ABSTRACT

Title of Dissertation: PROGRESSIVE SOURCE-CHANNEL CODING FOR MULTIMEDIA TRANSMISSION OVER NOISY AND LOSSY CHANNELS WITH AND WITHOUT FEEDBACK

Vinay Chande, Doctor of Philosophy, 2004

Dissertation directed by: Professor Nariman Farvardin
Department of Electrical and Computer Engineering

Rate-scalable or layered lossy source-coding is useful for *progressive* transmission of multimedia sources, where the receiver can reconstruct the source incrementally. This thesis considers "joint source-channel" schemes for such a progressive transmission, in the presence of noise or loss, with and without the use of a feedback link.

First we design image communication schemes for memoryless and finite state channels using *limited and explicitly constrained* use of the feedback channel in the form of a variable incremental redundancy Hybrid ARQ protocol. Constraining feedback allows a direct comparison with schemes without feedback. Optimized feedback based systems are shown to have useful gains.

Second, we develop a controlled Markov chain approach for constrained feedback Hybrid ARQ protocol design. The proposed methodology allows the protocol to be chosen from a collection of signal flow graphs, and also allows explicit control over the tradeoffs in throughput, reliability and complexity.

Next we consider progressive image transmission in the absence of feedback. We assign unequal error protection to the bits of a rate-scalable source-coder using rate compatible channel codes. We show that, under the framework, the source and channel bits can be "scheduled" in a single bitstream in such a way that operational optimality is retained for different transmission budgets, creating a rate-scalable joint source-channel coder.

Next we undertake the design of a joint source-channel decoder that uses "distortion aware" ACK/NACK feedback generation. For memoryless channels, and Type-I HARQ, the design of optimal ACK/NACK generation and decoding by packet combining is cast and solved as a sequential decision problem. We obtain dynamic programming based optimal solutions and also propose suboptimal, lower complexity distortion-aware decoders and feedback generation rules which outperform conventional BER based rules such as CRC-check.

Finally we design operational rate-distortion optimal ACK/NACK feedback generation rules for transmitting a tree structured quantizer over a memoryless channel. We show that the optimal feedback generation rules are embedded, that is, they allow incremental switching to higher rates *during the transmission*. Also, we obtain the structure of the feedback generation rules in terms of a feedback threshold function that simplifies the implementation.

PROGRESSIVE SOURCE-CHANNEL CODING FOR MULTIMEDIA TRANSMISSION OVER NOISY AND LOSSY CHANNELS WITH AND WITHOUT FEEDBACK

by

Vinay Chande

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Professor Nariman Farvardin, Chairman/Advisor
Professor K. J. Ray-Liu
Professor Rama Chellappa
Professor Min Wu
Professor Ben Kedem

DEDICATION

To

My parents

ACKNOWLEDGEMENTS

They say in ancient Indian texts that it is ignorance to believe that one is the (sole) doer of any activity. Ph.D., although sometimes perceived as an example of individual accomplishment, might be one of the best opportunities to realize that ancient precept over and over. Hence I take this page and the next to express my gratitude for getting this opportunity to do what has been a dream of all my life.

First and foremost, I would not have been able to make this journey, overcome my limitations and carry through to the finish line had it not been for Dr. Farvardin's kindness, generosity and patient encouragement. I have realized repeatedly that whenever I listened to him, good things happened. His feedbacks made papers more readable by an order of magnitude. His research world-view has shaped mine. On a more personal front, his personal leadership, and dynamic combination of sincerity and enthusiasm has been more than an inspiration for me. I noticed that after every meeting with him, I was filled with renewed vigor and almost palpable increased wakefulness. It has been an honor and privilege to work under his tutelage.

I thank my committee members, Professors Dr. Liu, Dr. Chellappa, Dr. Wu and Dr. Kedem for taking their time and being patient with me. I value their feedback immensely.

I am very happy to acknowledge my fellow researchers at the old and the new CSPL. In particular, Andres Kwasinski with whom I have some exciting continuing collaborative work, Mehdi Alasti, Hugh Brunk with whom I had long technical discussions, and Hamid Jafarkhani, a joint project with whom kick-started my research. I must also acknowledge Suki, Murari, Damianos, Kfir, Mike and other fellow graduate students who made CSPL and ECE an intellectually stimulating as well as fun environment. I also thank Dr. Liu and Dr. Wu for letting me attend their group meetings as part of the new CSPL.

I cannot say enough for the love and support I have gotten from my friends throughout these years. Sanjay, Sachin and Namrata have been directly involved in keeping me moving for a long time. Sumod, Aniruddha, Amol, Mukul, Siddarth, Arindam have been a dream of roommates each of whom has individually and collectively been there for direct help, and cheering as well. This has been a long time span, and I acknowledge, in almost chronological order, Niranjan, Anand, Sandeep, Sreekanth, Rajesh, GD and Anna for their cherished friendship and sanity-keeping.

In the end, I dedicate this thesis and these years to my parents, Shri Vijay and Sau Vijaya Chande, who together form that eternal, inexhaustible, source of love, from which I draw strength and power every moment. The infiniteness of their love, patience, encouragement and wisdom convinces one of the existence of the divine. Love you aaibaba.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Multimedia Sources over Noisy Channels

The past decade has been one of the most exciting times to be a communications engineer. Last ten years have seen an explosive growth in telecommunications technology and its deployment. The Internet has already become so indispensable that we sometimes wonder how people could do without it earlier.

The ultimate dream is that of complete connectivity across space and time, where a person anywhere on the globe, can instantly connect to every other person or institution, and has unrestricted, fast, up-to-date and economical access to collective knowledge and wisdom that humanity has to offer. In addition, such a person would like to be mobile without losing connectivity.

Along with data sources such as text, numbers, software programs and computer binaries, multimedia sources such as images, video, speech, music and graphics form significant part of the services that such a globally connected society would like to make available to its members. It has been predicted that the digital multimedia may soon become the dominant traffic on the Internet.

Digitally encoded multimedia sources, primarily images, video and audio, behave differently than data. Firstly, they are “high-bandwidth” sources, that is, in the raw form, they demand relatively large digital memory storage. Secondly, sources such as video and audio are real-time so they put real-time restrictions on delays and jitter. Thirdly, and most importantly for our discussion, unlike data, they are *loss tolerant*, that is, they allow approximate reproductions. They can be compressed in a “lossy” manner, *i.e.* they have a “rate-distortion” tradeoff in their digital representations. Also, this property introduces robustness as the information conveyed by them is not significantly altered if the reproduction at the receiver is not exactly what was transmitted.

This thesis deals with the techniques of *progressive* communication of such loss-tolerant multimedia sources over noisy channels. Progressive communication allows the receiver to reconstruct the source at increasing fidelity as it receives bits or channel symbols from the transmitter.

Though *embedded* or *rate scalable* source coders, whose output bit streams have a progressive reconstruction capability, exist, progressive transmission in the presence of channel impairments presents new challenges.

In this thesis we consider problems in joint source-channel framework and hence our principal objective is to maximize the end-to-end quality of the source reproduction at the receiver in a given *transmission budget* expressed in channels uses per source sample. We consider problems that fall in two broad categories. (i) First, we consider transmission of lossy sources over a noisy channel *when a feedback channel is available* from the receiver to the transmitter. (ii) Second, we consider progressive transmission of a lossy source over a channel in the absence of a feedback channel.

Before we embark on addressing the specific problems, in the following sections we discuss the research in the relevant topics, - namely joint source-channel coding, embedded or rate scalable source coding, progressive transmission and finally the use of feedback channel in communication problems.

1.2 Joint Source-Channel Coding

There is a large and still growing body of research in the area of Joint Source-Channel coding. Despite Shannon's "separation theorem" for memoryless channels [19], it is realized that for finite delays and non-asymptotic block lengths, it may be better to have some coupling between the compression schemes and the error control schemes, especially for loss tolerant sources like images and video. Throughout the thesis, by "source-coding" we refer to the map from the source domain to bits. It includes the quantizer as well as entropy coding if any. The source-coder output is a representation of the source at a certain encoding rate (or just "rate") that allows an approximate reconstruction of the source. The goodness of the approximation is measured by some distortion metric between the original and the reconstructed realization.

This coupling between the source-coding and the channel coding is implemented in a plethora of ways which can be classified broadly as follows. (i) *Tightly coupled systems:* Combined source-channel coding is where the source vectors are directly mapped to channel alphabet, and received channel symbols are directly used for estimating the source, without any explicit channel coding (*e.g.* [23]). Such approach, though optimal in operational rate distortion performance, is constrained by design and implementation complexity. (ii) *Source-aware channel encoding:* - Unequal Error Protection (UEP) is used when either the source, a transform or the compressed bit stream can be partitioned into portions with different sensitivity to channel noise and impairment. Error control codes of different strengths are assigned for different portions. Design procedure involve partitioning, sensitivity determination and resource allocation (*e.g.* [29, 46, 63]).

(iii) *Source-aware channel decoding:* - Such approaches use prior information (such as residual statistical dependence after compression) of the compressed source bitstream to obtain better estimates of channel coded bits (*e.g.* Source-Controlled Channel decoding [30, 60]). (iv) *Robust Source Encoding:* Modifying source coders to prevent error propagation is typically accomplished by fixed length quantization, packetization and resynchronization schemes *e.g.* [33], terminations for entropy coders, (*e.g.* [44]), source-interleavers(*e.g.* [51, 10]). (v) *Channel aware source-decoding:* Maximum A Posteriori (MAP) and Minimum Mean Squared Error (MMSE) estimation of the source, error detection and masking schemes, error concealment *e.g.* [69]), bad-frame masking, decoding for variable length codes.

The latest research in these areas focuses on efficient use of the available information at the decoding, turbo-like structures *e.g.* [25, 60], multiple description source-coding for networks (*e.g.* [54]), multicasting over noisy channels and delay constrained delivery (*e.g.* .[17]), and power and energy efficient source-channel coding (*e.g.* [41])

1.3 Rate-scalable or Embedded Source Coding and Progressive Transmission

The concept of rate-scalable source-coder is analogous to the decimal or binary expansion of a real number, where the real number is approximated more and more closely by adding more digits. A rate-scalable source-coder allows representation of the source at two or more different rates, where the representation at a lower rate is a prefix of that at the higher rate. Technically, all source coders are rate-scalable, as given any representation, some approximate reconstruction of the source, however bad, can always be obtained from any prefix of it. We are more concerned with *good* rate-scalable source coders which perform well at both the rates. Rate scalability is also referred to as SNR-scalability, and rate-scalable source-coding is also variously called successive approximation coding, layered coding, successively-refined coding, fine-grain scalable coding and embedded coding. In information theory, a successively refinable source is one for which a sequence of coding schemes exist which, asymptotically in blocklength, achieve minimum distortions at two different rates simultaneously. Not all sources are successively refinable in the information theoretic sense [21], but good rate scalable coders can still be designed.

Progressive transmission is the transmission of a multimedia source in layers, where the bits in "enhancement" layer further improve the quality of the reconstruction obtained by decoding the bits in the "base" layers. The size of the layers could be large - or it could be fine grained. In the absence of channel noise and impairments, the concept of progressive transmission is just semantically different from that of a layered or rate scalable source-coder. In the case where the transmission channel is noisy we distinguish between the source coding and the process of transmission.

1.4 Feedback Channel

Most modern communications systems allow simultaneous two way communication between the sender and the receiver on a link. The nature of the channels on forward and reverse links may be asymmetric, such as in communication from a stationary base station with high powered antenna to a mobile operating on low battery in a interference-ridden environment, or in a hybrid network with broadcasting satellite and a terrestrial uplink. But if such a channel is available, it can be exploited for efficient communication.

Again, the use of feedback is proved to have no effect on the information theoretic channel capacity of a discrete memoryless channel. It increases the capacity of a Gaussian channel only slightly [19]. Despite this result about asymptotic futility of feedback for increasing the capacity, Shannon indicated that feedback can be used to simplify the coding and communication. We find that, for schemes of comparable complexity, good transmission schemes using feedback indeed outperform schemes not using the feedback channel.

The techniques in literature which use feedback from the receiver to the transmitter can be classified as using the feedback in the form of (i) Information feedback [47, 48], (ii) Channel state feedback in the context of time varying channels (iii) Decision (ACK/NACK) feedback (*e.g.* [32], hosts of ARQ based methods [67])

Information Feedback: This is the most general form of feedback, where it is assumed that at each instant the receiver and the transmitter share the same information. This would be achieved if the receiver transmits all the raw received data (or observations or measurements) of the possibly corrupted received data back to the transmitter, instantly and accurately. In practice, this would imply that there is more traffic in the reverse direction than in the forward direction (*e.g.* in a BPSK encoded transmission of bits, information feedback would require that the floating point number generated by the matched filter for each transmitted bit, be sent back to the transmitter in an error free manner.) Though some clever schemes have been devised which make use of information feedback [47, 48], information feedback has limited applicability in the scenario of multimedia transmission to say, a mobile.

Channel State Feedback: In case of time varying channels, or even in case of memoryless channels, some side information about the channel behavior - such as observed channel SNR in mobile communication - may be known at the receiver at the time of the transmission.. This information can be made available to the transmitter by a feedback channel. This information is typically independent of the actual symbols being transmitted over the channel.

Decision (ACK/NACK) feedback: A complete information feedback is typically impractical. The reverse link may have limited data rate, probably a non-zero transmission delay, and may be error prone. In such cases receiver can use the feedback channel in a restricted way. A widely used feedback is Decision Feedback or ACK/NACK feedback. In such feedback, the receiver periodically generates a one bit feedback (ACK/NACK) about the acceptability of the received noisy symbols. In case of acceptability, ACK is sent or otherwise NACK is sent. Based on this feedback, the transmitter decides the next action, such as retransmission. ACK/NACK feedback, though restrictive, has the advantage that it is simple to generate and that it does not place too many demands on the reverse link. We shall exclusively look at ACK/NACK feedback in this thesis.

1.5 Contribution of the Thesis and its Overview

The thesis for the first time attempts to achieve progressive transmission of lossy sources in the presence of channel impairments and also addresses the ways to use a feedback channel. The contribution of the thesis can be categorized in in the following four categories, which form the four main chapters of the thesis.

(1) System design for progressive image transmission over noisy channels with feedback: Researchers have designed specific systems for transmission of images over noisy channels where they control the image coder, introduce robustness by carefully selecting the error protection for components of the image coder output and provide decoders which are targeted specifically towards images. All of the research did not use the feedback channel. We design a progressive image transmission system which uses the feedback channel. We design the transmission protocol to obtain the best end to end performance and then undertake direct comparison between the state-of-the-art image transmission systems which do not use feedback. We carry out the design for memoryless channels and for certain finite state channels. We observe an end-to-end gain of nearly 1 dB in average PSNR of the image for the channels and images selected. This work is presented in Chapter 2.

