probability for packets of any size.

We will also compare our scheme with the "block-weighted distortion estimate" (BWDE) [6], with the same two fixed-length packetization approaches. BWDE assumes that the current block is correctly received, while the MBs of the previous frame may be lost and concealed, thus the current block may have concealment distortion because it may be inter-coded using the previous frame. The estimate of decoder distortion is $\hat{D} = D_{q1}$ for intra mode and $\hat{D} = pD_c + (1-p)D_{q2}$ for inter mode, where D_{q1} is the quantization distortion of the current intra-coded pixel, D_c is the weighted average of the concealment distortion of the previous frame blocks that are mapped to the current MB, and D_{q2} is the quantization distortion of the residual for the current inter-coded pixel. The Lagrangian $J = \hat{D} + \alpha R$ is minimized among coding modes and QPs for each MB. Because this algorithm unrealistically assumes that the current block is always received, and because the distortion is not additive in its concealment and quantization components, performance with BWDE is expected to be worse than with modified ROPE.

In our simulation results, the system was evaluated using an H.263+ codec with standard QCIF (176×144) video sequences at frame rates of 10, 15 or 30 frames per second (fps). Various target transmission bit rates were tested ranging from 50kbps to 450kbps. A random packet loss generator was used to drop packets with variable erasure rates p. Different fixed packet lengths from 100 bits to 1000 bits were also

tested.

Fig. II.4 shows the PSNR performance versus packet erasure rate. Fig. II.4(a) is for the “Carphone” QCIF sequence at 200kbps and 30fps with packet length 400 bits. For a given distortion estimation method (ROPE or BWDE), variable-length packets outperform fixed-length packets, and re-sync per packet outperforms re-sync per GOB. For the ROPE algorithm, from $p = 5\%$ to $p = 30\%$, re-sync per fixed-length packet is about 0.2-0.4dB lower than variable-length packets, and about 1.0dB higher than re-sync per GOB. At $p = 0\%$, re-sync per packet performs slightly worse than re-sync per GOB because only rate penalty applies. For the same packing method (variable length, fixed length with re-sync per packet or per GOB), ROPE outperforms BWDE by about 2.0dB. Similar trends appear in Fig. II.4(b), which contains results for the “Container" QCIF image sequence at 100kbps and 15fps with 400-bit fixed-length packets.

Fig. II.5 shows PSNR versus transmission rate. Fig. II.5(a) is for “Carphone" at 30fps with packet length 400 bits and error rate $p = 10\%$. For the same distortion estimation method, as the transmission rate grows, the gap between variable-length packets and fixed-length with re-sync per packet is nearly constant. For ROPE, this constant is about 0.35dB. However, the gap between variable-length packets and fixed-length with re-sync per GOB increases dramatically, mostly due to the more serious division penalty as rate increases. For ROPE, it goes from 1.0dB at 100kbps up to 2.7dB at 450kbps. For the same packing method, ROPE beats BWDE by about 2.0-2.5dB, and the gap increases with rate. In Fig. II.5(b), which is for “Container” at 15fps with 400-bit fixed-length packets and $p = 5\%$, we observe similar trends.

Fig. II.6 shows PSNR versus packet length ranging from 100 bits to 1000 bits. Fig. II.6(a) is for “Carphone” at 200kbps and 30fps with packet loss rate $p = 10\%$, and Fig. II.6(b) is for “Container" at 100kbps and 15fps with $p = 5\%$. For the same distortion estimation algorithm, a larger fixed packet size leads to a smaller gap between variable-length and fixed-length packet results. Again, the ROPE algorithm yields consistent and significant gains over BWDE.

In summary, to integrate this source encoder with Forward Error Correction (FEC) to operate over a wireless/Internet channel, we change the variable-length packetization to fixed-length packetization, and modify the distortion estimation approach accordingly. In doing this, one pays three kinds of penalties. Experimental results demonstrated this PSNR downgrade of about 0.2-0.5dB. Simulation results also showed re-sync per packet outperformed re-sync per GOB.

II.C Source and Channel Coding over Wireless/Internet

The delivery of packet video over tandem Internet and wireless channels is discussed in this section. We assume the wireless channel introduces uniform random bit errors with rate P_b, and the Internet loses packets with erasure rate p. We assume P_b and p are constant and known at the encoder. In practice, this information may come from a test data sequence and tracking of channel conditions. The major resource shared between the source and channel encoders is the given target transmission rate. If the channel condition is poor (say, $P_b \geq 0.01$), more bits are needed for channel error detection and correction, thus a smaller bit rate is used for source encoding. The

system diagram is shown in Fig. II.7. In particular, the wireless component is modelled as the concatenation of an inner RCPC code and an outer error detection code, thus the tandem channel can be dynamically converted into a erasure channel for the source encoder, and the algorithm proposed in Section II.B can be easily re-used. We now discuss each component in detail.

II.C.1 The Source Encoder

The video source is encoded using the optimal inter/intra mode selection algorithm with fixed-length packets (re-sync per GOB and per packet are analyzed and compared). The mode selection algorithm was designed for a given output bit rate of the source coder and a given packet erasure rate. Here we are given instead the target transmission rate (that is, the output bit rate of the channel coder), and the wireless bit errors may increase the packet loss rate if the corrupted packets cannot be corrected and are thus discarded.

Given the bit error rate P_b, the channel coder (as discussed below) chooses a rate-compatible punctured convolutional (RCPC) code with channel code rate r from a family of RCPC codes so as to keep the probability of packet drop due to uncorrectable bit errors (p'_e) at about 1% for most of the transmission rates of interest. The packet erasure rate due to Internet congestion is p; thus, the total packet loss rate is $\hat{p} = p + p'_e - p \times p'_e \approx p + 0.01 - p \times 0.01 = 0.99p + 0.01$.

Knowing r, the transmission target rate R^* and frame rate f, as well as the fixed packet length, the source encoder determines the corresponding target source cod-

ing output bit rate R_S^*. With the target output bit rate of the source coder and the total packet loss rate $\hat{p}$, we may use the intra/inter mode selection algorithm directly as derived in Section II.B.

II.C.2 The Channel Encoder

We use a concatenated code consisting of a CRC outer coder and RCPC inner coder. That is, the grouped fixed-length y source information bits are appended with a 16-bit CRC and M zero ending bits to flush the memory and terminate the trellis decoding in the zero state. Then the $(y + 16 + M)$ bits are convolutionally encoded using a rate r RCPC coder [8].

CRCs provide error detection with low complexity and flexible block length. The optimal 16-bit CRCs for different packet lengths are proposed in [9, 10]. In particular, C_1, C_3 and C_4 [9] are typically used for packet lengths less than 151, between 151 and 257, and greater than 257 (and less than 28658 bits), respectively. All of these yield a very low probability of undetected error, typically less than 10^{-5}.

RCPC codes are a powerful extension of punctured convolutional codes [11, 12]. Here, the RCPC code is chosen adaptively to make the probability of packet drop due to uncorrectable bit error about 1%, under the given channel bit error rate P_b ($P_b \leq 0.15$) for most of the transmission rates of interest. As a practical matter, the 1% cannot be exactly achieved, and we used a rate 2/7 RCPC code when $P_b > 0.05$, a rate 2/3 RCPC code when $0.005 < P_b \leq 0.05$, a rate 8/9 RCPC code when $10^{-5} < P_b \leq 0.005$, and no channel coder is used if $P_b \leq 10^{-5}$. All of these RCPC codes have a memory

$M = 6$ and a puncturing period length 8. The details of their construction are given in Table II.2.

Table II.2: RCPC Codes Used in the System

RCPC Code Rate	**Mother Convolutional Code**			**Puncturing Table**
	Rate	Memory	Generation Matrix	
8/9	1/3	6	1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1	1 1 1 1 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
2/3	1/3	6	1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1	1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0
2/7	1/4	6	1 1 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0

To avoid an unacceptable corresponding packet loss rate, the FEC selection needs to guarantee that the bit error probability after correction is very small. Fig. II.8(a) shows the relationship between the bit error rate and the corresponding packet error rate without error correction. When bit error rate is very small ($\leq 5 \times 10^{-5}$), the packet error rate is roughly the product of the bit error rate and the fixed packet length. If the bit error rate is larger ($\geq 5 \times 10^{-4}$), the corresponding packet error rate goes up dramatically and reaches nearly 100% as the bit error rate goes to 0.02. Thus a powerful RCPC code is needed to avoid bad system degradation.

Simulations also show that it is reasonable to choose the packet drop rate due to uncorrectable bit error to be roughly 1%. Fig. II.8(b) shows the PSNR gap for different target packet drop rates, where PSNR gap (on the y-axis) refers to the average gap between the PSNR with zero packet drop rate and the PSNR under the given drop

rate over different wireless bit error rates. When the drop rate is high, the gap is large, but when the drop rate goes down to roughly 1%, the PSNR gap is very small. Returns diminish when the drop rate due to uncorrectable bit errors is pushed below 1%.

For the efficient detection of uncorrected errors, the serial list-Viterbi algorithm at the channel decoder was used with a list of 100 paths [12, 13]. The optimal path in the Viterbi decoding is chosen among those paths that satisfy the checksum equations. If at a given depth of trellis decoding, none satisfied the checksum equations, then an uncorrected error is declared and this packet is discarded. The corresponding MBs are then reconstructed from the previously received MBs using the concealment methods. Here we check 100 paths; increasing the number of paths does not necessarily improve the performance of the system, because we may reach a point where the probability of undetected errors becomes too high, and it is shown that dropping the uncorrected packet and using a proper concealment method may give a better result than using an uncorrected packet [13].

II.C.3 Performance Analysis and Simulation Results

This system was evaluated for the transmission of video over a tandem channel. The packet erasure rates tested were $p = 5\%$ and 10%, and bit error probabilities ranged from $P_b = 0$ to $P_b = 0.15$. The same error patterns were used for all algorithm versions. Again, we compare modified ROPE and BWDE distortion estimation.

Fig. II.9 shows PSNR versus bit error rate from $P_b = 0$ to $P_b = 0.15$. Fig. II.9(a) is for "Carphone" at 400kbps and 30fps with packet length 400 bits, and

$p = 10\%$. Fig. II.9(b) is for "Container" at 150kbps and 15fps with packet length 400 bits, and $p = 5\%$. Results are consistent with our predictions. With the same distortion estimation method (ROPE or BWDE), re-sync per packet yields better performance than does re-sync per GOB; with the same fixed packetization method, modified ROPE outperforms BWDE. Table II.C.3 shows parameters for the simulation for modified ROPE with re-sync per packet. Note that, as the bit error rate increases, a lower rate channel coder is used, and so the bit rate for source coding decreases. The estimates of the total packet loss rate at the encoder are close to the actual packet loss rate found at the decoder, consistent with our goal that the packet loss due to uncorrectable bit error is about 1%.

Table II.3: Parameters of Fig. II.9 for Modified ROPE and Re-sync per Packet

Carphone QCIF Re-synchronization per Packet						
Bit Error Rate	10^{-6}	10^{-5}	10^{-4}	10^{-3}	0.01	0.10
Source Bit Rate (kbps)	390	390	328	328	246	106
Assumed Packet Loss Rate (%)	10.9					
Total Packet Loss Rate (%) (Found at the Decoder)	10.04	10.35	10.06	10.58	10.89	11.52
PSNR (dB)	32.73	32.61	32.28	32.20	31.02	28.75

Container QCIF Re-synchronization per Packet						
Bit Error Rate	10^{-6}	10^{-5}	10^{-4}	10^{-3}	0.01	0.10
Source Bit Rate (kbps)	132	132	117	117	88	39
Assumed Packet Loss Rate (%)	5.95					
Total Packet Loss Rate (%) (Found at the Decoder)	5.04	5.38	5.06	5.61	5.93	6.61
PSNR (dB)	35.77	35.76	35.60	35.57	34.15	32.72

Fig. II.10(a) shows PSNR versus target transmission rate, and Fig. II.10(b) shows PSNR versus time (frame number) at 300kbps. The image sequence is "Salesman" at 10fps with packet length 800 bits, $p = 10\%$ and $P_b = 0.01$. Again re-sync per packet yields a much better performance than re-sync per GOB, and modified ROPE outperforms BWDE.

We also compare our system with a recent system [3] which uses a H.263+ source coder, and a concatenated FEC scheme employing interlaced Reed-Solomon (RS) codes and RCPC codes to protect the video data from packet loss and bit errors, respectively. We compare the performance of our system with the results given in Fig. 6 in [3], where the comparison system is operated over a wired IP and a wireless Rician fading channel with parameter K. Because sufficient interleaving is assumed to randomize the burst errors in [3], the SNR of the fade can be translated to a bit error rate as follows:

$$\begin{aligned} P_e &= \int_r P(e|r) f(r) dr \\ &= \int_0^\infty \varphi\left(-x\sqrt{SNR}\right)\left(xe^{-K+\frac{x^2}{2}} I_0(x\sqrt{2K}\,)\right) dx \end{aligned} \tag{II.19}$$

where $\varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$ and $I_0(x) = \sum_{n=0}^{\infty} \frac{(-1)^n (x/2)^{2n}}{(n!)^2} = \frac{1}{\pi} \int_0^\pi cos(xsin\theta) d\theta$, which are the cumulative Gaussian distribution function and the modified Bessel function of order zero, respectively. The simulation results are for "Susie" at 128kbps and 7.5fps. The comparison system generates 9 packets per frame, with the fixed packet length 128k/(7.5 $\times$ 9)=1896 bits, and our system is operated with an 800-bit packet. The results are shown in Fig. II.11. Over most bit error rates, our system outperforms

the comparison system by about 0.4dB. The comparison system outperforms ours in a small interval, perhaps because it selects among a larger set of RS and RCPC codes.

The sensitivity to mismatched channel status is examined in Fig. II.12, where the channel status used at the transmitter for the optimization mismatches the actual channel status in the network. The figures are for "Carphone" with transmission rate 400kbps and packet length 400 bits, with re-sync per packet. Fig. II.12(a) is for performance of mismatched bit error rate under a correct packet erasure rate estimate. The horizontal axis is the actual channel bit error rate; each curve represents the performance of the system that persists in using a particular rate RCPC code (so it is mismatched out of the correct bit error range). Performance drops dramatically when the actual bit error rate is higher than the estimate. The upper bound curve is the performance of a properly matched system. Fig. II.12(b) illustrates the mismatched packet erasure rate under a matched 0.001 wireless bit error rate. Again, each curve represents the performance where a particular packet erasure rate is assumed. At each actual channel status, the matched estimate yields the best performance, and poorer performance goes along with increasing mismatch. The upper bound curve shows performance of the matched system.

In many applications, both bit errors and packet erasures occur in bursts, and the Gilbert-Elliot model is good for capturing bursty loss patterns. A two-state Gilbert-Elliot model with the states named *Good* and *Bad* is illustrated in Fig. II.13. Note that the state transition characteristics are completely determined by the values P_{GG} and P_{BB}, where, for example, P_{GG} is the probability that the next state is *Good*, given the

current state is *Good*. Then the mean time durations (measured in number of steps) that the channel is in the *Good* and *Bad* states are $T_G = 1/(1-P_{GG})$ and $T_B = 1/(1-P_{BB})$, respectively. In Fig. II.14, we compare the performance of our system when used over a constant random channel to that when the channel is bursty. The top curve is the system performance for a constant channel with $p = 10\%$ and $P_b = 0.01$, which is the same as the top curve in Fig. II.10(a). The lower curve is the system performance for a channel with a constant $P_b = 0.01$, while the packet erasures are determined from a Gilbert-Elliot model utilizing the limiting per_state error probabilities of one and zero for each packet. We chose $P_{GG} = 0.9$ and $P_{BB} = 0.1$, thus $T_G = 10$, $T_B = 10/9$. The overall erasure rate over a long period of time, which is equal to the percentage of time that the channel is in the *Bad* state, is also 1%. Note that the performance degrades when the channel follows the Gilbert-Elliot model, because of the mismatch of the channel status, that is, the transmitter assumes the packet erasure is a constant 10%, while actually there are two states of erasure rate 0% and 100% with a certain coherent time.

II.D Performance over Time Varying Channels with Feedback

In the previous sections, we assumed the channel conditions (packet erasure rate and bit error rate) are known in advance by the transmitter, and stay constant. We also assumed that there is no feedback information from the receiver. However, real channels are usually time varying, and a backward channel from the receiver to the transmitter is available in many applications. Through this feedback channel, the receiver can signal to the transmitter its estimate of the current channel conditions and

the actual packet loss rate found at the decoder, and the transmitter can adapt its encoding choices accordingly. What is more, as indicated in [6], the backward channel can also specify lost packets via acknowledgement (ACK) or negative-acknowledgement (NACK), to obtain additional gain in the performance. We will extend our system to time varying channels and feedback.

For convenience, we assume the wireless bit error rate P_b and the packet erasure rate p are constant for the packets of the same frame, and they vary from frame to frame. We assume the transmitter knows the channel status correctly for the first frame. After that, it needs feedback to track channel variation. We also assume that the feedback link is error free.

II.D.1 Feedback of Channel Conditions

Here, we will not include channel estimation; we assume the decoder can estimate channel conditions correctly and instantaneously, and the transmitter will use this error free information, possibly with some delay, to choose intra/inter modes or adjust channel code rates.

If the feedback information arrives at the transmitter with negligible delay, the bit error rate and packet erasure rate used at the transmitter match actual channel conditions, so it should yield the upper limit of the performance of our system for the given channel model. In practice, there usually exists some feedback delay due to propagation time or buffering time. We assume a fixed feedback delay d. The transmitter knows the exact channel conditions of the $(n-d)$-th frame as it encodes the n-th frame. At that

time, all frames before the n-th are already transmitted.

Due to the memory in the channel, a natural guess is that the erasure rate and bit error rate seen by the packets of the n-th frame are the same as those seen by the $(n-d)$-th frame, as that is the newest feedback information obtained by the transmitter. With this information, the transmitter first selects the proper RCPC code according to the bit error rate, and then the modified ROPE algorithm does distortion estimation and selects the mode and quantization parameter (QP) that minimize the Lagrange (II.17).

When the ROPE algorithm estimates distortion for the n-th frame, it has the first and second moments of the expected distortions for each pixel in frame $(n-1)$. These are used in the recursive formulas to compute the estimates for frame n. With feedback, the transmitter knows the channel conditions for the $(n-d)$-th frame and its packet loss rate experienced at the decoder. Although the transmitter cannot use this information to re-encode frames $(n-d)$ through $(n-1)$, because they are already sent out, it can use the feedback information to refine the distortion estimate for these frames and therefore for the n-th frame as well.

The estimation refinement starts with the $(n-d)$-th frame, because now the transmitter has the exact channel conditions for this frame. For purposes of the recursive computations, it also temporarily assumes that the channel conditions stay constant at the conditions of the $(n-d)$-th frame up to the n-th frame. From frame $(n-d)$ up to and including frame $(n-1)$, the source transmitter recursively recomputes the first and second moments for each pixel according to this newest known packet loss rate. For frame n, the transmitter estimates distortion based on these refined estimates $E\{\tilde{f}_{n-1}^{i}\}$ and

$E\{(\tilde{f}^i_{n-1})^2\}$, and selects a mode. The refined computation prevents the accumulation of estimation error.

This refined estimation algorithm should yield better performance than the simple estimation method. The refined estimation method adjusts the estimates at each time interval, so only the moments of $\tilde{f}$ in the last $(d-1)$ frames may be incorrect because the transmitter does not yet have feedback information for these frames.

The computational complexity is higher than for the simple estimation case, because we need to re-compute the moments of the previous d frames. For d in the range of 0–20 (equivalently, 0–600ms for 30fps, and 0–200ms for 10fps), this complexity is modest. Also, the refined estimation algorithm needs more storage to store the moments of the $(n-1)$-th frame $E\{\tilde{f}^i_{n-1}\}$ and $E\{(\tilde{f}^i_{n-1})^2\}$. As in the simple estimation method, it needs to store the moments of the $(n-d-1)$-th frame $E\{\tilde{f}^i_{n-d-1}\}$ and $E\{(\tilde{f}^i_{n-d-1})^2\}$ and all the intra/inter mode selections and quantization step choices of each MB from the $(n-d)$-th frame through the current frame.

II.D.2 Feedback of ACK/NACK

Another kind of feedback information is to specify lost packets via ACK or NACK. This type of feedback information was used in [6], where the refined distortion estimation was proposed and shown to outperform simple estimation. For the packet erasure channel, the packet erasure rate of the channel can be inferred from the ACK/NACK feedback; while for the wireless or the tandem channels, the channel conditions cannot be inferred from the packet drop rate after the channel decoder, since different FEC is

used for different wireless channel conditions.

For a fixed feedback delay d, the transmitter can now exactly calculate the decoder reconstruction up to frame $(n-d)$, but the packet loss history from frame $(n-d+1)$ to frame n is still unknown. To use the feedback information, as shown in Section V of [6], the transmitter will recompute exactly the $(n-d)$th frame of decoder reconstruction by employing error concealment whenever the packets were lost; then the reconstructed frame is used to initialize the recursion formulas to estimate the distortion from frame $(n-d+1)$ up to frame n; at last, the refined estimates $E\{\tilde{f}_n^i\}$ and $E\{(\tilde{f}_n^i)^2\}$ are incorporated into the R-D optimization mode selection.

For the tandem varying channel, sending back *both* the channel conditions and the ACK/NACK information can result in further improvement of the performance, by decreasing the mismatch loss from tracking the channel variation, and employing the exact error concealment from the ACK/NACK information together.

Again, the computational complexity involved in updating all the intermediate frames may be a problem, and the performance degrades as the delay increases. When the delay is large, we can ignore the feedback information to reduce complexity with a relatively small penalty in performance.

II.D.3 Performance Analysis and Simulation Results

As before, the source encoder is implemented by modifying the H.263+ coder. The system is operated over a time varying tandem channel. The source is 300 frames from "Carphone" at 30fps with packet length 400 bits. The feedback performance is

compared with delays of zero, 10 and 20 frames.

In Fig. II.16, a channel with $P_b = 0$ and varying packet loss rate is considered. The variation of p over time (frame) is shown in Fig. II.15(a) in the range from 5% to 20%. Fig. II.16(a) shows the system performance with re-sync per packet over different target transmission rates. The top curve is for the instantaneous feedback of ACK/NACK; note that, for a pure packet erasure channel, the packet erasure rate can be inferred from the ACK/NACK information, so this curve actually corresponds to the use of both instantaneous ACK/NACK feedback information and channel condition feedback. The bottom curve is for the system without feedback; the encoder assumes a packet erasure rate equal to the average 12.5%. The other curves on the figure corresponds to feedback of only channel conditions with delay of 0, 10 and 20 frames; for the delayed feedback, simple and refined estimation are also compared. It is shown that the refined estimation method outperforms the simple estimation by more than 1dB, and the feedback of the additional information of ACK/NACK can yield a further gain in performane. In Fig. II.16(b), we show the PSNR of each frame for the system with re-sync per packet at the target transmission rate of 400 kbps, for the feedback of channel conditions with both instantaneous and 20 frame delay. The PSNR with refined feedback almost achieves the upper limit for instantaneous feedback, because the error model of this channel is piecewise constant with period longer than the fixed feedback time. The PSNR with simple estimation results in a larger gap.

Fig. II.17 shows the performance over a channel with varying bit errors, and packet erasure rate $p = 0$. The variation of P_b is shown in Fig. II.15(b). We chose

a smoothly varying curve so that it plausibly could represent a realization of a channel with memory. The performance versus transmission bit rate for various combinations of instantaneous feedback of channel conditions and of ACK/NACK are shown in Fig. II.17(a), for the system with re-sync per packet. For the case of no feedback, the transmitter assumes the channel bit error rate is always 0.01, thus keeps using the RCPC code with rate $2/3$. It is shown that combined feedback yields better performance than the use of only one type of feedback. In Fig. II.17(b) we show the PSNR versus transmission bit rate for 20-frame delayed feedback of both types of information. For feedback of channel conditions, the refined and simple estimation methods are compared. Again, combined feedback results in best performance, and refined estimation outperforms simple estimation. Note that the performance of the simple estimation scheme with feedback is worse than that of choosing an appropriate "average" channel condition in the absence of feedback.

Fig. II.18 shows the performance over a tandem channel model with time varying bit error rate and time varying packet erasure rate, which accounts for the conditions illustrated in Fig. II.15(a) and Fig. II.15(b). Fig. II.18(a) and Fig. II.18(b) show the PSNR performance versus transmission bit rate of various combinations of feedback information, in conjunction with either instantaneous feedback or 20-frame delayed feedback, respectively. We observe similar trends here; once again the advantage of combined feedback information and refined estimation is evident.

In summary, simulation results showed that combined feedback of both channel conditions and ACK/NACK information improve system performance compared to

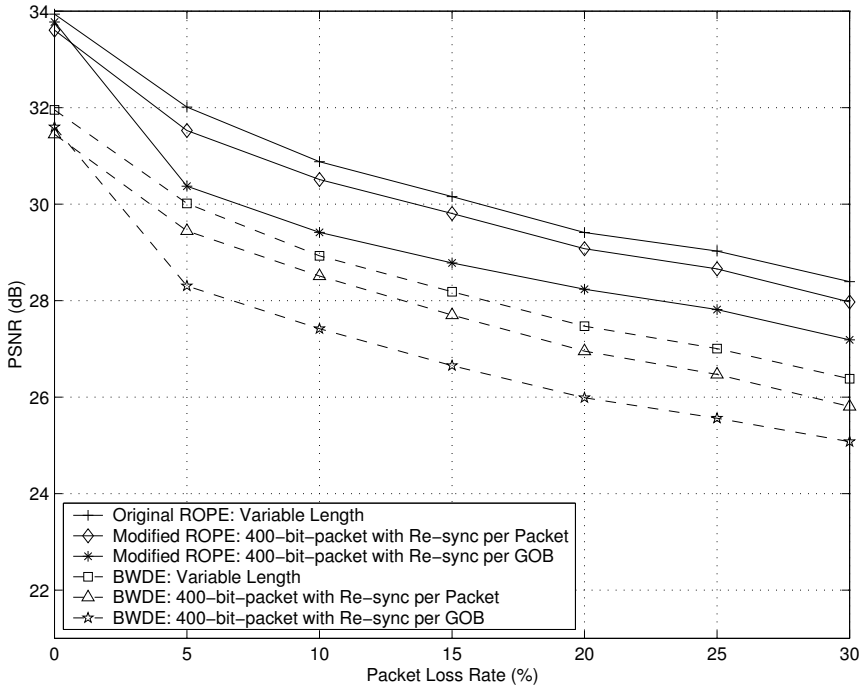
the feedback of just one type of information. For feedback of channel conditions, the refined estimation method substantially outperforms the simple estimation method.

II.E Conclusions

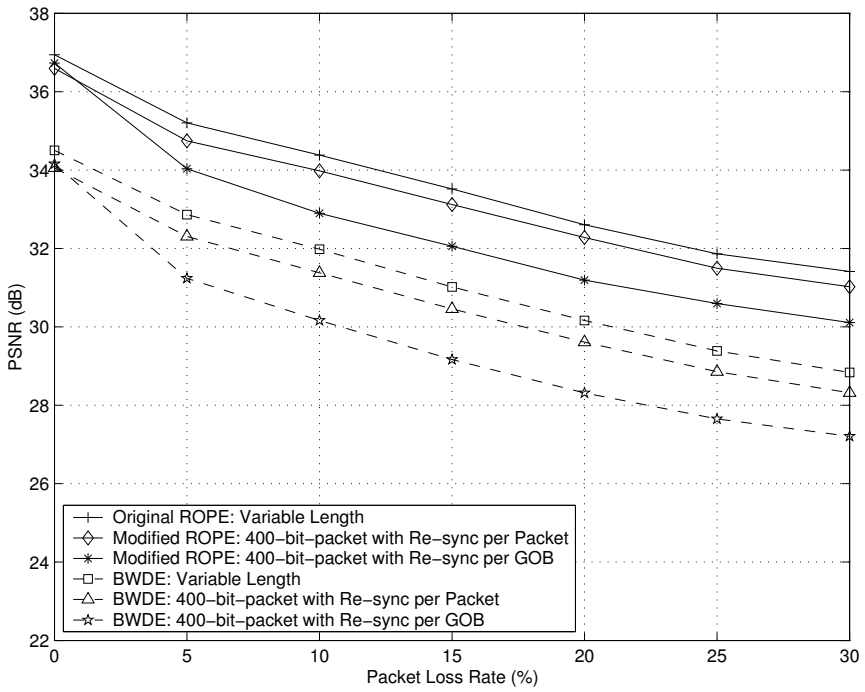
In this chapter, we present a transmission scheme for fixed length packet video. The transmission channel is a tandem channel which models both packet erasures and bit errors. We solve this tandem channel R-D optimization problem in two steps. First, we propose a video encoder using optimal inter/intra mode selection, operating over the wireline erasure-only channel. Then we added the wireless component. For this we used a concatenation of an inner RCPC coder and an outer CRC coder. Packets that fail the CRC check are dropped, so the tandem channel could be treated as a packet erasure channel. Detailed simulations were done to evaluate the performance over both constant and varying hybrid channel conditions. For the varying channel with delayed feedback information, it was shown that the refined estimation could dramatically improve the performance.

Acknowledgement

This chapter, in full, is a reprint of the material as it appears in Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Video Coding with Fixed Length Packetization for a Tandem Channel," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 273- 288, Feb. 2006. I was the primary author, and the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for this paper.

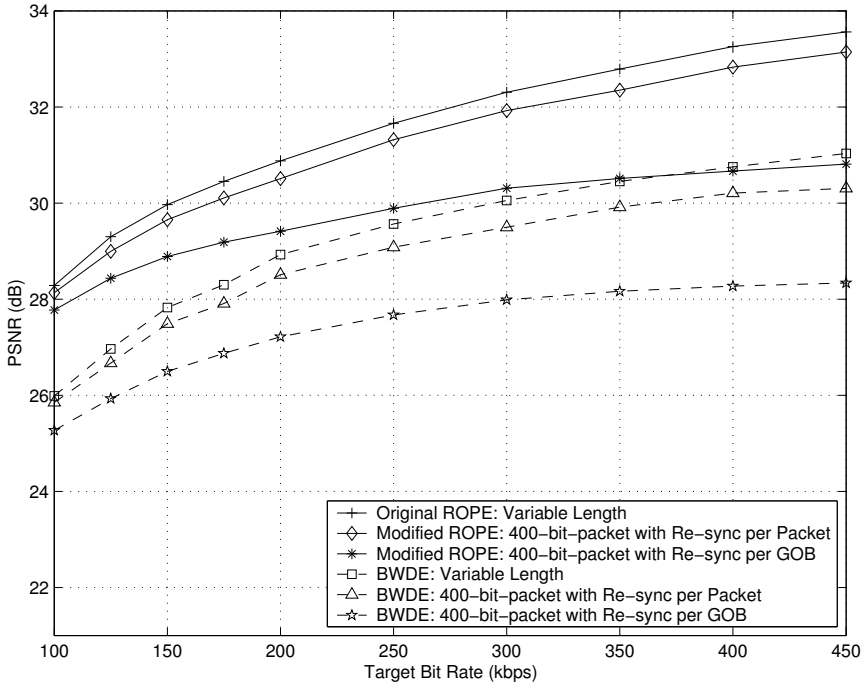


(a)

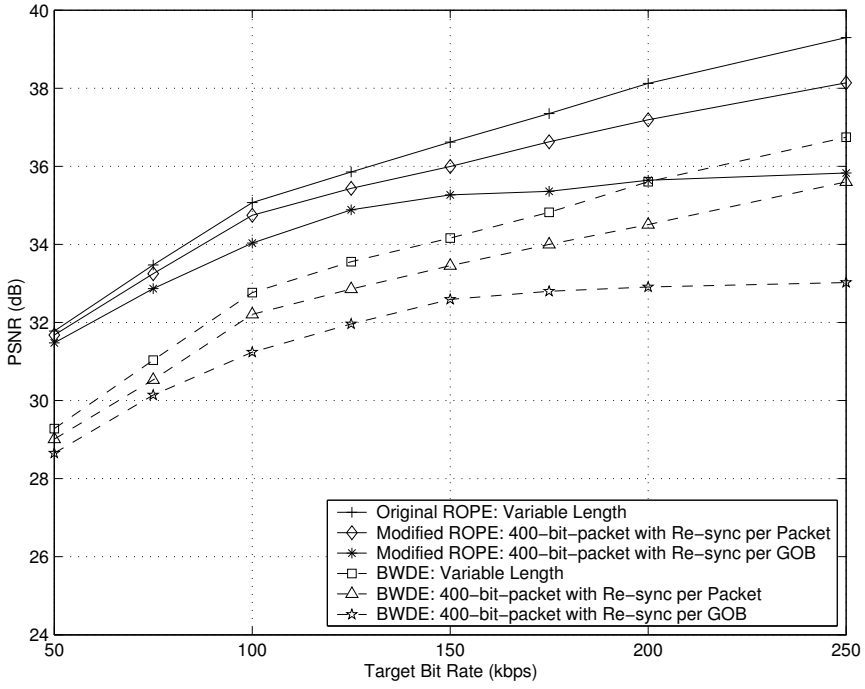


(b)

Figure II.4: PSNR performance versus packet loss rate. (a) Carphone QCIF at 200kbps and 30fps; (b) Container QCIF at 100kbps and 15fps.