(2) Constrained feedback HARQ design for error control: This work concretizes the methodology used in the previous chapter for packetized transmission of general data over noisy channels. The system in Chapter 2 is a hybrid Forward Error Correction/Automatic Repeat Query (HARQ) protocol for transmission. Specifying a HARQ protocol requires describing its components codes and the transmission strategy - *i.e.* the finite state machine describing the sequence in which the bits of the component codes are transmitted. The sequence of transmissions can be described by a signal flow graph. Conventionally Hybrid ARQ schemes are designed and analyzed by first selecting component codes and the transmission strategy, and then analyzing the graph of the protocol by signal flow graph techniques for different channel parameters[13]. If we know the channel statistics, something better can be done. Instead of choosing a fixed protocol - *i.e.* the component codes and the graph first, we consider a class of protocols - *i.e.* a collection of codes and graphs at once. This allows us to consider a more general class of Hybrid ARQ protocols - namely variable-rate incremental-redundancy hybrid ARQ protocols - where the number of bits transmitted between two ACK/NACKs is allowed to be different. We provide a Controlled Markov Chain based design scheme which, unlike existing design schemes for hybrid ARQ, allows optimization of parameters over a collection of graphs, and provides direct control over the tradeoff between main performance measures of a hybrid ARQ protocol - namely throughput and reliability. In addition, an important performance measure is the average usage of the feedback channel - which, by counting decoding attempts per information packet, is directly related to the computational complexity of the protocol. The controlled Markov Chain based design methodology, allows constraining the feedback usage too and hence is dubbed Constrained Feedback HARQ design. The ability to control the tradeoff between throughput, reliability and feedback channel usage, allows comparison of HARQ schemes with pure Forward Error Correction techniques too. This work forms Chapter 3.

(3) Progressive joint-source channel coder in the absence of feedback or design of unequal error protections for progressive transmission of rate scalable image coders: Typically, the bits output by a rate-scalable source coder have differing sensitivities to channel impairements. Hence, in the absence of a feedback channel, there is a need for unequal error protection of the source coder output bits. Also, the optimal allocation of unequal error protection turns out to be different for different transmission budgets, even for transmission over stationary and memoryless bit error channels. We provide an algorithm to obtain the optimal unequal error protection profile from a given family of embedded error protection codes, so as to maximize the quality of the image at a given transmission budget. In addition, we show a way to *schedule* the error protection bits and the source coder bits in such a way that the optimal unequal error protection profiles for different transmission budgets can be obtained from a single bit stream. In this sense we *extend the notion of a rate-scalable source coder to a rate-scalable joint source-channel coder.* Transmitting the output of the joint source-channel coder results in optimized progressive transmission of the source. This work, presented in Chapter 4 is a dual of Chapter 2, where a feedback channel is available to carry out progressive transmission of images. Chapter 4 also presents the results for transmission of images over stationary and memoryless bit-error channels. Chapter 5 presents a small extension of the technique and presents image transmission results for compound packet erasure channels.

(4) Optimal use of ACK/NACK feedback for joint source-channel decoding: Chapter 6 considers the transmission scenario with the feedback channel again. We go back to first principles and consider the problem of design of a source-channel decoder for transmission of a general vector quantized source (not necessarily a scalable coder or an image coder,) over a noisy memoryless channel with a retransmission based protocol such as ARQ or Type-I hybrid ARQ. Conventionally ACK/NACK feedback is generated at the receiver by means of an error detection mechanism such as cyclic redundancy check (CRC). This feedback generation, though computationally efficient, is suboptimal for distortion-rate tradeoff. We address the problem of designing "distortion aware" feedback generation rules which obtain the best possible distortion-rate tradeoffs in the case when the transmitter does a pure retransmission and the receiver does packet combining of the received noisy copies of codewords. First we show that the problem of design of optimal ACK/NACK generation and decoding by packet combining can be cast and solved as a sequential decision problem. The optimal solutions found by dynamic programming give feedback generation rules which depend explicitly on the distortion metric. The Lagrangian of rate and distortion is shown to be the Bayesian risk of the corresponding sequential decision problem. Consequently, the optimal scheme for feedback generation and decoding is obtained by dynamic programming over the state space of posterior probabilities of the transmit codewords. Next, based on the structure of the optimal solution, we propose suboptimal joint source-channel decoders and "distortion aware" feedback generation rules, which outperform conventional pure channel-decoders and CRC/BER based rules.

(5) Progressive transmission with ACK/NACK feedback and pruned TSVQ in the presence of noise: The last contribution of the thesis is Chapter 7 which extends the definition of NACK feedback. NACK feedback generally denotes that the receiver finds the received bits unacceptable or unreliable. A better way of looking at NACK feedback in the context of joint source-channel coding is as a *request to continue transmission about the same source symbols*. We consider an extended joint source-channel system with ACK/NACK feedback where a tree structured quantizer is transmitted with one feedback per stage. The best feedback generation schemes are those whose operating points lie on the lower convex hull of the operational rate-distortion region. We show that the convex hull, similarly to an analogous property of Pruned TSVQs [15], can be traced by a collection of feedback generation schemes - all of which are *embedded*, in the sense that a higher transmission rate operating point can never send NACK where an ACK was sent by a scheme operating at a lower transmission budget. We also characterize the operating feedback generation policies by a "feedback threshold function" which makes the implementation of the feedback generation scheme easier.

1.6 The Issue of Delay and Transmission of Real-Time Sources

Extensive literature exists that deals with the communication of real-time sources, speech, audio and video over noisy and lossy channels for either streaming or real-time interactive applications. In this thesis we do not consider the time based deadlines and real time sources directly. Still, the concepts of progressive transmission and the necessity of constraining the feedback channel usage in the context of ACK/NACK feedback has implications on delay performance of a multimedia communication system. These issues have been concurrently addressed by other researchers and the ideas presented in this thesis can be effectively combined with techniques for delivery of delay sensitive multimedia over error and loss prone channels and networks. Some of the works which are closely related to the ideas presented in the thesis and applied to delivery of real time sources are in Chou et al [16, 17]. An overview of the collection of techniques available for video transmission can be obtained from the books by Hanzo et al for wireless [31], Sun et al for compressed transmission of video over networks [62], and the review articles and special issues in [20, 69, 2, 7, 27].

1.7 Overview

Figure 1.1 describes how the different chapters in the thesis are related. The chapters are designed to be self contained and the necessary introduction and literature review is provided at the beginning of each chapter. Concluding remarks are presented in Chapter 8.

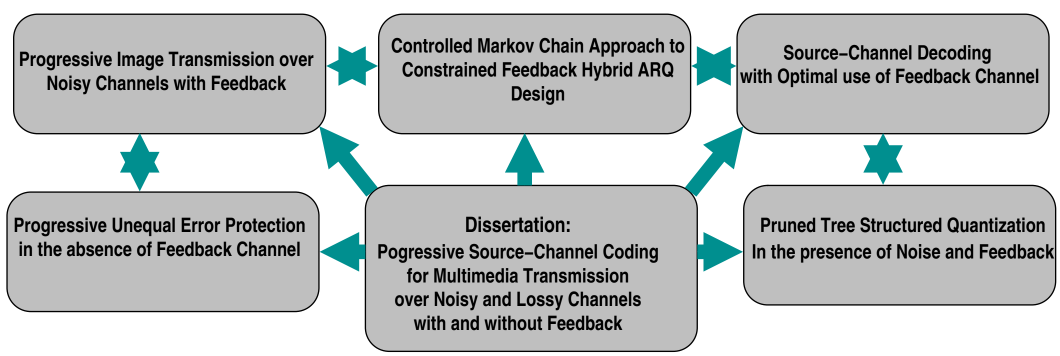

Figure 1.1: Thesis Organization and summary

Chapter 2

Image Communication over Noisy Channels with Feedback

2.1 Motivation

In addition to its evident relevance in delivery of multimedia to a wireless Internet user, digital image communication over noisy channels has applications in tele-medicine and modern battlefield. As argued in the introduction, an image is a loss-tolerant source, that is, typically, it can withstand errors and loss to a certain extent without compromising the visual information conveyed. It is of considerable interest to design efficient communication systems for image transmission over noisy and lossy channels. The problem has received much attention in the recent past A number of techniques have been suggested, which include suggestions for robust source coding (*e.g.* [14, 51]) Unequal Error Protection of subband coded images and recent works on error protection of progressively coded images [57, 59, 11, 58, 1, 39]. These techniques are primarily Forward Error Correction (FEC) based, and are designed for a one-way communication channel.

Most mobile communication systems allow two-way communication and hence there is a feedback channel available from the receiver to the transmitter. We address the problem of image transmission over noisy channels when such a feedback channel is available from the receiver to the transmitter. In this chapter, which describes the work at a system design level, we design an image communication system using limited feedback and obtain results superior to the state-of-the-art schemes not using feedback. We show how feedback can be effectively used in an image transmission system employing an embedded image compression algorithm like that of Said and Pearlman [52] and a family of embedded channel codes like Rate Compatible Punctured Convolutional (RCPC) codes [29]. We design the system for memoryless bit error channel and for 2-state Gilbert-Eliot channel. In the system design, we introduce the new concepts of (1) *variable incremental redundancy* hybrid ARQ-FEC protocol (2) *a Controlled Markov Chain approach* to design of such a protocol, *with constraints on the feedback channel usage*, (3) a quick design technique for such a protocol. Detailed discussion of the protocol design is provided in Chapter 3. In this chapter we describe the problem for memoryless and two state Gilbert-Eliot channels, describe the design, the optimization problem and its solution, followed by simulation results.

2.2 Transmission over Memoryless Channels

We first consider the problem of image transmission over a memoryless bit error channel with feedback.

2.2.1 The Feedback Channel

The challenge is to use the feedback channel in an efficient way so as to maximize the end-to-end quality of the image, for a given transmission budget (also called transmission rate) expressed in bits (or channel symbols) per pixel.

Information theory dictates that the capacity of a memoryless bit error channel (also known as binary symmetric channel) does not increase with feedback [19]. Notwithstanding this asymptotic result, in many practical systems, useful improvements in the throughput can be obtained by use of Hybrid Automatic Repeat ReQuest (ARQ)/Forward Error Correction (FEC) protocols instead of pure FEC protocols [29, 67].

There are ways of using the feedback channel which are more sophisticated than just the ACK/NACK feedback, such as a complete information feedback [47, 48], likelihood ratio feedback [65], and channel state feedback in the case of time varying channels. Complete information feedback is most general, but it may require a large data rate on the feedback channel. In fact, if the transmit information is binary and the received symbols are continuous valued then complete information feedback may require data rate much larger in the reverse direction than in the forward direction. Transmission of floating point numbers for the likelihood ratio feedback also has that drawback. Also, possibility of channel errors in the feedback channel also needs to be addressed satisfactorily.

Restricting the possible feedbacks to only two values of feedbacks has a possible drawback of sub-optimality. On the other hand, ACK/NACK feedbacks have the advantage that they are simple to generate, require low bandwidth to transmit over the feedback channel and, if necessary, can be protected easily by error correcting codes or by simple repetition. We use ACK/NACK feedbacks for our system. Consequently, for error control, we restrict our attention to the class of error control schemes which use Forward Error Correction as well as ACK/NACK feedbacks. Such a class of protocols is called Hybrid ARQ/FEC protocols or just HARQ protocols [67].

2.2.2 Selection of the Source-Coder

Consider a protocol for error control based on ACK/NACK feedbacks. In such a protocol, a ACK is sent from the receiver to the transmitter if the receiver is able to reliably recover the transmit information bits from the possibly corrupted received channel symbols. Otherwise a NACK is sent and additional transmissions for the same information bits are requested.

Note that such a protocol based on ACK/NACK feedbacks is inherently sequential. Also note that the number of channel symbols that need to be transmitted over the forward channel before a set of information bits is accepted by the receiver is a random variable. Conversely, the number of information bits recovered after the transmission of a fixed number of channel symbols is also a random variable.

The design objective is to maximize the average quality of the received image in a fixed transmission budget -expressed as total channel symbols transmitted over the forward channel. Clearly, this is accomplished if the quality of the received image is maximized for each channel realization. This will happen if the information bits recovered when the transmission budget is exhausted, give the best image representation for that rate.

The need for excellent image representation at a variable number of bit rates in the same stream is fulfilled by fine-grain rate-scalable or embedded image coders. The bitstream output by an embedded image coder is such that its every prefix, can be used to reconstruct the image, and the image quality improves with the length of the prefix, that is, a longer prefix results in a higher quality reconstruction. In addition, embedded image coders such as the SPIHT coder [52] are endowed with excellent rate distortion performance at all rates. The JPEG 2000 standard also incorporates highly efficient rate scalable image coding [44, 50]. The high flexibility in the selection of operating point on the operational rate distortion curve also makes them suitable for any transmission budget. The state-of-the-art image communication systems designed for noisy channel without feedback are designed as strong error protection applied to such embedded image coders [57, 59].

Consequently a high-performance fine-grain rate-scalable image coder is a natural choice for a source coder to be used with an ACK/NACK based error control protocol. We use the SPIHT image coder as the image coder of our choice.

One drawback of the embedded image coders such as SPIHT are such that, if some portion of the bitstream is not available or is irrecoverable from errors, then the bits that come after the missing portion cannot be used effectively in increasing the quality of the image, even if they are error free (see Figure 2.1). If some portion of the bitstream has undetected errors in it, then the bits following that portion may *decrease the quality* of the image.

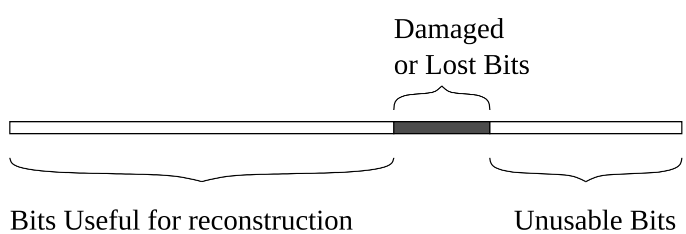

Figure 2.1: Typically, for a SPIHT like image coder, only the largest available prefix of the bitstream can be used for image reconstruction.

Therefore, on one hand, using a Hybrid ARQ protocol to transmit the output of an embedded source coder sequentially, will ensure that, the image is constructed to the highest possible quality from the successfully decoded bits, in every channel realization. On the other hand, the underlying protocol must have high reliability. That is, the probability of undetected post-decoding errors, *i.e.* the probability that an ACK is transmitted while the information bits are decoded incorrectly must be kept very low.

Therefore, given the choice of the source-coder, the task maximizing the quality of the received image reduces to the task of maximizing the throughput of the hybrid ARQ protocol, subject to high reliability.

2.3 Variable Incremental Redundancy Hybrid ARQ protocol

Keeping with the spirit of joint source-channel coding literature, we assume that the channel statistics (in this case, the bit error rate (BER)) are known. Hence for a given BER, we design a protocol which maximizes the throughput, subject to system constraint, which are, (i) computational constraints, (ii) available channel code family.

The block diagram of the transmission scheme is described in Figure 2.2.

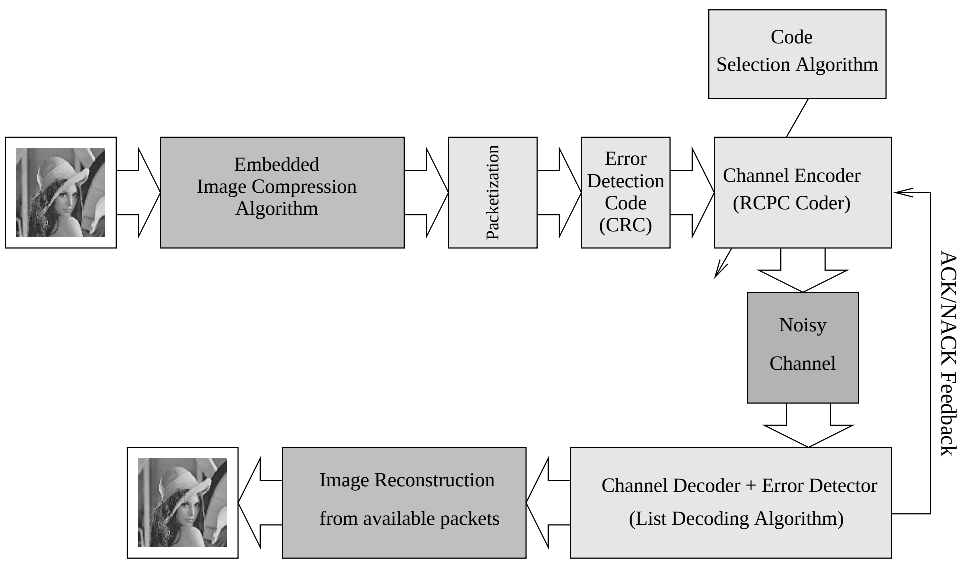

Figure 2.2: Block Schematic of Designed Scheme for Image Transmission

First, the output of an embedded image compression algorithm like that of Said and

Pearlman [52] is organized in fixed length packets; each packet is protected by error correcting codes and transmitted over a noisy channel. For now, let us consider the transmission of a single packet of r_s bits over the noisy channel. The protocol employed is as follows.

At the encoder, the packet is encoded by a concatenated channel code consisting of an outer error detection code (such as Cyclic Redundancy Check (CRC)) and an inner error correction code chosen from a family of RCPC codes. The output of the channel encoder is transmitted over the channel. Upon receiving the bits, the channel decoder attempts to correct the channel errors and recover the packet. The success or failure of decoding is determined by the error detection mechanism (we assume that the probability of undetected errors is zero). This result (success or failure) is conveyed back to the encoder by sending one bit (ACK/NACK) through the feedback channel (assumed to be error free).

On success, the encoder stops transmission for the current packet and proceeds with the transmission of the next packet. On failure, the encoder, according to a *decision policy,* switches to a stronger channel code and transmits on the channel *only the extra bits* needed for the chosen code. The decoder, on receiving the additional bits, makes another attempt at decoding the packet and verifies the outcome by the error detection mechanism. Because of rate compatibility of the underlying family of codes, the decoder can make use of the received bits from all *previous* transmissions to decode the packet. Again, the decoder conveys a success or failure bit back to the encoder through the feedback channel. The procedure is continued until, either the packet is successfully decoded, or the strongest channel code that can be chosen by the encoder's decision policy is used. In the latter event of *packet decoding failure*, the transmission for the packet is abandoned. We allow the number of these bits transmitted between two feedbacks to be different and hence dub the protocol a Variable Incremental Redundancy HARQ protocol.

The transmission of the image is stopped when the target transmission rate is reached. The source decoder reconstructs a replica of the image from the received packets.

2.3.1 Gain of using the Feedback Channel

Why should a scheme which uses a feedback channel in the way described above be expected to do better than a pure Forward Error Correction code and no feedback?

The gain obtained by using feedback is due to the fact that there is a nonzero probability that a packet protected by a weak channel code is decoded correctly by the decoder. This gain is maximized if we use one feedback bit for every transmitted bit. But note that, for schemes using decision feedback, the feedback channel usage is related to the computational complexity. For HARQ protocols, the generation of each feedback requires a channel-decoding operation, and hence a heavy use of feedback implies a large number of computations at the receiver.

Hence, a scheme which uses feedback in very small steps is impractical as it leads to a large feedback channel bandwidth, delay and most importantly, complexity. It turns out that most of the performance gain over pure FEC schemes can be obtained by a careful but limited use of feedback in a variable incremental redundancy HARQ protocol.

In the next section we design the protocol for given channel statistics, by developing a decision policy that minimizes the average number of bits transmitted on the channel for each packet under an explicit constraint on the feedback channel usage. This policy depends on the channel BER, the performance of each code in the family of RCPC codes, the bandwidth of the available feedback channel and the maximum tolerable probability of packet decoding failure.

2.4 The Design Problem and the Solution

We seek a decision policy for the encoder such that the average number of transmit bits per packet is minimized subject to (i) an upper bound on the number of feedback bits per source packet and (ii) an upper bound on the probability of packet decoding failure.

We define the *decision instant* for the encoder to be at the end of receiving a feedback bit. At a decision instant, the state of the encoder is described most generally by the sequence of channel codes used by it and the sequence of feedback bits received. Any general decision policy is described by specifying the next channel code to be used given the state of the encoder.

Using this notion of encoder state we translate the design problem to a a discrete optimization problem, which can be mapped to a finite horizon Markov Decision Process (MDP) problem [4]. The optimal policy obtained for the MDP (also called the controlled Markov chain) by dynamic programming yields a sequence of channel codes to be used in the protocol described in Section 2.3.

Let $\mathcal{C} = \{c_1, c_2, \ldots c_J\}$ denote a family of RCPC codes such that the code rates are decreasing, ***i.e.*** $r_c(c_1) > r_c(c_2) > \ldots > r_c(c_J)$. Let us also include in the family a 'null' or 'trivial' code c_0 which corresponds to transmitting nothing and hence has $r_c(c_0) = \infty$. For a fixed binary symmetric channel, let A_{c_j} denote the event that a packet encoded with code c_j is decoded successfully by the decoder. Let the probability of A'_{c_j} (complement of A_{c_j}) be denoted by $P_e(c_j)$. Clearly $P_e(c_0) = 1$. Then we make the following assumption.

Assumption: For the family of RCPC codes $\mathcal{C}$, assume that, $A_{c_i} \subset A_{c_j}$ whenever $i \leq j$.

This assumption implies $P_e(c_1) \geq P_e(c_2) \geq \ldots \geq P_e(c_J)$. Further, this assumption means that the probability of the event that a weaker code succeeds but a stronger code fails is zero. This is a reasonable assumption which is corroborated by simulations.

Consider the encoder at a decision instant. Let c_j be the last channel code used. If the last feedback bit denotes successful decoding, the decision of the encoder is clear; namely, that of stopping further transmission. So the decision policy must select a next channel code only if the last feedback bit has signaled a failure.

Under the above assumption, for $i < j < k$, the event A_{c_k} is conditionally independent of the event A_{c_i} given $A^{'}_{c_j}$ (or A_{c_j}). Therefore the encoder decisions need not depend on the complete sequence of channel codes used. This simplifies the notion of encoder state. We define the *encoder state* as the pair corresponding to the index of the last channel code used and the value of the last feedback received.

Therefore, the decision policy can be completely specified by *a sequence of channel codes of decreasing code rates.* So, by a policy π for transmitting a packet we mean an ordered subset of the collection C

$$\pi \overset{def}{=} (c_\pi^0, c_\pi^1, c_\pi^2, \dots c_\pi^{n(\pi)}), \tag{2.1}$$

where $c_\pi^0 = c_0$. The number of non-trivial codes used by the policy π is $n(\pi)$ and maximum number of feedback bits for the policy π is given by $n(\pi)-1$. The probability of packet decoding failure for the policy π is $P_e(c_\pi^{n(\pi)})$. Note that $n(\pi) = 1$ corresponds to *no feedback*. We impose a constraint that the packet decoding failure probability be less than a certain threshold p_e. To reflect the constraint on the feedback channel bandwidth, we require the maximum number of feedback bits per packet to be less than or equal to $M - 1$.

Now consider the transmission of a packet using a policy π. The event that transmission stops (*i.e.* ACK is received) after transmitting exactly $\frac{r_s}{r_c(c_\pi^k)}$ bits, is given by $\left(\cap_{i=1}^{k-1} A^{'}_{c_\pi^i}\right) \cap A_{c_\pi^k} = \left(\cup_{i=1}^{k-1} A_{c_\pi^i}\right)^{'} \cap A_{c_\pi^k}$, which, by the assumption, is $A^{'}_{c_\pi^{k-1}} \cap A_{c_\pi^k}$. Again, by the same assumption, the probability of this event is $P_e(c_\pi^{k-1}) - P_e(c_\pi^k)$.

Hence, given a policy π, the expected number of bits per packet to be transmitted on the channel can be written as,

$$R(\pi) \stackrel{def}{=} \sum_{k=1}^{n(\pi)} \frac{r_s}{r_c(c_\pi^k)}(P_e(c_\pi^{k-1}) - P_e(c_\pi^k)) + \frac{r_s}{r_c(c_\pi^{n(\pi)})} P_e(c_\pi^{n(\pi)}). \tag{2.2}$$

The last term represents the contribution of the event of packet decoding failure, namely $A'_{c_\pi^{n(\pi)}}$.

The problem of minimizing the average transmission rate is written as,

$$\min_\pi R(\pi) \text{ subject to } P_e(c_\pi^{n(\pi)}) \leq p_e \text{ and } n(\pi) \leq M. \tag{2.3}$$

We describe a simple dynamic programming based solution to this discrete optimization problem below. But it is particularly insightful to use a Controlled Markov chain framework. We develop the Controlled Markov Chain framework in detail in the next chapter.

To obtain the solution of problem (2.3), define, for any policy π, $1 \leq m \leq n(\pi)$ and $0 \leq j \leq m-1$,

$$r(\pi, j, m) \stackrel{def}{=} \sum_{k=j+1}^{m} \frac{r_s}{r_c(c_\pi^k)}(P_e(c_\pi^{k-1}) - P_e(c_\pi^k)) + \frac{r_s}{r_c(c_\pi^m)} P_e(c_\pi^m). \tag{2.4}$$

Define $r(\pi, m, m) \stackrel{def}{=} \frac{r_s}{r_c(c_\pi^m)} P_e(c_\pi^m)$. Also define, for codes $c_{init}, c_f \in \mathcal{C}$ with $r_c(c_f) < r_c(c_{init})$, $0 \leq j \leq m-1$,

$$r^*(c_{init}, c_f, j, m) \stackrel{def}{=} \min_{\pi: c_\pi^j = c_{init},\ c_\pi^m = c_f} r(\pi, j, m). \tag{2.5}$$

Then for $0 \leq j \leq m-1$, it is easy to see that $r(\pi, j, m)$ admits the following decomposition.

$$r(\pi, j, m) = \frac{r_s}{r_c(c_\pi^{j+1})}(P_e(c_\pi^j) - P_e(c_\pi^{j+1})) + r(\pi, j+1, m) \tag{2.6}$$

Now, as $r(\pi, j+1, m)$ does not depend on c_π^j, and c_π^{j+1} is the 'initial' code in $r(\pi, j+1, m)$, we have, by the optimality principle, the following dynamic programming equation.

$$r^*(c_{init}, c_f, j, m) = \min_{c \in \mathcal{C}: r_c(c_f) \leq r_c(c) < r_c(c_{init})} \frac{r_s}{r_c(c)} (P_e(c_{init}) - P_e(c)) + r^*(c, c_f, j+1, m) \tag{2.7}$$

Notice that, in (2.3), the constraint $P_e(c_\pi^{n(\pi)}) \leq p_e$ is satisfied by selecting the weakest (highest $r_c(c)$) code with desired error performance, *i.e.* $c_\pi^{n(\pi)} = arg\max_{c \in \mathcal{C},\ P_e(c) \leq p_e} r_c(c)$. Let this code be denoted by c_f^*. Then by the notation developed, the solution to problem (2.3) is $r^*(c_0, c_f^*, 0, M)$. The optimal policy is obtained by recursively solving eq. (2.7) and setting the j^{th} code in the policy to be the one achieving the minimum in (2.7).

Again, as the output of the image compression system is embedded, minimizing the (average) number of transmit bits per packet is equivalent to maximizing the (average) number of source packets for a target transmission rate in bits/pixel, which, in turn, is equivalent to maximizing the (average) Peak Signal-to-Noise-Ratio (PSNR) for a target transmission rate.

2.5 Results for Memoryless Channel

In the paper by Sherwood and Zeger, [57], the authors reported some of the best results for transmission of images over memoryless bit error channels in the absence of feedback. We undertook the task of determining the performance improvement over the scheme in [57] that can be achieved by feedback and the scheme described in the previous sections. The average PSNR (dB) results of transmitting the grey-scale test-image LENNA over binary symmetric channels for different target transmission rates for different channel BER's are reported in Figure 2.3 and the values of PSNR for some transmission budgets are tabulated in Table 2.1.

The image Lenna, of dimensions 512x512 was compressed with the Said and Pearlman coder [52] with arithmetic coding. The family of channel codes, $\mathcal{C}$, was chosen from Rate Compatible Punctured Convolutional (RCPC) codes in [29]. The source-coder output was divided into source-packets of size 32 bytes each ($b = 256$). A two-byte CRC was used as an outer error-detection code. The inner error-correction code family were the collection of RCPC codes similar to those used in [57]. Specifically, a mother code for BER of 0.1 was a 64 state rate 1/3 code, while that for BER of 0.01 was a 16 state rate 1/4 code taken from [29]. List Viterbi (LV) Decoding ([55],[57],) was used with hamming distance as path-metric and a search depth of 100 in both cases. That is, for error detection with LV decoding, a feedback of NACK is sent from the receiver to the transmitter if the CRC is not satisfied in the top 100 paths of the trellis. The system with a maximum feedback of zero bits corresponds closely to that in [57].

One can observe from Table 2.1 that irrespective of the target transmission rate, a carefully chosen feedback of just a few bits per source packet, (less than 0.01 feedback bits per source bit) can consistently improve the PSNR by about 1 dB over a system which uses no feedback. Notice that most of the gain is obtained by introducing the feedback of just one bit per source-packet. Additional gains are obtained by allowing more feedbacks. The gain is nearly 1.2 dB the case of BER 0.01 with only 1 feedback bit per source packet. It was observed that the improvement in performance on further increasing the feedback was negligible.

From Figure 2.3, it can be observed that the gain of a system with feedback over one without feedback *increases* with transmission rate. This phenomenon can be explained by the following. At high source-coding rates, the PSNR(dB)-source-coding-rate curves for coders like SPIHT are nearly linear. For noisy channels, the higher throughput obtained by introducing feedback, yields a PSNR-transmission rate curve with a steeper slope than one for system with no feedback. Hence the performance gap between the two systems widens with transmission rate.

Maximum Feedback in bits per source packet	Transmission Rate bits/pixel		
BER 0.1	0.25	0.5	1.0
3	29.53	32.48	35.53
2	29.45	32.36	35.41
1	29.32	32.09	35.21
0	28.56	31.50	34.36
BER 0.01			
1	33.16	36.26	39.37
0	31.98	35.07	38.12

Table 2.1: PSNR (dB) Results for Image LENNA over BSC's.

2.6 Extension to Finite State Channels

The gains obtained in the previous section over systems not using feedback indicate that it is indeed worthwhile to use a feedback channel if one is available, even for memoryless channels.

In this section we extend the results for memoryless channels to the Gilbert-Elliot channel. Such finite-state Markov channel models have been shown to be good approximations for binary transmission over slowly varying flat fading channels [66].

Gilbert-Elliot channel model is a Markov channel model with two states. The channel is assumed to be a binary symmetric channel (BSC) in each state. The bit-error rates (BER) in the good state G and the bad state B are denoted by ϵ_G,and ϵ_B respectively. The transitions between the states are Markov and are assumed to be unknown at the receiver or the transmitter. The model is specified by the BERs ϵ_G, ϵ_B, the steady state probability of state B, P_B, and the average sojourn time of state B, T_B.

The Gilbert-Elliot model is depicted in Figure 2.4. We shall refer to the BSC's with BER ϵ_G and ϵ_B as BSC_G and BSC_B respectively.