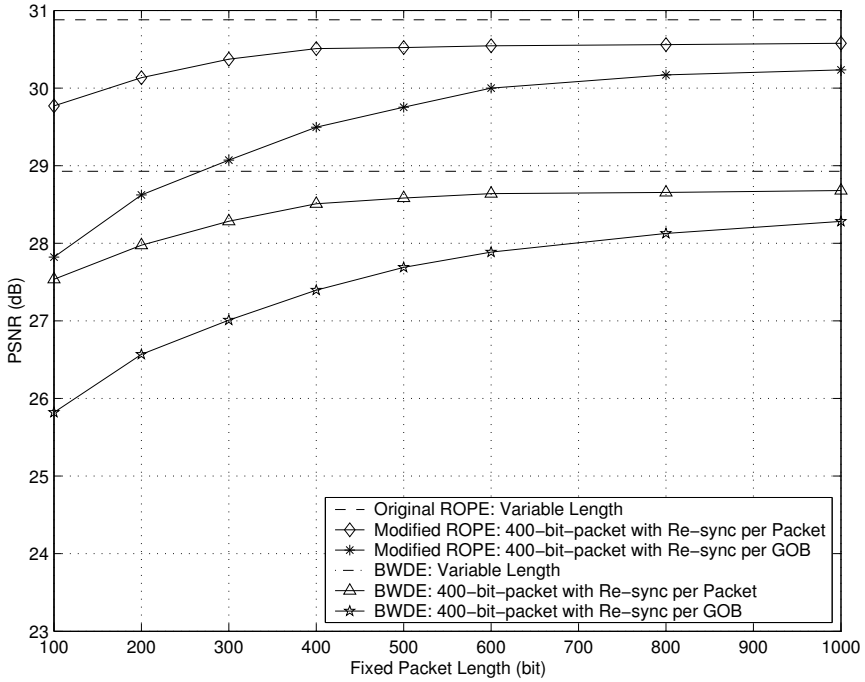


(a)

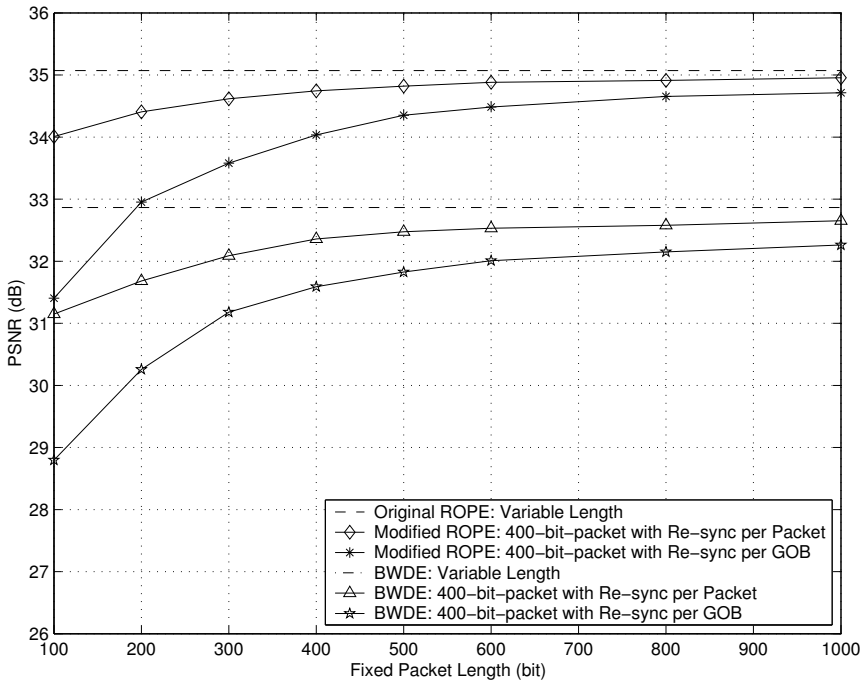


(b)

Figure II.5: PSNR performance versus target bit rate. (a) Carphone QCIF at 30fps, with packet erasure rate p=10%; (b) Container QCIF at 15fps, with packet erasure rate p=5%.



(a)



(b)

Figure II.6: PSNR performance versus fixed packet length. (a) Carphone QCIF at 200kbps and 30fps, with packet erasure rate p=10%; (b) Container QCIF at 100kbps and 15fps, with packet erasure rate p=5%.

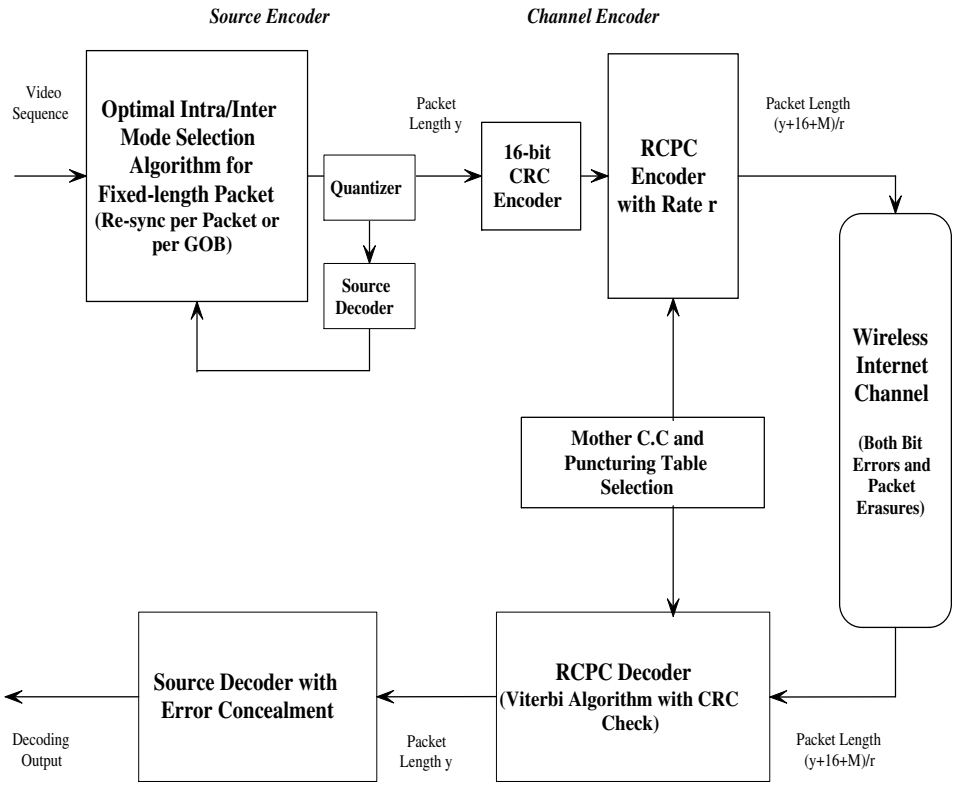
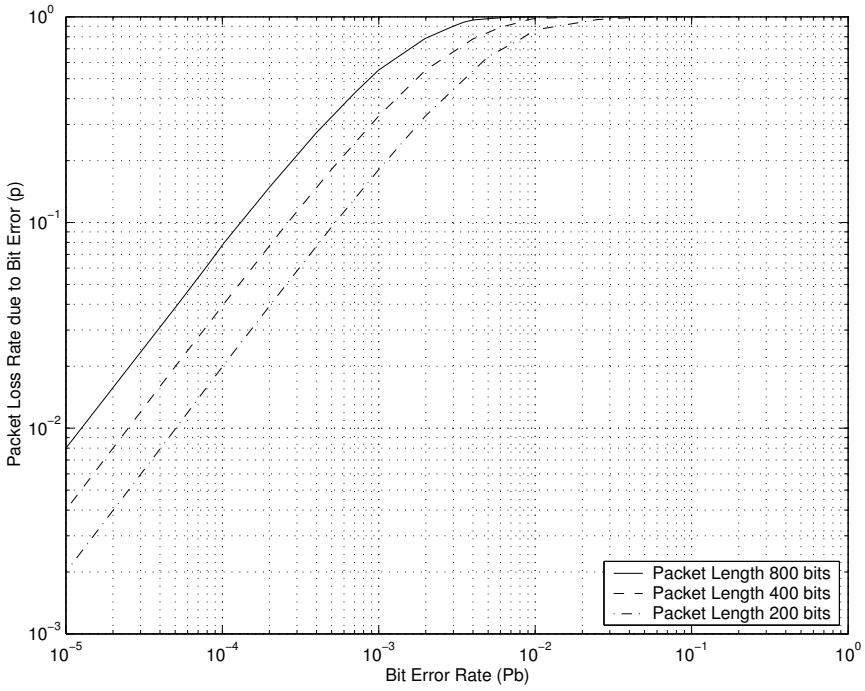
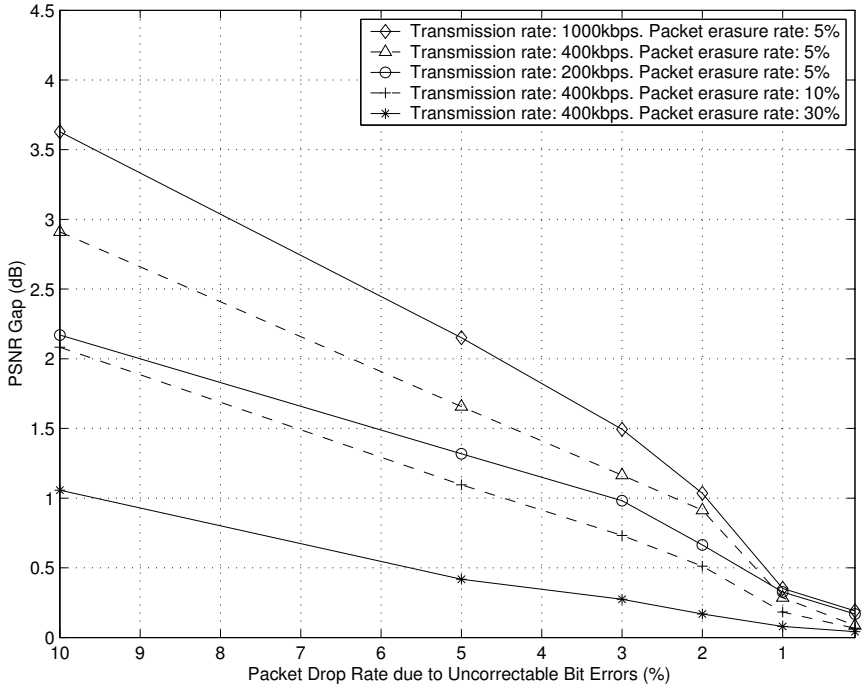


Figure II.7: System Overview

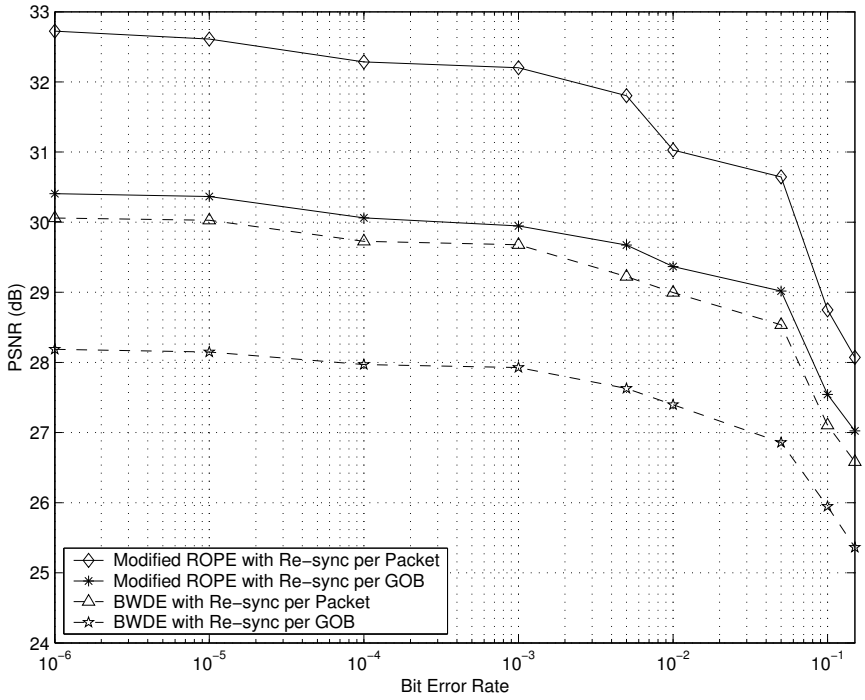


(a)

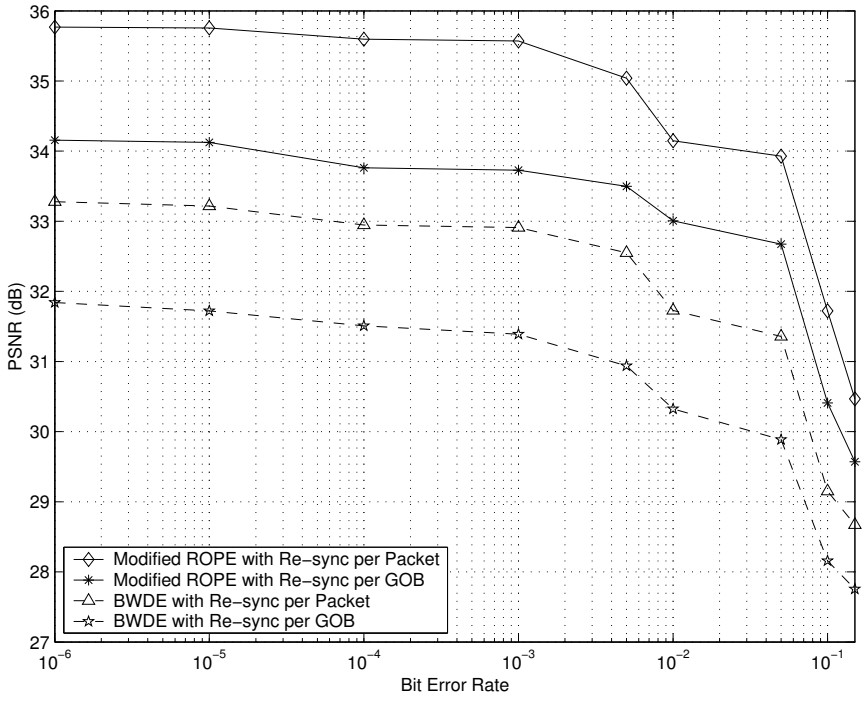


(b)

Figure II.8: The illustration of why we chose the packet drop rate due to uncorrectable bit errors to be 1%. (a) Bit error rate vs. corresponding packet loss rate without error correction; (b) PSNR gap for different target packet drop rates, "Carphone" QCIF sequence at 10fps and fixed-packet length 400.

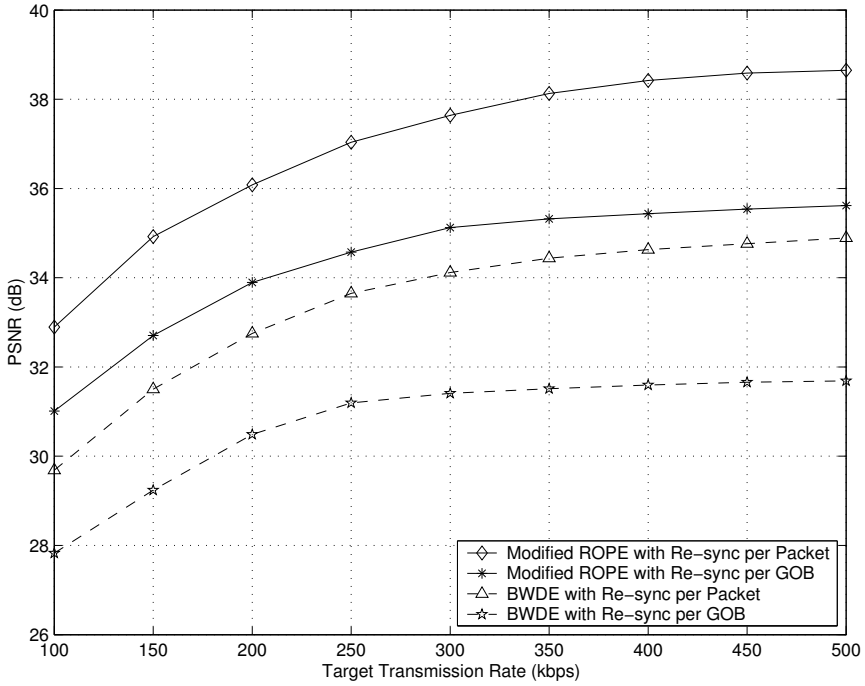


(a)

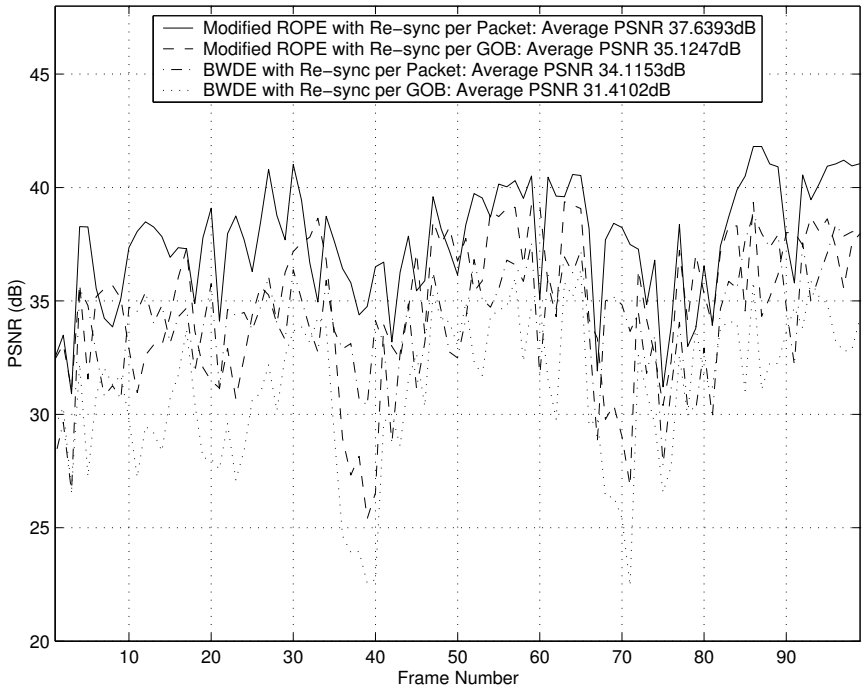


(b)

Figure II.9: PSNR performance versus bit error rate. (a) Carphone QCIF at 400kbps and 30fps, p=10% and packet length 400 bits; (b) Container QCIF at 150kbps and 15fps, p=5% and packet length 400 bits.



(a)



(b)

Figure II.10: PSNR performance versus transmission rate and versus frame number. (a) Salesman QCIF at 10fps, p=10% and P_b=0.01, packet length 800 bits; (b) Salesman QCIF at 300kbps and 10fps, p=10% and P_b=0.01, packet length 800 bits.

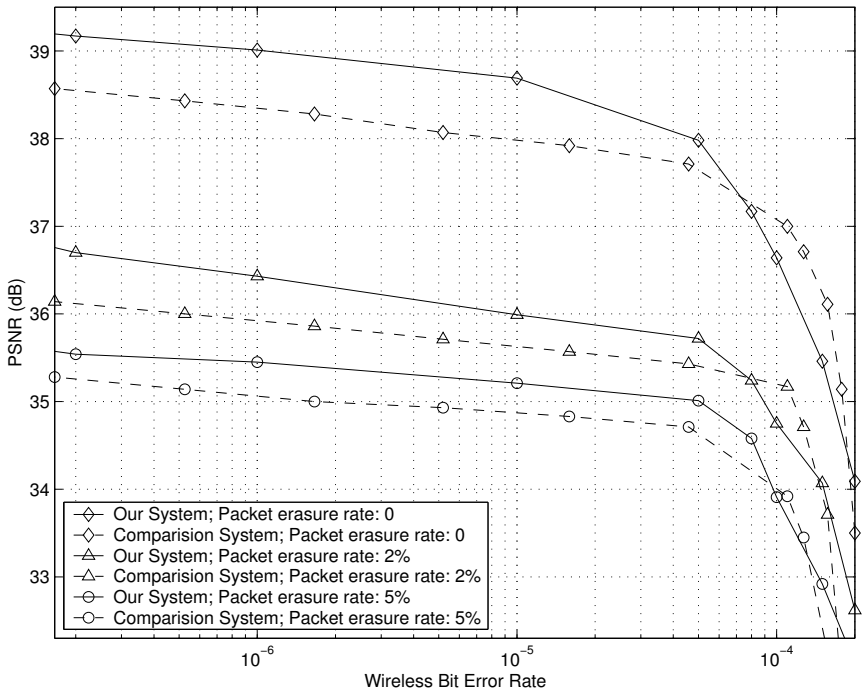
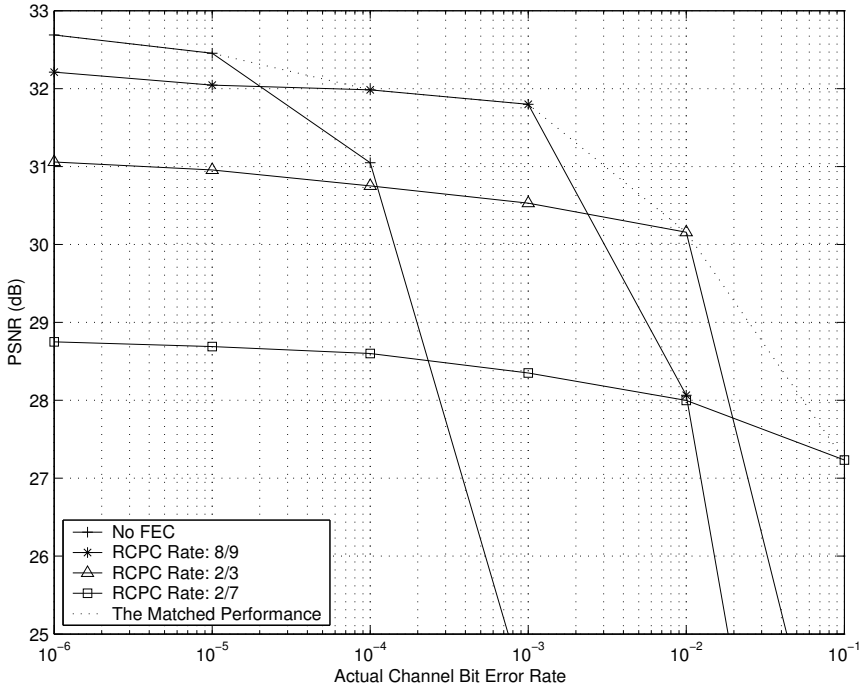
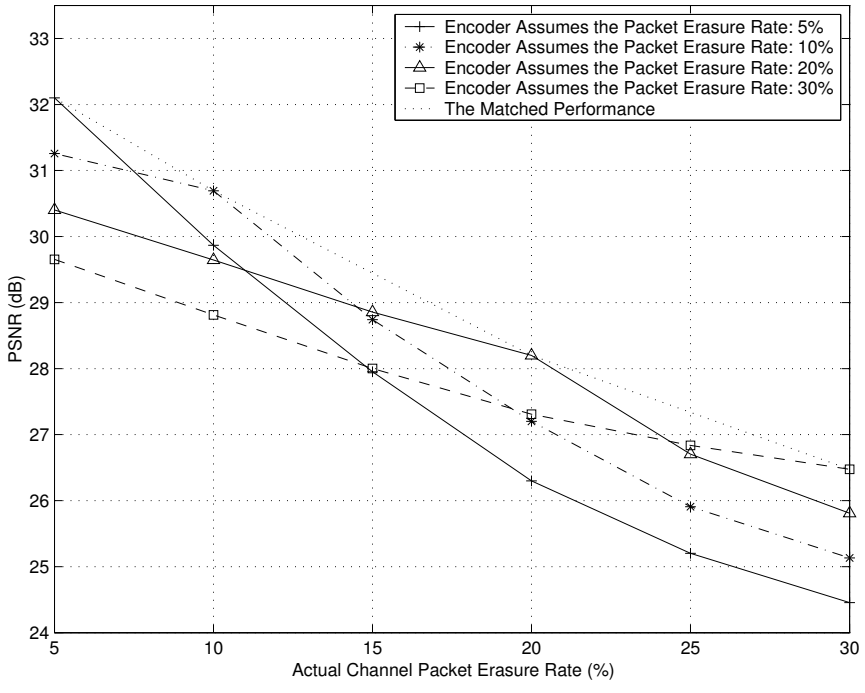


Figure II.11: PSNR performance versus wireless bit error rate, Susie QCIF with 128kbps and 7.5fps, 800-bit fixed packet length for our system, 9 packets per frame for the comparison system.



(a)



(b)

Figure II.12: PSNR performance for mismatched system, Carphone QCIF at 400kbps and 15fps, with packet length 400 bits. (a) P_b=0.001 and p=0%; (b) P_b=0 and p=5%.

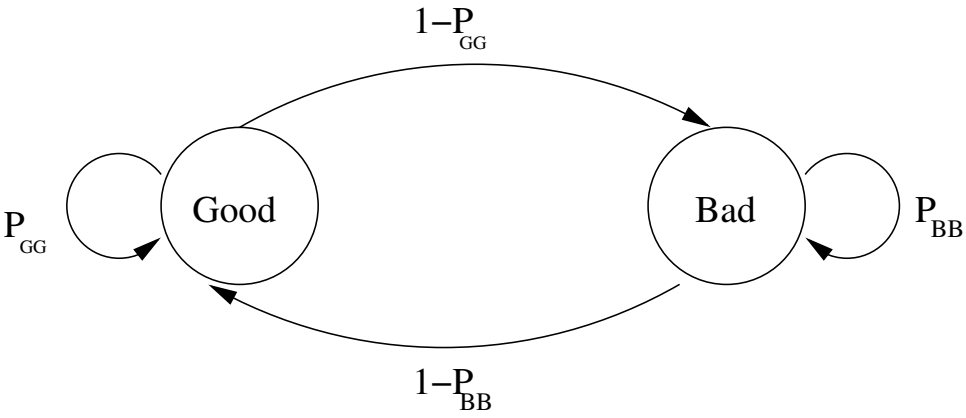


Figure II.13: The two-state Gilbert-Elliot model.

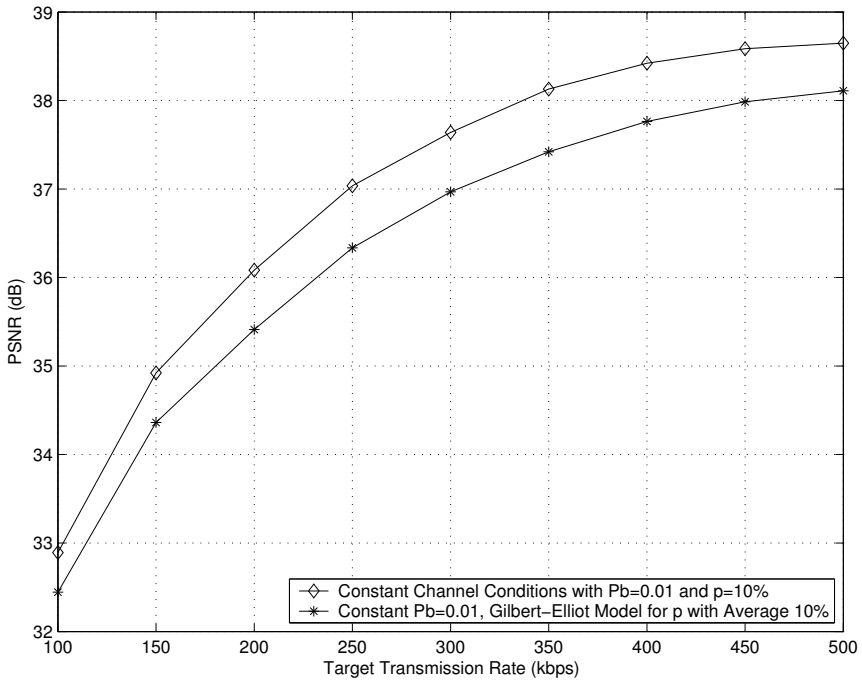
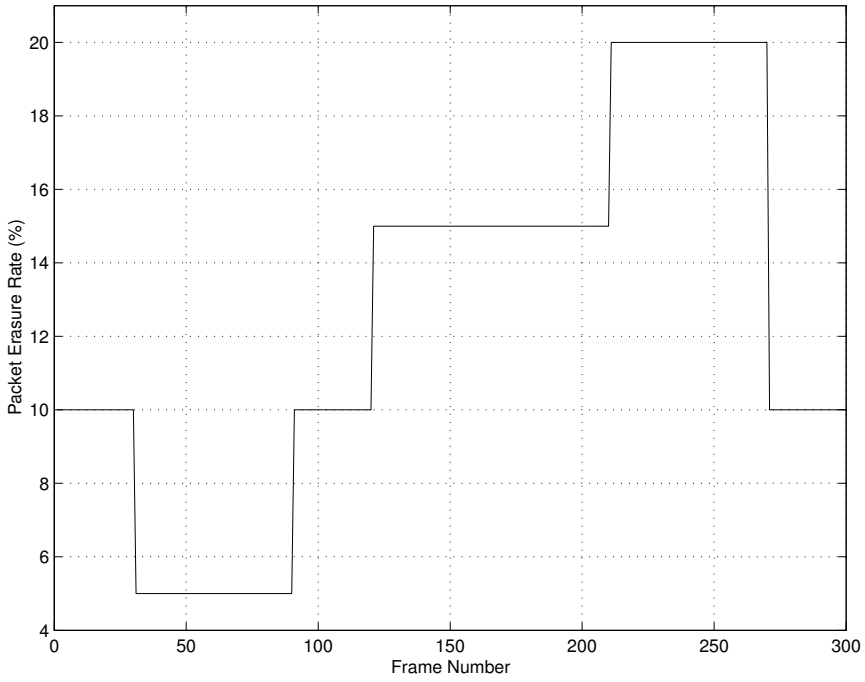
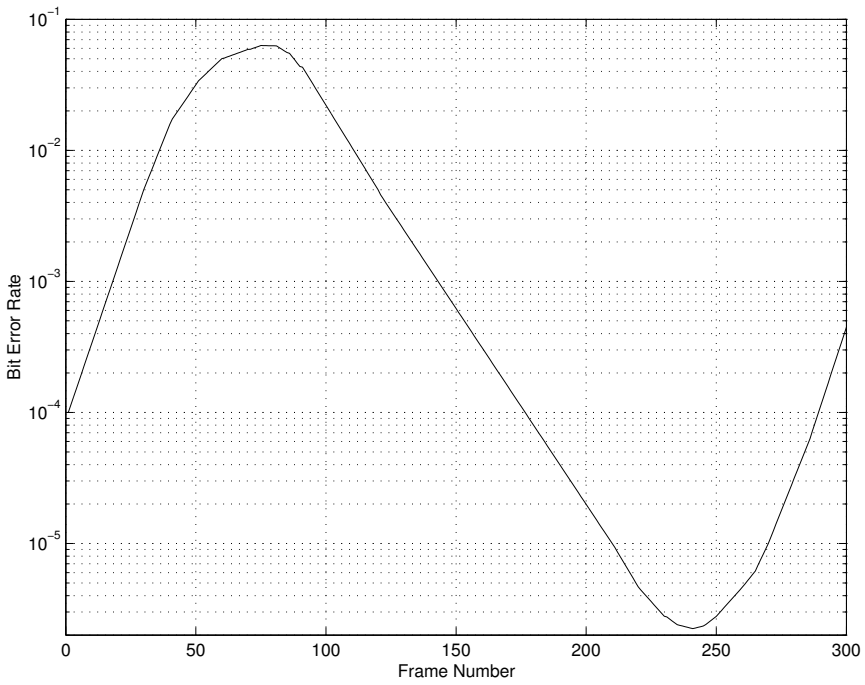


Figure II.14: PSNR performance versus transmission rate, Salesman QCIF at 10fps, re-sync per packet, with packet length 800 bits.