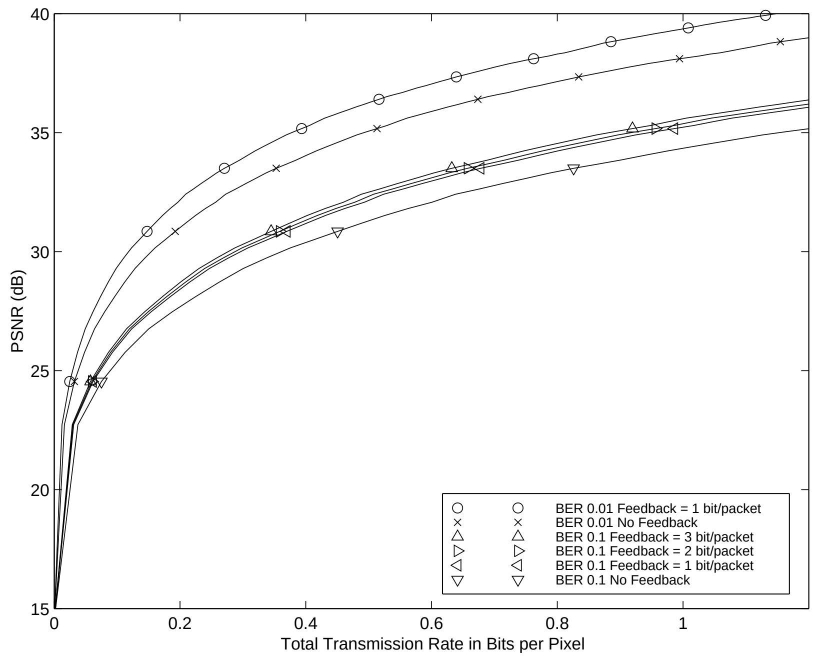

Figure 2.3: Performance Comparison for progressive transmission of image LENNA over BSC, with and without a feedback channel. BER 0.01 and 0.1

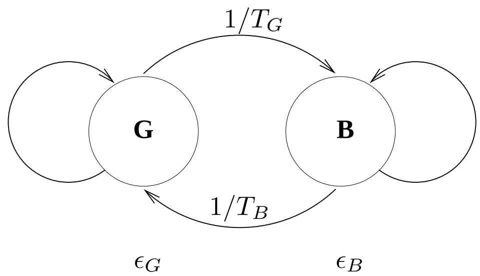

Figure 2.4: Gilbert-Elliot channel

The system schematic for this channel can also be described by Figure 2.2. The transmission protocol is slightly modified and is described in the next section.

As earlier, the usage of the feedback channel is an additional design parameter for such a scheme. We demonstrate that in this case, significant performance gains over a system without feedback, can be achieved with moderate use of the feedback channel.

2.6.1 Gain of using the Feedback Channel

As we have seen in previous sections, careful use of feedback achieves throughput gains for memoryless channels. In addition, for finite state channels, the gain obtained by the proposed schemes which use feedback, can be attributed to another factor. Essentially, the proposed combination of embedded source-coder and a HARQ protocol accomplishes an implicit adaptation of the instantaneous allocation of source-coding rate and channel-coding rate according to the channel conditions. The adaptation of the channel-coding rate, according to the channel conditions, is accomplished *without explicit transmission of the channel state information* using decision feedbacks (ACK/NACK) from the receiver to the transmitter. We shall see that the system with feedback outperforms , with larger gains compared with the memoryless case, the state-of-the-art pure Forward Error Correction systems designed for the channel, such as [59].

2.7 Changes in the Protocol

The protocol for finite state channels is a slight modification of the protocol for the memoryless channel described in Section 2.3. Again, Let $C = \{c_1, c_2, \ldots, c_J\} \cup \{c_0\}$ denote the available family of rate compatible channel codes used for error correction. Each code $c \in C$ is a $(\frac{b}{r_c(c)}, b)$ block code where channel code-rate $r_c(c)$ includes the code-rate for the error detection code. Then the HARQ protocol employed is described by specifying a *policy* $\pi \stackrel{def}{=} (c_\pi^0, c_\pi^1, c_\pi^2, \ldots, c_\pi^{n(\pi)})$, which is a subset of $\mathcal{C}$ ordered by *decreasing code-rates* and where $c_\pi^0 = c_0$. A fixed length source-packet is transmitted using the variable incremental redundancy HARQ protocol described in Section 2.3. Because the finite state channel can go into a severe state, *it is possible that the strongest channel code in the policy may not be able to correct all the errors.* We call this event a *policy-failure* for the source-packet. In the event of a policy-failure, *all the received bits for the packet are discarded and the transmission for the source-packet is started from the beginning, i.e.* from the first code in the policy. In other words, for each packet the system emulates a generalization of Type I HARQ [67] where retransmission of a codeword is done in several steps of incremental redundancy, as determined by the policy. This modification is chosen because it yields a tractable throughput approximation. The same policy is applied to the transmission of all the source-packets.

The transmission of the source-packets is stopped when the transmission budget is exhausted. The image is reconstructed from the successfully received source-packets, which form an error-free representation of the source at some rate. This way the system dynamically trades source-bits for channel-bits whenever necessary.

2.8 The New Problem and the Solution

Now let us consider the design of a transmission policy with at the most M steps. Among all the allowed policies, (all the subsets of $\mathcal{C}$ with M or less elements), the task is to select a policy so that, in a given transmission budget in bits per pixel, the average number of source-bits that are delivered reliably at the receiver is maximized. We look at a normalized version of the above objective function, namely, the *throughput* of the policy over the channel. The throughput $\eta(\pi)$ of a policy π is defined as the average number of source-bits correctly received per channel-bit transmitted. It is independent of the transmission budget. Hence, the best policy with M steps is the one which solves,

$$\max_{\pi} \eta(\pi) \text{ subject to } n(\pi) \leq M. \tag{2.8}$$

Note that $M = 1$ corresponds to the conventional Type I HARQ, while $M = 2$ and higher are the schemes based on decoding by code-combining ([67]).

To limit further the average number of feedbacks sent per source-packet, we may impose the following implicit constraint on the average number of feedbacks per source-packet for the allowed policies. Let $P_e^G(\pi)$ denote the probability of policy-failure when a source-packet is transmitted over BSC_G while using the policy π. Similarly define $P_e^B(\pi)$. Then, we may require,

$$\max(P_e^G(\pi), P_e^B(\pi)) \leq p_e, \tag{2.9}$$

for some small number p_e. Any policy π, satisfying the constraint (2.9) has an average feedback less than $n(\pi)/(1-p_e)$ per source-packet in each state of the Gilbert-Elliot channel and hence in the channel itself.

2.8.1 Throughput Estimation

Let $E(\mathcal{R}(\pi)|G)$ and $E(\mathcal{R}(\pi)|B)$ denote the expected number of channel-bits transmitted for successful transmission of a source packet by policy π, over channels BSC_G and BSC_B respectively. Now, for the Gilbert-Elliot channel, if the sojourn times of states G and B are much larger than $E(\mathcal{R}(\pi)|G)$ and $E(\mathcal{R}(\pi)|B)$ then the throughput $\eta(\pi)$ of the policy π can be approximated by,

$$\eta(\pi) \approx \frac{bP_G}{E(\mathcal{R}(\pi)|G)} + \frac{bP_B}{E(\mathcal{R}(\pi)|B)}. \tag{2.10}$$

Here P_G and P_B are the steady state probabilities of the two channel states.

This can be simply seen as follows. If the sojourn times of states G and B are long compared to $E(\mathcal{R}(\pi)|G)$ and $E(\mathcal{R}(\pi)|B)$, the transmission of single packet does not encounter a channel-state change. So in essence, for a fraction P_B of time, the transmission is like that over BSC_B. The throughput in that case is $\frac{b}{E(\mathcal{R}(\pi)|B)}$. Similarly, throughput for the portion of time when channel is in state G is $\frac{b}{E(\mathcal{R}(\pi)|G)}$. Averaging by the steady state probabilities, we get the expression in eq. (2.10).

The values $E(\mathcal{R}(\pi)|G)$ and $E(\mathcal{R}(\pi)|B)$ are estimated by the technique outlined in 2.4 as follows. Let $P_e^G(c)$ denote the probability that a source-packet encoded with channel code $c \in C$ and transmitted over the channel BSC_G could not be decoded successfully. Then for the policy π, $E(\mathcal{R}(\pi)|G)$ can be approximately written as

$$E(\mathcal{R}(\pi)|G) \approx \frac{\left(\sum_{k=1}^{n(\pi)} \frac{b}{r_c(c_\pi^k)}(P_e^G(c_\pi^{k-1}) - P_e^G(c_\pi^k)) + \frac{b}{r_c(c_\pi^{n(\pi)})} P_e^G(c_\pi^{n(\pi)})\right)}{(1 - P_e^G(c_\pi^{n(\pi)}))} \tag{2.11}$$

Here $P_e^G(c_\pi^0) = 1$. Other probabilities $P_e^G(c)$ are obtained by simulation. Also $P_e^G(\pi) \approx P_e^G(c_\pi^{n(\pi)})$. Similarly $E(\mathcal{R}(\pi)|B)$ and $P_e^B(\pi)$ can be computed. This is an approximation as it is based on the assumption in Section 2.4.

Very interestingly, if eq. (2.11) is used for $E(\mathcal{R}(\pi)|G)$, then the estimate of the throughput, described by eq. (2.10) remains valid under a weaker assumption. Instead of assuming that the channel state does not change during the transmission of an entire source-packet, if we just assume that the channel state does not change before a packet decoding failure, we get the same expression for throughput. This is so because, as the protocol discards all the previously received bits after a packet decoding failure, a packet decoding failure is a renewal instant.

The optimal policy solving equation (2.8) is obtained by exhaustively searching over all the policies meeting the desired constraints.

2.9 Simulation Results for Gilbert Elliot Channels

For simulations, image Lenna, was compressed with the Said and Pearlman coder [52] with arithmetic coding. The family of channel codes, $\mathcal{C}$, was chosen from Rate Compatible Punctured Convolutional (RCPC) codes in [29]. The source-coder output was divided into source-packets of size 32 bytes each ($b = 256$). A two-byte CRC was used as an outer error-detection code. The inner error-correction code family was the collection of RCPC codes obtained from a 16 state, rate 1/4 code taken from [29]. List Viterbi (LV) Decoding (*e.g.* [57],) was used with hamming distance as path-metric and a search depth of 10. For error detection with LV decoding, a feedback of NACK is sent from the receiver to the transmitter if the CRC is not satisfied in the top 10 paths of the trellis.

The simulation results are presented for a class of channels with the following parameters: (1) Bit Error Rates $\epsilon_B = 0.1, \epsilon_G = 0.001$, (2) Different steady state probabilities $P_B \in \{0.1, 0.2\}$, (3) Different average sojourn times for state B in bits $T_B \in \{400, 2000, 10000\}$. We compare three systems in this section. System A is a scheme which uses feedback, when the implicit constraint on the feedback, given by equation (2.9), is not applied. The scheme chooses to maximize the throughput estimate given by equation (2.8) over all the allowed policies. System B is a scheme which puts a constraint on the feedback channel usage, irrespective of the channel state, by requiring that (2.9) be satisfied. System C is a scheme without feedback given in the paper by Sherwood and Zeger [59]. For error correction, it uses a product code of RCPC-CRC code and Reed Solomon codes, with interleaving for recovery from burst errors induced by the channel entering in bad state. The results are obtained from the throughput calculations reported in [59].

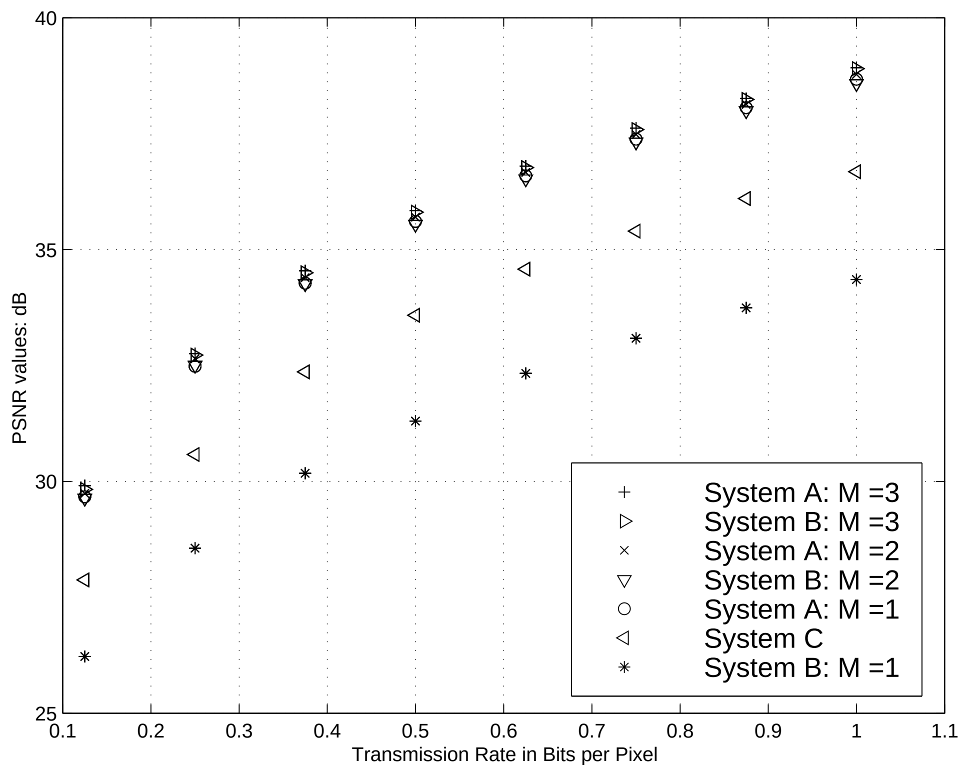

Figure 2.5: Average PSNR (dB) Performance comparison for different schemes for Lenna: Gilbert-Elliot Channel with $P_B = 0.1$, $T_B = 400$ bits.

Figure 2.9 shows the average PSNR performance of Systems A, B and C for a Gilbert-Elliot Channel with parameters $P_B = 0.1, T_B = 400bits, \epsilon_B = 0.1, \epsilon_G = 0.001$ for the transmission of the image Lenna, as a function of transmission budget in bits per pixel. Let us first look at the designed system with unconstrained feedback, System A, and the system with no feedback, System C. First notice that for this channel, even the simplest case of System A, namely the one with $M = 1$, can perform significantly (up to 2 dB) better than System C which, to our knowledge, reports the best results reported for a scheme without feedback. Increasing the number of steps in the policy, (*i.e.* making $M > 1$,) gives further, though relatively small,(up to 0.3 dB) gains at all transmission rates. This can be explained as follows. For the given channel, System A with $M = 1$ chooses a high-rate code of code-rate 0.82. This code is sufficient to recover the source-packet reliably when the channel is in the good state ($\epsilon_G = 0.001$), but almost always fails when the channel is in the bad state ($\epsilon_B = 0.1$). When the channel is in a bad state, the policy repeatedly request a retransmission, until the bad state is over. This way, the policy automatically implements an "outage", which, in this case, is favorable for the throughput. Therefore, despite the high throughput, the usage of the feedback channel is high, especially in the bad state.

The feedback channel usage is explicitly controlled in system B, where constraint (2.9) is satisfied for $p_e = 0.01$. For M=1, this results in a highly conservative system, designed for the worst case, such that its performance for the channel is inferior to that of System C, which has no feedback. But if a single intermediate step is allowed in the policy (M=2), then the constrained feedback scheme, system B, performs close to System A. The optimal policy for system B, M =2 contains codes with rate 0.82 and 0.264. Hence Systems A and B with a policies designed this way can switch adaptively between channel code rates so as to suit the channel state. Note that no explicit channel state information is obtained or transmitted.

Table 2.2 lists the average PSNR performance of the systems A and B for the image Lenna, for different channel parameters, for transmission rates 0.25 and 1.00 bits per pixel. Increasing the number of steps from $M = 1$ to $M = 3$, yields PSNR gains of up to 0.25 dB at all transmission rates over System A with $M = 1$. Table 2.3 gives the comparison of the observed throughput with the estimated throughput given by eq. (2.10). As expected the analytical approximation of the throughput becomes closer to the true throughput as the average sojourn time for the bad state increases. Table 2.4 gives illustrative results for the average number of feedbacks needed per source-packet for the two schemes, for the channel with $P_B = 0.2$ and different values of T_B. It is evident that System A, for $M = 1$, requires a large number of retransmissions. On the other hand, System B for $M = 1$ requires a very small number of retransmissions, but provides a low throughput. Allowing one intermediate step in transmission, *i.e.* $M = 2$, increases the throughput of both the systems, while *reducing* the feedback channel usage for System A radically. It can also be observed that, for the cases $M > 1$, the additional implicit feedback constraint, eq. (2.9), does not reduce the feedback by a large margin.

Table 2.5 gives illustrative results for the Mean Absolute Difference (MAD) and Standard Deviation (STD) of the observed PSNR for two different transmission rates for the image Lenna. The variations in PSNR decrease with increasing transmission rate and increasing M, though the latter trend is not quite consistent.

2.10 Conclusion

In the presence of a feedback channel, the combination of an embedded image coder, a rate compatible family of channel codes, and transmission using a HARQ protocol, provides a simple and efficient scheme for image transmission. The system trades source-bits for channel bits achieving adaptive and dynamic allocation of source coding rate

P_B	T_B	$n(\pi)$	Syst. A		Syst. B	
			0.25bpp	1.0bpp	0.25bpp	1.0bpp
0.1	400	1	32.48	38.67	28.56	34.35
		2	32.64	38.77	32.50	38.58
		3	32.75	38.92	32.72	38.90
	2000	1	32.75	38.88	28.55	34.35
		2	32.89	39.06	32.87	39.01
		3	32.98	39.12	32.94	39.10
	10000	1	32.72	39.01	28.55	34.35
		2	32.85	39.04	32.76	39.05
		3	33.08	39.16	32.96	39.15
0.2	400	1	31.61	37.86	28.55	34.35
		2	32.04	38.18	31.64	37.79
		3	32.24	38.38	32.19	38.27
	2000	1	31.90	38.40	28.55	34.35
		2	32.45	38.76	32.39	38.54
		3	32.56	38.85	32.49	38.74
	10000	1	31.94	38.40	28.55	34.34
		2	32.41	38.76	32.53	38.68
		3	32.58	38.85	32.64	38.80

Table 2.2: PSNR (dB) Performance of optimized policies over G-E channel with different parameters: 1) System A - unconstrained feedback 2) System B - constrained feedback, Image: Lenna.

and channel coding rate for a realization of the channel. The use of feedback can yield significant improvement in the quality of the received image over a system not using feedback, especially for time varying channels such as the Gilbert-Elliot channel. The complexity of the system and the usage of the feedback channel for the proposed systems can be controlled by constraining the search space for policies appropriately. We obtain nearly 1 dB gain in average received PSNR over state of the art systems not using feedback in the case of memoryless channels. The gains are over 2 dB for the Gilbert Elliot channel. Simulation results indicate that a system with constrained but carefully designed feedback can achieve a large fraction of gains with small usage of the feedback channel and consequently a small number of decoding attempts. The transmission of the image is progressive by design. Overall, it may be worthwhile to exploit the feedback

P_B	T_B	$n(\pi)$	Syst. A		Syst. B	
			Est. η	η	Est. η	η
0.1	400	1	0.740	0.685	0.264	0.261
		2	0.769	0.700	0.766	0.673
		3	0.773	0.721	0.773	0.718
	2000	1	0.740	0.716	0.264	0.261
		2	0.769	0.744	0.766	0.737
		3	0.773	0.754	0.772	0.752
	10000	1	0.740	0.738	0.264	0.261
		2	0.769	0.744	0.766	0.745
		3	0.773	0.762	0.772	0.760
0.2	400	1	0.658	0.559	0.264	0.261
		2	0.716	0.595	0.710	0.562
		3	0.723	0.635	0.722	0.632
	2000	1	0.658	0.622	0.264	0.261
		2	0.716	0.683	0.710	0.669
		3	0.723	0.698	0.722	0.695
	10000	1	0.658	0.644	0.264	0.260
		2	0.716	0.701	0.710	0.690
		3	0.723	0.715	0.722	0.706

Table 2.3: Throughput observed vs. estimated, for G-E channel with different parameters: 1) System A - unconstrained feedback 2) System B - constrained feedback.

channel for image transmission if it is available.

T_B	$n(\pi)$	Syst. A	Syst. B
400	1	1.46	1.000
	2	1.217	1.198
	3	1.235	1.231
2000	1	1.288	1.001
	2	1.114	1.105
	3	1.159	1.151

Table 2.4: Average number of feedbacks per source-packet for Gilbert Elliot Channel with $P_B = 0.2$.

T_B	$n(\pi)$	0.25 bpp		1.00bpp	
		MAD	STD	MAD	STD
400	1	0.18	0.24	0.12	0.15
	2	0.11	0.13	0.07	0.08
	3	0.10	0.12	0.04	0.05
2000	1	0.24	0.30	0.15	0.19
	2	0.16	0.20	0.08	0.10
	3	0.18	0.23	0.08	0.08

Table 2.5: Variation of the PSNR (dB) from the mean value for System A, $P_B = 0.1$ for Lenna.

Chapter 3

Constrained Feedback Hybrid ARQ Design

3.1 Introduction

Transmissions over wireless channels experience large bit error rates due to fading and interference. Hence strong error control needs to be employed. In situations when two-way communication is possible, the error control protocols can make use of the feedback channel for better or more efficient error correction. It has been established that the information theoretic capacity of a memoryless channel is not increased in the presence of a feedback channel [19]. But in practice a combination of Forward Error Correction and Automatic Repeat Query, called Hybrid FEC/ARQ (HARQ) can have better throughput than pure ARQ and pure FEC for comparable reliability [35, 67].

HARQ protocols are typically implemented by transmission of incremental redundancy for an embedded (rate-compatible) family of channel codes at the transmitter (*e.g.* Rate Compatible Punctured Convolutional Codes [29], or punctured Reed Solomon Codes [68]) and by code-combining [13] at the decoder. The performance of a HARQ protocol is measured by *throughput* and *reliability*. In HARQ, the generation of each feedback bit requires a decoding operation. Hence the average number of feedback bits for the channel is a measure of the complexity of the protocol.

The techniques to analyze the performance of fixed HARQ protocols for different channel conditions are well developed in literature [67, 35, 36]. They make extensive use of the underlying signal flow graph of the protocol (*e.g.* [38, 67]), and compute the performance measures such as the throughput and the reliability from its transfer function. In this chapter we address the dual problem, namely, that of designing the best HARQ protocol from a collection of protocols for *a given channel condition.* That is, we consider packetized transmission over a memoryless noisy channel with known Bit Error Rate or Symbol Error Rate and investigate the design of the best protocol for that channel from a collection of HARQ protocols with possibly different underlying signal flow graphs. Recognizing the fact that forcing the the number of bits between two ACK/NACK feedback to be equal is too restrictive, we allow them to be variable, *i.e.* we consider Variable Incremental Redundancy HARQ protocols. It increases the complexity of buffer management slightly but results in gains in throughput.

The conventional analysis approach focuses on a single signal flow graph and hence is inadequate as the search space of the protocols contains protocols with different underlying graphs.

Our methodology allows us to address the problem of HARQ design when there is a constraint on the *average feedback channel usage.* This is relevant, as, although incremental transmission of redundancy in very fine increments has maximal throughput, it may not be computationally feasible. Also, in a multicasting scenario, such a design might result in a feedback implosion.

We show that, for a fixed channel, both the problems - namely the task of choosing the optimal HARQ protocol from a given family of channel codes, under unconstrained or constrained feedback, can be mapped to a Markov Decision Process (MDP) with discrete states, alternatively called a controlled Markov chain (CMC) [4] problem. The protocol maximizing the throughput is obtained by solving the optimization of the controlled Markov chain through dynamic programming. The constraint on the feedback is achieved by the use of Lagrange Multipliers. The Lagrangian, the weighted sum of transmission costs and feedback costs, also arises naturally when carrying of the performance computation with transmission delays and overheads.

The contributions of the chapter are, (i) the variable incremental redundancy constrained feedback HARQ protocol, with useful performance improvement over conventional Type I or Type II HARQ protocols, (ii) the MDP or CMC framework for design of such a protocol, which allows operationally optimal tradeoffs between performance metrics. The methodology improves over the conventional signal flow graph approach. In addition, we illustrate our methodology by designing HARQ protocols with Reed Solomon Codes. We also develop analytical expressions and approximations for estimating the transition probabilities.

The chapter is organized as follows. In the next section, Section 3.2, we describe general ARQ and HARQ protocols. Section 3.3 we describe the design problem as a Controlled Markov Chain. Section 3.4 describes how throughput, reliability and average feedback are calculated. In section 3.5 the underlying optimization problem is set up and the solution is described. In section 3.6, simulation results using Punctured Reed Solomon Codes are presented. 3.8 describes the analytical expressions for calculation/approximation of the transition probability in a Punctured Reed Solomon code family. Section 3.7 is the concluding section.

3.2 ARQ and Hybrid ARQ Protocols

ARQ based protocols have been extensively used at the link layer level for point to point communication on a noisy two way link. Retransmissions are also used in end-to-end error recovery at the transport layer, for communication over a lossy packet based network[64, 5]. In a *pure ARQ* protocol for packetized transmission over a noisy channel, the transmitter encodes every packet with an error detection code. A packet is transmitted repeatedly until it is received "correctly" by the receiver as decided by the error detection code and as conveyed to the transmitter by ACK/NACK feedback. The three standard flavors of a pure ARQ protocol are the basic *Stop and Wait (SW)* scheme, the *Go-Back-N (GBN)* scheme, which requires buffers at the transmitter, and the *Selective Repeat (SR)* Scheme, which requires buffers at the transmitter and the receiver. The GBN and the SR schemes are ways of statistical multiplexing across packets, to keep the channel busy and achieve higher throughput in the presence of propagation/queuing delays.

In a HARQ protocol [67, 35], the transmitter encodes every packet with an error correcting code (FEC) which also allows error detection at the receiver. When such a channel code fails to correct the errors at the receiver, the error detection mechanism is used to detect the failure. The result is conveyed to the transmitter by ACK/NACK feedback.

The simplest HARQ protocol is a Type-I hybrid ARQ protocol, where, like a pure ARQ, copies of a packet encoded by a fixed channel code are transmitted repeatedly till ACK is received. A generalization of HARQ protocol is obtained when the the protocol allows transmission of the channel codeword (information symbols and parity check symbols) in increments. Mandelbaum proposed this technique of *incremental transmission of redundancy* in [43] where he recognized the usefulness of MDS property of Reed Solomon Codes for this purpose. Rate Compatible Punctured Convolutional (RCPC) codes [29] are convolutional codes which allow such incremental transmission.

The notion dual to such incremental transmission of redundancy at the transmitter, is *packet combining* or *code combining* at the receiver [13, 67]. In packet combining or diversity combining, several noisy *copies of the same codeword* are combined at the receiver to decode (estimate) the transmit packet better. Code combining is a generalization of packet combining and is a concept similar to sensor fusion. A receiver is said to do code combining when it combines several noisy codewords or codeword fragments, obtained by encoding the same packet by possibly *different* channel codes, in order to decode the packet.

HARQ protocol for transmission of a single packet, over a memoryless noisy channel can be described by a finite state machine or a signal flow graph. The protocol starts in a state s_0, and if necessary, goes through states $s_1, s_2, s_3, \ldots, s_N$ in a prespecified order, according to the underlying signal flow graph, before terminating in state s_T.

Figure 3.1 shows the signal flow graph of a variation of Type II HARQ protocol. Figure 3.2 shows the bare-bones of signal flow graph of a general HARQ protocol under the assumption of error free feedback and no timeout.

Under these assumptions, for such a protocol, in each state s the transmitter transmits a prespecified set of bits $g(s)$ for the packet. The receiver receives a noisy version of it and decodes it and sends a ACK/NACK feedback. Figure 3.2 also shows the bare-bones of the protocol of Figure 3.1 and that of a Type I HARQ protocol. In the next section, we see how the figure can be interpreted as the State-Action diagram of a policy of a controlled Markov Chain.

3.3 Controlled Markov Chain for HARQ

We consider a basic transmission scheme which is similar to a Selective-Repeat HARQ system with ACK/NACK feedback except that we allow a variable number of bits to be transmitted between two feedback requests. We assume that the buffer-size at the transmitter and receiver is infinite, so that the propagation delay does not affect the throughput.

Consider the transmission of a single k bit long source-packet over a *memoryless* noisy channel. We are provided with a family of channel codes $C = \{c_1, \dots c_J\}$, some of which are embedded (rate compatible). We assume that each channel code is equipped with an error detection mechanism. The source packet is encoded with a channel code and transmitted over the noisy channel. The decoder attempts a decoding and checks the success of its decoding by the error detection mechanism. On success, it transmits a ACK on the feedback channel. Else it transmits a NACK. We assume that the feedback channel is error and loss free and hence, all ACKs and NACKs are received correctly.

The Controlled Markov Chain framework is clear when we realize that, on receiving a NACK, the encoder can take one of the following *actions*: (i) transmitting additional parity check bits, (ii) transmitting copies of some of the previously transmitted bits for the packet, (iii) transmitting the packet encoded with a different channel code, according to a *policy* until an ACK is received. By allowing the decoder to combine previous transmissions for the packet, the above scheme can emulate code-combining, diversity-combining and Type-I and Type-II HARQ systems [67]. If the decoder has the ability to combine output from at most b previous transmissions, then the indices of the last b channel codes and previous b feedbacks form the *state* of the encoder. At each decision instant, *i.e.* after receiving a feedback, the encoder and the decoder share the same knowledge of the state.[1]

Let the collection of states be $\mathcal{S}^* = \mathcal{S} \cup \{s_0, s_T\}$, where s_0 and s_T denote the starting state and the terminating state respectively. When in state $s \in \mathcal{S} \cup \{s_0\}$, the encoder takes an action u from possible set of actions $U(s)$, puts $g(s,u)$ bits on the channel and receives one bit feedback. With probability $P_s^{s_T}(u)$ it receives an ACK and the terminates in state s_T. With probability $P_s^{\tau(u,s)}(u) = 1 - P_s^{s_T}(u)$, it receives a NACK and makes a transition to a unique state $\tau(u,s) \in \mathcal{S}$. Let $h(s,u)$ be the probability that action u results in a ACK with an *undetected error*. Under this framework, we see that the transmission of a packet is a controlled Markov chain, which starts in state s_0, and with probability 1 terminates in the absorbing state s_T. Let us call $g(s,u), h(s,u)$ and $f(s,u)$, the *transmission cost, reliability cost* and *feedback cost* respectively.

[1]Note that this notion of the "state" is limited and is applicable only for tracking of the protocol at the transmitter and the receiver. Firstly, this notion of the state indicates that the action taken by the encoder, which governs the evolution of the protocol, depends only on the information provided by the knowledge of this state. Note that the encoder has access to the actual information bits but it is allowed to use them only for transmission and not for controlling the protocol. Secondly, this notion of the state is also not used for error correction purposes at the decoder as it does not form or contribute to the sufficient statistics of the information bits encoded in the packet. In principle the sufficient statistics for error correction purposes are the posterior probabilities of the information bits given the received channel symbols. Thirdly, to keep the state space finite and small, later we shall resort to some approximations. In that case, even for a memoryless channel, the states may not be Markovian, that is, they may not decorrelate the past and the future evolution of the protocol perfectly. But in the chapter we assume that the states are defined so that they are Markovian.