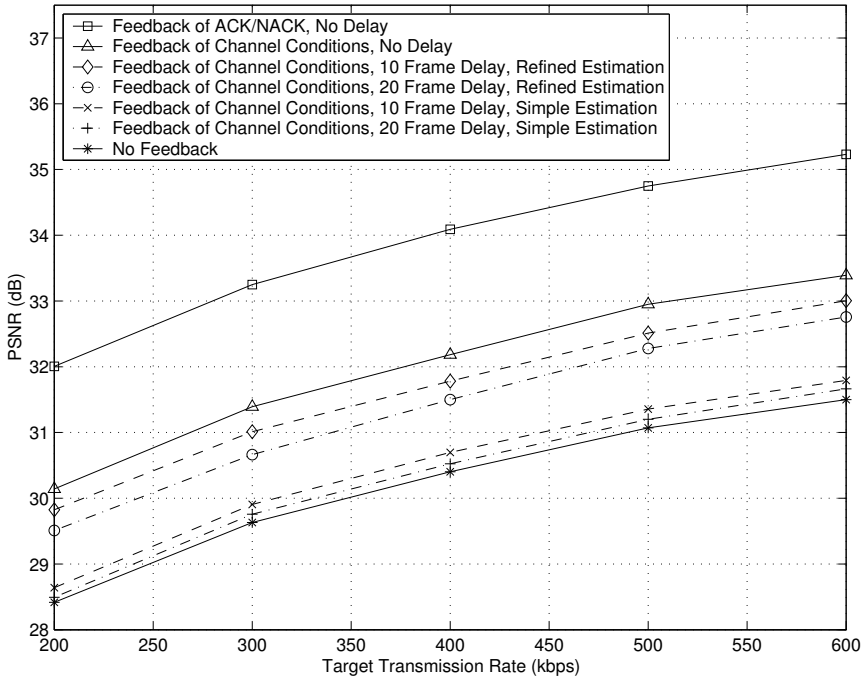


(a)

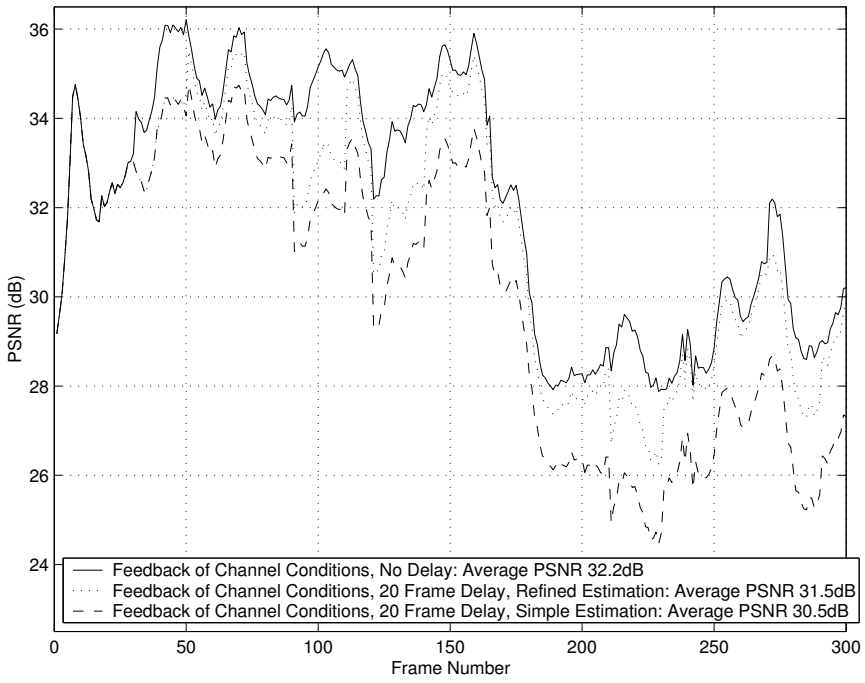


(b)

Figure II.15: Channel variation model over time. (a) Time varying channel packet erasure rate over time; (b) Time varying channel bit error rate over time.

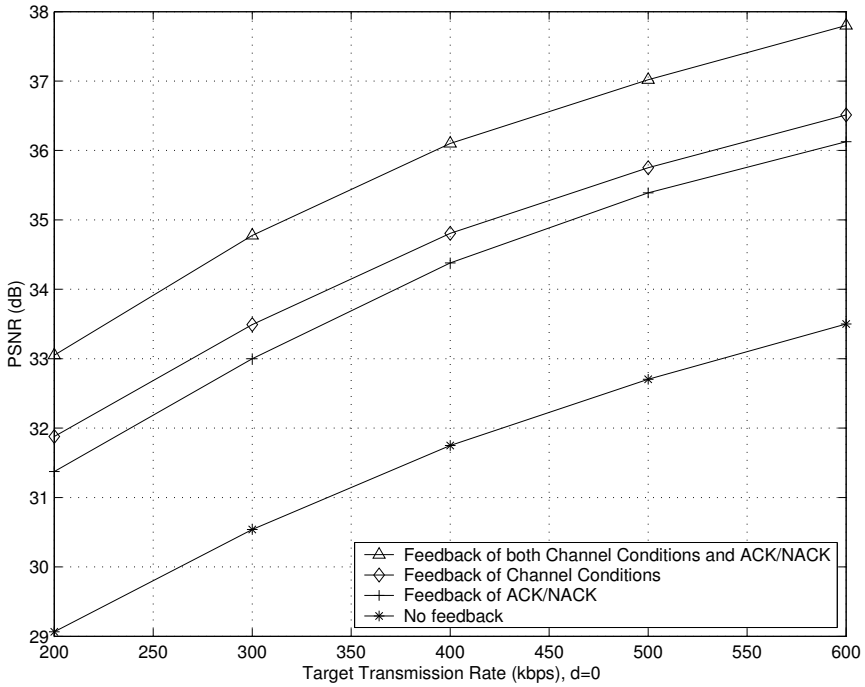


(a)

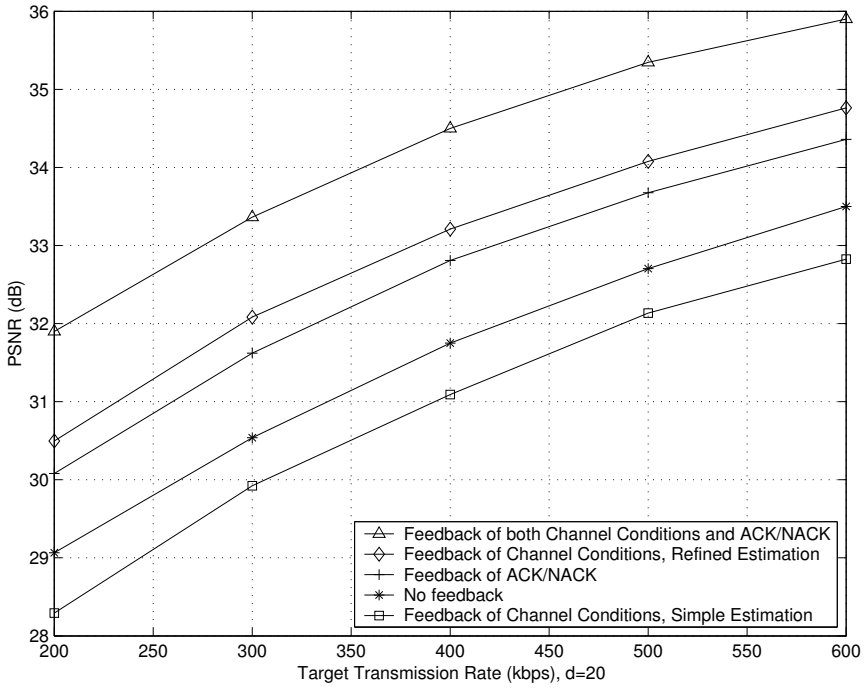


(b)

Figure II.16: PSNR performance over the time-varying pure packet erasure channel given in Fig. II.15(a), system with re-sync per packet, Carphone QCIF 30fps and packet length 400 bits. (a) PSNR performance versus transmission rate; (b) PSNR performance versus frame number at 400kbps.

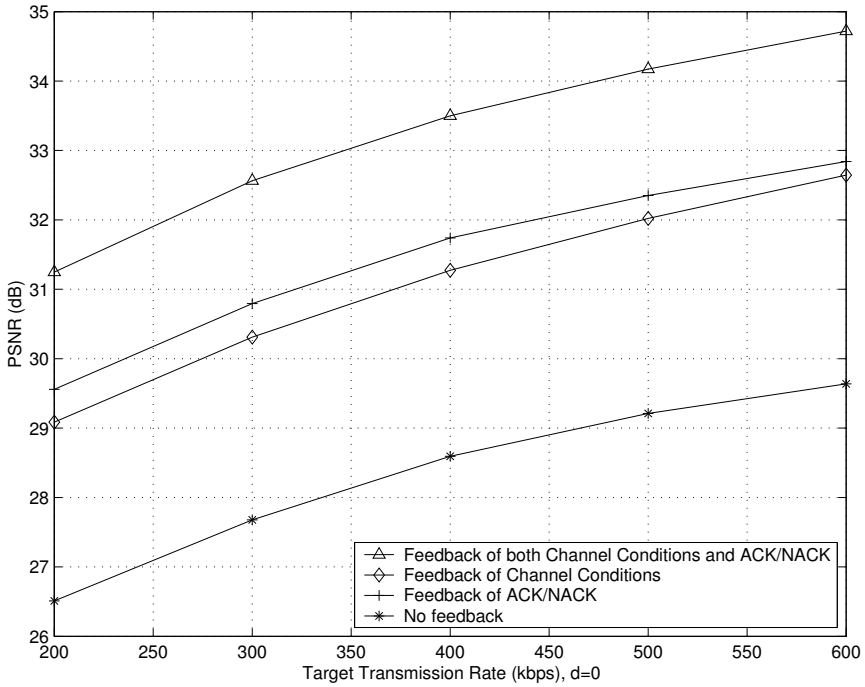


(a)

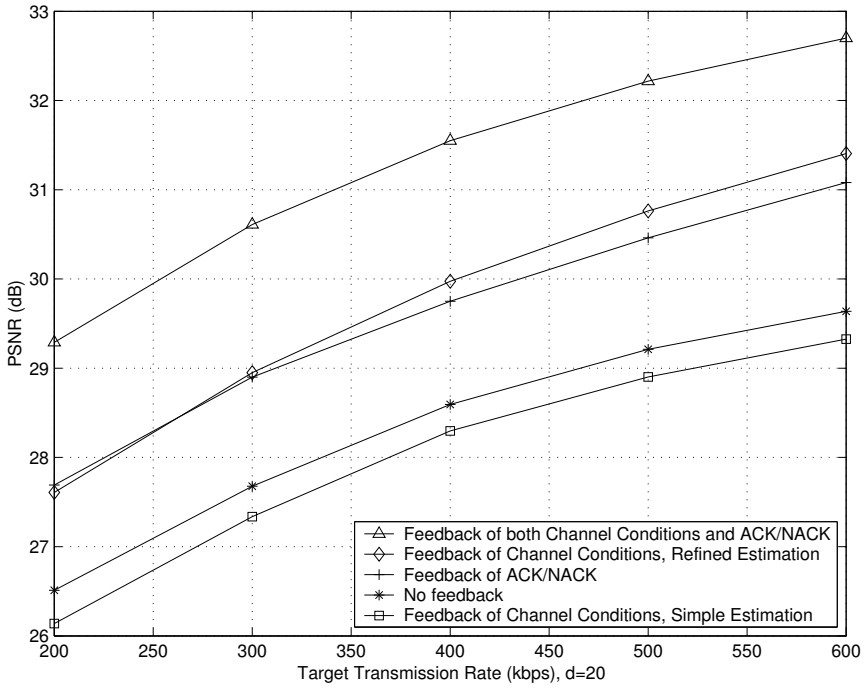


(b)

Figure II.17: PSNR performance over the time-varying pure wireless bit error channel given in Fig. II.15(b), system with re-sync per packet, Carphone QCIF 30fps and packet length 400 bits. (a) PSNR performance versus transmission rate, with instantaneous feedback; (b) PSNR performance versus transmission rate, with 20 frames delayed feedback.



(a)



(b)

Figure II.18: PSNR performance over a tandem channel with both time varying packet erasure rate and bit error rate, which is the combination of Fig. II.15(a) and Fig. II.15(b), system with re-sync per packet, Carphone QCIF 30fps and packet length 400 bits. (a) PSNR performance versus transmission rate, with instantaneous feedback; (b) PSNR performance versus transmission rate, with 20 frames delayed feedback.

III

Bandwidth Allocation for Video Communications over CDMA Networks

III.A Introduction

Direct-sequence code-division multiple-access (DS-CDMA) technology is useful because it exhibits robust performance against channel fading and interference, as well as good multiple access capacity [14]. Source coding, channel coding and spread spectrum are the three main components in most CDMA communications systems. Source coding frees up bandwidth for both channel coding and spreading, while channel coding and spreading protect the transmitted bits from noise, interference and fading [15].

The major resource shared among the three components is the total bandwidth,

which is equivalent to the transmission chip rate. We denote by R_s the source bit rate (in bits per second) and by W the chip rate (in chips per second); also, we denote by r_c the channel code rate and by M the processing gain. The variables R_s, r_c, and M are constrained as follows:

$$\frac{M}{r_c} R_s = W. \tag{III.1}$$

Another constraint in video communications is the total transmission time, which is equivalent to achieving an average target frame rate f_s (frames per second). Our goal is, for a target chip rate W and a target frame rate f_s, to find the optimal bandwidth allocation (R_s^*, r_c^*, M^*) for each packet, such that the expected distortion over all frames between the transmitted video bitstream and the reconstructed video bitstream at the decoder is minimized.

In [16] and [17] , bandwidth allocation algorithms were proposed for image transmission, where a simplified source, with either a uniform or a Gaussian distribution, and an optimal scalar quantizer, were assumed. In [18], [19], [20] and [21] , bandwidth allocation was discussed for image or video communications over either a single channel or multiple channels, and the allocation of system parameters was chosen for the entire transmission duration. In this chapter, we present a robust CDMA scheme with efficient bandwidth allocation for the transmission of packet video over a single wireless channel, which is assumed to be frequency selective with Rayleigh amplitude statistics. The choice of bandwidth allocation is made adaptively at the packet level, and incorporates the effects of both the changing channel characteristics and the current video content. This algorithm can be easily extended to a tandem channel with both wireline

and wireless links.

This chapter is organized as follows. In Section IV.B, the system model and channel model are briefly introduced. In Section III.C, we analyze the statistics of the received signal, and present a theoretical bound on the block error rate. Based on the results, Section III.D presents strategies for channel coding and/or spreading selection, given a target packet drop rate. In Section III.E, the algorithm to allocate bandwidth for each packet is presented, which solves the optimization problem by a two step approach based on the results illustrated in Sections IV.B-III.D. The statistical characteristics of channel estimates are derived and analyzed in Section III.F. System performance and its sensitivity to channel estimation errors are presented numerically in Section III.G. Lastly, Section III.H concludes the chapter.

III.B System Model

The system diagram discussed in this chapter is shown in Fig. III.1, and we will discuss the components in detail in this section.

III.B.1 Source Coding

The source encoder in Fig. III.1 uses an algorithm with fixed-length packetization, which is also a modified version of the original ROPE algorithm [6]. However, the modification proposed in this chapter difer from the one in Chapter II, because it operates on chips rather than bits.

Given the packet loss rate at the receiver and a target source bit rate, ROPE

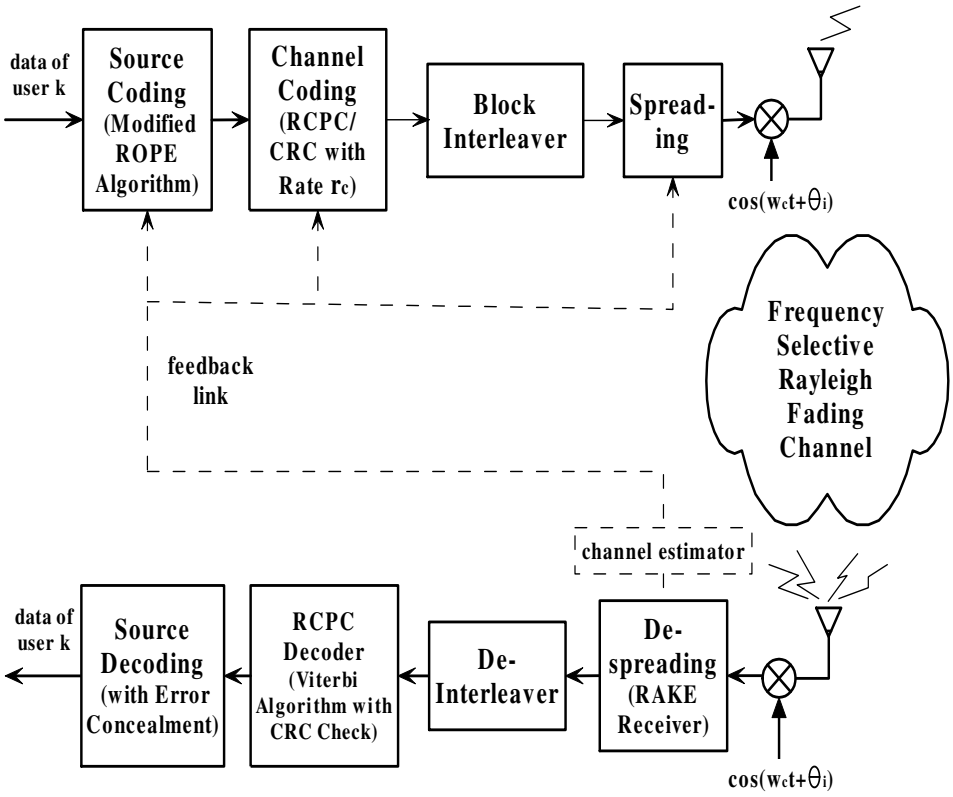


Figure III.1: System overview.

and modified-ROPE algorithms optimally chooses the inter/intra mode and quantization step for each MB according to the expected distortion at the pixel level.

Over a wireless channel, fading, noise and interference will cause packets to be dropped, due to uncorrectable bit errors. We denote the packet drop rate by p_p. It is shown in Section III.C that p_p is a function of r_c and M, given the other system parameters and channel characteristics.

III.B.2 Channel Coding

As in Chapter II, a concatenated code is used for FEC, which consists of a RCPC inner coder and a CRC outer coder. The encoder chooses a RCPC code from a

family of RCPC code candidates for each packet. Denoting by r_{RP} the rate of a non-packetized RCPC code, the channel coding rate of the resultant block code is

$$r_c = \frac{y - 16 - Z}{y} r_{RP}, \tag{III.2}$$

where y is the fixed packet length at the input of the RCPC encoder, 16 bits are used for the CRC check, and Z is the number of zero tail bits to terminate the packet [22]. In our system, the RCPC code candidates have rate r_{RP} equal to 1/3, 2/3, 8/9 and 1, with memory $Z = 6$ [15, 22]. The serial list-Viterbi algorithm at the channel decoder is used to find the best candidate in the trellis, and the packet is discarded if none of the first 100 paths with the maximal metric satisfies the CRC checksum equations [22].

III.B.3 Signal Spreading

Before going through the wireless channel, an interleaver is used to randomize the error bursts in the transmitted data block. The interleaved data stream is then spread using direct sequence with a long spreading code by a factor of M (processing gain), and transmitted using BPSK modulation.

We consider an asynchronous CDMA up-link system with K users [15]. Denote by $a_k(t)$ the signature sequence waveform for the kth user ($0 \leq k \leq K-1$), and by $a_{k,j}$ the corresponding sequence element, where $a_{k,j} \in (+1, -1)$. Similarly, denote by $b_k(t)$ the data signal waveform for the kth user, and by $b_{k,j}$ the corresponding sequence element, where $b_{k,j} \in (+1, -1)$. Also denote by T_s and T_c the time duration of each symbol (i.e., channel bit) and each chip, respectively, with $T_s = MT_c$. Then

$a_k(t) = \sum_{j=-\infty}^{\infty} a_{k,j}\ P_{T_c}(t - jT_c)$ and $b_k(t) = \sum_{j=-\infty}^{\infty} b_{k,j}\ P_{T_s}(t - jT_s)$, where $P_{T_c}(t)$ is the pulse shape function so that $P_{T_c}(t) = 1$ for $0 \leq t < T_c$ and zero elsewhere, and $P_{T_s}(t)$ is the pulse shape function with $P_{T_s}(t) = 1$ for $0 \leq t < T_s$.

Denote by E_b, E_s and E_c the energy-per-information bit, per-channel bit and per-chip, respectively. Also, denote by A the amplitude of the transmitted signal. We assume perfect power control, thereby implying that all users have the same received power. Then we obtain

$$E_c = \frac{E_s}{M} = \frac{r_c}{M} E_b = \frac{1}{2} A^2 T_c. \tag{III.3}$$

Because both the total video transmission time and the bandwidth (thus the chip duration T_c) are fixed, keeping the total transmission energy constant is equivalent to keeping the energy per chip (E_c) constant. Lastly, the transmitted signal for the kth user is given by

$$s_k(t) = Re[S_k(t)e^{j\omega_c t}], \text{ where } S_k(t) = A a_k(t) b_k(t). \tag{III.4}$$

III.B.4 Wireless Channel Model

A linear tapped delay line filter is used to model the frequency selective Rayleigh fading channel, with a lowpass equivalent impulse response for the kth user $h_k(t) = \sum_{l=1}^{L} c_{k,l}(t)\ \delta(t - lT_c - \tau_{k,l}(t))$, where $\delta(t)$ is the Dirac delta function, L is the number of resolvable paths (which is assumed constant over time), and $c_{k,l}(t) = \alpha_{k,l}(t)e^{j\theta_{k,l}(t)}$ is the complex gain. The $\alpha_{k,l}$'s, $\theta_{k,l}$'s and $\tau_{k,l}$'s are the random path amplitudes, phases and delays. Assume the $\alpha_{k,l}$'s, $\theta_{k,l}$'s and $\tau_{k,l}$'s are independent for different k (but correlated over time).

We assume slow fading, that is, $T_c/T_{coh} \ll 1$, where $T_{coh} = c/(f_c v)$ is the coherence time of the channel, v is the speed of the mobile and c the speed of light. Considering a scenario where f_c is 900MHz, a typical T_{coh} is 10^{-2} seconds at highway speeds (75mph) and 2.5×10^{-2} at local speeds (30mph). For a packet size of $y = 400$ bits, and a source bit rate R_s greater than 50kbps, T_{coh} is typically equal to the transmission time of at least 2 packets. It means the fading is slow enough that the bits inside the same packet see the same fading amplitude and phase.

The received signal for the kth user over the kth channel is $r_k(t) = s_k(t) * h_k(t) + n_k(t)$, and the composite signal at the output of the channel is $r(t) = Re[R(t)e^{j\omega_c t}]$, where

$$R(t) = \sum_{k=0}^{K-1} \sum_{l=1}^{L} \alpha_{k,l} S_k(t - lT_c - \tau_{k,l}) e^{j\varphi_{k,l}} + N(t). \quad \text{(III.5)}$$

In (III.5), $\varphi_{k,l} = \theta_{k,l} - \omega_c \tau_{k,l}$, and $N(t)$ is complex Gaussian noise with two-sided power spectral density N_0 [14]. The $\varphi_{k,l}$'s are independent identically distributed (i.i.d.) random variables (rvs), uniformly distributed in $[0, 2\pi)$, the $\tau_{k,l}$'s are i.i.d. rvs, uniformly distributed in $[0, T_s)$, and the $\alpha_{k,l}$'s are independent Rayleigh rvs with density function $p(\alpha_l) = \frac{\alpha_l}{\sigma_l^2} exp(-\frac{\alpha_l}{2\sigma_l^2})$. Let $\Omega_l = E[(\alpha_l)^2] = 2\sigma_l^2$ and Ω_1=1. For an exponential multipath intensity profile (MIP), i.e., one where the energy of each path is decaying exponentially, we have $\Omega_l = \Omega_1 e^{-\nu(l-1)}$ and thus $\sum_{l=1}^{L} \Omega_l = (1 - e^{-\nu L})/(1 - e^{-\nu})$.

III.C System Analysis

In this section, we derive the statistics of the received signal, and the theoretical bound on the packet drop rate p_p due to wireless fading and interference. As a result,

it is shown that, conditioned on the system parameters and channel characteristics, p_p is a function of the coding rate r_c and spreading gain M.

III.C.1 Statistics of received signal

As shown in Fig. III.2, we use an L_R-finger CDMA RAKE receiver [14, 15], where the matched filter is matched to the reference user's signature code and is assumed to have achieved time synchronization. Note that L_R may be less than L due to complexity constraints and the omission of the weakest paths; however, here we choose $L_R = L$ for simplicity. Let the 0th user be the reference user. In the current analysis, we assume the tap weights and phases are perfect estimates, so that $\hat{c}^*_{k,l} = c^*_{k,l} = \alpha_{k,l}e^{-j\theta_{k,l}}$; equivalently, we assume $\theta_{0,l} = \tau_{0,l} = 0$, and thus $\varphi_{0,l} = 0$. Later, in Section III.F, the effects of estimation errors will be considered.

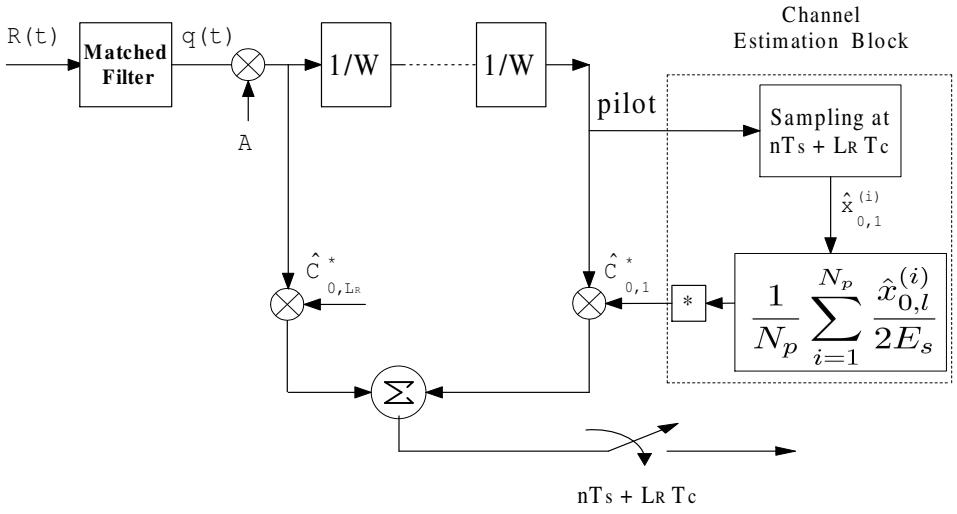


Figure III.2: RAKE receiver, with the channel estimation block.

The lowpass output statistic of the RAKE receiver is given by $g_n(t) = \sum_{m=1}^{L} A\, c_{0,m}^{*}\, q\,(t-(L-m)T_c)$, where $q(t)$ is the output of the matched filter, given by $q(t) = R(t) * a_0(T_s - nt)$. We can express $g_n(t) = g_S(t) + g_{I_s}(t) + g_{I_m}(t) + g_N(t)$, where it is seen that $g_n(t)$ includes four components: the signal $g_S(t)$, the self-interference $g_{I_s}(t)$, the multi-access-interference $g_{I_m}(t)$ and the Gaussian noise $g_N(t)$. At the sampling time $t = nT_s + LT_c$, we simplify the notation to $g_n = g_S + g_{I_s} + g_{I_m} + g_N$.

Conditioned on $b_0 = 1$, the signal component, g_S, is

$$g_S = A^2 M T_c \sum_{m=1}^{L} \alpha_{0,m}^2 = 2E_s \sum_{m=1}^{L} \alpha_{0,m}^2. \tag{III.6}$$

Conditioned on $\{\alpha_{0,m}\}_{m=1}^{L}$, the noise component, g_N, is a conditional complex zero-mean Gaussian rv with variance

$$\sigma_{g_N}^2 = E[g_N g_N^{*} | \{\alpha_{0,m}\}] = 4E_s N_0 (\sum_{m=1}^{L} \alpha_{0,m}^2). \tag{III.7}$$

If the number of users K is sufficiently large (say, $K \geq 8$), g_{I_m} is asymptotically a conditional zero-mean Gaussian rv with variance

$$\sigma_{g_{I_m}}^2 = \frac{8E_s^2}{3M}(K-1)(\sum_{l=1}^{L} \Omega_l)(\sum_{m=1}^{L} \alpha_{0,m}^2), \tag{III.8}$$

and g_{I_s} becomes negligible compared to g_{I_m} [14, 23].

Thus, for large K, the decision variable, $Re[g_n]$, is a Gaussian rv with conditional mean g_S and conditional variance $\frac{\sigma^2}{2}$, where $\sigma^2 = \sigma_{g_{I_s}}^2 + \sigma_{g_{I_m}}^2 + \sigma_{g_N}^2$. Combining

(III.3-III.8), the signal-to-noise-plus-interference-ratio (SNIR) at the receiver (denoted by γ_r), conditioned on $\{\alpha_{0,m}\}_{m=1}^{L}$, is given by [23]

$$\begin{aligned}\gamma_r &= \frac{g_S^2}{\sigma^2} \\ &= \Big(\sum_{m=1}^{L} \alpha_{0,m}^2\Big) \frac{1}{\frac{2(K-1)}{3M}\frac{1-e^{-\nu L}}{1-e^{-\nu}} + \frac{N_0}{r_c E_b}} \\ &= \Big(\sum_{m=1}^{L} \alpha_{0,m}^2\Big) \frac{1}{\frac{2(K-1)}{3M}\frac{1-e^{-\nu L}}{1-e^{-\nu}} + \frac{N_0}{M E_c}}. \end{aligned} \tag{III.9}$$

As a result, given the parameters (K, L, ν and chip-energy-to-noise ratio E_c/N_0) and the current channel conditions $\{\alpha_{0,m}\}_{m=1}^{L}$, γ_r is a function of r_c and M.

III.C.2 Packet error rate

A method was presented in [24] to find the weight enumerators, $T(x) = \sum_{d=d_{min}}^{y} A_d x^d$, of any binary linear block codes formed from a family of RCPC codes for a given packet length y, where d is codeword weight, A_d is the number of codewords with weight d, and d_{min} is the minimum weight (distance) of the block code. More details will be explained thoroughly in Chapter VI.

Furthermore, conditioned on the receiver SNIR, γ_r, a tight bound on the packet error rate introduced by the wireless link is given by [24]

$$p_p \leq \sum_{d=1}^{y} A_d Q(\sqrt{2d\gamma_r}). \tag{III.10}$$

For example, for a packet length y=400, the relationship between packet error rate p_p and γ_r for the four RCPC code candidates of our system is shown in Fig. III.3.