3.4 Performance Computation for a HARQ Protocol

Under the CMC framework, an HARQ protocol can be completely described by specifying the action to be taken in each state.We define a protocol or a *policy* π to be a map from $\mathcal{S} \cup \{s_0\}$ to $\cup_{s \in \mathcal{S} \cup \{s_0\}} U(s)$, defined such that $\pi(s) \in U(s), \forall s$. A policy tells the next set of bits to be transmitted for the packet given the current state. Figure 3.1 shows the state-action diagram for a variation of Type-II Hybrid ARQ protocol [68].

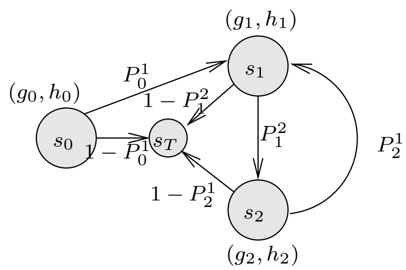

Figure 3.1: State-action diagram for Type-II HARQ with direct combination

The throughput, the reliability and the average number of feedback bits can be calculated as follows.

For $s \in \mathcal{S} \cup \{s_0\}$, let $V^\pi(s)$, $H^\pi(s)$ and $F^\pi(s)$ denote, respectively the expected transmission cost, expected reliability cost and expected feedback cost for a source-packet when the system starts in state s and terminates into state s_T while following a policy π. Then the throughput of policy π is given by $\eta(\pi) = \frac{k}{V^\pi(s_0)}$. The probability of undetected packet error is given by $H^\pi(s_0)$ and the average number of decoding attempts is given by $F^\pi(s_0)$.

$V^\pi(s_0), H^\pi(s_0)$ and $F^\pi(s_0)$ are computed either from the transfer function obtained by applying Mason's Gain Formula [38] to the signal flow graph or by direct computa-

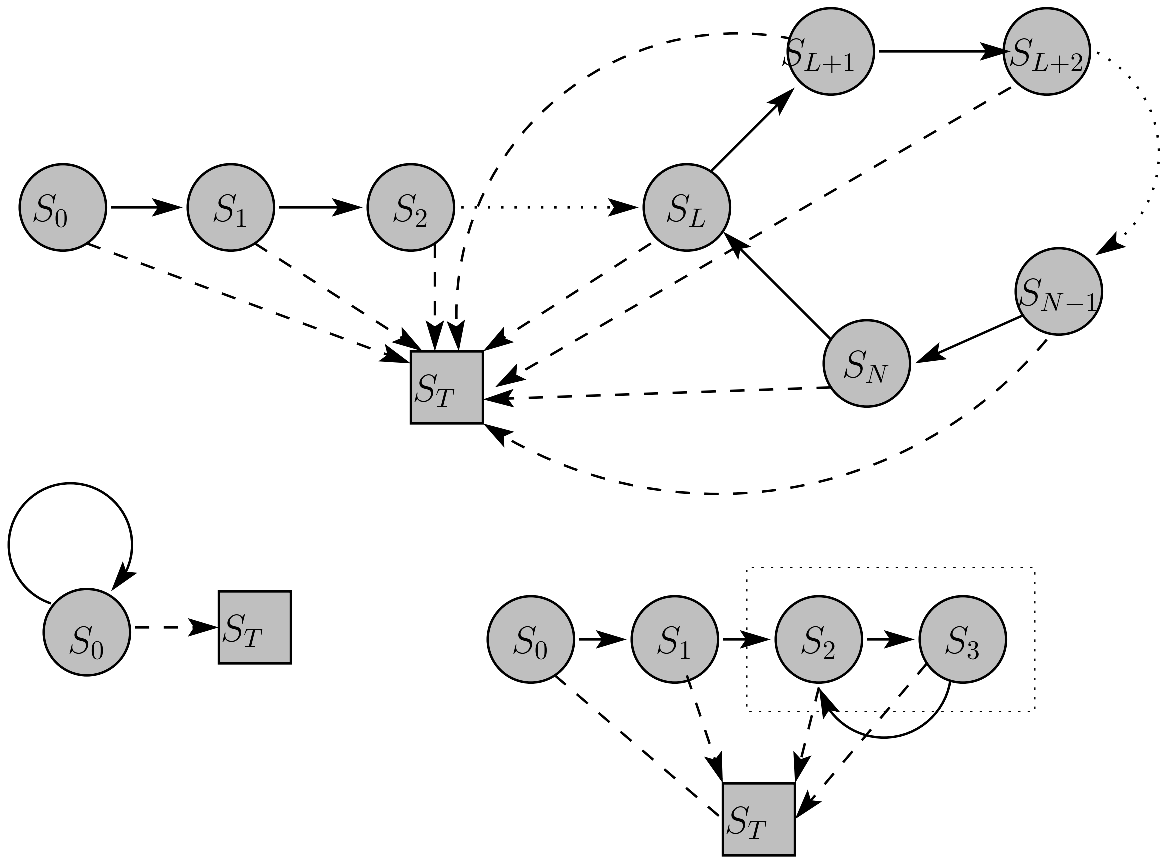

Figure 3.2: State-action diagram for general HARQ with error free feedback and no timeouts

tion from the following equations. For all $s \in \mathcal{S} \cup \{s_0\}$, if $u = \pi(s)$,

$$V^\pi(s) = g(s,u) + P_s^{\tau(s,u)}(u) V^\pi(\tau(u,s)) \tag{3.1}$$

$$H^\pi(s) = h(s,u) + P_s^{\tau(s,u)}(u) H^\pi(\tau(u,s)) \tag{3.2}$$

$$F^\pi(s) = f(s,u) + P_s^{\tau(s,u)}(u) F^\pi(\tau(u,s)) \tag{3.3}$$

These are linear equations with almost decoupled structure and can be solved straightforwardly.

In the next section we see that the CMC structure can be used to obtain optimal

HARQ policy from a collection of policies.

3.5 Constrained Feedback HARQ Design

An intuitive justification for the superior performance of HARQ over pure FEC, even for memoryless channels, was provided in Chapter 2. Suppose the application can tolerate an undetected error probability of 10^-6 per packet. In Pure FEC, only one decoding attempt is allowed per packet. Hence the error correction must be strong enough to correct the channel induced errors, to the desired reliability, with probability 1 for the first transmission of the packet. On the other hand, in HARQ protocol, which uses incremental transmission of parity check bits, the first transmission need not be strong enough to correct all errors, so long as the uncorrected errors are *detected* with high reliability. Hence the number of parity check bits in the first transmission can be less (sometimes significantly so) than the case for pure FEC. If the first transmission is able to correct all errors with, say, probability 0.5, (and detect the uncorrected errors with probability approaching 1), the extra parity check bits do not need to be transmitted with probability 0.5, and hence higher throughput is achieved. Incremental transmission of redundancy, hence, is designed to "build up" the error correction code till it is strong enough to correct all errors up to the desired reliability. This argument clearly indicates that, in principle, the highest throughput will be achieved if the transmission of redundancy is done in small increments, such as one channel symbol per transmission.

But the argument presented above fails to consider the following drawbacks of redundancy transmission in fine increments.

- **Complexity:** Note that each incremental transmission requires one ACK/NACK feedback from the decoder, and each feedback generation requires a decoding operation. Hence, if a scheme transmits a packet in 5 increments, the decoding complexity is increased 5 times over pure FEC transmission.

- **Delay:** Each independent incremental transmission may suffer a separate transmission delay and queuing delay. If a scheme transmits a packet in 5 increments, then, irrespective of the buffering scheme used, the delay before an ACK is generated can be nearly 5 times that of a pure FEC transmission.

- **Overhead:** Each incremental transmission may go over the channel as a separate logical entity (such as an IP packet) and hence may need to be provided with separate header and sequence number. This overhead will diminish the promised throughput.

Nevertheless, pure FEC is only at one end of the spectrum of complexity vs. throughput tradeoff and if it is possible, the available feedback channel must be exploited for better performance. The proposed methodology provides a way of achieving this tradeoff in an operationally optimal fashion. As the usage of feedback channel, *i.e.* the number of ACK/NACKs per packet is directly related to complexity, delay and overheads, we would like to design an HARQ protocol for a given channel such that the throughput is maximized but the use of feedback channel is constrained. The feedback channel usage can limited directly by constraining either (i) the maximum number of feedbacks allowed per packet (similar to Chapter 2) or (ii) the average number of feedbacks allowed per packet. For this chapter we consider the latter technique. The proposed methodology also allows a direct control of the reliability of the protocol, provided appropriate probability computations can be done. We take a Lagrangian approach where we express the constraints by minimizing a weighted sum of reciprocal of throughput, feedback channel usage and probability of undetected error.

Hence the design of an HARQ protocol involves, finding from the set of all allowed HARQ policies, a policy which yields maximum throughput without violating the constraints on maximum tolerable probability of undetected error and on the average number of feedback bits. That is, solving following design problem.

CHARQ Protocol Design Problem:

$$\min_{\pi} V^{\pi}(s_0) \text{ subject to } F^{\pi}(s_0) \leq F_0 \text{ and } H^{\pi}(s_0) \leq H_0 \tag{3.4}$$

Equivalently, the optimal policy must be an unconstrained minimizer of a Lagrangian cost,

$$\min_{\pi} V^{\pi}(s_0) + \lambda_{pu} H^{\pi}(s_0) + \lambda_{fb} F^{\pi}(s_0) \tag{3.5}$$

for some Lagrange multipliers $\lambda_{pu} \geq 0, \lambda_{fb} \geq 0$.

Equation (3.5) is a problem of minimization of total expected cost before termination for a controlled Markov chain. The search for optimal policy is accomplished by dynamic programming [4].

For the given $\lambda_{pu}, \lambda_{fb}$, the optimal policy π^* satisfies the following Bellman equations of optimality. For each $s \in \{s_0\} \cup \mathcal{S}$,

$$\begin{aligned} & V^{\pi^*}(s) + \lambda_{fb} F^{\pi^*}(s) + \lambda_{pu} H^{\pi^*}(s) \\ = \quad & \min_{u \in U(s)} (g(s,u) + \lambda_{fb} f(s,u) + \lambda_{pu} h(s,u) + \\ & P_s^{\tau(u,s)}(u)(V^{\pi^*}(\tau(u,s)) + \lambda_{fb} F^{\pi^*}(\tau(u,s)) + \\ & \lambda_{pu} H^{\pi^*}(\tau(u,s)))). \end{aligned} \tag{3.6}$$

The set of equations (3.6) is solved by the algorithms of value iteration or policy iteration [4]. Let numerical superscripts denote the iteration index. Then the algorithm is described as follows.

Value Iteration Algorithm:

1. Set $k = 0$. Set, arbitrarily, $V^k(s) = 0, F^k(s) = 1$ and $H^k(s) = 1$.

2. For all $s \in \{s_0\} \cup \mathcal{S}$, set *values*, $J^k(s) = V^k(s) + \lambda_{pu} H^k(s) + \lambda_{fb} F^k(s)$.

3. For all $s \in \{s_0\} \cup \mathcal{S}$, set

   $$J^{k+1}(s) = \min_{u \in U(s)} \big((g(s,u) + \lambda_{fb} f(s,u) + \lambda_{pu} h(s,u)) + P_s^{\tau(u,s)}(u)(J^k(\tau(u,s)))\big)$$

4. Set $\pi^{k+1}(s)$ to the minimizer action (u) in the above minimization.

5. For some small number ϵ, if $\max_{s \in \{s_0\} \cup \mathcal{S}} |J^{k+1}(s) - J^k(s)| < \epsilon$, stop, and select $\pi^* = \pi^{k+1}$. Else, increment k and go to step 3.

Policy Iteration Algorithm:

1. Set $k = 0$. Initialize $\pi^k(s) = u$ for some arbitrary $u \in U(s)$.

2. Obtain *steady state values for* π^k, $J^k(s)$ for $s \in \{s_0\} \cup \mathcal{S}$ by solving of linear equations given by,

   $$\begin{aligned} J^k(s) &= g(s, \pi^k(s)) + \lambda_{fb} f(s, \pi^k(s)) + \lambda_{pu} h(s, \pi^k(s)) \\ &\quad + P_s^{\tau(s,\pi^k(s))}(\pi^k(s)) J^k(\tau(\pi^k(s), s)). \end{aligned}$$

3. Set

   $$\pi^{k+1}(s) = arg \min_{u \in U(s)} \big(g(s,u) + \lambda_{fb} f(s,u) + \lambda_{pu} h(s,u) + P_s^{\tau(u,s)}(u) J^k(\tau(u,s))\big)$$

4. If, for all $s \in \{s_0\} \cup \mathcal{S}$, $\pi^{k+1}(s) = \pi^k(s)$ then stop and select $\pi^* = \pi^{k+1}$. Else increment k and go to step 2.

The values of individual performance parameters can be obtained by solving equations 3.1 for the selected protocol.

3.5.1 Interpretation of the Lagrangian

Note that the optimization problem could have been set in alternative ways. For example, to maximize the throughput subject to a constraint on the feedback, we could have solved,

$$\min_{\pi} -\frac{k}{V^{\pi}(s_0)} + \lambda F^{\pi}(s_0). \tag{3.7}$$

But the optimization problem in eq. (3.7) does not yield itself to elegant solution by the theory of Controlled Markov Chains unlike the problem in eq. (3.5).

The Lagrangian in equation (3.5) also arises naturally when analyzing the HARQ protocol in the following situation.

Delay and Overhead Analysis in Stop and Wait based HARQ protocol: Consider a Stop and Wait based HARQ protocol executing a policy π. Suppose at every transmission step the transmitter must append a header of length l_h bits to the (partial) channel codeword. This is an overhead that grows with the number of steps needed for transmission. The total number of bits put on the channel before receiving an ACK is given by $V^{\pi}(s_0)+l_hF^{\pi}(s_0)$. Similarly, let T_s denote the baud period, *i.e.* the time taken to put one bit over the channel. Let the transmission delay for each step be T_d and let the decoding delay - the delay for generating a feedback be T_{dec}. Let T_f denote the time taken for the feedback to reach the transmitter. Then, the total delay from the start of transmission of a packet to the time when an ACK is received by the receiver, is computed as,

$$T_{total} = V^{\pi}(s_0)T_s + (l_hT_s + T_{dec} + T_d + T_f)F^{\pi}(s_0). \tag{3.8}$$

Similar expression holds for expected total delay when the delays are not deterministic but are independent random variables with finite means. The total channel usage $V^{\pi}(s_0)+l_hF^{\pi}(s_0)$ as well as the total delay in eq. (3.8) are of the form of the Lagrangian in eq. (3.5). Hence the Lagrangian has a physical meaning in this situation.

3.5.2 Feasibility

The second part of the design procedure is the search for Lagrange Multipliers which will make the solutions meet the constraints. The problem stated in eq. (3.4) may not have a solution at all. Note that, as a pure FEC transmission is a special case of the HARQ protocol, all values of $F_0 \geq 1$ in the problem (eq. 3.4) can be met by some policy. The reliability constraint is harder to meet and some values of H_0 may not have any solution in the set of policies.

If such a solution exists, the determination of the two parameters λ_{fb} and λ_{pu} requires solving a linear program. Also, the dynamic range of numbers for probability of undetected error is much smaller than that for feedback, and hence the sensitivity of the two Lagrange multipliers is widely different.

A faster method can be devised if one is willing to tolerate approximate meeting of the reliability constraint. Note that in practice, reliability constraint, or probability of undetected error, is typically specified in logarithmic scale, or described by "orders of magnitude" such as 10^{-5} and 10^{-6}. Consider a protocol given by policy π. A close upper bound on the probability of undetected error $H^{\pi}(s_0)$ is given by,

$$H^{\pi}(s_0) \leq F^{\pi}(s_0)(\max_{s \in \{s_0\} \cup \mathcal{S}} h(s, \pi(s))).$$

This upper bound allows us to drop the reliability constraint in the Lagrangian by incorporating it directly in the search. If the action set $U(s)$ at each state s is modified to $U'(s) \stackrel{def}{=} \{u : u \in U(s), h(s,u) \leq H_0\}$, then the solution obtained by setting $\lambda_{pu} = 0$ (unconstrained reliability), will satisfy $H^{\pi}(s_0) \leq F^{\pi}(s_0)H_0$, which has the same "order of magnitude" as H_0. Modifying the action set to $U''(s) \stackrel{def}{=} \{u : u \in U(s), h(s,u) \leq$

$\frac{H_0}{F_0}\}$, results in a solution guaranteed to meet the reliability constraint. Determination of a single Lagrange multiplier λ_{fb} can be handled by the relatively quick descent or bisection techniques.

3.6 Results with Reed Solomon Codes

The design procedure of previous section yields the optimal policy or protocol for a given memoryless channel for essentially arbitrary selection of channel codes and error detection mechanisms. The essential part of the design is knowledge of the transition probabilities of the Controlled Markov Chain. These probabilities can be obtained analytically or by simulation.

We illustrate the technique by using a family of channel codes which consist of punctured codes obtained from a mother Reed-Solomon Code. (Rate Compatible) Punctured Reed-Solomon codes have been considered good codes for wireless error control, especially for hybrid ARQ. This is because of several reasons. Firstly, they have an optimality property that they are Maximum Distance Separable (MDS), i.e. they meet the Singleton bound [6] with equality and each additional symbol increases the minimum distance of the code by one [68]. Secondly, the Berlekemp-Massey decoding algorithm can be used when there are symbol errors as well as symbol erasures [67]. This is especially suitable for a fading channel where a deep fade, if detected, results in symbol erasure. Thirdly, the weight distribution of MDS codes is completely determined. It can be used to analytically compute or estimate the transition probabilities.

We consider a family of (n, k) punctured RS-codes over $GF(q)$ for $n_{min} \leq n \leq n_{max}$. These are punctured versions of a (n_{max}, k) parent-code. We use *bounded distance decoding* as the decoding method. With each block length $n, n_{min} \leq n \leq n_{max}$, there is a decoding diameter $d_{dec}(n)$. A received word $\bar{y}^{(n)}$ is accepted if

$$d_H(\bar{y}^{(n)}, c) \leq \lfloor d_{dec}(n)/2 \rfloor \text{for some codeword } c \tag{3.9}$$

where d_H is the hamming distance. (Decoding *radius* is analogously defined as $r_e(n) \stackrel{def}{=} \lfloor d_{dec}(n)/2 \rfloor$). We assume a *symbol-symmetric channel* with symbol error rate p_e, ***i.e.*** For $\alpha, \beta \in GF(q), \beta \neq \alpha$

$P[y = \alpha | x = \alpha] \stackrel{def}{=} 1 - p_e$ and $P[y = \beta, | x = \alpha] = \frac{p_e}{q-1}$.

The Markov Chain is set up as follows. Exploiting the MDS property of RS-codes, we define the states as $\{s_0, s_T\} \cup \{s_n, n_{min} \leq n \leq n_{max}\}$, where the system is in state s_n if blocklength n was used in the decoding for generating the last feedback. An action $u \equiv (n_1, n_3)$ in a state s_n, consists of discarding n_1 symbols and requesting n_3 new symbols. Under these definitions, the probability of retransmission $1 - P_s^{s_T}(u)$ and probability of termination with undetected error $h(s, u)$ can be computed or approximated from the distance properties of the MDS codes. Please note that, this notion of state, as information decorrelating the past and the future, is an approximation, which is exact in the first two transmissions but remains a good approximation for further transmissions. In Section 3.8, we derive the analytical expressions and obtain approximations for the computation of transition probabilities.

Figures 3.3 to 3.10 and show the results obtained for a symbol symmetric channel over $GF(32)$. The RS code family used an $(n, 8)$ code family over $GF(32)$ obtained by puncturing a $(31, 8)$ RS code.

The schemes indexed with the prefixes $T1$ and $T2$ are the conventional Type I and Type II Hybrid ARQ schemes. The schemes indexed by CF and CR are respectively the proposed schemes for different values of the Lagrangian penalties λ_{fb} and λ_{pu}. The constrained feedback schemes indexed by CF have $\lambda_{fb} > 0$ and $\lambda_{pu} = 0$. (Still, the decoding radii of the code family are chosen to maintain a minimum level of reliability of 7, (that is $h(s,u) < 10^{-7} \forall s$ and u, as in the discussion in Section 3.5.2). The constrained reliability schemes CR have $\lambda_{pu} > 0$.

Refer to Figures 3.3 to 3.5, which are 2 dimensional projections of performance triplets (Table 3.1) in the space of Throughput, Reliability and Feedback, for a channel with symbol error probability of 0.1.

It is evident that the proposed approach captures the tradeoff between the three competing requirements, namely high throughput, high reliability and low computation, quite well. Constrained Feedback schemes, such as CF-2, achieve about 20% gain in throughput over the closest conventional Type-II scheme (TF-5), while maintaining reliability over 7 but allowing nearly 0.5 NACKS on an average. If the NACKS are allowed to increase, a scheme such as CR-3 achieves this throughput gain while retaining reliability better than 8, albeit at the expense of increased feedback. .

Figures 3.6 to 3.8 and table 3.2 show similar trend and trade offs for $p_e = 0.05$. Similar, though, not as prominent tradeoffs are observed in channels with lower symbol error probabilities. Figures 3.9 and 3.10 are condensed versions of similar results for channels with symbol error probability of 0.01,0.001 and 0.0001.

3.7 Conclusion

We propose a dynamic programming based technique for design of Hybrid ARQ system for error control in wireless channels [9]. It is more flexible than the conventional signal flow graph based techniques in the sense that it allows tighter control over throughput/feedback and throughput/reliability tradeoff. The results indicate that, if the system can support a little extra complexity and more feedback then significant improvements in throughput can be obtained by careful design.

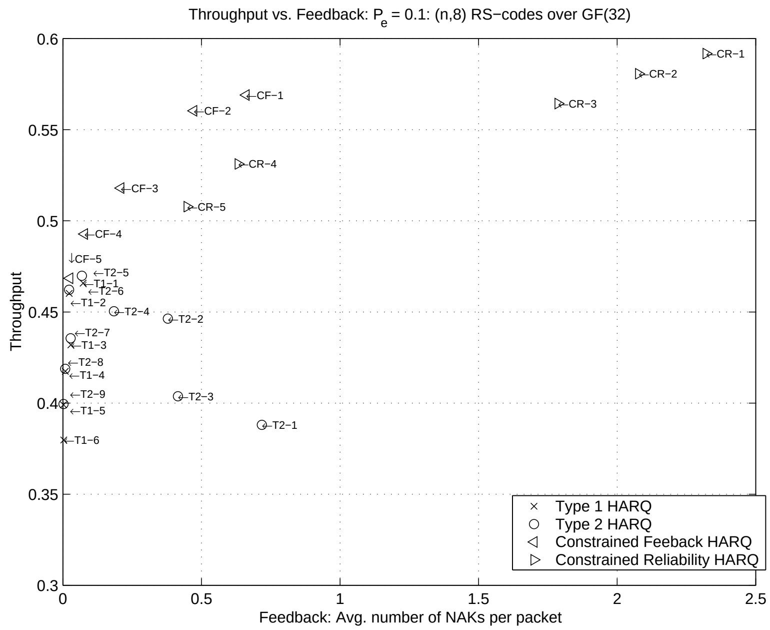

Figure 3.3: Throughput Vs. Feedback Performance of Various Schemes:$P_e = 0.1$

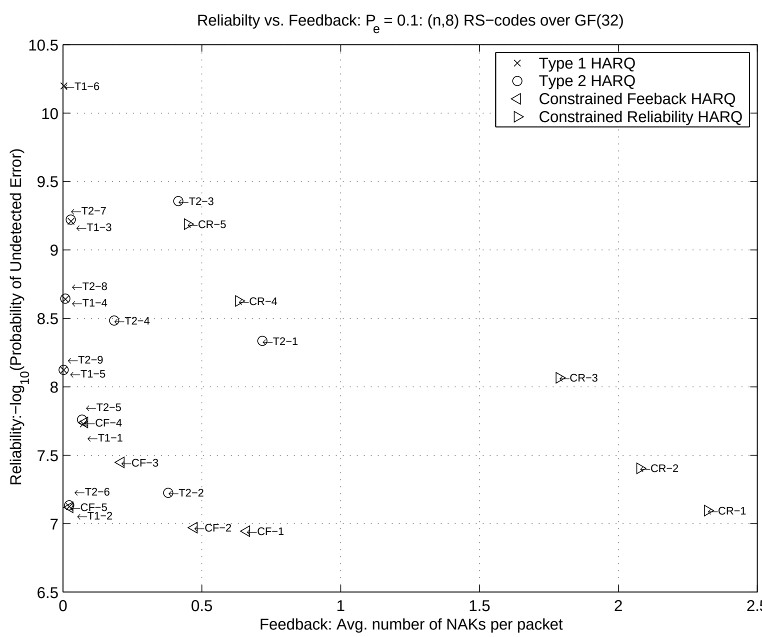

Figure 3.4: Reliability Vs. Feedback Performance of Various Schemes:$P_e = 0.1$

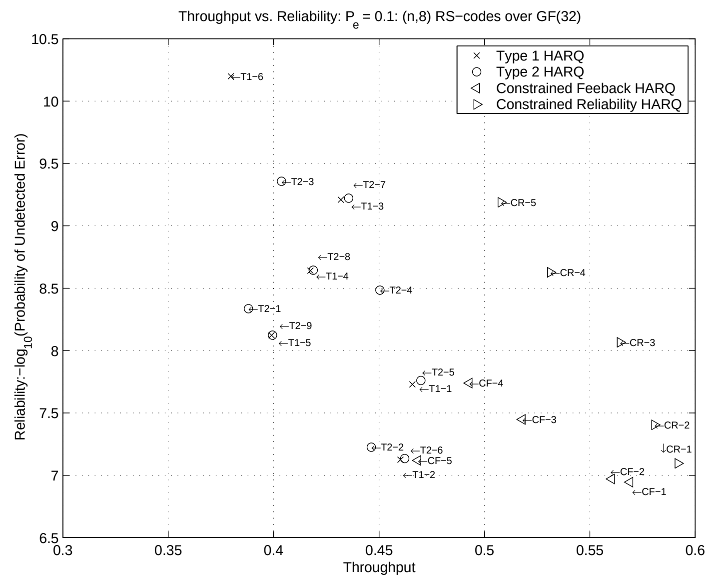

Figure 3.5: Reliability Vs. Throughput Performance of Various Schemes: $P_e = 0.1$

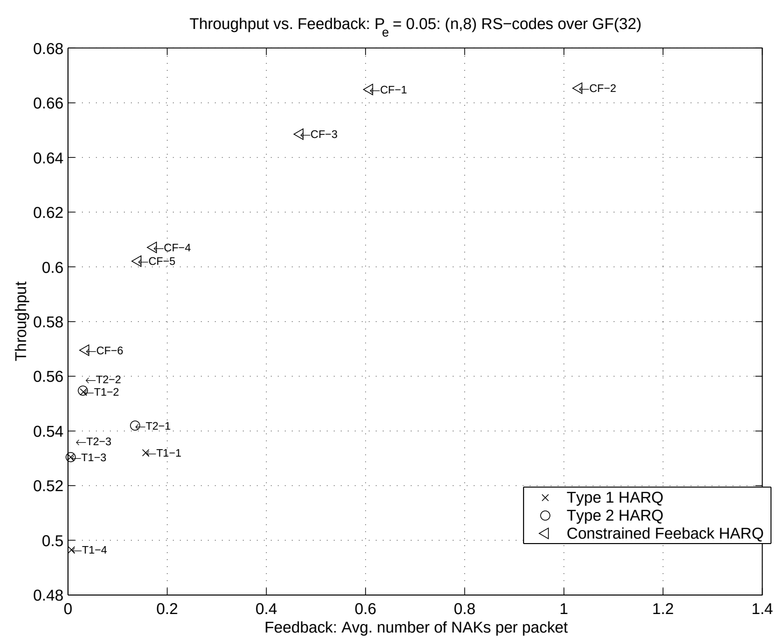

Figure 3.6: Throughput Vs. Feedback Performance of Various Schemes: $P_e = 0.05$

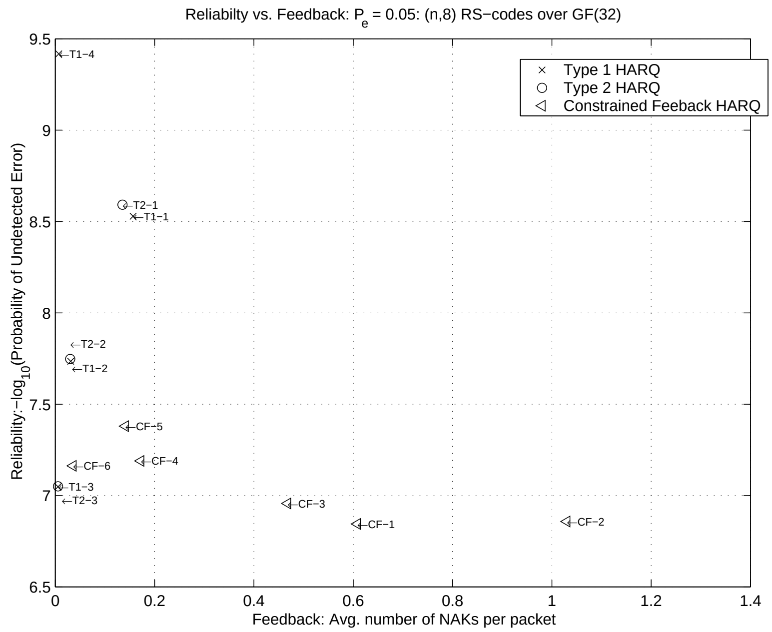

Figure 3.7: Reliability Vs. Feedback Performance of Various Schemes: $P_e = 0.05$

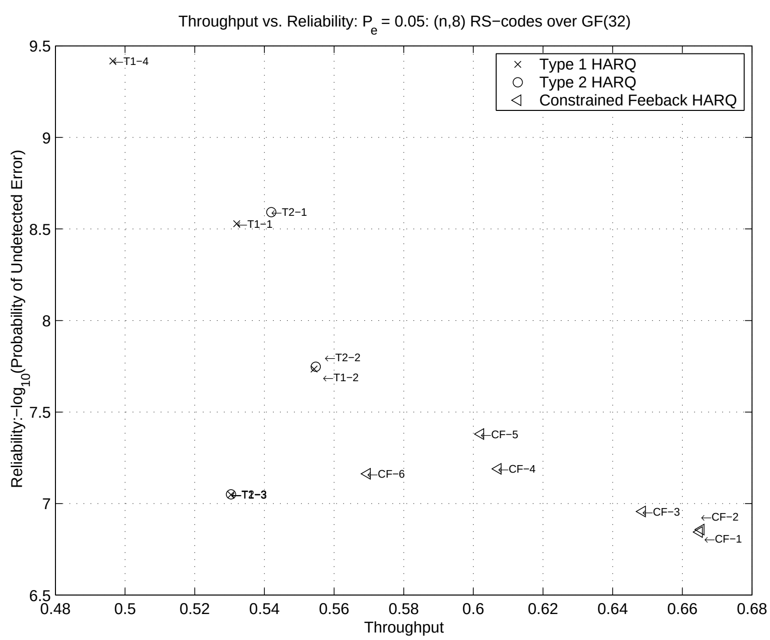

Figure 3.8: Reliability Vs. Throughput Performance of Various Schemes: $P_e = 0.05$