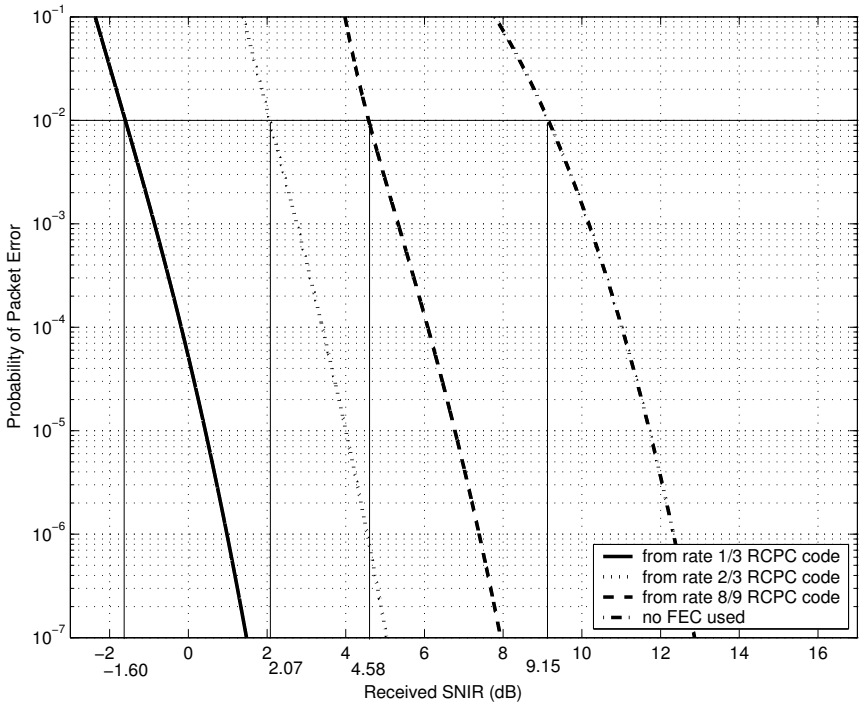


Figure III.3: The probability of packet error versus the received SNIR for the 4 candidate RCPC codes.

In short, p_p is determined by the weight enumerator of the truncated RCPC code and the receiver SNIR, where the former is determined by the structure of the RCPC code (characterized by r_c) and packet length y, and the latter is a function of r_c and M, given the system parameters and channel characteristics. We conclude that, given the channel and system parameters, and conditioned on the current channel conditions, p_p is fully determined by r_c and M.

III.D Tradeoff between Channel Coding and Spreading under a Target Packet Drop Rate

In this section, the tradeoff between channel coding and spreading is illustrated under a predetermined target packet drop rate (denoted by p_p^*). That is, besides the chip rate constraint (IV.1), we impose another constraint: for each packet, the expected packet drop rate due to the wireless link will not be larger than p_p^*. For example, in [22] and [12], a target $p_p^* = 1\%$ was used. In the next section, we will relax this constraint, and allow the packet drop rate to vary based on video content.

In Section III.C, it was shown that p_p is determined by r_c and M, given the system parameters and channel characteristics. Also, from the perspective of the source encoder, given the packet drop rate at the receiver, maximizing the source bit rate R_s is an intuitive strategy, although it has not been shown to be optimal. We speculate that, for a video encoder with a convex operational rate-distortion (RD) curve, under a given packet drop rate, distortion would be reduced by transmitting more source bits. This we confirmed empirically. We conducted 1000 simulations for the modified ROPE video encoder [22] and we found that, in every case, for the given packet drop rate, the distortion was minimized when the source rate was maximized. Therefore, our baseline strategy for bandwidth allocation, for a given p_p^*, is to choose the $(\hat{r_c}, \hat{M})$ for each packet that maximizes R_s, among all the (r_c, M) combinations that achieve p_p^*. In other words, we will choose the $(\hat{R_s}, \hat{r_c}, \hat{M})$ with maximal R_s among all the combinations that satisfy the p_p^* target.

Assume the number of users K, the packet length y, and the channel param-

eters L and ν are known in advance. The transmitter pre-calculates the relationship between p_p and γ_r for each RCPC code before the data transmission, and thus knows the SNIR threshold, $TH_{\gamma_r}(r_c)$, above which the target p_p^* is achieved. For example, for our system with 4 RCPC codes and $p_p^* = 1\%$, as shown in Fig. III.3, the thresholds are as follows:

$$TH_{\gamma_r}(r_c) = \begin{cases} 9.15dB & r_c = 0.945 \ (r_{RP} = 1) \\ 4.58dB & r_c = 0.84 \ (r_{RP} = 8/9) \\ 2.07dB & r_c = 0.63 \ (r_{RP} = 2/3) \\ -1.60dB & r_c = 0.315 \ (r_{RP} = 1/3). \end{cases} \tag{III.11}$$

Note that the relationship between r_c and r_{RP} is specified in (III.2). During the data transmission, the receiver estimates the $\alpha_{0,m}$'s, calculates $(\sum_{m=1}^{L} \alpha_{0,m}^2)$ and sends this number back to the transmitter. Depending on whether the processing gain (M) is fixed or variable, the transmitter selects the RCPC code and/or M according to the strategies given in the following subsections, encodes the data, and sends the packet out.

Note that instead of the receiver sending back the number $(\sum_{m=1}^{L} \alpha_{0,m}^2)$ and the transmitter doing the calculations to select the best RCPC code and/or M, the calculations can be done by the receiver and thus only the RCPC code and/or M must be sent back to the transmitter.

III.D.1 Strategy with fixed processing gain

For certain scenarios, the spreading gain M is pre-determined and fixed for all packets during the transmission process. Under this situation, given the system param-

eters and channel characteristics, γ_r is solely determined by r_c, and to maximize R_s we must maximize r_c (see (IV.1)). Thus, we choose the RCPC code with highest coding rate that achieves $p_p \leq p_p^*$.

Specifically, the encoder will first try to use an uncoded bit stream, calculate $\gamma_r(r_c)$ by (III.9), and check whether it is larger than the corresponding threshold $TH_{\gamma_r}(r_c)$. If so, this RCPC code is used for the current packet; if not, it will try the RCPC code with second highest r_c, and repeat the procedure until the lowest rate code is considered.

III.D.2 Strategy with variable processing gain

If both the channel coding rate r_c and the spreading gain M are variable for each packet, choosing the pair $(\hat{r_c}, \hat{M})$ that maximizes R_s among those that achieve $p_p \leq p_p^*$, is equivalent to choosing $(\hat{r_c}, \hat{M})$ with the highest (r_c/M) ratio among all the (r_c, M) pairs such that $\gamma_r(r_c, M) \geq TH_{\gamma_r}(r_c)$.

Specifically, for each candidate code rate r_c, the encoder finds the $\hat{M}(r_c)$ which is the smallest integer that satisfies the equation $\gamma_r(r_c, M) \geq TH_{\gamma_r}(r_c)$. Then, the pair $(\hat{r_c}, \hat{M})$ is chosen from all the $(r_c, \hat{M}(r_c))$ pairs such that it has the highest $(r_c/\hat{M}(r_c))$ ratio.

This strategy allows the encoder to select both r_c and M for each packet. However, it can be shown that the encoder will actually select the same r_c and only vary M most of the time. To see this, from (III.9) with constant $\frac{E_c}{N_0}$, we have $\hat{M}(r_c) =$

$\bar{M}(r_c) + \delta$, where

$$\bar{M}(r_c) = \left(\frac{\frac{2}{3}(K-1)\frac{1-e^{-\nu L}}{1-e^{-\nu}} + \frac{N_0}{E_c}}{\sum_{m=1}^{L} \alpha_{0,m}^2}\right) TH_{\gamma_r}(r_c), \tag{III.12}$$

and $\delta \in [0, 1)$ is a fraction that makes $\hat{M}(r_c)$ an integer. Because, typically, $\bar{M}(r_c) \gg \delta$,

$$\begin{aligned} \frac{r_c}{\hat{M}(r_c)} &\approx \frac{r_c}{\bar{M}(r_c)} \\ &= \left(\frac{\sum_{m=1}^{L} \alpha_{0,m}^2}{\frac{2}{3}(K-1)\frac{1-e^{-\nu L}}{1-e^{-\nu}} + \frac{N_0}{E_c}}\right) \frac{r_c}{TH_{\gamma_r}(r_c)}. \end{aligned} \tag{III.13}$$

Thus, given K, L, ν and $(\sum_{m=1}^{L} \alpha_{0,m}^2)$, maximizing $(r_c/\hat{M}(r_c))$ is equivalent to using the RCPC code with the highest $(r_c/TH_{\gamma_r}(r_c))$ ratio. For our example of 4 RCPC codes with $p_p^* = 1\%$, the ratio $(r_c/TH_{\gamma_r}(r_c))$ is 0.455, 0.391, 0.292 and 0.115 for the rate-1/3, rate-2/3, rate-8/9, and rate-1 RCPC codes, respectively. So the transmitter will always use the rate-1/3 RCPC code, and only M is adjusted to account for the changing channel conditions. This result illustrates that FEC is more important than spreading in a scenario where the number of users is large enough so that the self-interference due to the RAKE receiver is negligible compared to the multi-access-interference.

III.E Bandwidth Allocation Algorithm

The tradeoff between channel coding and spreading under a predetermined p_p^* was discussed in Section III.D. However, packets are not equally important. We want to choose the p_p^* for each packet according to the video content. For example, a

packet with static MBs should be relatively less sensitive to errors because of the ease of concealment, and thus a higher p_p^* should be acceptable.

In this section, we propose an algorithm for bandwidth allocation adaptively at the packet level, which incorporates the effects of both the changing channel conditions and the current source content. Our target is to find the bandwidth allocation among source coding, channel coding and spreading to optimize the overall performance, under the given bandwidth constraint and a target frame rate. We select a predetermined set of target packet drop rates, denoted $[p_{p1}, p_{p2}, ..., p_{pN}]$. During transmission, for each packet, the bandwidth allocation parameters are obtained using the following two steps.

Step I: We first trade off r_c and M for a given target packet drop rate due to the wireless link. That is, for each $p_{pi} \in [p_{p1}, p_{p2}, ..., p_{pN}]$, we trade off r_c and M using the algorithm of Section III.D, so that the target p_{pi} is achieved and R_s is maximized under the chip rate constraint. As a result, we obtain a set of 4-tuples $(p_{pi}, R_{si}, r_{ci}, M_i)$.

It is possible that no (r_c, M) combination can achieve a certain p_{pi} under the current channel conditions. Then the encoder will discard this p_{pi} as a candidate allocation for the current packet. Also, if no member of $[p_{p1}, p_{p2}, ..., p_{pN}]$ can be achieved, the encoder will declare deep fading and temporarily send nothing (except the pilot bits for channel estimation); however, this will not happen often if the range of p_{pi} is large enough.

Step II: We now trade off the packet drop rate (p_p) and source coding rate (R_s) to maximize the overall performance under the chip rate constraint and frame rate target.

For each $(p_{pi}, R_{si}, r_{ci}, M_i)$ 4-tuple, the video encoder encodes the current video

content (MB by MB) based on the Modified-ROPE algorithm [22] until the encoded bits fill the fixed length packet. That is, for each encoded MB, the encoder exhaustively tries all combinations of inter/intra mode and possible quantization steps, and finds the optimal mode/quantization decision.

Denote the pixel to be encoded by f, and the recovered pixel at the decoder by $\tilde{f}$, where $\tilde{f}$ is treated as a rv at the encoder. The expected distortion for each pixel is given by

$$d = E\{(f - \tilde{f})^2\} = (f)^2 - 2fE\{\tilde{f}\} + E\{(\tilde{f})^2\}. \tag{III.14}$$

The formulas to calculate the two moments of $\tilde{f}$ were derived in [22], given the coding mode (inter or intra), the quantization parameter (QP), the packet drop rate, and the concealment method.

For each combination of mode and quantization step, the encoder calculates the distortion of each pixel based on the packet drop rate (p_{pi}) according to (III.14), and further calculates the total distortion of the current MB (denoted by D_{MB}), the number of source bits used to encode the current MB (denoted by R_{MB}), and the number of chips needed to transmit the current MB (denoted by W_{MB}), where $W_{MB} = (R_{MB}/r_{ci})\ M_i$.

The optimal inter/intra mode and QP for the current MB are obtained by minimizing a Lagrange variable that is determined by both expected distortion and rate usage. Unlike [6, 22], our constraint is on chip rate rather than source bit rate. Given the frame rate f_s (frames per second) and the chip constraint W (chips per second), the target average number of chips for each MB over time, denoted by W_{Target}, is equal to the constant $W/(N_{MB}f_s)$, where N_{MB} is the number of MBs per frame ($N_{MB} = 99$ for

QCIF video). The extended Rate-Distortion (RD) framework is to choose the mode and the QP to minimize

$$\min_{(mode,QP)} J_{MB} = \min_{(mode,QP)} (D_{MB} + \lambda W_{MB}). \quad \text{(III.15)}$$

Assuming the current encoding MB is the nth MB from the beginning, rate control is achieved by updating λ using the following algorithm modified from [6] and [22] to use chips rather than bits:

$$\lambda(n) = \lambda(n-1)\ [\ 1 + \alpha(W_{used}(n-1)\ -\ (n-1)W_{Target})\ ], \quad \text{(III.16)}$$

where $\alpha = 1/(5W_{Target})$, $W_{used}(n)$ is the total number of chips used up to the nth MB, and $W_{used}(n) = W_{used}(n-1) + W_{MB}(n)$.

As a result, for each 4-tuple $(p_{pi}, R_{si}, r_{ci}, M_i)$, the encoder uses the Modified-ROPE algorithm (with the modified rate control scheme for CDMA) to encode MBs until the encoded bits fill the fixed length packet. Note that this encoding strategy guarantees satisfaction of the frame rate target, under the chip rate constraint, over time.

Because fixed length packetization is employed, the end of the packet usually is not the end of the encoded bits of a MB. A typical diagram of the packetization is shown in Fig. III.4. As shown in Fig. III.4, y denotes the fixed packet length in bits, A and a denote the number of total bits and tail bits of the last MB of the previous packet, respectively, B and b denote the number of total bits and tail bits of the current MB, respectively, c denotes the number of encoded MBs in the current packet, and $D_{MB}(j)$

($j = 0, \cdots, c$) is the corresponding expected distortion ($D_{MB}(0)$ is the distortion of the last MB of the previous packet). Note that y is a constant over the entire transmission; A, a and $D_{MB}(0)$ are the same for all $(p_{pi}, R_{si}, r_{ci}, M_i)$ 4-tuples when encoding the current packet; b and $D_{MB}(j)$ ($j = 1, \cdots, c$) are different for each tuple, and c may differ as well.

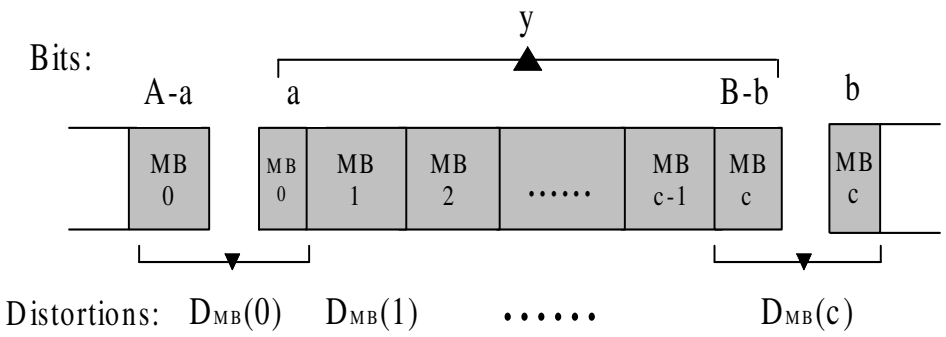


Figure III.4: Illustration of a typical packetization after source encoding.

The last step is for the encoder to choose among the $(p_{pi}, R_{si}, r_{ci}, M_i)$ 4-tuples so that the performance is optimized. Although the performance one would like to optimize is the PSNR of the entire transmission, this is not feasible both because we do not have access to the entire video and future channel conditions in advance, and because the overall optimization would be computationally intractable in any case. So instead, our greedy encoding strategy minimizes the distortion-per-time unit. The encoder chooses the 4-tuple that minimizes a parameter defined as the average expected distortion-per-time unit, which is equal to the total expected distortion across the packet divided by the total transmission time of the packet, and is approximately determined by

$$\frac{\text{Packet Distortion}}{\text{Packet Transmission Time}} \approx$$

$$\frac{(\frac{a}{A})D_{MB}(0) + \sum_{j=1}^{c-1} D_{MB}(j) + (\frac{B-b}{B})D_{MB}(c)}{y/R_{si}}. \qquad \text{(III.17)}$$

After the 4-tuple that minimizes (III.17) is determined, the corresponding packet will be sent out.

Note that (III.17) is approximate due to the terms $(a/A)D_{MB}(0)$ and $((B - b)/B)D_{MB}(c)$, which approximately represent the corresponding distortion of the partial MBs inside the current packet. Because the tails are much shorter than y , the effect of the imprecision is relatively negligible.

It is worth discussing the computational complexity of the algorithm, as well as the method to choose the set of possible p_p^*s. With the two-step approach, the complexity burden of the first step is negligible compared to both the second step and motion vector (MV) searching. In the second step, for each $(p_{pi}, R_{si}, r_{ci}, M_i)$ 4-tuple, the encoder chooses the optimal encoding mode and QP for each MB, by searching all possible mode/QP combinations (a total of $2 \times 31 = 62$ combinations). This complexity is comparable to that of the original ROPE algorithm and the DCT operations in the H.263+ codec [6, 22]. Assuming the number of possible p_p^*s is N_1, the total computational burden of the second step will be N_1 times higher than that of the algorithm with a single p_p^*. In [22], it was shown that for the same R_s, the performance gain from decreasing p_p diminishes dramatically for $p_p < 1\%$, and becomes negligible for $p_p < 0.1\%$. Also, we have found that if the SNIR falls below a value corresponding to a packet drop rate of roughly 5%, that packet drop rate increases very rapidly as the SNIR decreases further. Thus, we chose the p_p^*s in the region $[0.1\%, 5\%]$. In particular, after many simulation

runs for which the chip rates varied from 2 to 30Mcps, and E_c/N_0 varied from -18 to 0dB, we chose the following set of packet drop rates: [0.2%, 0.6%, 1%, 1.5%, 3%]. Adding additional p_p^*s only yielded trivial performance improvement. As a result, our algorithm uses $N_1 = 5$, and thus has about five times higher complexity than that of the original and modified ROPE algorithms described in [6, 22]. Using a smaller N_1 will decrease the complexity at the expense of performance degradation.

The coherence time of the channel tells us the correlation of channel conditions between adjacent packets. In Section III.B.4, we showed that the coherence time under most situations is at least the transmission time of two packets, thus we assume a constant channel across one packet. Furthermore, in our system design and analysis, we assume that the estimation feedback is instantaneous and error-free. We also assume instantaneous encoding and zero encoder buffer delay. Under these ideal assumptions, the encoder has the correct channel conditions for each packet instantly, and the encoder does not need knowledge of the coherence time. However, in the absence of these assumptions, knowledge of the coherence time of the channel can be usefully incorporated into the system design for predicting the changing channel conditions [25].

In all our numerical results, we assume that the RAKE receiver employs all resolvable paths of the selective fading channel (i.e., $L_R = L$). Such a receiver is practical in narrowband direct-sequence (DS) systems, where the number of resolvable paths is small. However, in wideband DS systems, and especially in ultra-wideband (UWB) systems, the use of less than the full number of resolvable paths (i.e., $L_R < L$), known as generalized selection combining (GSC), is typically employed. For perfect channel

state information, the results when $L_R = L$ serve as an upper bound on the performance achievable when $L_R < L$. However, with noisy channel estimates, eliminating the weak paths is usually beneficial, because the larger channel estimation errors experienced by the weak paths often outweighs the enhanced diversity achievable by using them [25, 26].

As to the optimality of the proposed algorithm, *Step II* is optimal in terms of minimizing distortion, since we select the bandwidth allocation 4-tuple from a set corresponding to the minimum distortion. *Step I*, however, has not been proved mathematically to be optimal. In our simulations, we found that once the packet drop rate was fixed, among those (R_s, r_c, M) triples which achieved that packet drop rate, the one with largest R_s invariably produced the lowest distortion. Note that this is not necessarily an optimal strategy for a variety of reasons: because the video encoder does not strictly have a convex RD curve, since adding one bit does not always decrease the distortion; because time correlation exists between dropped packets; and because the target packet drop rate may not be exactly achieved, since the choice of RCPC codes is finite. So while not provably optimal, it was found empirically that the strategy of *Step I* of choosing the largest R_s among those (R_s, r_c, M) triples which achieve each given packet drop rate was always the best choice.

This algorithm can be also extended to video communications over a tandem channel, which models both packet losses due to network congestion or buffer overflows on the wired link, and burst bit errors due to noise, fading and interference on the wireless component. Denote by p_e the packet erasure rate introduced by the wired link, and

assume it is known at the encoder. To use the proposed bandwidth allocation algorithm, instead of using the packet drop rate p_p of the wireless link, we need to use the total packet drop rate due to both wired and wireless links, which is given by

$$P_R = p_e + p_p - p_e \times p_p. \tag{III.18}$$

III.F The Effect of Imperfect Channel Estimation

In this section, the bias and variance of the channel estimates are derived, and the system sensitivity to channel estimation errors will be presented numerically in the next section. We assume the channel is characterized by a flat MIP, that is ν=0, and $\sum_{l=1}^{L} \Omega_l = L$. We also assume the data bits and pilot bits are time multiplexed. Specifically, for each packet, N_p pilot bits are appended to the end of that packet. As indicated in Fig. III.2, the sample mean at the output of each tap of the RAKE receiver is used as the channel estimate. That is, if $k = 0$ corresponds to the desired user, then for each resolvable path l ($1 \leq l \leq$ L) and each pilot bit i ($1 \leq i \leq N_p$), $\hat{x}_{0,l}^{(i)}$ denotes the output of the sampler, and the estimate of the complex channel gain of the lth resolvable path, $\hat{c_{0,l}}$, is given by

$$\hat{c}_{0,l} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{\hat{x}_{0,l}^{(i)}}{2E_s}. \tag{III.19}$$

Following the analysis of Section III.C.1, it can be shown that $\hat{x}_{0,l}^{(i)}$ is composed of four components, the signal $\hat{x}_s^{(i)}$, the self-interference $\hat{x}_{I_s}^{(i)}$, the multi-access-interference $\hat{x}_{I_m}^{(i)}$, and the Gaussian noise $\hat{x}_N^{(i)}$. The signal component $\hat{x}_S^{(i)} = 2(\alpha_{0,l} e^{j\varphi_{0,l}}) E_s$.

Conditioned on the current channel conditions of the desired user (that is, $c_{0,l} = \alpha_{0,l}e^{j\theta_{0,l}}$, for $1 \leq l \leq$ L), $\hat{x}_N^{(i)}$ is a conditional complex zero-mean Gaussian rv with variance $\sigma_N^2 = 4E_sN_0$. In the following analysis, we assume the number of users K is sufficiently large, so that $\hat{x}_{I_m}^{(i)}$ is asymptotically a conditional zero-mean Gaussian rv with variance $\sigma_M^2 = \frac{8E_s^2}{3M}(K-1)L$, and $\hat{x}_{I_s}^{(i)}$ becomes negligible. As a result, for large K, $(\hat{x}_{0,l}^{(i)}/2E_s)$ is asymptotically a conditional complex Gaussian rv with conditional density $\hat{x}_{0,l}^{(i)} \sim N(\alpha_{0,l}e^{j\varphi_{0,l}}, \sigma_x^2)$, where

$$\sigma_x^2 \approx \frac{\sigma_N^2 + \sigma_M^2}{(2E_s)^2} = \frac{2(K-1)L}{3M} + \frac{N_0}{E_s}. \tag{III.20}$$

Ignoring the self-interference components, the channel estimate is given by

$$\hat{c}_{0,l} \approx \frac{1}{N_p}\sum_{i=1}^{N_p}\frac{1}{2E_s}(\hat{x}_s^{(i)} + \hat{x}_{I_m}^{(i)} + \hat{x}_N^{(i)}), \tag{III.21}$$

where the $\hat{x}_N^{(i)}$ are independently and identically distributed (i.i.d) conditional zero-mean Gaussian rvs, $\hat{x}_{I_m}^{(i)}$ and $\hat{x}_N^{(i)}$ are independent of each other, and $\hat{x}_{I_m}^{(i)}$ and $\hat{x}_{I_m}^{(j)}$ are independent if $|i-j| \geq 2$, because the two regions of integration will see two independent random data bits. Thus, $\hat{c}_{0,l}$, the average of a set of jointly conditional Gaussian rvs, also is a conditional complex Gaussian rv, with a conditional density given by $\hat{c}_{0,l} \sim N(\alpha_{0,l}e^{j\varphi_{0,l}}, \sigma_c^2)$, where

$$\sigma_c^2 \approx \frac{1}{4N_p^2E_s^2}\left(\sum_{i=1}^{N_p}(\sigma_N^2 + \sigma_M^2) + \sum_{i=1}^{N_p-1}E(\hat{x}_{I_m}^{(i)}\hat{x}_{I_m}^{(i+1)})\right). \tag{III.22}$$

Note that the correlation between $\hat{x}_{I_m}^{(i)}$ and $\hat{x}_{I_m}^{(i+1)}$ (i.e., $E(\hat{x}_{I_m}^{(i)}\hat{x}_{I_m}^{(i+1)})$) is not zero, but is negligible compared to σ_M^2 because of the relatively low values of the partial autocorrelations of the spreading sequences [14], so we approximate σ_c^2 as

$$\sigma_c^2 \approx \frac{1}{N_p^2}\sum_{i=1}^{N_p}\sigma_x^2 \approx \frac{1}{N_p}\left(\frac{2(K-1)L}{3M}+\frac{N_0}{E_s}\right). \tag{III.23}$$

In Section III.D, $(\sum_{l=1}^{L}\alpha_{0,l}^2)$ was the parameter that the decoder feeds back to the encoder. Now, the decoder feeds back $F=\sum_{l=1}^{L}|\hat{c}_{0,l}|^2$. If we assume the resolvable paths are independent, the $\hat{c}_{0,l}$s are i.i.d conditional complex Gaussian rvs, and F follows a non-central chi-square distribution with $2L$ degrees of freedom. It is shown in the Appendix that

$$E\{F\}=(\sum_{l=1}^{L}\alpha_{0,l}^2)+L\sigma_c^2, \tag{III.24}$$

$$Var\{F\}=2\sigma_c^2\left(\sum_{l=1}^{L}\alpha_{0,l}^2\right)+L\sigma_c^4. \tag{III.25}$$

As a result, F is a biased estimator of $(\sum_{l=1}^{L}\alpha_{0,l}^2)$ with a deviation

$$\begin{aligned}\Delta&=F-(\sum_{l=1}^{L}\alpha_{0,l}^2)=L\sigma_c^2\\ &\approx\frac{L}{N_p}\left(\frac{2(K-1)L}{3M}+\frac{N_0}{E_s}\right).\end{aligned} \tag{III.26}$$

As the number of pilot bits per packet (denoted by N_p) increases, both the deviation Δ and the variance $Var\{F\}$ decrease and eventually go to zero. A typical illustration of the relationship between $Var\{F\}$ and N_p is shown in Fig. III.5, under the scenario

$y = 400$, $K = 20$, $L = 4$, $E_c/N_0 = -8$dB, $r_{RP} = 1/3$ and $M = 15$. If we use the estimator $\hat{F} = F - L\sigma_c^2$, it will be unbiased, but a given estimate could be negative. Since the term being estimated is non-negative, a better estimate is $F^* = max(\hat{F}, 0)$.

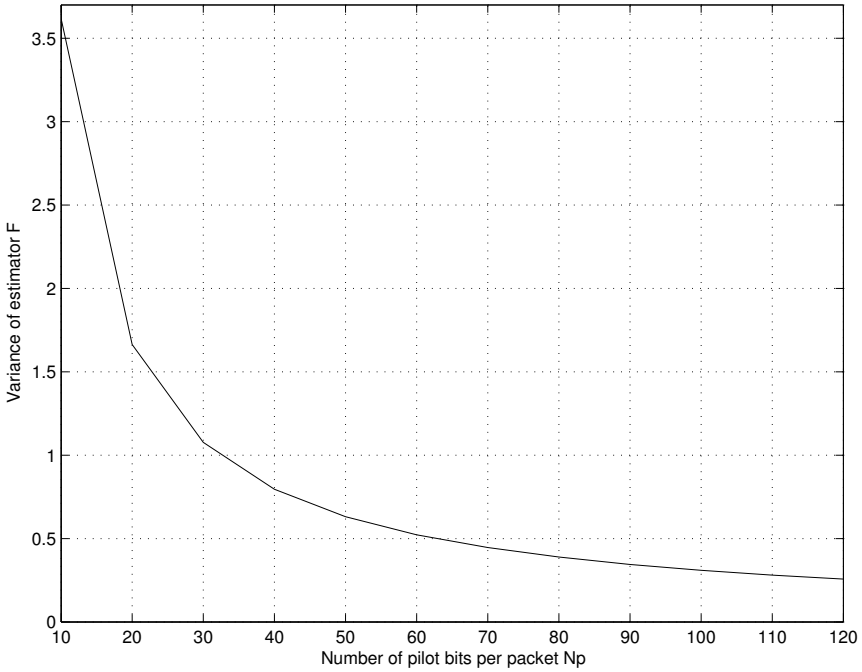


Figure III.5: A typical illustration of variance of the estimator F versus the number of pilot bits per packet N_p.

The negative effect of using a larger N_p is that pilot bits will occupy more bandwidth, reducing that available to data transmission, and thus yielding a lower source rate. Comparing to (III.2), the relationship between r_c and r_{RP} with N_p pilot bits per packet is determined by

$$r_c = \frac{y - 16 - Z}{y + N_p} r_{RP}. \tag{III.27}$$

The best choice of N_p is determined by a tradeoff between the estimate accuracy and the rate efficiency.

III.G Simulation Results

The proposed system was evaluated using a modified H.263+ codec. In our simulation, we set the packet length $y = 400$ bits, the number of users $K = 20$, the number of resolvable paths $L = 4$, and the normalized Doppler spread $f_D T_c = 3 \times 10^{-6}$. We also assumed the channel was characterized by a flat MIP. The Jakes model [27] was used to generate time-correlated Rayleigh fading parameters for each independent path of each user. Standard QCIF (176×144) video sequences were used at frame rate 15fps. The end-to-end distortion was measured by the peak signal-to-noise ratio (PSNR), which is defined as $10log_{10}(Peak^2/MSE)$, where $Peak$ is the peak value of the source and MSE is the mean square error over all frames.

Figs. III.6, III.7 and III.8 illustrate the PSNR performance of the system versus the transmission chip rate W, which varies from 2Mcps to 30Mcps. In all the figures, $E_c/N_0 = -8$dB. The figures are for the relatively high motion sequences "Carphone" and "Coastguard", and for a very low motion sequence "Akiyo". In each figure, the performance is shown for a system using the proposed bandwidth allocation with multiple target packet error rates, a system having a bandwidth allocation with a fixed target packet error rate of 1%, and a system having a bandwidth allocation with both a fixed target packet error rate of 1% and a fixed spreading gain $M = 15$. The system with multiple packet drop rates outperforms the one with a fixed target packet error rate by about 1.0 to 1.7dB, and outperforms the one with both fixed target packet error rate and fixed spreading gain by about 2.1 to 3.6dB.

Fig. III.9 illustrates the performance versus E_c/N_0, as E_c/N_0 varies from

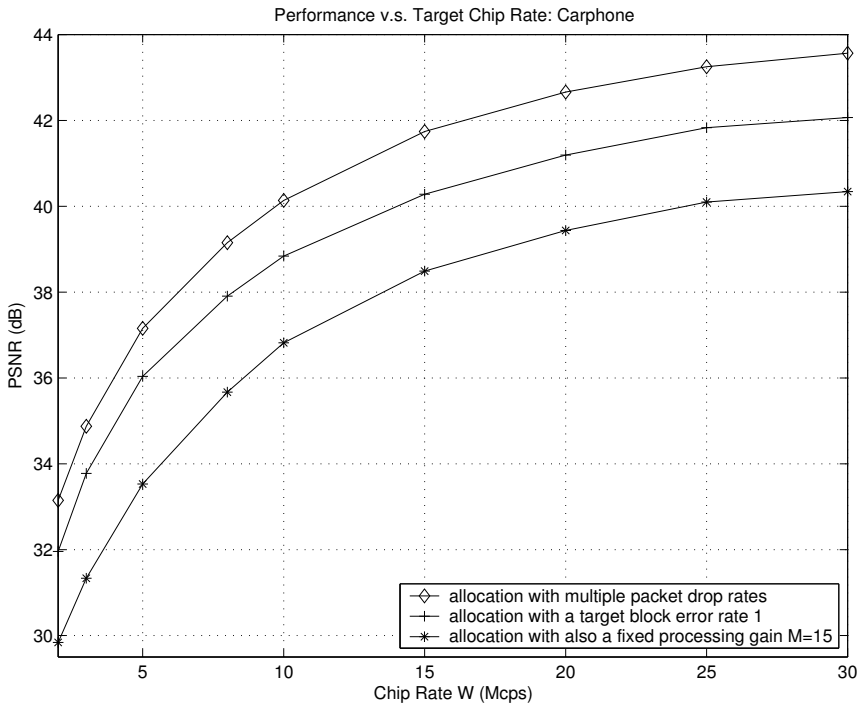


Figure III.6: PSNR performance versus target transmission chip rate, Carphone QCIF, and $E_c/N_0 = -8$dB.

-18dB to 0dB, for "Carphone" QCIF at the transmission chip rate W=15Mcps. The system with multiple packet drop rates outperforms the one with a fixed target packet rate by about 1.4 to 1.7dB, and outperforms the one with both fixed target packet error rate and fixed spreading gain by about 3.2 to 4.8dB. Also, the benefit of a higher E_c/N_0 diminishes when $E_c/N_0 \geq -8$dB.

During the simulations, it was observed that for the same test sequence, and with other conditions the same, operating at a higher chip rate usually leads to a greater use of low target packet error rates. This is because the larger available bandwidth improves the performance by increasing both the source encoding accuracy and the error correction capability. Fig. III.10 illustrates this trend, by showing the percentage of total packets that end up employing the corresponding target drop rate in the bandwidth

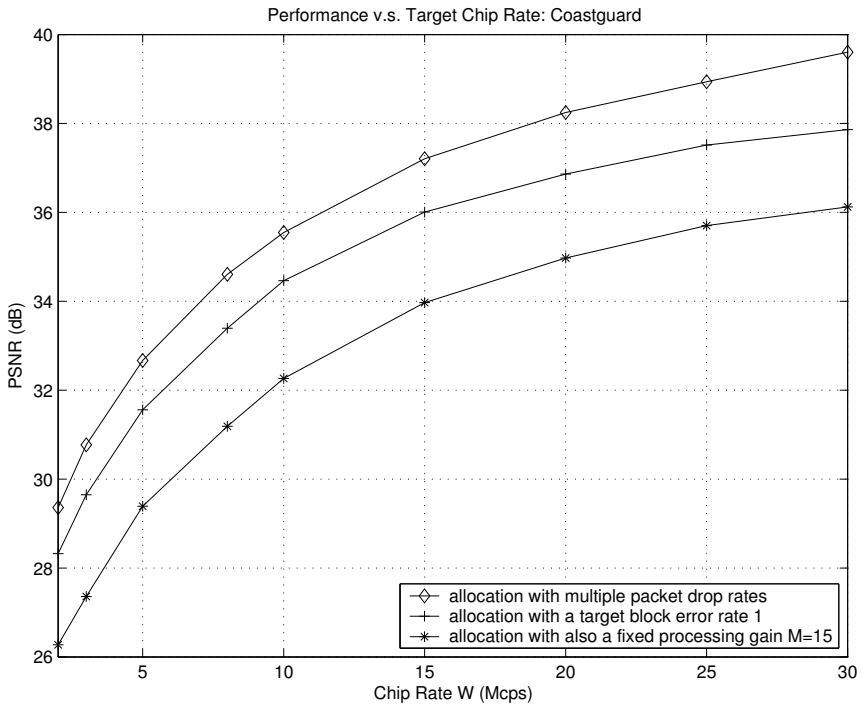


Figure III.7: PSNR performance versus target transmission chip rate, Coastguard QCIF, and $E_c/N_0 = -8$dB.

allocations for "Carphone" QCIF with $E_c/N_0 = -8$dB, operating at chip rate equal to 2Mcps, 15Mcps and 30Mcps. Similarly, under the same scenario, increasing E_c/N_0 also leads to a greater use of low target packet error rates.

In Fig. III.11, histograms for the target drop rate are shown for two high motion sequences "Carphone" and "Coastguard", and two low motion sequences "Mother and Daughter" and "Akiyo", all at chip rate 15Mcps and $E_c/N_0 = -8$dB. It is seen that, for this particular scenario, the low motion sequences use relatively more high packet error rates, and the high motion sequences use relatively more low packet error rates. This may be because low motion sequences are more tolerant of packet errors due to the ease of concealment.

We also compare our system with a recent system [28] which did the band-

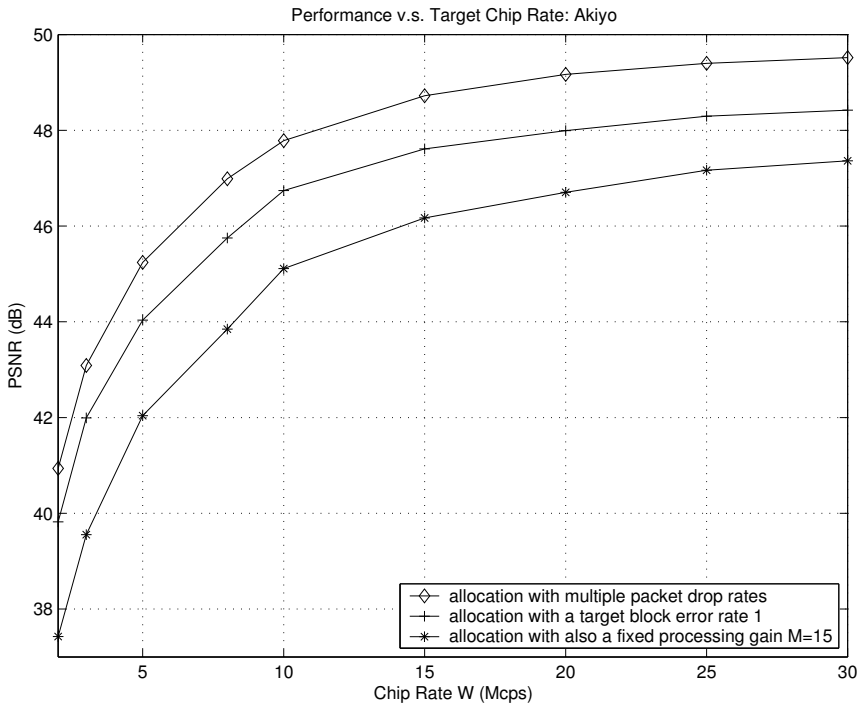


Figure III.8: PSNR performance versus target transmission chip rate, Akiyo QCIF, and $E_c/N_0 = -8$dB.

width allocation for scalable video transmission over single or dual DS-CDMA channels with universal rate-distortion characteristics (URDC) [19, 28]. In [28], bandwidth allocations were chosen from a finite set of possible source bit rates, a finite set of possible coding rates, and a finite set of possible spreading lengths. This was done once for the whole transmission period. We compare the performance of our system with the results given in Fig. 1 of [28], where the comparison system is operated over block fading channels with the number of resolvable paths L=3 and 8 interferers. To keep the same total transmission power, a single 8dB channel and two 4.98dB channels were compared. The simulation results were for "Foreman" at 10fps, and the performance was measured by MSE (which we convert to PSNR). As shown in Fig. III.12, our system with multiple packet drop rates outperforms the comparison systems of both single and dual channels

over most chip rates.

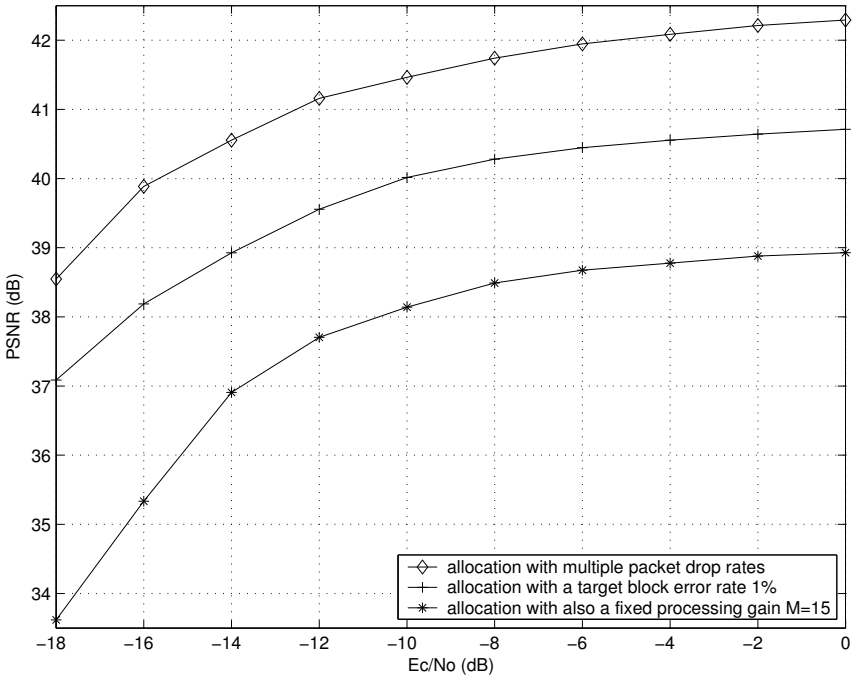


Figure III.9: PSNR performance versus E_c/N_0, Carphone QCIF, and chip rate 15Mcps.

Finally, the system performance with imperfect channel estimation is presented. In Fig. III.13, PSNR versus the number of pilot bits per packet, N_p, is shown, for "Carphone" QCIF at chip rate 15Mcps and $E_c/N_0 = -8$dB, with the proposed algorithm using the channel estimators F and F^*. The top dashed line is an upper bound on performance, under the assumption that a genie informs the transmitter of the actual channel conditions ($N_p = 0$). It is seen that the estimator F^* outperforms the biased estimator F, especially when N_p is small. The optimal choices of N_p under this scenario are 85 and 105 for the algorithm using F^* and F, respectively, which accounts for about 17.5% and 20.8% of the total packet length, respectively. (The number of information bits is $y = 400$ per packet.) It is also seen that the performance degradation at the optimal N_p is about 2.15dB for the system using F^* , and about 3.2dB for the system using

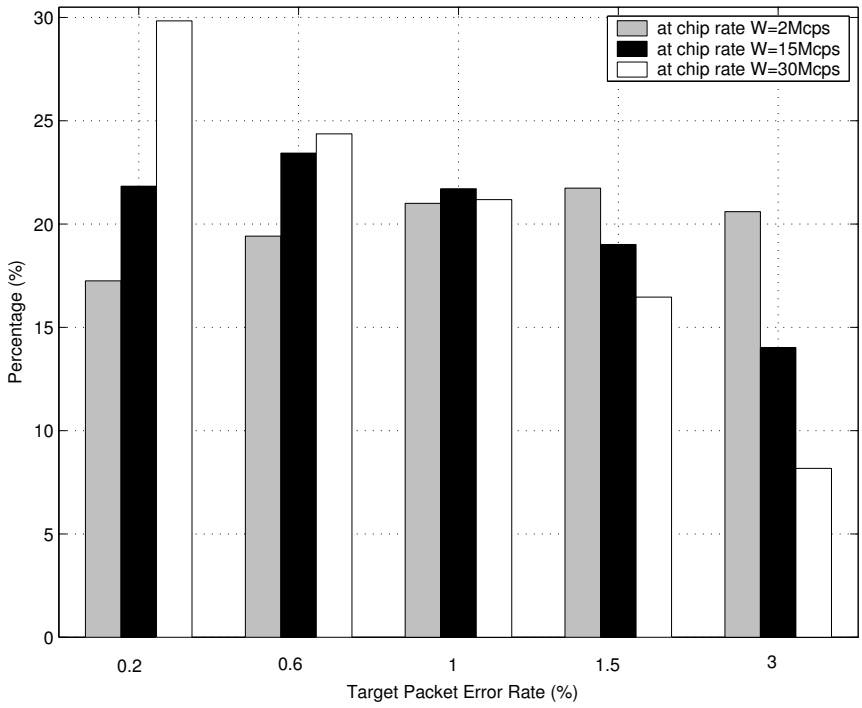


Figure III.10: Percentage of the total packets that employ the corresponding target drop rate in bandwidth allocation, for Carphone QCIF at $E_c/N_0 = -8$dB, and chip rate 2Mcps, 15Mcps and 30Mcps.

F. This suggests that the simple pilot averaging technique for channel estimation is not very efficient, and more elaborate techniques (e.g., Wiener filtering) may be needed.

III.H Conclusions

In this chapter, we proposed a robust video transmission scheme with efficient bandwidth allocation among source coding, channel coding, and spreading, that operates at the packet level. The algorithm is proposed for use in a CDMA network, and can be extended to operate over a tandem channel with both packet erasures and burst bit errors. The optimality problem among three parameters is solved using a two-step tradeoff strategy, by introducing the target packet drop rate at the decoder.

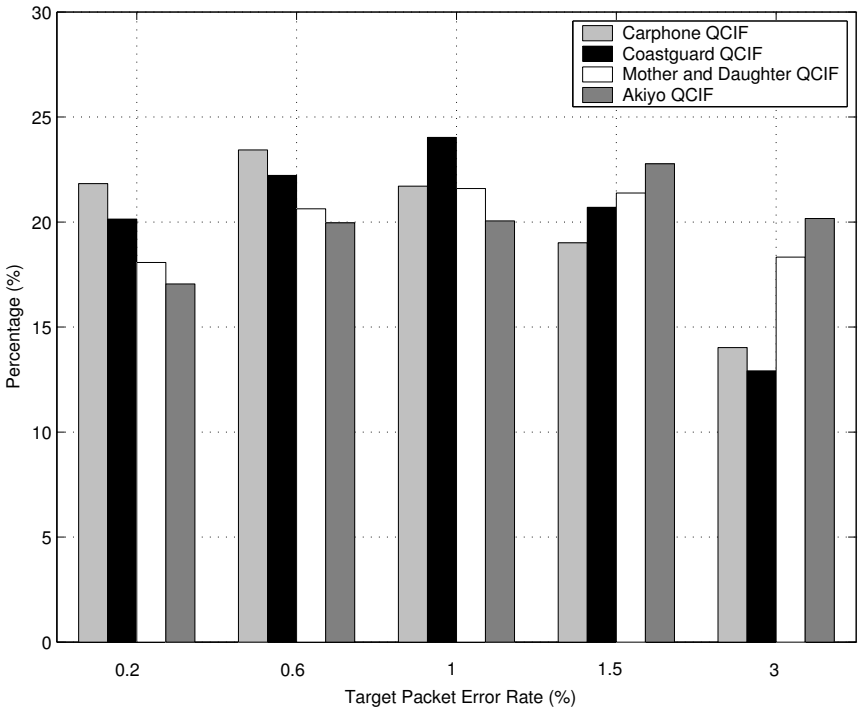


Figure III.11: Percentage of the total packets that employ the corresponding target drop rate in bandwidth allocation, at chip rate 15Mcps and $E_c/N_0 = -8$dB, for Carphone QCIF, Coastguard QCIF, and Mother and Daughter QCIF, and Akiyo QCIF.

Based on the statistics of the received signal, and a theoretical bound on the packet error rate due to the wireless link, p_p, we found that, conditioned on appropriate system parameters and channel conditions, p_p is fully determined by the channel coding rate, r_c, and spreading gain, M. Furthermore, the algorithms to trade off channel coding and spreading for a given packet drop rate p_p^* were derived for systems both with fixed and with variable M. It was also shown that for a system with variable r_c and M, in a scenario where the number of users is large so that the self-interference is negligible, the transmitter should use the lowest rate channel code and only vary M to adapt to different channel conditions.

As to the video source encoding, we used the modified-ROPE algorithm [6,

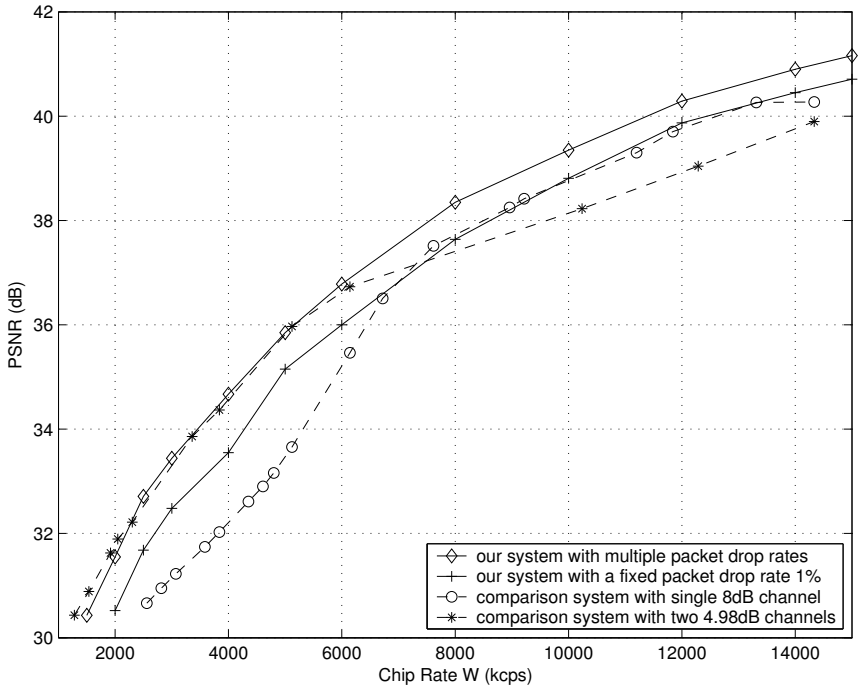


Figure III.12: PSNR performance versus target transmission chip rate for our system and comparison system, Foreman QCIF, frame rate 10fps.

22], and extended the rate control scheme to operate on chips, rather than bits, so that the effect of variable spreading gain could be accounted for. Likewise, we extended the rate-distortion framework for optimal mode selection (and QP selection) to operate at the chip rate, rather than the bit rate.

We then developed an algorithm with a two-step tradeoff for choosing the bandwidth allocation parameters. The first step was to trade off between channel coding and spreading, given a target packet drop rate, such that we maximized the source bit rate. The second step was to trade off between the packet drop rate and the source bit rate, based on the channel coding and spreading determined by the first step, such that the overall expected distortion per time unit was minimized. This optimization process was done for each packet, and incorporated the effects of both the changing channel

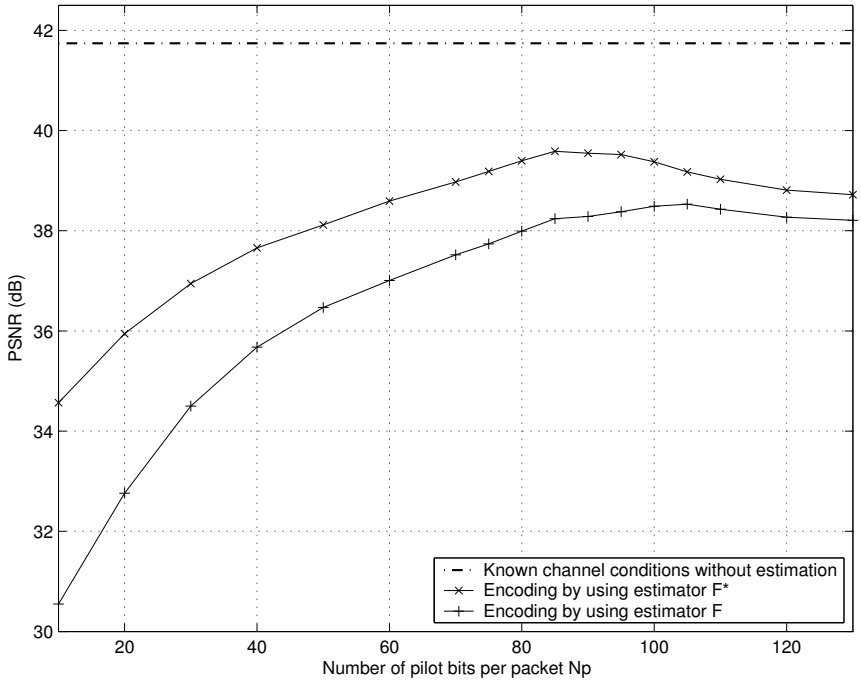


Figure III.13: PSNR performance versus the number of pilot bits per packet N_p, for "Carphone" QCIF at chip rate 15Mcps and $E_c/N_0 = -8$dB, with the proposed bandwidth allocation algorithm using the channel estimators F and F^*. The reference line is the system performance with known channel conditions.

characteristics and the current video content. The effect of imperfect channel estimation on system performance was also studied.

Results were simulated for a variety of sequences. It was shown that the proposed system which allows all components to vary offered about a 1.4dB gain over a scheme using a fixed packet drop rate, and up to 4dB gain over a scheme using a fixed spreading gain. The scheme also outperformed a comparable system in the literature [28] that adapted source coding rate, channel coding rate, and spreading gain, but which did not operate on a packet basis.

III.I Appendix: Calculation of $E\{F\}$ and $Var\{F\}$

In this appendix, we derive the conditional mean and variance of the term $F = \sum_{l=1}^{L} |\hat{c}_{0,l}|^2$, where the $\hat{c}_{0,l}$s are i.i.d conditional complex Gaussian rvs, with conditional mean $E\{\hat{c}_{0,l}\} = c_{0,l} \ = \alpha_{0,l} e^{j\varphi_{0,l}}$ and variance σ_c^2, as given by (III.23). Thus, F has a conditional non-central chi-square distribution with $2L$ degrees of freedom. The conditional mean of F is given by

$$\begin{aligned} E\{F\} &= \sum_{l=1}^{L} E\{|\hat{c}_{0,l}|^2\} \\ &= \sum_{l=1}^{L} \left((E\{|\hat{c}_{0,l}|\})^2 + \sigma_c^2 \right) \\ &= (\sum_{l=1}^{L} \alpha_{0,l}^2) + L\sigma_c^2, \end{aligned} \tag{III.28}$$

and the conditional variance

$$\begin{aligned} Var\{F\} &= E\{F^2\} - (E\{F\})^2 \\ &= \sum_{l=1}^{L} E\{|\hat{c}_{0,l}|^4\} + \sum_{l_1=1}^{L} \sum_{l_2=1, l_2 \neq l_1}^{L} \\ &\left\{ E\{|\hat{c}_{0,l_1}|^2 |\hat{c}_{0,l_2}|^2\} - \left(\sum_{l=1}^{L} E\{|\hat{c}_{0,l}|^2\} \right)^2 \right\}. \end{aligned} \tag{III.29}$$

Each $\hat{c}_{0,l}$ can be expressed as $\hat{c}_{0,l} = \hat{c}_{0,l}^Q + j\hat{c}_{0,l}^I$, where $\hat{c}_{0,l}^Q \sim N(\alpha_{0,l} cos(\varphi_{0,l}),$ $\sigma_c^2/2)$, and $\hat{c}_{0,l}^I \sim N(\alpha_{0,l} sin(\varphi_{0,l}),\ \sigma_c^2/2)$. Also, $\hat{c}_{0,l}^Q$ and $\hat{c}_{0,l}^I$ are conditionally independent, because, from (III.21), $\hat{c}_{0,l}$ is the average of three components, among which $\hat{x}_s$ is a real number, $\hat{x}_N$ is conditionally a symmetric complex Gaussian rv, and $\hat{x}_{I_m}$ is an

asymptotically, conditionally symmetric, complex Gaussian rv.

It is straightforward to show that for any Gaussian rv $x \sim (\mu, \sigma^2)$, $E\{x^3\} = \mu^3 + 3\mu\sigma^2$, and $E\{x^4\} = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4 = \left(E\{x^2\}\right)^2 + 4\mu^2\sigma^2 + 2\sigma^4$. Also, $E\{(\hat{c}_{0,l}^Q)^2\} + E\{(\hat{c}_{0,l}^I)^2\} = \alpha_{0,l}^2 + \sigma_c^2 = E\{|\hat{c}_{0,l_1}|^2\}$. Thus, $E\{|\hat{c}_{0,l}|^4\}$ is given by

$$\begin{aligned}
E\{|\hat{c}_{0,l}|^4\} &= E\left\{\left((\hat{c}_{0,l}^Q)^2 + (\hat{c}_{0,l}^I)^2\right)^2\right\} \\
&= E\{(\hat{c}_{0,l}^Q)^4\} + E\{(\hat{c}_{0,l}^I)^4\} + \\
&\quad 2E\{(\hat{c}_{0,l}^Q)^2\,(\hat{c}_{0,l}^I)^2\} \\
&= (E\{(\hat{c}_{0,l}^Q)^2\})^2 + 2\sigma_c^2\alpha_{0,l}^2 cos^2(\varphi_{0,l})) + \sigma_c^4/2 \\
&\quad + (E\{(\hat{c}_{0,l}^I)^2\})^2 + 2\sigma_c^2\alpha_{0,l}^2 sin^2(\varphi_{0,l})) \\
&\quad + \sigma_c^4/2 + 2E\{(\hat{c}_{0,l}^Q)^2\}E\{(\hat{c}_{0,l}^I)^2\} \\
&= 2\sigma_c^2\alpha_{0,l}^2 + \sigma_c^4 + \left(E\{(\hat{c}_{0,l}^Q)^2\} + E\{(\hat{c}_{0,l}^I)^2\}\right)^2 \\
&= 2\sigma_c^2\alpha_{0,l}^2 + \sigma_c^4 + (E\{|\hat{c}_{0,l_1}|^2\})^2. \qquad \text{(III.30)}
\end{aligned}$$

Plugging (III.30) into (III.29), we get

$$\begin{aligned}
Var\{F\} &= \sum_{l=1}^{L}\left(2\sigma_c^2\alpha_{0,l}^2 + \sigma_c^4\right) + \sum_{l=1}^{L}(E\{|\hat{c}_{0,l_1}|^2\})^2 \\
&\quad + \sum_{l_1=1}^{L}\sum_{l_2=1, l_2\neq l_1}^{L} E\{|\hat{c}_{0,l_1}|^2|\hat{c}_{0,l_2}|^2\} \\
&\quad - \left(\sum_{l=1}^{L} E\{|\hat{c}_{0,l}|^2\}\right)^2 \\
&= 2\sigma_c^2\left(\sum_{l=1}^{L}\alpha_{0,l}^2\right) + L\sigma_c^4. \qquad \text{(III.31)}
\end{aligned}$$

Acknowledgement

This chapter, in full, is a reprint of the material as it appears in Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Error Resilient Video Communications over CDMA Networks with a Bandwidth Constraint," which will appear in the *IEEE Transactions on Image Processing*, Oct. 2006. I was the primary author, and the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for this paper.

IV

Delay Allocation for Wireless Video Communications

IV.A Introduction

Delay partitioning is a fundamental tradeoff problem in the cross-layer design of a communications system. This problem is especially important in realtime video communications such as video conferencing or video telephone, in which there exists a tight end-to-end delay constraint. For example, interactive video telephone should have a maximum end-to-end delay of no more than around 300 msec. Once the receiver begins displaying the received video, the display process must continue without stalling. In other words, in order to be useful, frame data entering the source encoder at time t must be displayed at the decoder by time $(t + T)$, where T is the delay constraint, that is, an upper bound for end-to-end delay of the system. In addition, the available data rate on the channel is constrained by the available bandwidth.

In [29] and [30], the design of rate control schemes for low delay video transmissions was studied for a noiseless channel. In [31] and [32], the efficient design of an interleaver for a fading channel was investigated. In [33], specific tandem and joint source-channel coding strategies with complexity and delay constraints were analyzed and compared. In [34]-[35], delay-constrained wireless video transmission schemes were proposed for different application scenarios. In [36] and [37], tradeoffs between delay and video compression efficiency were discussed for a motion-compensated temporal filtering (MCTF) video codec, and for hierarchical bi-directional (B-frames) schemes, respectively. In [38], the tradeoff between the long-term average transmission power and the average buffer delay incurred by the traffic was analyzed mathematically over a block-fading channel with delay constraints. And in [39], the tradeoff between the network capacity and the end-to-end queueing delay was studied for a mobile ad hoc network.

In this literature, either design strategies with delay constraints were investigated without considering any tradeoff issue, or certain tradeoff problems with delay constraint were discussed for different purposes and contexts than those in this chapter. In this chapter, we study delay partitioning for video communications over a Rayleigh fading channel. In particular, we focus on the delay allocation between the source encoder buffer and the interleaver as we vary various parameters, such as the motion of the video content, the rate of variation of the channel, the end-to-end delay constraint, the channel bit rate, and the channel code rate.

The system model we study is shown in Fig. IV.1. Typically, video frames

arrive at the video encoder at a constant frame rate. The frames are compressed to a variable bit stream and passed on to the video encoder output buffer from which bits are drained at a constant rate. To protect against channel errors, forward error coding (FEC) is employed on the compressed bitstream coming out of the encoder buffer. This is followed by interleaving to provide robustness to channel fading. Finally, the bit stream coming out of the interleaver is modulated and sent over the wireless channel. At the receiver, the bitstream is demodulated, de-interleaved, decoded, and then passed on to the video decoder input buffer (henceforth called the decoder buffer). The video decoder extracts bits from the decoder buffer at a variable rate to display each frame at its correct time and at the same constant frame rate at which they were available to the video encoder. A rate control mechanism is used at the video encoder to control the number of bits allotted to each frame so that the encoder buffer and the decoder buffer never overflow or underflow, while maintaining acceptable video quality at all times. Note that we assume there is no video encoder input buffer, and no video decoder output buffer, hence the video encoder output buffer and the video decoder input buffer are called the encoder buffer and decoder buffer, respectively, throughout this chapter.

The chapter is organized as follows: In Section IV.B, the system model is introduced in detail. In Section IV.C, we formulate the delay partitioning problem mathematically, and end up with a relationship among the three main delay components, i.e., source encoding buffer delay, interleaving delay, and channel decoding delay, under a delay constraint. Simulation results of the tradeoff between the source encoder buffer and the interleaver are shown and analyzed in Section IV.D, for different video

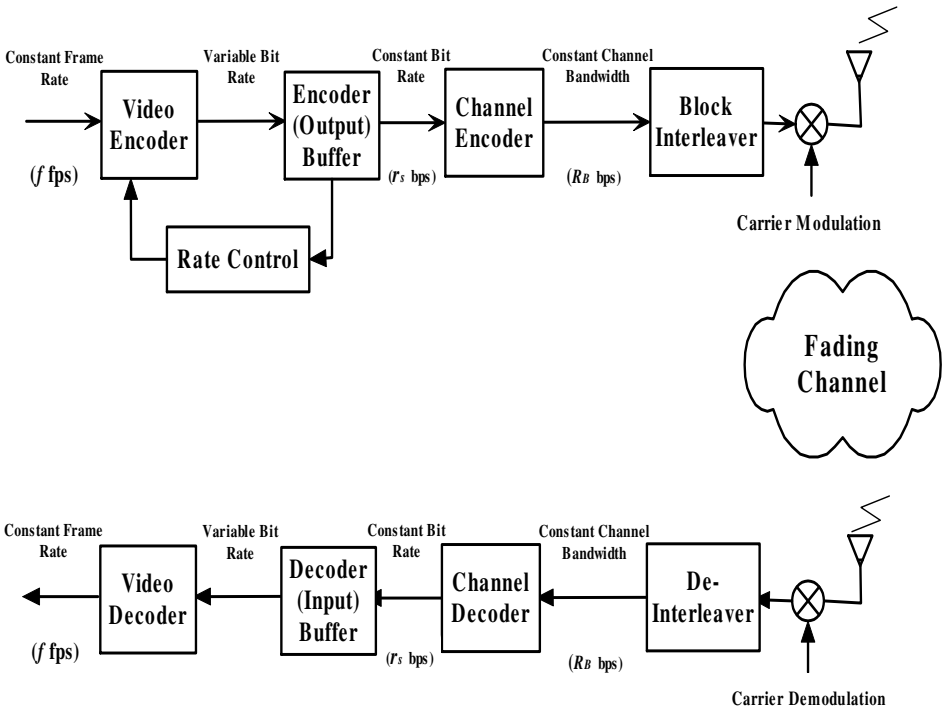


Figure IV.1: System overview.

sequences over Rayleigh fading channels. In particular, we study how the tradeoff will be affected by the motion of the video content, the rate of variation of the channel, the delay constraint and the channel bit rate. Lastly, Section IV.E concludes the chapter.

IV.B System Model with Delay Constraint

In this section, we will discuss the components in Fig. IV.1 in detail.

IV.B.1 Source Coding

In real time video communications, the end-to-end delay for transmitting video data needs to be very small, particularly for interactive two-way applications such as video conferencing and gaming. Since H.263 is the video codec standard designed for

low delay and low bit rate applications, we choose it as the source codec here. Video data enters the H.263 source encoder at a constant rate of f frames per second (fps), where it first undergoes block based motion compensated (MC) prediction, followed by DCT transformation of the residual block. The DCT coefficients are quantized by appropriately choosing the quantization parameter, and the quantized values are then run length and Huffman coded. Assume the transmission bit rate is R_B bits per second (bps), and the source coded bit stream leaves the encoder buffer at r_s bps.

Whenever a frame occupies more than r_s/f bits, bits will accumulate in the source encoder buffer and increase the encoder buffer delay experienced by the incoming bits. If this trend continues for several frames, the buffer may fill up because the buffer size is limited. When the number of bits in the buffer is more than a predetermined threshold, it will lead to frame skipping as will be discussed later. On the other hand, whenever a frame occupies less than r_s/f bits, the encoder buffer fullness level decreases. If this trend continues for several frames, the encoder buffer may run empty, thereby wasting channel bandwidth.

By sensing the buffer fullness and keeping an estimate of the available bit budget, the rate control chooses the quantization step-size and seeks to prevent buffer overflow and underflow while maintaining acceptable video quality. If either the remaining bit budget is small or the buffer is getting full, the rate control resorts to coarse quantization. If either the remaining bit budget is large or the buffer is getting empty, the quantization step size is reduced (i.e., fine quantization). A large delay budget for the source encoder allows the use of a large encoder buffer, which tends to result in higher

quality video because the rate control has more freedom. Typically, the increased number of bits resulting from finely encoding a complex scene can be easily accommodated in the large buffer. However, when tight delay constraints exist, the system must operate with a small encoder delay budget, or equivalently a small encoder buffer, which tends to reduce the quality of the video, as the functioning of the rate control is more constrained. In extreme cases, the encoder buffer may fill up several times, leading to loss of data through repeated frame skipping.

On the decoder side, the incoming stream of video data is buffered in a source decoder buffer. Once the source decoder starts displaying the frames, the delay constraint becomes operational. If T denotes the upper bound for end-to-end delay of the system, a frame entering the encoder at time t must be displayed at the decoder at time $(t + T)$, and all the video data corresponding to this particular frame must be available at the decoder accordingly. A video frame that is not able to meet its delay constraint is useless and is considered lost. We assume that the source decoder has knowledge of the frame numbers skipped by the source encoder and that it holds over the immediately preceding displayed frame and displays it in place of the skipped frame.

In H.263, the rate control performs the bit allocation by selecting the encoder's quantization parameter for each block of 16x16 pixels. We choose the TMN-8 (Test Model Number 8) rate control [29, 30] recommended for low delay applications. The TMN-8 rate control is a two step approach: a frame layer control first selects a target bit count for the current frame, followed by a macroblock (MB) layer rate control which selects the quantization step size for each MB in the frame. The TMN-8 rate control

has a threshold for frame skipping. Whenever the number of bits in the encoder buffer increases beyond this threshold, typically one frame is skipped so that the number of bits in the buffer falls below the threshold. For each skipped frame, buffer fullness reduces by r_s/f bits. We assume the first frame in the video sequence is coded as an *I* frame, and all subsequent frames as *P* frames, since this is a common strategy for video communications with a tight delay budget. We also assume the *I* frame is transmitted error free to the decoder and the decoder does not start the display until the first *I* frame is completely buffered. The rate control starts with the first *P* frame. Once the *I* frame is displayed, the delay constraint becomes operational and all subsequent frames must meet their delay constraint.

From the point of view of the system engineer, the parameter of interest is the threshold for frame skipping (denoted by S_t). However, for the hardware engineer, the buffer size (denoted S_b) is more important. These two quantities are closely related, as explained now. As shown in Fig. IV.2, we modify the rate control such that, while encoding the frame, if the buffer fullness level exceeds S_t, the remaining MBs of the frame are all skipped. If a particular sequence comprising N_{skip} bits is used to inform the decoder of this situation, the buffer size required is $S_b = S_t + N_{skip}$. Because N_{skip} is usually much smaller than S_t, for example $N_{skip} = 24$ in our system while S_t is at least several thousands, for simplicity, we assume the threshold for frame skipping and the buffer size are the same and are equal to S, i.e., $S_b = S_t = S$.

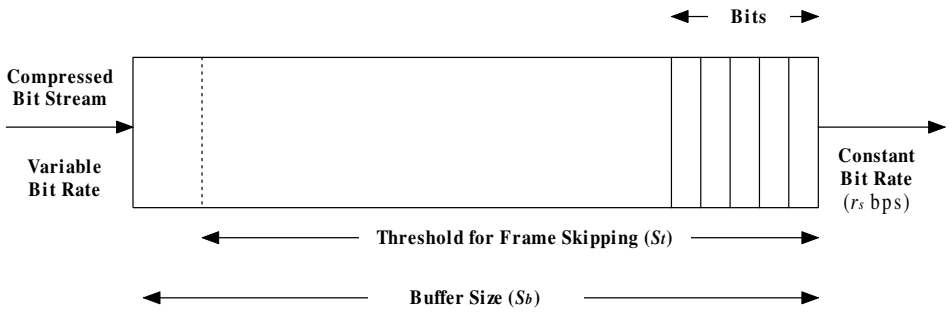


Figure IV.2: Source Encoder Buffer.

IV.B.2 Channel Coding

The information bitstream coming out of the source encoder buffer is channel coded using a rate-compatible punctured convolutional (RCPC) code with rate r_c and constraint length ν [11]. At the receiver, the Viterbi algorithm is used to find the best candidate in the trellis for the received bitstream. The delays encountered in the channel encoder and decoder are called the channel encoding delay and the channel decoding delay. Together these make the delay budget of channel coding. When using a convolutional code with constraint length ν, the channel decoding delay is approximately the decision depth of the Viterbi decoder, which is about 5ν in bits. The decision depth for punctured convolutional codes is generally longer. If the puncturing period for the RCPC code is P, the decision depth can be bounded by $5P\nu$ [40]. We note, however, that when using channel encoding schemes such as turbo coding that require iterative decoding at the receiver, the channel coding delay budget may use up a significant portion of the overall delay budget.

Bandwidth is a major resource shared between source coding and channel coding. A bandwidth constraint limits the available rate on the channel. Allocating

more bandwidth to the source encoder allows more information from the source to be transmitted, resulting in better quality video. However, the bandwidth available for channel coding is reduced, leading to increased errors on the channel and thus a reduced probability of achieving high video quality. Let R_B bps be the total available rate on the channel, and r_s and r_c be the average source coding rate and the channel code rate, respectively. Then the bandwidth constraint is expressed as [15, 41]

$$\frac{r_s}{r_c} = R_B. \tag{IV.1}$$

IV.B.3 Interleaving and Fading Channel Model

We consider coherent BPSK over a flat fading channel, where flat fading means that there is a constant gain across the bandwidth of the received signal. Therefore, the effect of the channel is a multiplicative gain term on the received signal level. We use the channel model suggested by Jakes [27], in which the envelope of the fading process is assumed to be Rayleigh distributed. The Doppler spectrum is given by

$$S(f) = \frac{1}{\sqrt{1-(f/f_D)^2}}, \tag{IV.2}$$

where f_D is the Doppler frequency and is given by $f_D = f_c v/c$, where f_c is the carrier frequency, v is the mobile velocity, and c is the speed of light. The covariance function of the fading process for this channel model can be shown to be given by the first order Bessel function, namely

$$R_\alpha(\tau) = J_0(2\pi f_D|\tau|), \tag{IV.3}$$

where τ is the time separation between the two instances when the channel is sampled. Thus, the correlation between two consecutive symbols with separation T_s is $J_0(2\pi f_D T_s)$, where T_s is the symbol time. The product $f_D T_s$ is usually called the normalized Doppler frequency.

Error control coding works well when the code symbols used in the decoding process are affected by independent channel conditions. Correlated fading is one of the sources of channel memory on the land mobile channel. Interleaving is used to break up channel memory, and it is an essential element in the design of error control coding techniques for the land mobile channel. A block interleaver formats the encoded data in a rectangular array of N_1 rows and N_2 columns. The code symbols are written in row-by-row and read out column-by-column. On the decoder side, the received symbols are first de-interleaved before they enter the decoder. As a result of this reordering, the fading samples of two consecutive symbols entering the decoder are actually $N_1 T_s$ apart in time, and the correlation between two consecutive channel instances is now given as $J_0(2\pi f_D N_1 T_s)$. The parameter N_1 is often referred to as the depth of the interleaver.

The inverse of the normalized Doppler frequency roughly equals the coherence time, $N_{coh} = 1/(f_D T_s)$, of the channel in bits, and is a measure of the number of consecutive bits over which the channel remains correlated. The amount of interleaving required depends on the channel. If the channel is slower, the coherence time is larger and consequently a larger interleaver is required. When there is no limit on the size of the interleaver, perfect interleaving can be achieved for mobile channels, which ensures that the fading envelopes are uncorrelated. However, both interleaving and de-

interleaving introduce delay in the system, called the interleaving delay. Both of these delays are equal to $N_1 N_2 T_s$ seconds. In a practical system, the interleaving delay budget is constrained not only by the overall delay budget, but also by the delay budget necessary for the robust functioning of the source coding and the channel coding.

For convolutionally coded systems, the dimensions of the interleaver are chosen to maximize the interleaving depth N_1, which should ideally be N_{coh} to ensure nearly independent fading conditions for consecutive symbols. More important, N_2 should be chosen at least large enough to avoid the wrap around effect [42, 43, 44]. The wrap around effect means that the length of an error event exceeds the number of columns in the interleaver. This results in more than one symbol being affected by virtually the same channel conditions, and thus degrades performance. As a rule of thumb, the number of columns is chosen slightly larger than the length of the shortest error event of the code.

Interleaving, in conjunction with FEC, is a mechanism to achieve time diversity, where, by transmitting consecutive symbols sufficiently separated in time, nearly independent fading is ensured. As with any diversity technique, the performance improvement shows diminishing returns with increased diversity order. Note that the effective order of diversity is a nondecreasing function of N_1. Various rules of thumb are available in the literature to determine the interleaver depth sufficient to extract nearly independent fading case performance [31, 32].

In [31], simulations were used to demonstrate that fully interleaved performance is approximately achieved for BPSK over exponentially correlated channels when

the interleaver depth is chosen to satisfy $f_D T_s N_1 > 0.1$. This rule, however, does not apply to correlated fading channels with other auto-correlations, such as the Jakes' model. In [32], a simple figure of merit for evaluating the depth of the interleaver was obtained for Rician channels, and a variety of channel auto-correlation functions. However, as shown in our simulations, this figure of merit does not hold true for the Jakes' fading model with low κ factor (κ is the ratio of signal energy in direct and diffused signal components) Rician channels and the limiting Rayleigh fading case.

IV.C The Delay Constraint Formulation

The end-to-end delay constraint of each frame, T, is the upper bound to the delay that a frame may experience and still be able to be displayed on time, where by delay, we mean the time difference between when the video frame is captured for encoding and when it reaches the video decoder. Consider frame i captured at time t. Without loss of generality, we assume t to be zero. Further, we assume that each frame has the same number of MBs, and denote this number by M (e.g., for video with QCIF format, $M = 99$). We also denote the MB index by k ($k = 0, 1, 2, \cdots, M-1$), and we let $b_i(k)$ be the number of bits in the kth MB of the ith frame.

Frames arrive at the video encoder at some constant frame rate, thus the processor has to process each frame in the same amount of time, because we assume there is no video encoder input buffer. Each frame has the same number of MBs, and we assume each MB has to be processed in the same amount of time. At the video decoder, frames are displayed at a constant frame rate, and we assume there is no video decoder

output buffer.

For frame i to meet its delay constraint, the kth MB's decoding must begin at time $T-(M-k)T_d$, where T_d is the time required to decode a MB (source decoding only, i.e., excluding the FEC decoding) and is assumed to be the same for all MBs. Also, the kth MB becomes available for encoding only after time kT_e, where T_e is the encoding time of a MB (source encoding only, i.e., excluding the FEC encoding) assumed to be the same for all MBs. Thus, if the kth MB is to meet its decoding deadline, the following must be true:

$$T_{eb}(k) + T_{enc}(k) + T_{int}(k) + T_c(k) + T_{CH} + T_{dein}(k) + T_{dec}(k) + T_{db}(k) = T - (k+1)T_e - (M-k)T_d, \quad \text{(IV.4)}$$

where $T_{eb}(k)$ is the encoder buffer delay, i.e., the time the kth MB waits in the encoder buffer before it starts moving out to the channel encoder, $T_{enc}(k)$ is the FEC encoding delay for the kth MB, $T_{int}(k)$ is the delay caused by interleaving for the kth MB, $T_c(k)$ is the transmission time for the kth MB, T_{CH} is the channel propagation delay, assumed to be a known constant, $T_{dein}(k)$ is the delay caused by de-interleaving for the kth MB, $T_{dec}(k)$ is the channel decoding delay, and finally $T_{db}(k)$ is the decoder buffer delay for the kth MB, i.e., the time it waits in the decoder buffer before its decoding begins for display.

A few simplifications can be made. We have earlier explained the logic for assuming that the video encoding time, T_e, and the video decoding time, T_d, are the same for all MBs. We also assume they are equal to each other, which is essentially the

same as saying that the MBs arrive at the encoder buffer and depart from the decoder buffer as a stream with each MB spaced T_{MB} seconds apart, where $T_{MB} = 1/(Mf)$ and f is the frame rate. As a consequence of the above assumption, notice that the right hand side of (IV.4) becomes independent of k. We ignore the delay caused by channel encoding (i.e., $T_{enc}(k) \approx 0$), because it is negligible compared to the delay caused by channel decoding and the delay caused by source encoding. For Viterbi decoding of RCPC codes with constraint length ν and period P , the decoder has a latency of approximately $T_{dec}(k) = 5P\nu/B$ bits [11, 40]. Also, since we are assuming a rate r_c channel code and a fixed channel rate of R_B bps on the channel, the transmission time for the kth MB can be expressed as $T_c(k) = b_i(k)/r_s$. We assume that each MB has enough bits to span the width of the interleaver at least once, i.e., $b_i(k) \geq N_2$. The sum of the interleaving and the de-interleaving delays is then approximately given as $T_{int}(k) + T_{dein}(k) \approx 2N/B$. Incorporating all these simplifications, (IV.4) can be written as:

$$T_{eb}(k) + \frac{2N}{R_B} + \frac{b_i(k)}{r_s} + T_{CH} + \frac{5P\nu}{R_B} + T_{db}(k) = T - (M+1)T_{MB}. \quad \text{(IV.5)}$$

Furthermore, the term $b_i(k)/r_s$ is typically of the order of a few milliseconds. For example, with $r_s = 48$kbps, $f = 10$fps and $M = 99$, the average number of bits-per-MB is $b_i(k) \approx 50$, and thus $b_i(k)/r_s \approx 1$ msec. Because the delay budgets in the multimedia applications we study are typically equal to or greater than 100 msec, the term $b_i(k)/r_s$ can be neglected. Assuming a constant channel propagation delay T_{CH},

and noting that we need $T_{db}(k) > 0$ to guarantee the source decoder buffer does not run empty, (IV.5) can be rewritten as

$$T_{eb}(k) + \frac{2N}{R_B} + \frac{5P\nu}{R_B} \leq C, \tag{IV.6}$$

where $C = T - (M+1)T_{MB} - T_{CH}$, is a constant.

The encoder buffer delay experienced by each MB in each frame must satisfy the above inequality in order for the corresponding frame to meet its display deadline. As explained previously, the maximal number of source coded bits in source encoder buffer is equal to S, and they leave the buffer at a rate r_s bps, thus $T_{eb}(k) \leq S/r_s$. As a result, (IV.6) is always true whenever the following is true:

$$\frac{S}{r_s} + \frac{2N}{R_B} + \frac{5P\nu}{R_B} \leq C, \tag{IV.7}$$

where S/r_s can be viewed as the delay budget for source coding, $2N/R_B$ as the delay budget for interleaving and $5P\nu/R_B$ as the delay budget for channel decoding. As a result, the delay partitioning problem is to allocate the delay budget among these three components under the constraint (IV.7), such that the overall distortion of the video transmission is minimized.

IV.D Simulation Results and Discussion

IV.D.1 The Effect of Interleaver Depth on System Performance

An interleaver is important to remove the channel memory when error control codes designed for memoryless channels are applied to channels with memory. Before we consider the tradeoff in delay allocation in wireless multimedia, we first study the effect of interleaver design without a delay budget restriction.

The performance of an interleaver is governed by its interleaving depth N_1. As mentioned in Subsection IV.B.3, simulation results in [31] demonstrated that fully interleaved performance is approximately achieved for BPSK over exponentially correlated channels when $N_1 \geq 0.1N_{coh}$ is satisfied, and [32] further extended this result to Rician channels and a variety of channel auto-correlation functions by proposing a simple figure-of-merit for evaluating interleaver depth. Our simulations confirm this result for the Jakes' fading model with high κ factor Rician channels. In Fig. IV.3, we show simulation results for a system with a channel code of rate $r_c = 1/2$ and minimal distance $d_{min} = 10$, an interleaver with $N_2 = 100$ columns, and the Jakes' fading spectrum with $f_D T_s = 0.01$. The two bottom dashed lines are drawn for the Rician channel with $\kappa = 5$ (or 7dB), with interleaver depth $N_1 = 14$ and ideal interleaving (i.e., $N_1 = \infty$). These results match the results of Fig. 4 in [32], which illustrates that $N_1 = 14$, which is slightly larger than $0.1N_{coh} = 10$, gives performance close to ideal (infinite) interleaving.

However, further simulations illustrate that this figure-of-merit does not hold true for the Jakes' fading model with low κ factor Rician channels. Lowering the κ

factor of the Rician channel makes the fading more severe, and the channel is Rayleigh when $\kappa = 0$ (or $-\infty$dB), where the direct signal component is totally absent. Clearly with decreasing κ, the performance degrades, and a larger interleaver depth may be required. In Fig. IV.3, the performance when $\kappa = 0$ is shown, by utilizing interleavers with depth $N_1 = 14$, $N_1 = 0.7N_{coh} = 70$, $N_1 = N_{coh} = 100$, and infinite interleaving. As seen from the four top plots, substantial gains in performance are achieved over $N_1 = 14$, with an improvement by an order of magnitude, especially at middle and high SNR. On the other hand, although the performance improves significantly from $N_1 = 14$ to $N_1 = 70$, there is not much gain in further increasing after $N_1 \geq 70$. This is the typical characteristic of any diversity system, where with increasing diversity order, the improvement in performance shows diminishing returns.

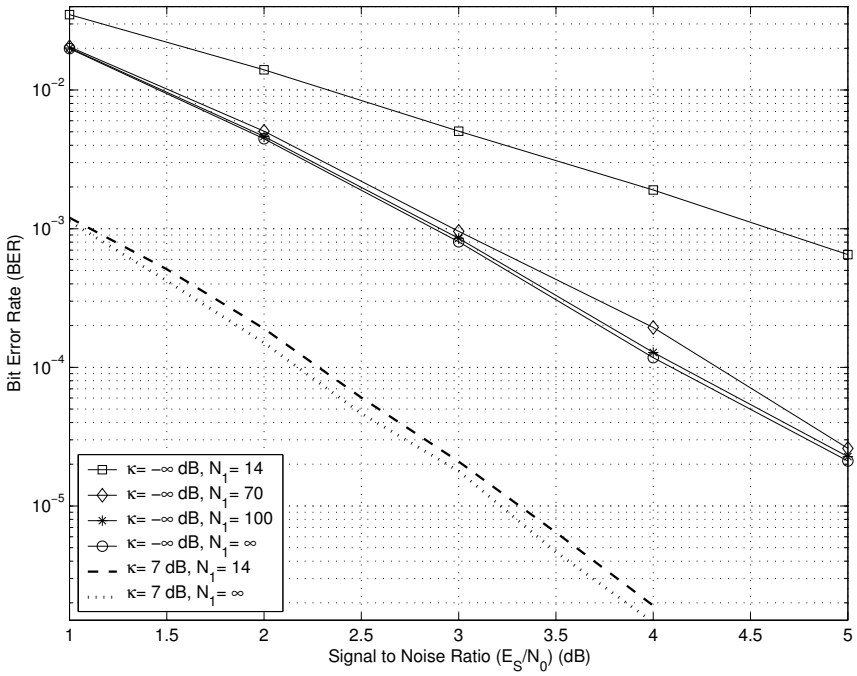


Figure IV.3: Performance comparison for evaluating the effect of interleaver depth: bit error rate (BER) versus the signal to noise ratio (E_s/N_0), channel code with rate $r_c = 1/2$ and $d_{min} = 10$, interleaver with $N_2 = 100$, and the Jakes' fading spectrum with $f_D T_s = 0.01$.

Fig. IV.4 further illustrates this point, by showing bit error performance versus interleaver depth, with a convolutional channel code having $r_c = 1/3$, constraint length $\nu = 6$ and $d_{min} = 14$ [11, 24], over a Rayleigh fading channel with $f_D T_s = 0.005$ (i.e., $N_{coh} = 200$). N_2 is fixed to be 16, which is slightly greater than d_{min} [42, 43]. We again note the sharp fall in bit error rate (BER) as N_1 increases from 0 to 80, and that the performance begins to flatten out around $N_1 = 140$ onwards, which is again the depth corresponding to $0.7N_{coh}$.

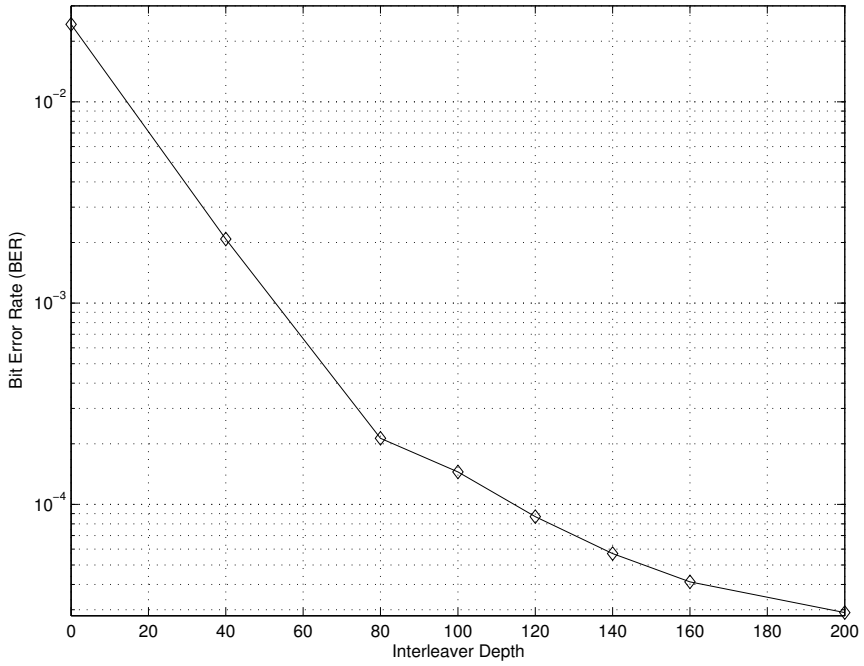


Figure IV.4: Bit error rate (BER) versus the interleaver depth (N_1), $E_s/N_0 = 3$dB, channel code with rate $r_c = 1/3$ and $d_{min} = 14$, interleaver with $N_2 = 16$, and the Jakes' fading spectrum with $f_D T_s = 0.005$.

As a result, our simulation results suggest that: for Rician channels with high κ factor, fully interleaved performance is approximately achieved when the interleaver depth $N_1 \geq 0.1N_{coh}$; while for Rician channels with low κ factor, in particular for a Rayleigh fading channel, fully interleaved performance is approximately achieved when

$N_1 \geq 0.7N_{coh}$. Also, the number of columns (N_2) should be greater than the minimal distance (d_{min}) of the channel code to avoid the wrap around effect.

IV.D.2 Delay Allocation between the Source Encoder Buffer and the Interleaver, for Fixed Delay Budget, Channel Bit Rate and FEC Code

We will discuss the delay allocation between the source encoder buffer and the interleaver in this and the next subsections. In all our simulations, we encoded QCIF size video sequences at $f = 10$ fps. Also, for all comparisons, we kept the ratio of the energy-per-coded bit to the noise power spectral density, E_s/N_0, constant at 3dB. For each set of system and channel parameters, we ran ten thousand realizations of the time-correlated Rayleigh fading channel, which were generated using the Jakes' model [27]. We computed the cumulative distribution function (CDF) of the average peak signal to noise ratio (PSNR), where PSNR is calculated by first averaging the mean square error (MSE) for the entire decoded video sequence, and then converting to PSNR. The system performance can be gauged once the CDF curves for each possible set of parameters in the set of interest are available. For example, Fig. IV.5 illustrates what the CDF curves could look like. Whenever two CDF curves do not intersect (e.g., curves C_1 and C_3 in Fig. IV.5), the lower curve is superior because it always has a higher probability of achieving any given average PSNR. When there are crossovers between two curves (e.g., curves C_1 and C_2 in Fig. IV.5), then one curve may be superior for one application but not for another. Comparison between the curves may then involve criteria such as minimizing the area under the curve, perhaps with some weighting. In this chapter, as

shown in Fig. IV.5, to evaluate the system performance, we adopted the criterion from [45] of minimum area under the CDF curve to the left of a certain threshold x_h defined later in this section, i.e., the value $\int_0^{x_h} F_c(x)dx$.

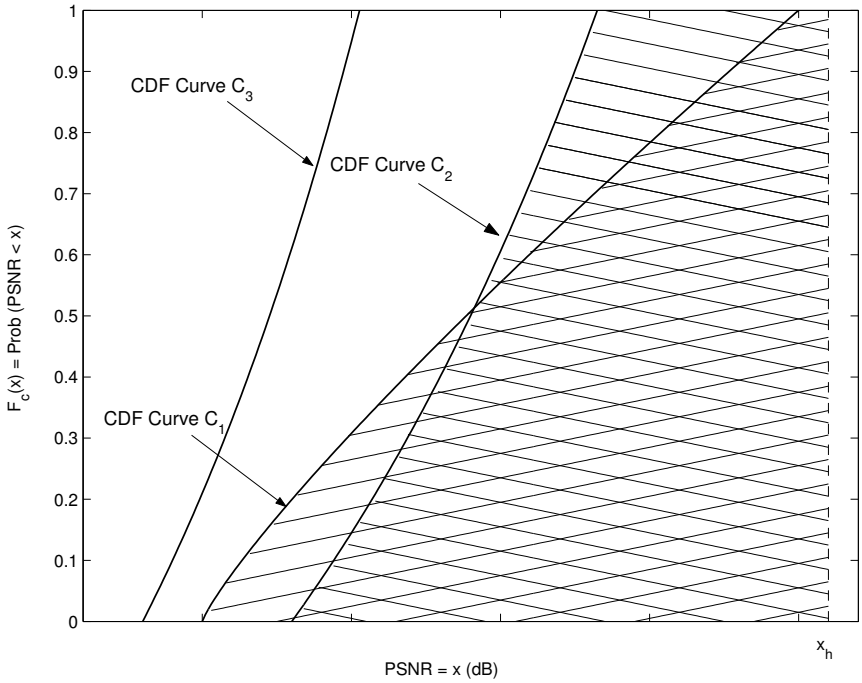


Figure IV.5: Comparison of PSNR CDF curves.

In this subsection, we analyze the delay partition between the source encoder buffer and the interleaver, for a fixed delay budget C, a given channel bit rate R_B and a fixed RCPC code with rate r_c. As explained in Section IV.B, the delay budget of the source encoder is determined by the threshold for frame skipping S. Given R_B and a RCPC code with rate r_c, the source coding rate, r_s, is determined by (IV.1), and the channel decoding delay, which is roughly equal to $(5P\nu/R_B)$, is also fixed. Under this scenario, increasing the delay budget of the source encoder comes at the cost of reducing the interleaver delay budget, i.e., using a smaller interleaver. In general, given the total delay budget C and channel bit rate R_B, the choice of S is affected by the

source encoding rate r_s and the video content, and the choice of interleaver depth N_1 is related to the channel fading characteristics (N_{coh} and channel model) and the video content. Therefore, we will focus on how this tradeoff will be affected by the motion of the video content, the rate of variation of the channel, the delay constraint and the channel bit rate.

In the following simulations, we used the rate $r_c = 1/3$ RCPC code with $\nu = 6$ and $d_{min} = 14$ [11, 24] for channel coding, and N_2 was fixed at 16. We ran the simulations with different parameters, for example, video sequences with high, medium or low motion, channels with fast, medium or slow fading, delay constraints that are tight, medium or loose , and different channel bit rates.

First, we assume a delay constraint $C = 150$ msec and a channel bit rate $R_B = 144$ kbps (thus $r_s = 48$ kbps). We simulated the system for a medium motion sequence "Foreman" QCIF over a medium fading channel with normalized Doppler frequency $f_D T_s = 0.005$ ($N_{coh} = 200$ bits). The candidate delay allocations we tested are summarized in Table IV.1, which were calculated based on (IV.7). Fig. IV.6 shows the CDF curves of the PSNRs for these delay allocations, and the areas under the CDF curves are plotted as the solid line in Fig. IV.7, where the x-axis is the interleaver delay budget expressed as a fraction of the total delay budget. It is seen that, as the interleaver delay budget increases from $N_1 = 67$, the system performance initially improves because of the increased diversity gain. However, the diversity gain shows diminishing returns, and at some point the reduction in source encoder delay budget starts having more of an effect, and the system performance degrades. It is seen that $(N_1 = 151, S = 5500)$

is the optimal delay allocation for this case, where N_1 is about $\frac{3}{4}N_{coh}$.

Table IV.1: Delay allocations for tradeoff between S and N_1, used in the simulations for Fig. IV.6 and Fig. IV.7.

Interleaver depth N_1 (in bits)	67	87	123	140	151	170	217
Source encoder buffer size S (in bits)	6400	6190	5800	5620	5500	5300	4800
Interleaver delay budget as a fraction of the total delay budget $(\frac{2N_1N_2}{R_B})/C$ (in %)	9.93	12.89	18.22	20.74	22.37	25.19	32.15

To see the effect of the motion of the video content, we also simulated a very high motion sequence "Mobile" QCIF and a very low motion sequence "Akiyo" QCIF, with the other parameters the same ($C = 150$ msec, $R_B = 144$ kbps and $N_{coh} = 200$). The system performances measured by the areas under the CDF curves are plotted and compared in Fig. IV.7, where the threshold value x_h was set to be the maximal PSNR value observed among all the realizations in the test for that individual video sequence. For example, in Fig. IV.6, the largest PSNR achieved by any of the systems is 33.01dB, so for the purposes of generating the curve corresponding to "Foreman" QCIF in Fig. IV.7, we compute the areas under the CDF curves and to the left of $x_h = 33.01$ for the curves in Fig. IV.6. Because different x_h values were used for the three curves corresponding to the three different video sequences, the performance comparison (i.e., y-axis values) is only meaningful within a curve, but not between different curves. It is observed that, given the above parameters, a higher motion video sequence requires

a higher source encoder buffer size S, at the cost of a smaller interleaver depth. For example, Fig. IV.7 shows the optimal choices of N_1 are 170, 151 and 140, for "Akiyo", "Foreman" and "Mobile", respectively. In compressing video, some frames may need more bits than other frames because of the presence of fine detail. In addition, for a high motion video, some frames may need a significantly larger number of bits than others to well represent the occurrence of high motion, and the performance may degrade more seriously during concealment for frame skipping. As a result, a larger source encoder buffer is needed. To further illustrate this point, in Fig. IV.8, we assumed an unconstrained encoder buffer size, and recorded the number of bits accumulated in the buffer for the three video sequences when the source rate was $r_s = 48$ kbps. Note that, although the buffer size is unlimited here, the number of bits accumulated is not infinite because the system is still subject to rate control. As expected, Fig. IV.8 illustrates that a higher motion sequence, at times, needs a larger buffer size than a lower motion sequence.

We also simulated the system for different channel variation rates, with the same $C = 150$ msec and $R_B = 144$ kbps. Fig. IV.9 and Fig. IV.10 show the performance results for a slowly fading channel with $f_D T_s = 0.0035$ ($N_{coh} = 286$ bits), and Fig. IV.11 and Fig. IV.12 are for a fast fading channel with $f_D T_s = 0.01$ ($N_{coh} = 100$ bits). Again, Fig. IV.9 and Fig. IV.11 show the CDF curves of the PSNRs for "Foreman" QCIF, and Fig. IV.10 and Fig. IV.12 compare the areas under the CDF curves of all three video sequences. Again, the x_h values were set to the maximal PSNRs observed for the corresponding video sequences. It is seen that, given the same set of system

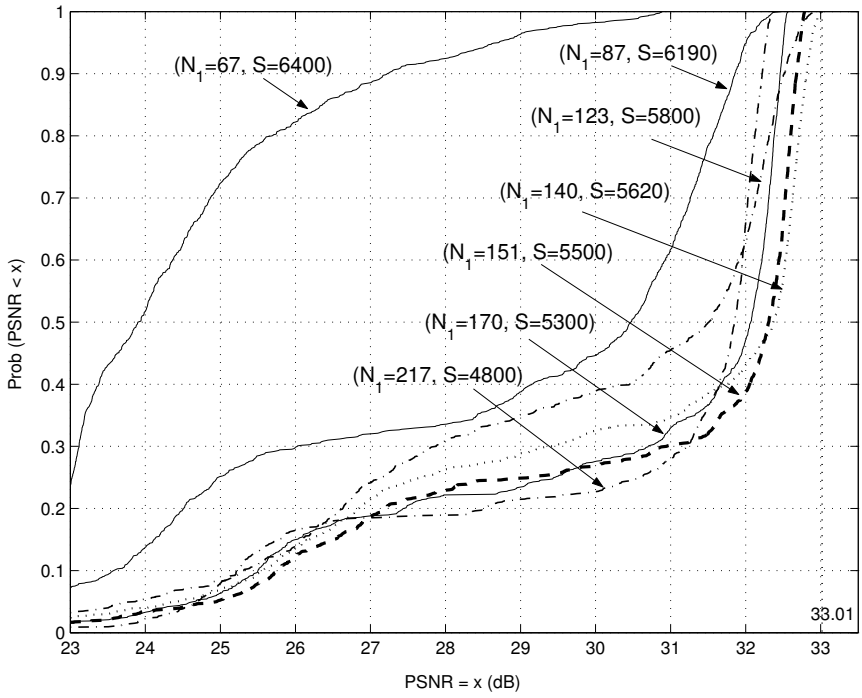


Figure IV.6: CDF curves of the PSNRs for the various delay allocations, for Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.005$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps.

parameters, a larger N_1 is preferable for a slowly fading channel, in order to break the channel memory, whereas a smaller N_1 is preferable for a fast fading channel to free up more of the delay budget for the source encoder buffer. Again, Fig. IV.10 and Fig. IV.12 show that the higher motion video sequence favorites to allocation more delay budget to source encoder buffer, which is consistent with our previous observations.

Next, we simulated the system for different delay budgets and different channel bit rates. Fig. IV.13 shows the system performance for "Foreman" QCIF at $R_B = 144$ kbps and $f_D T_s = 0.005$, with a tight delay constraint $C = 100$ msec, a medium constraint $C = 150$ msec and a very loose constraint $C = 250$ msec. In order to compare the performance not only along each curve in Fig. IV.13, but also across curves, the same x_h value, set to be the maximal observed PSNR value in all the simulations for Fig. IV.13,

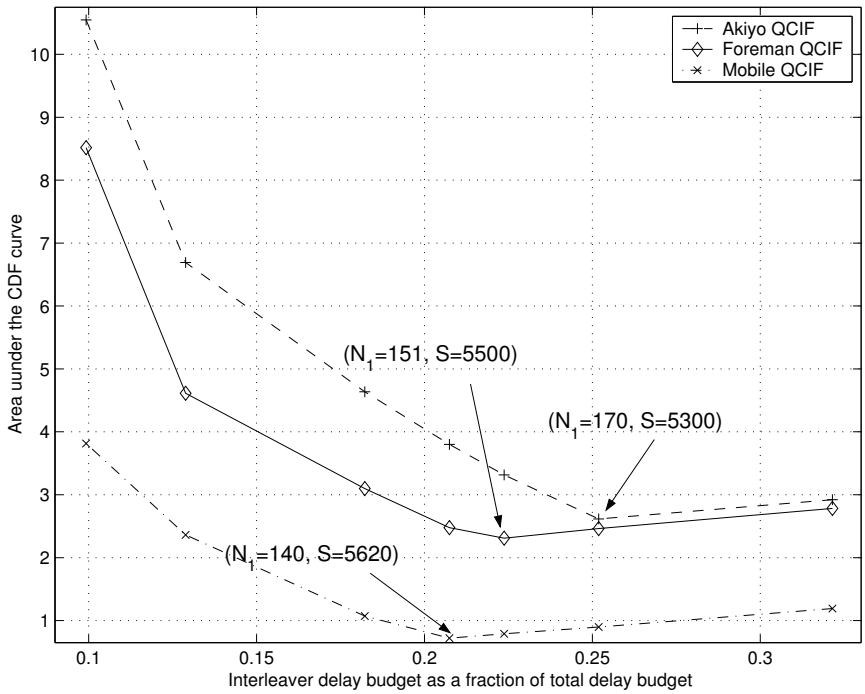


Figure IV.7: System performance, as measured by the areas under the CDF curves, versus the fraction of the interleaver delay budget, for different video sequences, Rayleigh fading channel with $f_D T_s = 0.005$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps. The curve for Foreman QCIF is derived from Fig. IV.6 with $x_h = 33.01$.

was applied for the area calculations. It is seen that, for the three constraints, the optimal choices of N_1 are 135, 151 and 180, respectively, while the corresponding optimal ratios of the interleaver delay budget to the total delay budget are 30.0%, 22.4% and 16.0%, respectively. In other words, as the delay budget C increases, the optimal interleaver depth N_1 increases, because of more available resources; while the corresponding ratios of the interleaver delay to the total delay budget decrease, because of the diminishing returns of the diversity gain. Also, it is seen that the system performance with the best (N_1, S) choice improves, i.e., has a smaller area (y-axis value), as C increases. Similar trends occur when the channel bit rate R_B increases, holding other system parameters constant. As shown in Fig. IV.14, which plots the system performance for "Foreman"

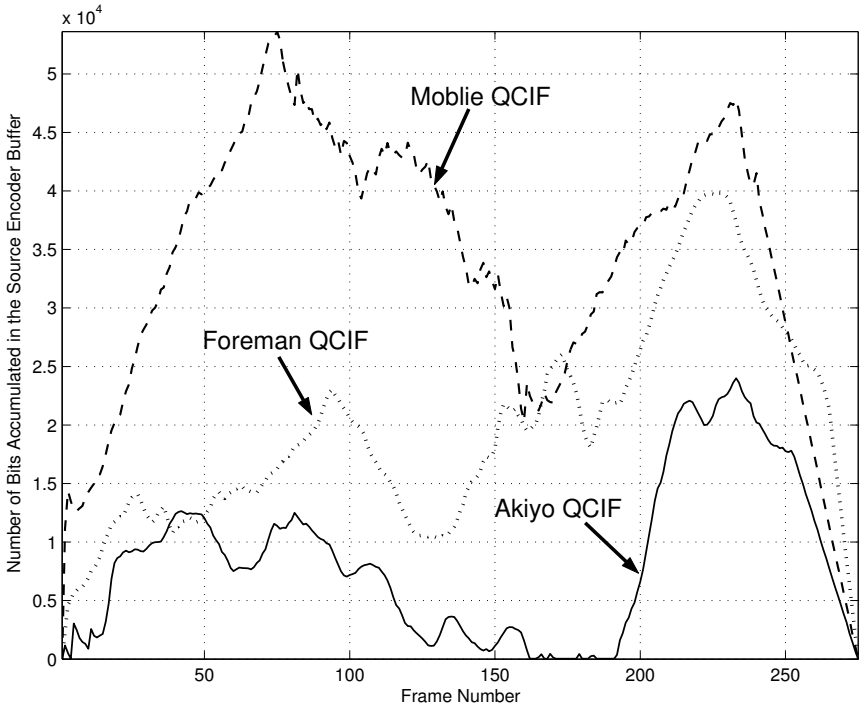


Figure IV.8: The number of bits accumulated in a source encoder buffer with unlimited size versus the frame number, for different video sequences at the source coding rate $r_s = 48$ kbps.

QCIF at $C = 150$ msec and $f_D T_s = 0.005$, with different channel bit rates, the optimal choices of N_1 are 135, 151 and 170, for $R_B = 96$ kbps, $R_B = 144$ kbps, and $R_B = 168$ kbps, respectively, and the corresponding ratios of the interleaver delay budget to the total delay budget are 30.0%, 22.4%, 21.6%, respectively. Also, the system performance with best (N_1, S) choice improves when R_B increases.

Examining the results shown in figures from Fig. IV.6 to Fig. IV.12, as well as our other simulation results, we see the following trends. *First*, the normalized Doppler frequency is the key parameter in the delay partitioning, and a system operating over a fast fading channel prefers a smaller interleaver depth N_1. And as shown in all the above simulation results, it seems that about $0.7N_{coh}$ (more precisely, from $0.6N_{coh}$ to $0.9N_{coh}$) is a safe choice for N_1. This result is consistent with our conclusion in

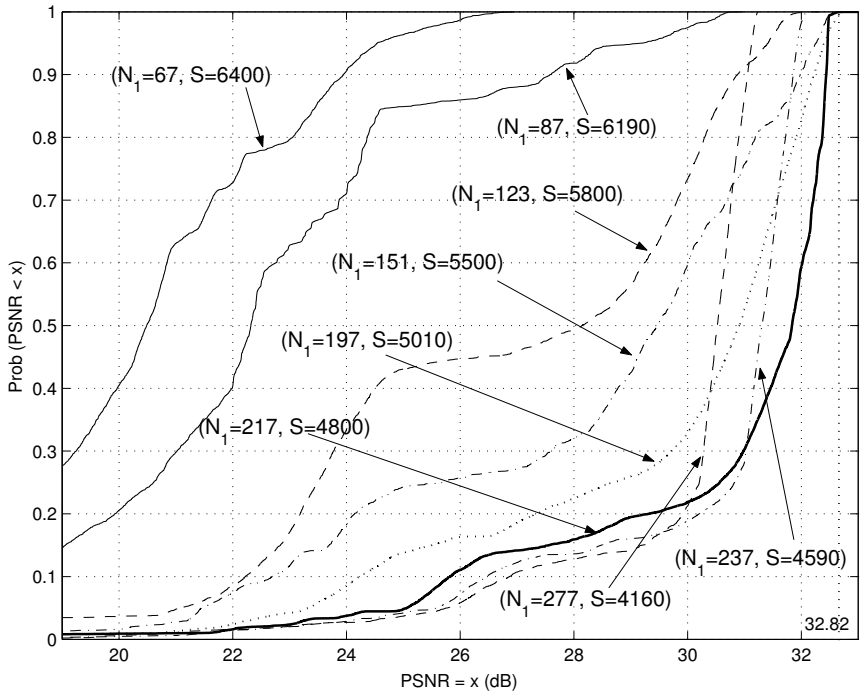


Figure IV.9: CDF curves of the PSNRs for various delay allocations, for Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.0035$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps.

Subsection IV.D.1, which illustrates that the maximum gain from the interleaver is approximately achieved when $N_1 \geq 0.7N_{coh}$ in a Rayleigh fading channel. *Second*, the video content also affects the delay partitioning; a sequence with higher motion content usually prefers a larger source encoder, and thus a smaller N_1. *Third*, either fast fading, or a larger total delay budget C, or a larger channel bit rate R_B, improves the system performance on the average, holding other parameters the same. For example, Fig. IV.6, Fig. IV.9 and Fig. IV.11 show that, for a given set of system parameters, the highest PSNR achieved improves from about 32dB to about 34dB when the channel varies more rapidly. Note that the performance improvement for a larger C or a larger R_B is due to the system having additional available resources, while the performance improvement for fast fading is due to additional channel diversity. However, the last

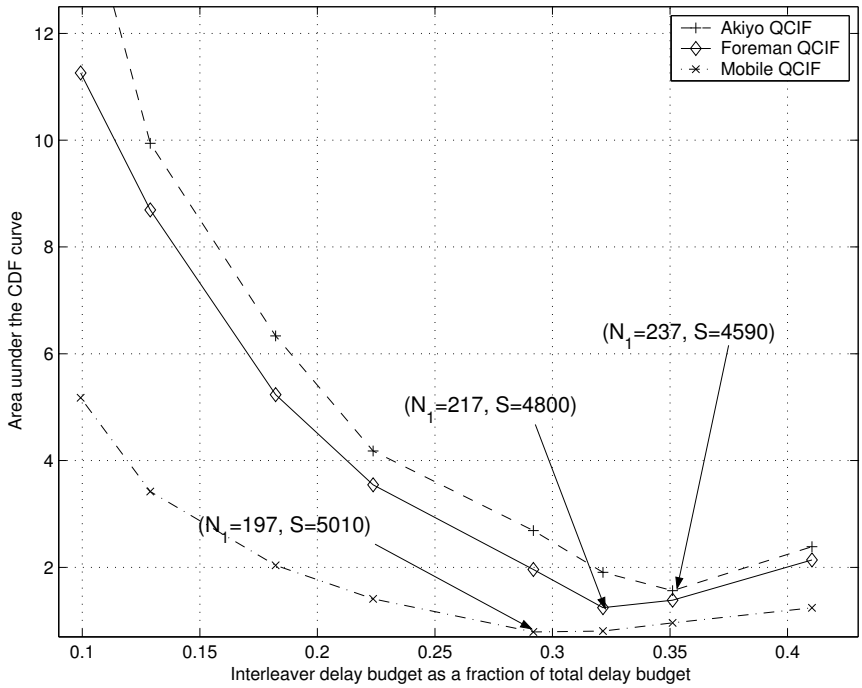


Figure IV.10: System performance, as measured by the areas under the CDF curves, versus the fraction of the interleaver delay budget, for different video sequences, Rayleigh fading channel with $f_D T_s = 0.0035$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps. The curve for Foreman QCIF is derived from Fig. IV.9 with $x_h = 32.82$.

conclusion is valid only under our assumption of perfect channel estimation. *Lastly*, the gaps between the performances of the optimal delay allocation and various sub-optimal delay allocations decrease when the channel varies faster. For example, in Fig. IV.10 (a slowly fading channel), the performance of the optimal allocation and those of other allocations varies by a factor of 10, while in Fig. IV.12 (a fast fading channel), the differences are limited to a factor of 1.2. This implies that the delay allocation issue is more important when the channel varies slowly. When the channel varies fast enough, different allocations may not affect the performance as much.

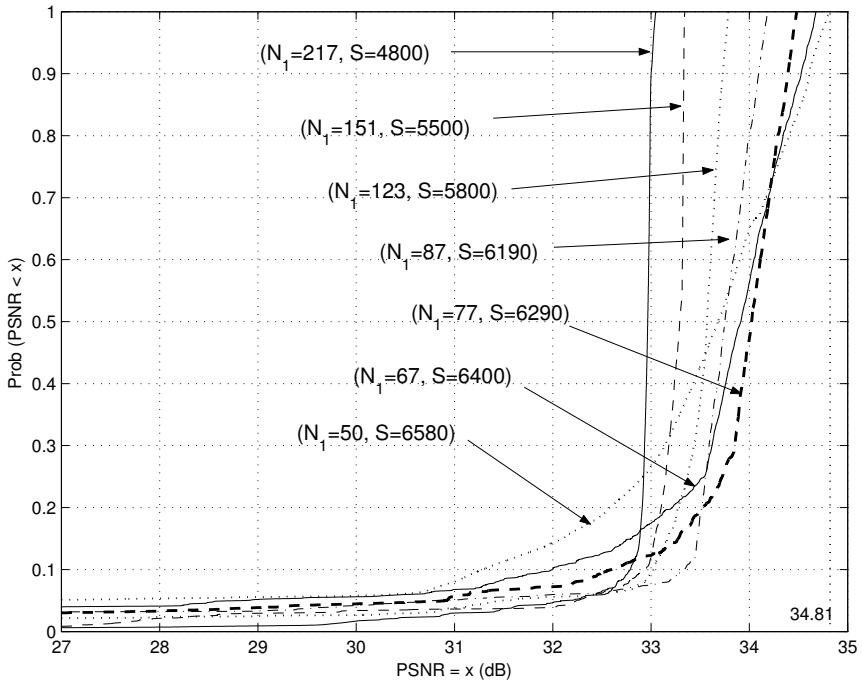


Figure IV.11: CDF curves of the PSNRs for various delay allocations, for Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.01$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps.

IV.D.3 Bandwidth Allocation between Source Coding and Channel Coding and Delay Allocation between the Source Encoder Buffer and the Interleaver, for Fixed Delay Budget (C) and Channel Bit Rate (R_B)

In this subsection, we vary the channel coding rate, r_c, to analyze the bandwidth partition between source coding and channel coding, together with the delay partition between the source encoder buffer and the interleaver, for a fixed delay budget, C, and a given channel bit rate, R_B.

Again, the rates r_s and r_c must satisfy the bandwidth constraint (IV.1). Also, we note from the delay constraint (IV.7) that, for a fixed R_B, the interleaver delay, which is equal to $2N/R_B$, and the channel decoding delay, which is equal to $5P\nu/R_B$, do not change by changing r_s. This implies that increasing S proportionately with r_s will

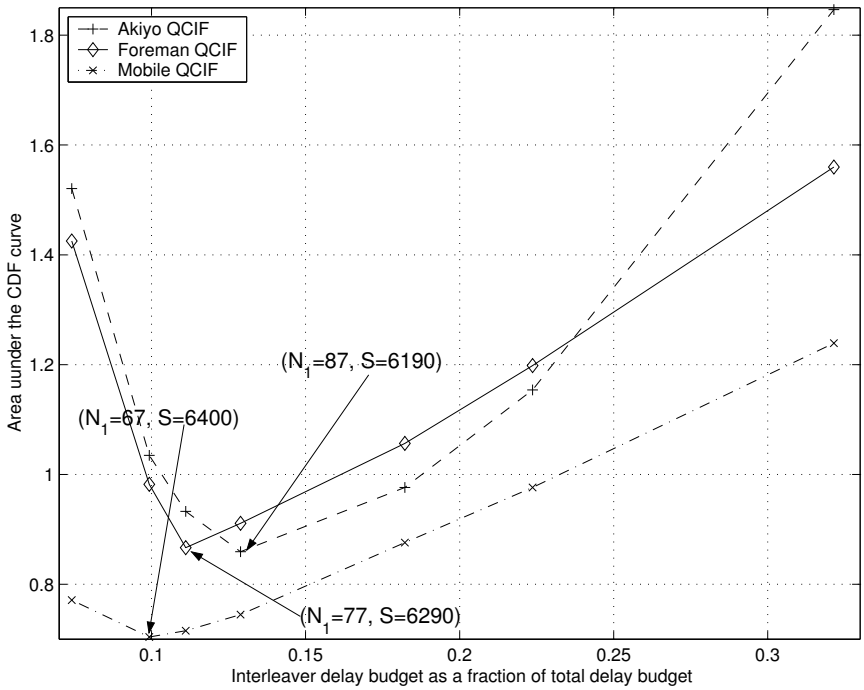


Figure IV.12: System performance, as measured by the areas under the CDF curves, versus the fraction of the interleaver delay budget, for different video sequences, Rayleigh fading channel with $f_D T_s = 0.01$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps. The curve for Foreman QCIF is derived from Fig.IV.11 with $x_h = 34.81$.

ensure that the same delay allocation is maintained. However, maintaining the same delay allocation is not necessarily desirable. With a change in r_s and r_c, the optimal delay allocation may change.

Assume there are N_c candidate channel codes with rates $\{r_c\}$. The optimal bandwidth partition and delay partition, i.e., the best (r_c, r_s, N_1, S) 4-tuple, can be determined by a two-step optimization method:

Step I: For each channel code candidate with rate r_c, calculate the corresponding r_s from (IV.1). For each (r_c, r_s) pair, among the candidate delay partition pairs (N_1, S), find the one for this bandwidth allocation that minimizes the area under the CDF curve, as illustrated in Section. IV.D.2. This yields N_c 4-tuples, with correspond-

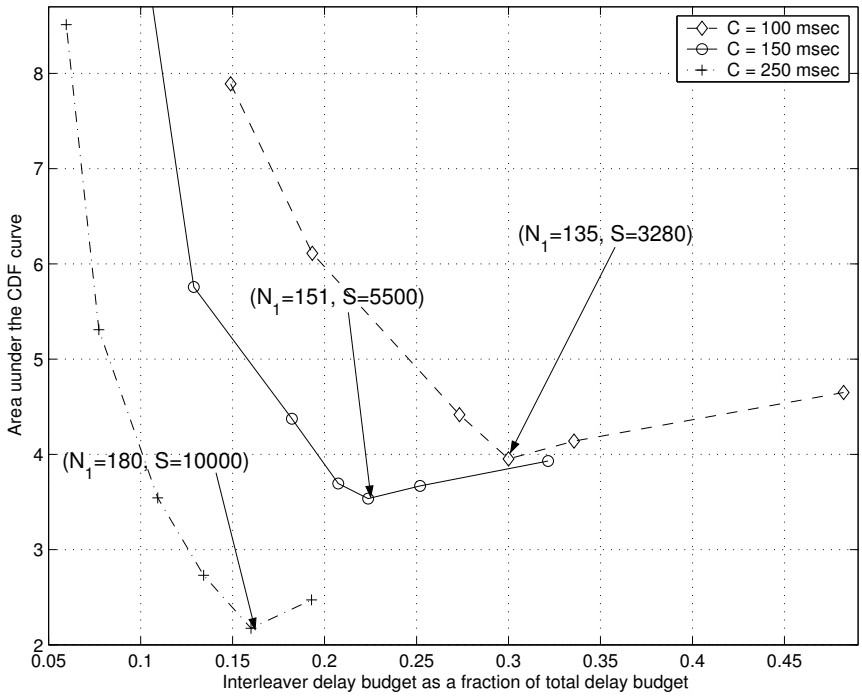


Figure IV.13: System performance, as measured by the areas under the CDF curves, versus the fraction of the interleaver delay budget, for delay budgets $C = 100$ msec, $C = 150$ msec and $C = 250$ msec, Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.005$, and channel bit rate $R_B = 144$ kbps. All the areas are calculated with $x_h = 34.16$, and the curve for $C = 150$ msec is derived from Fig. IV.6.

ing PSNR CDF curves.

Step II: Among the N_c 4-tuples, find the one with the smallest area under its CDF curve, using a common threshold value, x_h. This (r_s, r_c, N_1, S) 4-tuple is the one with best bandwidth and delay allocations.

To illustrate this procedure, we simulated the system for different channel codes in the same RCPC family, with rates equal to $1/3$, $4/11$, $2/5$, and $4/9$ [11], for "Foreman" QCIF at $f_D T_s = 0.005$, $R_B = 144$ kbps, and $C = 150$ msec. From (IV.1), the corresponding source coding rates are 48 kbps, 52.4 kbps, 57.6 kbps, and 64 kbps, respectively. For each (r_c, r_s) pair, different (N_1, S) pairs that satisfied (IV.7) were simulated, and the (N_1, S) pair that minimized the area under the CDF curve was selected.

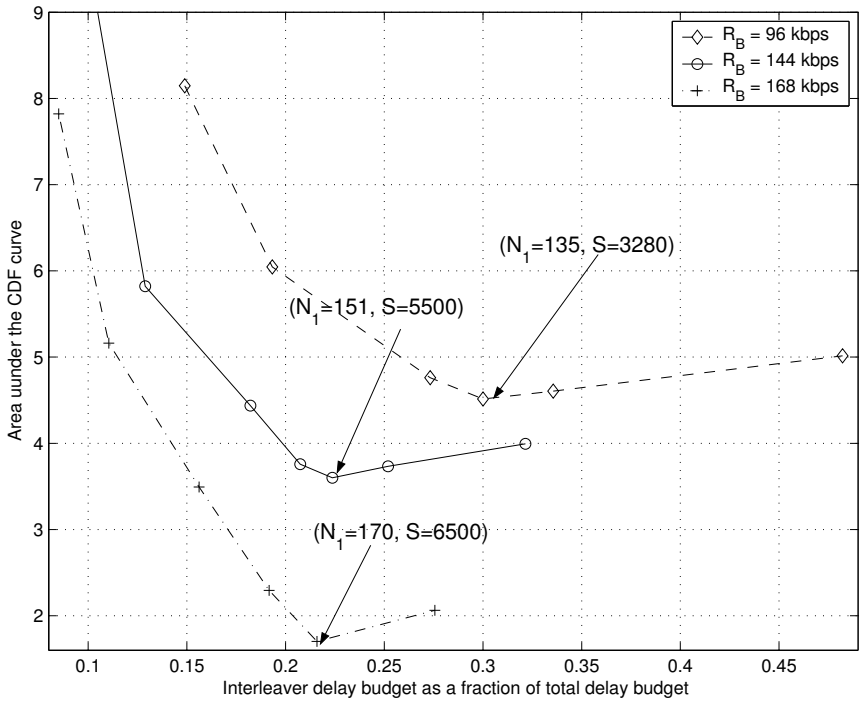


Figure IV.14: System performance, as measured by the areas under the CDF curves, versus the fraction of the interleaver delay budget, for channel bit rates $R_B = 96$ kbps, $R_B = 144$ kbps, and $R_B = 168$ kbps, Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.005$, and delay budget $C = 150$ msec. All the areas are calculated with $x_h = 34.22$, and the curve for $R_B = 144$ msec is derived from Fig. IV.6.

For example, the pair $(N_1 = 151, S = 5500)$ was selected for the bandwidth allocation $(r_c = 1/3, r_s = 48\text{k})$, where the areas of CDF curves were derived from Fig. IV.7. In Fig. IV.15, we show, for four possible (r_c, r_s) allocations, the CDF curve for the corresponding best (N_1, S) pair. They are $(N_1 = 151, S = 5500)$, $(N_1 = 170, S = 5780)$, $(N_1 = 190, S = 6110)$, and $(N_1 = 217, S = 6400)$, for $r_c = 1/3$, $r_c = 4/11$, $r_c = 2/5$, and $r_c = 4/9$, respectively. Then, the best bandwidth and delay partition 4-tuple was selected among the four candidates shown in Fig. IV.15. In Fig. IV.16, we plot the areas under all the CDF curves, wherein all curves were calculated with the same threshold $x_h = 35.78$. It is seen that the $(r_c = 1/3, r_s = 48\text{k}, N_1 = 151, S = 5500)$ 4-tuple yields the best overall performance.

In Fig. IV.16, we show that, all other parameters being the same, increasing r_c, and thus increasing r_s in accordance with (IV.1), the optimal ratio of the interleaver delay to the total delay budget increases, and both the optimal interleaver depth N_1 and the optimal source buffer S increase. This is because, first, both channel coding and interleaving are used to combat the channel fading and to protect the information sequence, so when a channel code with higher r_c is used, it is willing to use a larger N_1 to compensate for the loss from a less powerful channel code. Second, with r_s increasing, the source encoder needs a larger buffer. As shown in (IV.7), for a fixed R_B, the channel decoding delay, $5P\nu/R_B$, is fixed for all the RCPC codes in an RCPC family, since all the codes are formed from the same mother code with the same period P and constraint length ν. When r_c increases, the source encoding delay, S/r_s, becomes smaller, given the same S, because r_s increases with r_c according to (IV.1). This additional delay resource will be shared by both the source encoder and the interleaver, both of which want a larger delay budget. It turns out that the best selection is one that results in a larger S and a larger N_1. Further, the optimal ratio of the interleaver delay to the total delay budget, which is equal to $(2N_1N_2)/(R_BC)$, also increases, because N_1 increases, while C, R_B and N_2 are kept constant.

Lastly, Fig. IV.16 shows that, when increasing r_c, the system performance with the best delay partition degrades. For example, the performance gaps between that of the optimal delay allocation for $r_c = 1/3$ and those for $r_c = 4/11$, $r_c = 2/5$, and $r_c = 4/9$, are about a factor of 0.34, 2.50, and 8.81, respectively. It is seen that, under the scenario we studied here, the system always prefers to use the strongest channel code. This is

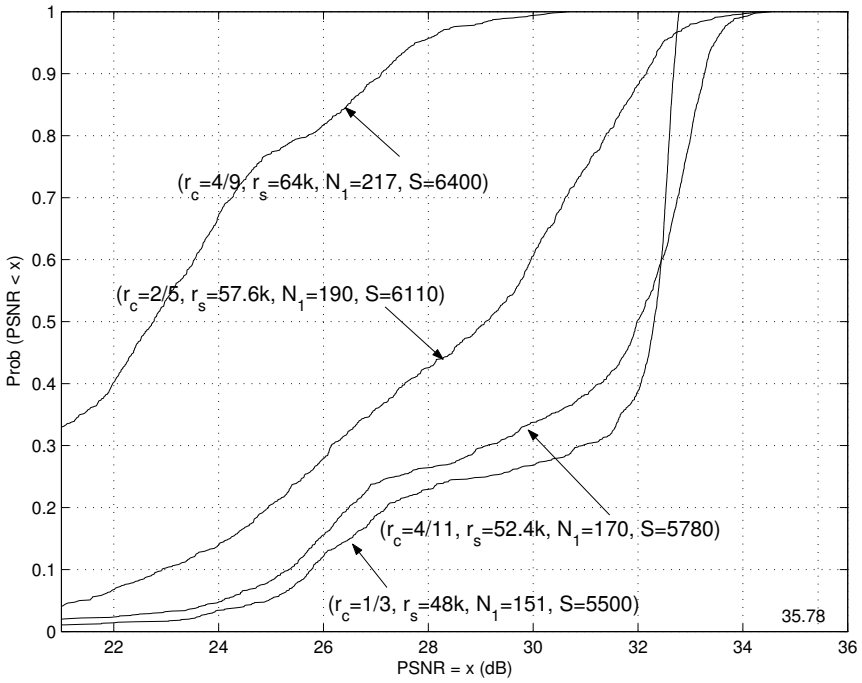


Figure IV.15: CDF curves of the PSNRs for the best (N_1, S) choices of different channel coding rates $\{r_c\}$, Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.005$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps.

probably because the E_s/N_0 value is 3dB, which is relatively low. Under better channel conditions, a higher rate RCPC code would most likely be preferred.

IV.E Conclusions

We analyzed the performance of a wireless video communication system operating over a fading channel, under both an end-to-end delay constraint and a bandwidth constraint. We showed that the main delay components in the system include the queuing delay in the source encoder output buffer, the delay caused by interleaving and deinterleaving, and the delay caused by channel decoding. The relationship among these three components, restricted by the delay constraint, was derived mathematically. We then focused on the delay partitioning between the source encoding and the interleaving.

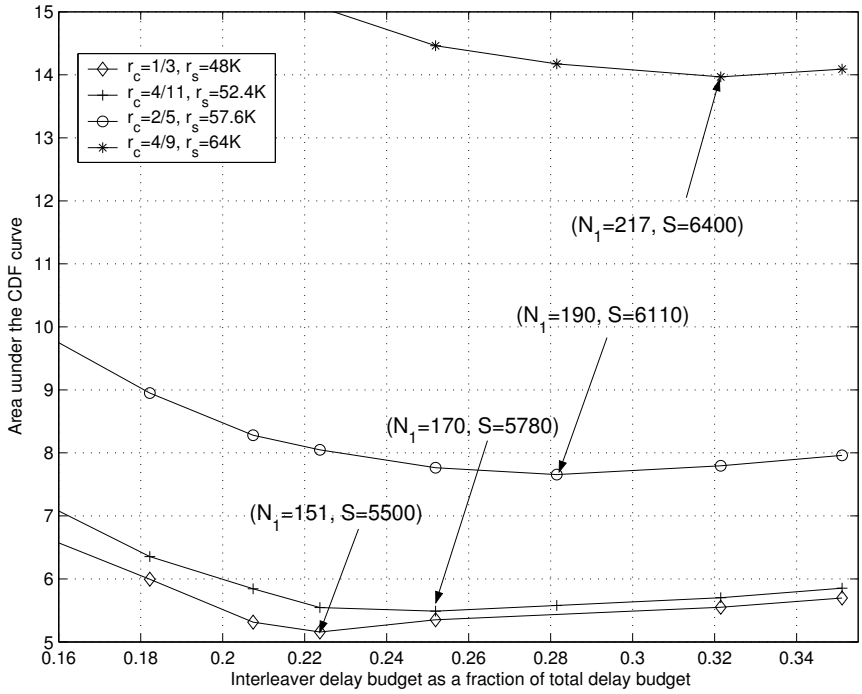


Figure IV.16: System performance, as measured by the areas under the CDF curves, versus the fraction of the interleaver delay budget, for different channel coding rates $\{r_c\}$, Foreman QCIF, Rayleigh fading channel with $f_D T_s = 0.005$, delay budget $C = 150$ msec, and channel bit rate $R_B = 144$ kbps. All the areas are calculated with $x_h = 35.78$, and the optimal performance points, corresponding to the minimal areas on the respective curves, are derived from Fig. IV.15.

Simulation results of the tradeoff between the delay of the source encoder buffer and the interleaver were compared. In particular, we studied how this tradeoff is affected by parameters such as the Doppler frequency of the fading channel, the motion of the video content, the delay constraint, the channel bit rate, and the channel code rate.

It was shown that the normalized Doppler frequency of the fading channel (i.e., N_{coh}) is the key parameter in the delay partitioning. Given other parameters held constant, a system operating over a fast fading channel prefers a smaller interleaver depth N_1, and thus a smaller ratio of the interleaver delay to the total delay budget. From our results for various QCIF sequences over a Rayleigh fading channel with different

bandwidth and delay constraints, we found that optimal values for the interleaver depth N_1 ranged from the integer part of $0.6N_{coh}$ to the integer part of $0.9N_{coh}$, and that, in general, the integer part of $0.7N_{coh}$ is a safe choice for N_1. Also, we showed that the system performance is more sensitive to the delay partitioning when it operates over a slow fading channel.

Other system parameters also affect the delay partitioning between the source encoding and interleaving. In general, for a sequence with higher motion content, because of a larger variation in the number of bits used to describe each frame, a larger source encoder buffer size S and a smaller interleaver depth N_1 are preferable, and thus a smaller ratio of the interleaver delay to the total delay budget. For a system with a larger total delay budget C, or a larger channel bit rate R_B, because of the additional resources, both a larger S and a larger N_1 are preferable, and our results indicate that the corresponding ratio of the interleaver delay to the total delay budget becomes smaller. Lastly, for a system with a higher channel code rate (i.e., a weaker channel code), because of the increase of source rate and the loss of error correction capability, both a larger S and a larger N_1 are again preferable, but now our results indicate that the corresponding ratio of the interleaver delay to the total delay budget becomes larger.

We also showed that either a larger total delay budget C, or a larger channel bit rate R_B, or fast fading (i.e., a smaller N_{coh}), improves the system performance on the average, holding other parameters the same. Notice that the conclusion for fast fading is valid only under our assumption of perfect channel estimation. Also, a two-step procedure was proposed to determine the optimal bandwidth partition and delay

partition, from a finite set of possible RCPC codes. The best allocation depends on both the channel conditions and the video content.

In conclusion, we mention several possible directions in which this work can be extended. We used a video encoder with single-frame prediction. One may involve the use of more sophisticated source encoding strategies, such as hierarchical bi-directional prediction (B-pictures) and long-term frame prediction with pulsed quality, which are more efficient but will introduce additional source coding delay. Also, the channel codes we studied are from a family of RCPC codes. One may use instead codes based upon iterative decoding, such as turbo codes and low density parity check (LDPC) codes, which are more powerful but can result in a larger delay. Finally, our analysis assumed perfect channel estimation. One can relax this assumption, and study the effect on the delay allocation when noisy channel estimates are used.

Acknowledgement

This chapter, in full, is a reprint of the material as it appears in Kartikeya Mehrotra, Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Delay Budget Partitioning in Wireless Multimedia," which is submitted to the *IEEE Transactions on Image Processing* and is currently under review. Both Kartikeya Mehrotra and I were primary authors and contributed equally to this paper, the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for this paper.

V

Conclusions

In this dissertation, we investigated techniques for wireless video communications, which spans the breadth of communications from physical layer communications theory to image and video processing at the application layer. The results are in the general category of cross-layer optimization, that is, they span multiple layers of the standard network protocol stack.

In Chapter II, we presented a transmission scheme for fixed length packet video over a tandem wireless Internet channel. We solved this tandem channel R-D optimization problem in two steps. First, we proposed a video encoder using optimal inter/intra mode selection, operating over the wireline erasure-only channel. Then we added the wireless component. For this we used a concatenation of an inner RCPC coder and an outer CRC coder. Packets that fail the CRC check are dropped, so the tandem channel could be treated as a packet erasure channel. The system performance was both analyzed and simulated, over both constant and time-varying hybrid channel conditions. For the varying channel with delayed feedback information, both instantaneous

feedback and delayed feedback were evaluated, and an improved method of refined distortion estimation for encoding was presented. It was shown that the refined estimation could dramatically improve the performance.

In Chapter III, we proposed a robust video transmission scheme with efficient bandwidth allocation among source coding, channel coding, and spreading, that operates at the packet level. The algorithm was proposed for use in a CDMA network, and can be extended to operate over a tandem channel with both packet erasures and bursty bit errors. The optimization problem among the three parameters was solved using a two-step tradeoff strategy. Results were simulated for a variety of sequences. It was shown that the proposed system which allows all components to vary offered about a 1.4dB gain over a scheme using a fixed packet drop rate, and up to 4dB gain over a scheme using a fixed spreading gain.

Delay tradeoff in video wireless communication systems were investigated in Chapter IV. The key tradeoff elements include the queuing delay in the source encoder output buffer, the delay caused by interleaving and deinterleaving, and the delay caused by channel decoding. The relationship among these three components, restricted by the delay constraint, was derived mathematically. We then focused on delay partitioning between the source encoding and the interleaving, in particular, determining how much certain key parameters affect the performance when the video is transmitted over a time-varying Rayleigh fading channel. These parameters included motion of the video, relative velocity between transmitter and receiver, channel characteristics, and total bandwidth. It was shown that the normalized Doppler frequency of the fading

channel (i.e., N_{coh}) is the key parameter in the delay partitioning.

In conclusion, we mention several possible directions in which this work can be extended. In the bandwidth allocation issue for wireless video communications, one may consider joint bandwidth and power allocation schemes, where we both allocate bandwidth among source coding, channel coding and spreading adaptively on packet basis, and also assign power adaptively across packets. Further, for video transmission over a wireless Internet channel, a so-called "product channel code" can be used, by adding an erasure protection mechanism across packets. For example, in [20], each column code is an RS code, and each row code is a concatenation of an outer CRC code and an inner RCPC code. As a result, the bitstream is protected against both the errors and the erasures.

In the delay allocation issue, one may consider the use of more sophisticated source encoding strategies, such as hierarchical bi-directional prediction (B-pictures) and long-term frame prediction with pulsed quality, which are more efficient but will introduce additional source coding delay [36] [37]. One may also consider the use of channel codes based upon iterative decoding, such as turbo codes and low density parity check (LDPC) codes, which are more powerful but can result in a larger delay [46]. Finally, our analysis assumed perfect channel estimation. One can relax this assumption, and study the effect on the delay allocation when noisy channel estimates are used.

VI

Weight Distribution of a Class of Binary Linear Block Codes Formed from RCPC Codes

In this appendix chapter, we study the weight enumerator and the numerical performance of a class of binary linear block codes formed from a family of rate-compatible punctured convolutional (RCPC) codes. Also, we present useful numerical results for a well-known family of RCPC codes.

RCPC codes, first introduced by Hagenauer [11], are a powerful form of punctured convolutional codes, having flexible rates and requiring an adaptive decoder. Any binary (punctured) convolutional code can be transmitted as a fixed length binary block code, and the knowledge of the weight distribution of linear codes is crucial in its error performance analysis. Methods to obtain the weight distribution of linear block codes formed from a convolutional code were presented in [47, 48]. In this paper, we extend

the previous results to compute the weight enumerator of a family of RCPC codes.

VI.A RCPC Codes: Encoding and Decoding

RCPC codes are a special case of punctured convolutional codes, obtained by adding a rate-compatibility restriction which implies that a high rate code is embedded in the lower rate codes [11]. Mathematically, a family of RCPC codes is described by a mother code and a sequence of puncturing matrices. Assume the generator matrix is $G = (g_{i,j})_{S\times(M+1)}$, with rate $R = 1/S$ and memory order M. Also assume the puncturing matrices are $a(l) = (a_{i,j}(l))_{S\times P}$ for $l = 1, \cdots, (S-1)P$, with the puncturing period P, and $a_{i,j}(l)\epsilon(0,1)$ where 0 implies puncturing. The rate-compatibility restriction implies

$$\text{if } a_{i,j}(l_0) = 1, \text{ then } a_{i,j}(l) = 1 \text{ for all } 1 \leq l_0 \leq l.$$

Note the rate of a RCPC code is $R(l) = P/(P+l)$, so a code with a larger value of l has more powerful error correction capability.

A simple example of a family of RCPC codes is given in [11], where a rate $1/2$ convolutional code with $M = 2$ is punctured periodically with $P = 4$. The generator polynomial of the mother code is $G(D) = \{D^2 + D + 1, D^2 + 1\}$, and a sequence of

puncturing tables is

$$a(1) = \begin{pmatrix} 1\,1\,1\,0 \\ \\ 1\,0\,0\,1 \end{pmatrix}, \quad a(2) = \begin{pmatrix} 1\,1\,1\,0 \\ \\ 1\,1\,0\,1 \end{pmatrix},$$

$$a(3) = \begin{pmatrix} 1\,1\,1\,1 \\ \\ 1\,1\,0\,1 \end{pmatrix}, \quad a(4) = \begin{pmatrix} 1\,1\,1\,1 \\ \\ 1\,1\,1\,1 \end{pmatrix},$$

with code rates $4/5$, $4/6$, $4/7$ and $4/8$, respectively.

On the receiving side, the decoder can use the Viterbi algorithm (VA) with a trellis modified by the current puncturing matrix $a(l)$. Suppose x is sent and y is received. For binary transmission over an an additive white Guassian noise (AWGN) channel, the VA will find the path $\hat{x}^m$ which satisfies

$$\max_m \left(\sum_{j=1}^{J} \sum_{i=1}^{S} a_{i,j}\ \hat{x}_{i,j}^m\ y_{i,j}\right) \tag{VI.1}$$

where $a_{i,(j+P)} = a_{i,j}$ is the (i,j)-th entry of $a(l)$, and J is the trellis length.

VI.B Transition Matrix Sequence

The transition matrix of a convolutional code is used to describe the state transition possibilities and corresponding output weight of the code [47]. For a convolutional code with rate $R = 1/S$ and memory M, the transition matrix is a 2^M by 2^M matrix. Assuming $i_n \epsilon (0,1)$ is the weight of the n-th output, and $H = \sum_{n=1}^{S} i_n$ is the

Hamming weight of the entire output. Denote by $A_{i,j}$ the (i,j)-th entry of the transition matrix, $A_{i,j} = D^H$ if there is an input (either zero or one) that takes the encoder from state i to state j; otherwise, $A_{i,j} = 0$. For example, the transition matrix of the convolutional code given in Section VI.A is

$$A = \begin{pmatrix} D^0 \cdot D^0 D^1 \cdot D^1 & 0 & 0 \\ 0 & 0 & D^1 \cdot D^0 D^0 \cdot D^1 \\ D^1 \cdot D^1 D^0 \cdot D^0 & 0 & 0 \\ 0 & 0 & D^0 \cdot D^1 D^1 \cdot D^0 \end{pmatrix} = \begin{pmatrix} 1 & D^2 & 0 & 0 \\ 0 & 0 & D & D \\ D^2 & 1 & 0 & 0 \\ 0 & 0 & D & D \end{pmatrix}. \quad \text{(VI.2)}$$

As stated in [47], the (i,j)-th element of the K-th power of A, $(A^K)_{i,j}$, gives the output weight enumerator, given that the encoder starts in state i, ends in state j, and K binary digits are fed into the encoder.

For RCPC codes, the output information changes periodically due to the periodic puncturing. Therefore, we need a transition matrix sequence to describe the state transition possibilities and the weights of the outputs. We denote this sequence by $A_1, A_2, \cdots, A_x, \cdots$, where $A_{x+P} = A_x$. Each matrix A_x is obtained from the structure of the mother code and the λ-th column of the puncturing matrix $a(l)$, where $\lambda \equiv x \pmod{P}$ and $\lambda \, \epsilon \, (1, \cdots, P)$. Specifically, the (i,j)-th entry of A_x is equal to D^h, if there is an input that takes the encoder from state i to state j, and h is the Hamming weight of the punctured output using the λ-th column of $a(l)$; otherwise $(A_x)_{i,j} = 0$.

For example, for the family of RCPC codes described by Section VI.A, de-

fine matrix A as in Equation (VI.2), and define matrices B and C as follows, which correspond to the first and second output, respectively:

$$B = \begin{pmatrix} 1\, D\, 0\, 0 \\ 0\, 0\, D\, 1 \\ D\, 1\, 0\, 0 \\ 0\, 0\, 1\, D \end{pmatrix}, \quad C = \begin{pmatrix} 1\, D\, 0\, 0 \\ 0\, 0\, 1\, D \\ D\, 1\, 0\, 0 \\ 0\, 0\, D\, 1 \end{pmatrix}.$$

The matrix sequences of the four RCPC codes are

$$a(1):\ A_1 = A,\ A_2 = B,\ A_3 = B,\ A_4 = C,\ \cdots$$

$$a(2):\ A_1 = A,\ A_2 = A,\ A_3 = B,\ A_4 = C,\ \cdots$$

$$a(3):\ A_1 = A,\ A_2 = A,\ A_3 = B,\ A_4 = A,\ \cdots$$

$$a(4):\ A_1 = A_2 = A_3 = A_4 = \cdots = A.$$

We define matrix Φ^K by $\prod_{x=1}^{K} A_x$, which yields the output information for K continuous steps of the RCPC code. In particular, the (i, j)-th element of the matrix Φ^K, $(\Phi^K)_{i,j}$, gives the output weight enumerator, given that the encoder starts in state i, ends in state j, and K binary digits are sent into the RCPC encoder.

VI.C Weight Enumerator

Several different methods for constructing binary linear block codes from a convolutional code were presented in [47], along with a way to find the corresponding weight enumerator $T(D) = \sum_d A_d D^d$, where A_d is the number of codewords of weight d. These methods can be applied to the block codes formed from a RCPC code. Denote by R_p and R_{block} the rate of an unterminated punctured convolutional code and of the resultant block code, respectively, and denote by K and N the fixed block length of the input and output of the block RCPC encoder, respectively. As an example from [47], for the zero tail (ZT) method, $T(D)$ is given by $(\Phi^k)_{1,1}$, $R_{block} = \frac{(K-M)}{K} R_p$, and $N = K/R_p = (K-M)/R_{block}$.

Having the weight enumerator $T(D)$ of a linear block code, we may use it to evaluate its performance. Denote by d_{min} the minimum distance of the block code, and by E_s/N_0 the energy-per-symbol divided by the noise power density. Note that $E_s/N_0 = R_{block} \cdot E_b/N_0$ where E_b/N_0 is the energy-per-bit divided by the noise power density. The union bound on the block error rate of a RCPC code for binary transmission over an AWGN channel is given by

$$P_{block} \leq \sum_{d=d_{min}}^{N} A_d Q(\sqrt{2d\frac{E_s}{N_0}}). \tag{VI.3}$$

Furthermore, a good approximation to the union bound of the bit error probability P_{bit} is obtained by scaling each term in the sum of Equation (VI.3) by (d/N) [49].

VI.D Numerical Examples

In this section, we show the results for the block codes formed from the "Good" RCPC codes with $M = 6$ and $P = 8$ [11], whose encoder is shown in Fig. VI.1. In particular, we examine the block codes formed from the rate $8/9$, $2/3$ and $1/3$ codes in this family [12].

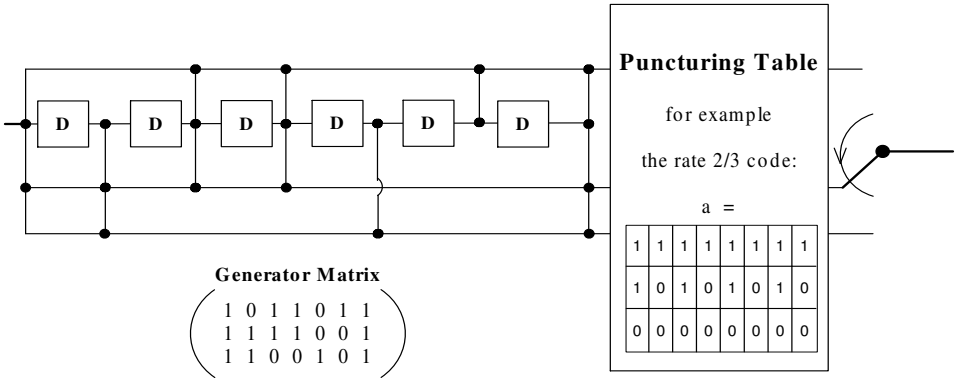


Figure VI.1: The 64-state RCPC codes with puncturing period $P = 8$.

With the ZT method, using the method illustrated in this paper, the weight distributions of the block codes with different input block lengths K are given in Table I. The block error rates and bit error rates of these block codes with $K = 400$ for an AWGN channel are shown in Fig. VI.2.

In summary, we illustrated how to compute the weight enumerator and evaluate the performance of binary linear block codes formed from a family of RCPC codes. The concept of the transition matrix sequence was introduced and explained. Numerical results for a well-known family of RCPC codes were also presented.

Acknowledgement

This chapter, in full, is a reprint of the material as it appears in Yushi Shen, Pamela C. Cosman, and Laurence B. Milstein, "Weight Distribution of a Class of Binary Linear Block Codes Formed from RCPC Codes," *IEEE Communications Letters*, vol. 9, no. 9, pp. 811-813, Sept. 2005. I was the primary author and the co-authors Dr. Cosman and Dr. Milstein directed and supervised the research which forms the basis for this paper.

Table VI.1: Weight Distribution of Block Codes formed from the 64-state RCPC Codes, with Period $P = 8$.

Weight Distance	Rate 8/9 RCPC Code			
	K=200	K=400	K=600	K=800
0	1	1	1	1
3	94	194	294	394
4	1390	2965	4540	6115
5	17247	37822	58397	78972
6	195637	455037	724437	1003837
7	2254907	5634757	9329607	13339457
8	25932510	70104784	120872684	178236209

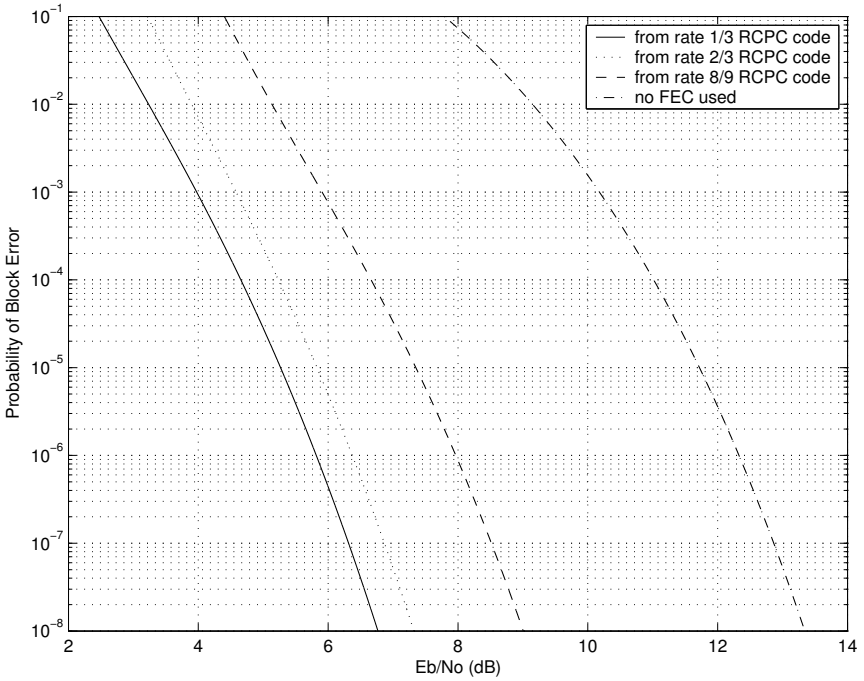
(a) $d_{min} = 3$

Weight Distance	Rate 2/3 RCPC Code			
	K=200	K=400	K=600	K=800
0	1	1	1	1
6	96	196	296	396
7	1509	3109	4709	6309
8	4447	9247	14047	18847
9	14350	30150	45950	61750
10	57369	121569	185769	249969
11	213677	457177	700677	944177
12	794911	1726461	2668011	3619561

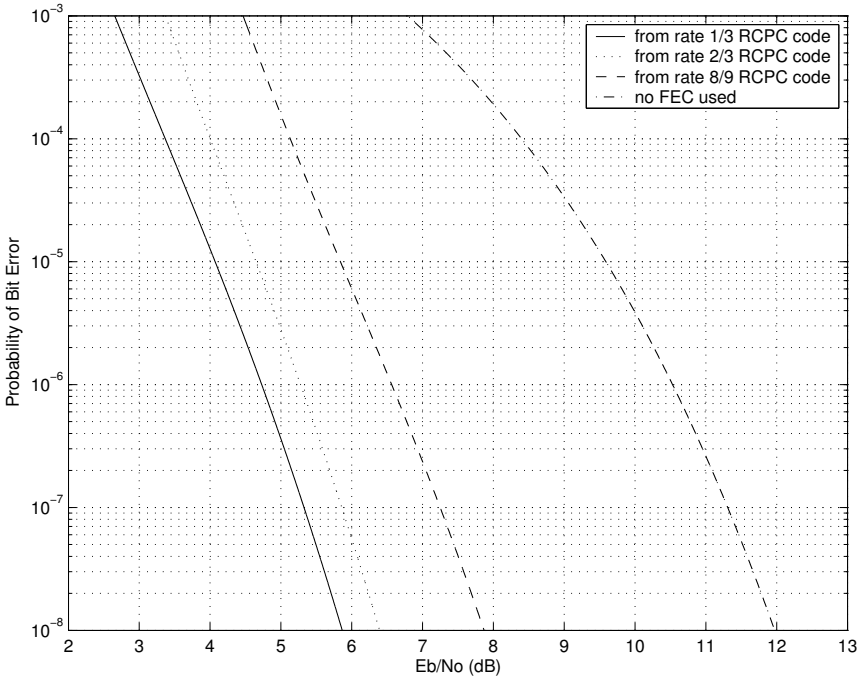
(b) $d_{min} = 6$

Weight Distance	Rate 1/3 RCPC Code			
	K=200	K=400	K=600	K=800
0	1	1	1	1
14	194	394	594	794
16	1338	2738	4138	5538
18	2072	4272	6472	8672
20	6546	13546	20546	27546
22	16698	34698	52698	70698
24	51209	107009	162809	218609
26	147582	309782	471982	634182

(c) $d_{min} = 14$



(a)



(b)

Figure VI.2: Performance evaluation for the ZT codes formed from the RCPC codes, as shown in Fig. VI.1, with fixed length $K = 400$. The corresponding weight distributions are given in Table I. (a) Block error prob. vs. the energy-per-bit divided by the noise power density. (b) Bit error prob. vs. the energy-per-bit divided by the noise power density.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 7, pp. 379–423 and 623–656, 1948.

[2] H. Yousefi'zadeh, H. Jafarkhani, and F. Etemadi, "Distortion-optimal transmission of progressive images over channels with random bit errors and packet erasures," *Proceedings of the IEEE on Data Compression Conference*, pp. 132–141, 2004.

[3] Y. Pei and J. W. Modestino, "Use of concatenated fec coding for real-time packet video over heterogeneous wired-to-wireless ip networks," *Proceedings of the 2003 International Symposium on Circuits and Systems*, vol. 2, pp. 25–28, 2003.

[4] R. Anand, C. Podilchuk, and H. Lou, "Progressive video transmission over a wired-to-wireless network," *Vehicular Technology Conference Proceedings*, vol. 3, pp. 2424–2428, May 2000.

[5] Y. Shen, P. C. Cosman, and L. B. Milstein, "Video communications with optimal intra/inter-mode switching over wireless internet," *37th Asilomar Conference on Signals, Systems and Computers*, pp. 1548–1552, Nov. 2003.

[6] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, 2000.

[7] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[8] S. Lin and D. J. Costello, "Error control coding: Fundamentals and applications," *Englewood Cliffs, New Jersey: Prentice-Hall*, 1983.

[9] G. Castagnoli, J. Ganz, and P. Graber, "Optimum cyclic redundancy-check codes with 16-bit redundancy," *IEEE Trans. Commun.*, vol. 38, pp. 111–114, Jan. 2000.

[10] T. V. Ramabadran and S. S. Gaitonde, "A tutorial on crc computations," *IEEE Micro*, vol. 8, pp. 62–75, Aug. 1998.

[11] J. Hagenauer, "Rate-compatible punctured convolutional codes (rcpc codes) and their applications," *IEEE Trans. Commun.*, vol. 36, pp. 389–400, Apr. 1998.

[12] P. G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Trans. Commun.*, vol. 4, no. 7, pp. 189–191, July 1997.

[13] M. Roder and R. Hamzaoui, "Fast list viterbi decoding and application for source-channel coding of images," *Konstanz University Review*, no. 182, Dec. 2002.

[14] T. Eng and L. B. Milstein, "Coherent ds-cdma performance in nakagami multipath fading," *IEEE Trans. Commun.*, vol. 43, pp. 1134–1143, Apr. 1995.

[15] Q. Zhao, P. C. Cosman, and L. B. Milstein, "Tradeoffs of source coding, channel coding and spreading in frequency selective rayleigh fading channels," *Journal of VLSI Signal Processing*, vol. 30, pp. 7–20, 2002.

[16] R. Annavajjala, P. C. Cosman, and L. B. Milstein, "On source coding, channel coding, and spreading tradeoffs in a ds-cdma system operating over frequency selective fading channels with narrow-band interference," *IEEE J. Select. Areas Commun.*, vol. 23, no. 5, pp. 1034–1044, May 2005.

[17] Q. Zhao, P. C. Cosman, and L. B. Milstein, "Optimal allocation of bandwidth for source coding, channel coding and spreading in cdma systems," *IEEE Trans. Commun.*, vol. 52, no. 10, pp. 1797–1808, Oct. 2004.

[18] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," *Proceedings SPIE*, Jan. 2000.

[19] L. P. Kondi, S. N. Batalama, D. A. Pados, and A. K. Katsaggelos, "Joint source/channel coding for scalable video over ds-cdma multipath fading channels," *IEEE Int. Conf. Image Processing*, pp. 994–997, Oct. 2001.

[20] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Real-time error protection of embedded codes for packet erasure and fading channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 8, pp. 1064–1072, Aug. 2004.

[21] S. Zhao, Z. Xiong, and X. Wang, "Optimal resource allocation for wireless video over cdma networks," vol. 4, pp. 56–67, Jan. 2005.

[22] Y. Shen, P. C. Cosman, and L. B. Milstein, "Video coding with fixed length packetization for a tandem channel," *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 273–288, Feb. 2006.

[23] M. K. Simon and M-S. Alouini, "Digital communications over fading channels: a unified approach to performance analysis," *Wiley-IEEE Press*, 2004.

[24] Y. Shen, P. C. Cosman, and L. B. Milstein, "Weight distribution of a class of binary linear block codes formed from rcpc codes," *IEEE Commun. Lett.*, vol. 9, no. 9, pp. 811–813, Sept. 2005.

[25] L. L. Chong and L. B. Milstein, "The effects of channel-estimation errors on a space-time spreading cdma system with dual transmit and dual receive diversity," *IEEE Trans. Commun.*, vol. 52, pp. 1145–1151, July 2004.

[26] T. Eng, N. Kong, and L. B. Milstein, "Comparison of diversity combining techniques for rayleigh fading channels," *IEEE Trans. Commun.*, pp. 1117–1129, Sept. 1996.

[27] P. Dent, G. E. Bottomley, and T. Croft, "Jokes' model revisited," *Electronics Letters*, vol. 29, no. 13, pp. 1162–1163, 1993.

[28] D. Srinivasan, L. P. Kondi, and D. A. Pados, "Scalable video transmission over wireless ds-cdma channels using minimum tsc spreading codes," *IEEE Signal Processing Lett.*, vol. 11, no. 10, pp. 836–840, Oct. 2004.

[29] J. R. Corbera and S. Lei, "Rate control in dct video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, Feb. 1999.

[30] C. Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE J. Select. Areas Commun.*, vol. 17, no. 5, pp. 172–185, May 1999.

[31] J. W. Modestino and S. Y. Mui, "Convolutional code performance in the rician fading channel," *IEEE Trans. Commun.*, vol. 24, no. 6, 1976.

[32] M. Rice and E. Perrins, "A simple figure of merit for evaluating interleaver depth for the land-mobile satellite channel," *IEEE Trans. Commun.*, vol. 49, no. 8, Aug. 2001.

[33] J. Lim and D. L. Neuhoff, "Joint and tandem source-channel coding with complexity and delay constraints," *IEEE Trans. Commun.*, vol. 51, no. 5, pp. 757–766, May 2003.

[34] S. Aramvith, C. Lin, S. Roy, and M. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 558–565, 2002.

[35] G. Su and M. Wu, "Efficient bandwidth resource allocation for low-delay multiuser video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1124–1137, Sept. 2005.

[36] A. Leontaris and P. C. Cosman, "End-to-end delay for hierarchical b-pictures and pulsed quality dual frame video coders," *ICIP Conference*, Oct. 2006.

[37] G. Pau, B. Pesquet-Popescu, M. Schaar, and J. Vieron, "Delay-performance tradeoffs in motion-compensated scalable subband video compression," *Proceedings Advanced Concepts for Intelligent Vision Systems*, Sept. 2004.

[38] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inform. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[39] M. J. Neely and E. Modiano, "Capacity and delay tradeoffs for ad hoc mobile networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 1917–1937, 2005.

[40] Y. Yasuda, K. Kashiki, and Y. Hirata, "High rate punctured convolutional codes for soft decision viterbi decoding," *IEEE Trans. Commun.*, vol. 32, pp. 315–319, Mar. 1984.

[41] Y. Shen, P. C. Cosman, and L. B. Milstein, "Error resilient video communications over cdma networks with a bandwidth constraint," *IEEE Trans. Image Processing*, no. 10, Oct. 2006.

[42] L. Wilhelmsson and L. B. Milstein, "On the effect of imperfect interleaving for the gilbert-elliott channel," *IEEE Trans. Commun.*, vol. 47, no. 5, May 1999.

[43] K. Tang, P. H. Siegel, and L. B. Milstein, "On the performance of turbo coding for the land mobile channel with delay constraints," *Proceedings of 33rd Asilomar Conference*, pp. 1659–1665, Oct. 1999.

[44] L. Wilhelmsson, "On using the gilbert-elliott channel to evaluate the performance of block coded transmission over the land mobile channel," *Ph.D. Dissertation, Univ. of California, San Diego*, 1997.

[45] P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Trans. Image Processing*, vol. 9, no. 6, 2000.

[46] S. Noh and S.-C. Park, "Trade-off between performance and complexity with various candidates in ldpc coded mimo-ofdm iterative decoding," *Advanced Communication Technology, ICACT*, vol. 2, pp. 1291–1293, Feb. 2006.

[47] J. Wolf and A. Viterbi, "On the weight distribution of linear block codes formed from convolutional codes," *IEEE Trans. Commun.*, vol. 44, pp. 1049–1051, 1996.

[48] M. P. C. Fossorier, S. Lin, and D. Costello, "On the weight distribution of terminated convolutional codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1646–1648, July 1999.

[49] M. P. C. Fossorier and S. Lin, "Bit error probability for maximumlikelihood decoding of linear block codes and related soft-decision decoding methods," *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 3083–3090, Nov. 1998.