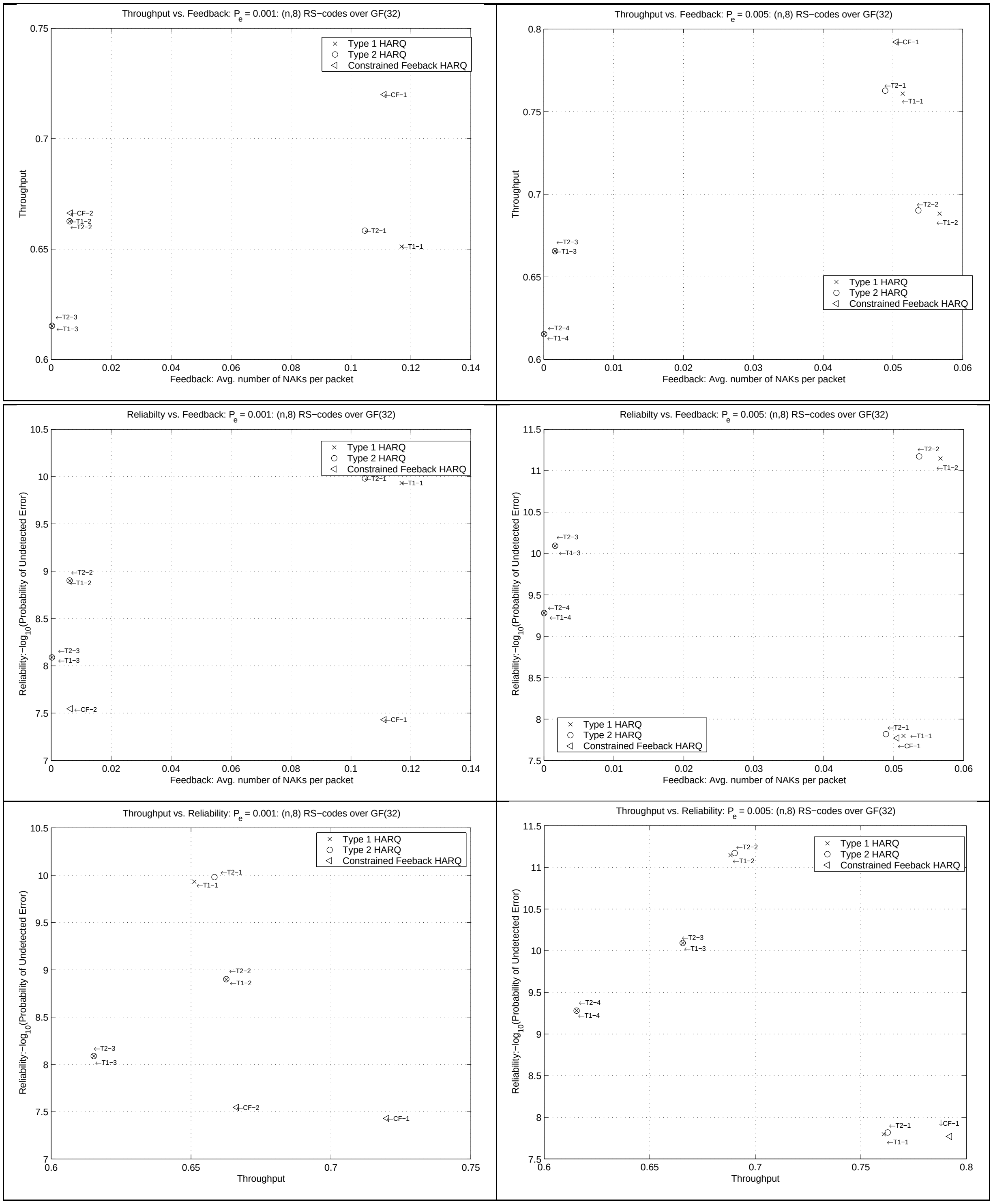

Figure 3.9: Performance of various schemes for channels with $P_e = 0.01$ and $P_e = 0.0005$.

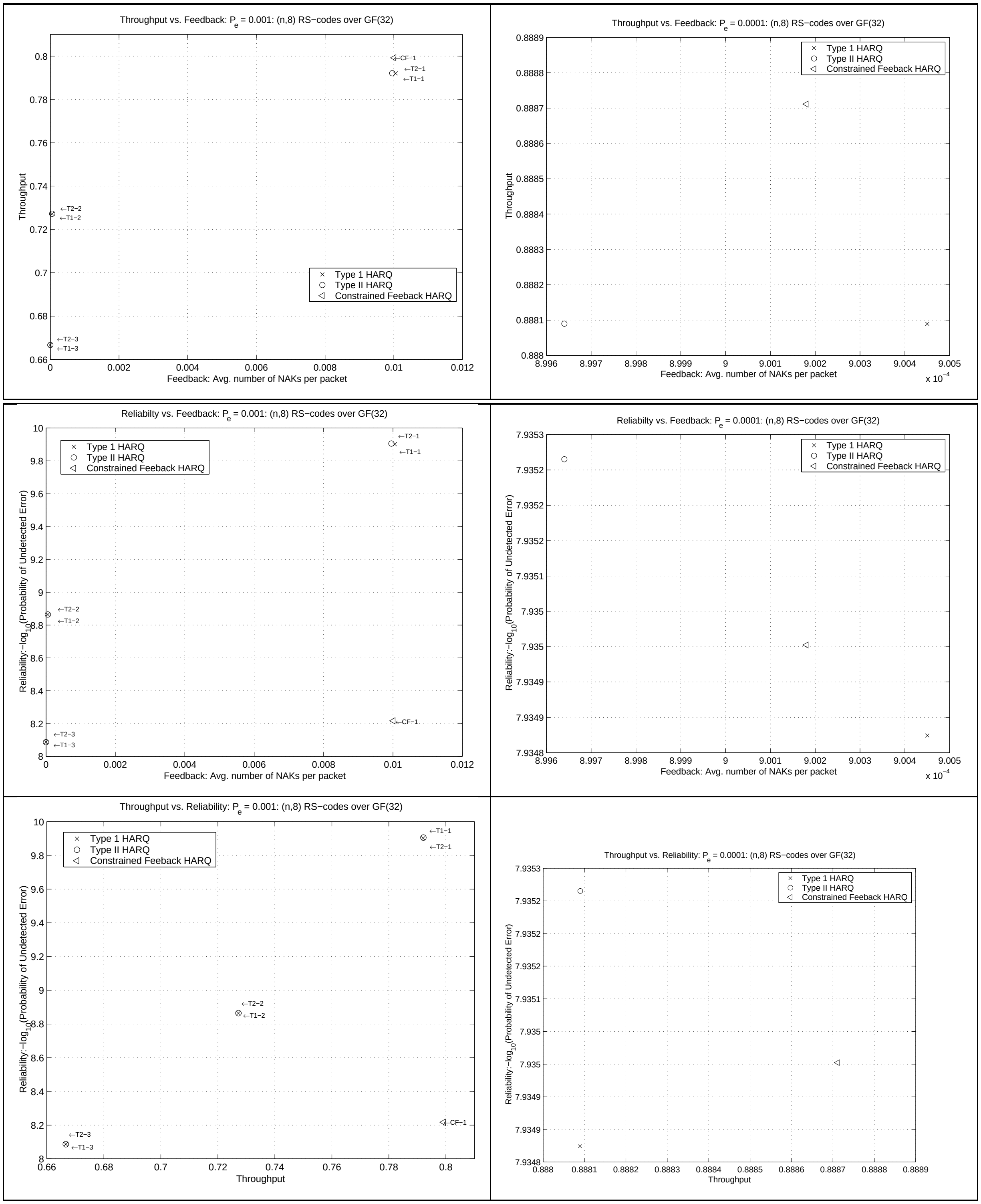

Figure 3.10: Performance of various schemes for channels with $P_e = 0.001$ and $P_e = 0.0001$.

3.8 Appendix: Transition Probabilities for CHARQ with Reed Solomon Codes

In this section we derive the formulas for computation/approximation of the transition probabilities of the controlled Markov chain. We need the following results.

Weight Distribution of Reed Solomon Codes: Reed Solomon codes are Maximum Distance Separable (MDS), and their weight distribution is completely determined[6, 45]. For a (n, k) MDS code over $GF(q)$, let $A_j(n, k, q)$ denote the number of codewords of weight j. Then $A_0 = 1$, $A_j = 0$ for $j = 1, \dots d^* - 1$, where $d^* = n - k + 1$, and for $j \geq d^*$,

$$A_j(n, k, q) = \binom{n}{j}(q-1)\sum_{i=0}^{j-d^*}(-1)^i\binom{j-1}{i}q^{j-d^*-i} \tag{3.10}$$

A very interesting property of the MDS codes is their symmetry with respect to distribution of zero symbols in a codeword. The number of codewords of weight j with zeros in fixed $n-j$ locations, does not depend of the location of zeros - or the "zero-distribution-pattern". Therefore, denote by $M_j(n, k, q) = \frac{1}{\binom{n}{n-j}}A_j(n, k, q)$ the number of codewords of weight j with a fixed zero-distribution pattern.

This property helps us calculate 2-step and 3 step Weight Distribution functions, useful for the calculation of transition probabilities.

Proposition 1 *Let $A^2_{j_1 j_2}(n_1, n_2, q, k)$ denote 2-step Weight Distribution function, that is, the number of codewords of a $(n_1 + n_2, k)$ RS code over $GF(q)$, which have weight j_1 in first n_1 coordinates, and j_2 in next n_2 coordinates. Let $A^3_{j_1 j_2 j_3}(n_1, n_2, n_3, q, k)$ denote the number of codewords of a $(n_1 + n_2 + n_3, k)$ RS code over $GF(q)$, which have weight j_1 in first n_1 coordinates, j_2 in next n_2 coordinates and j_3 in next n_3 coordinates. Then,* *by the symmetry of zero distributions the following holds.*

$$A^2_{j_1 j_2}(n_1, n_2, q, k) = \frac{\binom{n_1}{j_1}\binom{n_2}{j_2}}{\binom{n_1+n_2}{j_1+j_2}} A_{j_1+j_2}(n_1+n_2, q, k). \tag{3.11}$$

$$A^3_{j_1 j_2 j_3}(n_1, n_2, n_3, q, k) = \frac{\binom{n_1}{j_1}\binom{n_2}{j_2}\binom{n_3}{j_3}}{\binom{n_1+n_2+n_3}{j_1+j_2+j_3}} A_{j_1+j_2+j_3}(n_1+n_2+n_3, q, k). \tag{3.12}$$

Puncturing: An (n, k) block code can be punctured to obtain a (n_1, k) code by dropping $n - n_1$coordinates of the codewords. It can be easily shown that punctured versions of MDS codes are also MDS. Consequently, the distance properties of the punctured codes are independent of the *puncturing table,* that is, the coordinates dropped. The weight distribution of the punctured code is again given by the expression above.

Counting error patterns within decoding spheres: [6] Consider an (n, k) code over $GF(q)$. Let $T(n, j, w, s)$ denote number of error patterns of weight w at a Hamming distance s from a fixed codeword of weight j. Then

$$T(n, j, w, s) = \sum_{0 \le a \le n \; 0 \le b \le n; a+2b+w=s+j} \binom{n-j}{b+w-j}\binom{j}{a}\binom{j-a}{b}(q-1)^{b+w-j}(q-2)^a \tag{3.13}$$

Figure 3.11 shows how equation (3.13) can be derived.

Let $0 \le s \le \lfloor \frac{d^*-1}{2} \rfloor$. Let $\zeta(n, j, w, s)$ denote the set of error patterns of weight w, which are at a Hamming distance s from at least one codeword of weight j. Then $|\zeta(n, j, w, s)|$, the size of the set, is given by $A_j(n, k, q)T(n, j, w, s)$.

Symmetry in error patterns: As the code is symmetric with respect to zero-distribution patterns, so is the set $\zeta(n, j, w, s)$. Hence the number of error patterns of weight w, which have exactly z_1 non-zero symbols in first n_1 coordinates, is given by

$$\frac{\binom{n_1}{z_1}\binom{n-n_1}{w-z_1}}{\binom{n}{w}} A_j(n, k, q)T(n, j, w, s) \tag{3.14}$$

These equations are valid under the convention that $\binom{m}{z} = 0$ whenever $m < 0$ or $z < 0$ or $z > m$.

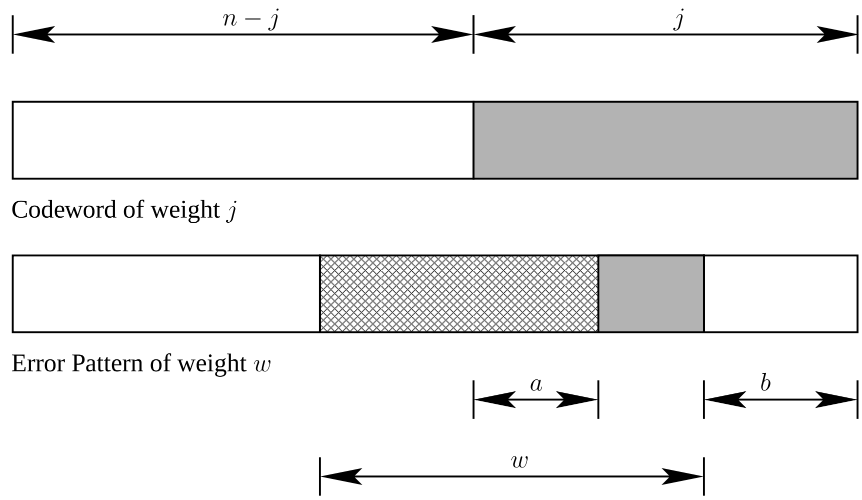

Figure 3.11: Error Pattern of weight w, and codeword of weight j. Non-zero coordinates in the error pattern disagree at a places. Zero coordinates disagree at b places.

Symbol Symmetric Memoryless Channel: Consider transmission of a codeword over a symbol symmetric channel with symbol error probability p_e. The probability that w out of n symbols are received in error is given by $\binom{n}{w} p_e^w (1-p_e)^{n-w}$.

Bounded Distance Decoding: For every codeword length $n,$ we associate a decoding radius $r_e(n) \leq \lfloor \frac{d^*-1}{2} \rfloor$. A codeword c is decoded if the Hamming distance between the received word and the codeword is less than $r_e(n)$. If no such codeword is found, a decoding failure is declared, which will be used to generate NACK feedback in the HARQ protocol.

States of the Controlled Markov Chain: As described earlier, we define the states as $\{s_0, s_T\} \cup \{s_n, n_{min} \leq n \leq n_{max}\}$, where the system is in state s_n if blocklength n was used in the decoding for generating the last feedback.

Deriving probabilities of the controlled Markov Chain: At any decision instant, which has resulted in a NACK, the receiver must take an action. The action consists of receiving n_3 additional symbols, discarding n_1 past received symbols, while retaining past n_2 symbols. We shall calculate the first step transition probabilities as follows. Without loss of generality, we assume that the all zero codeword of length $n_1+n_2+n_3$ is transmitted over a memoryless symbol symmetric channel with symbol error probability p_e. Let $\underline{e}_1, \underline{e}_2$ and $\underline{e}_3$ denote the (random) received error patterns of lengths $n_{1,}n_2$ and n_3 respectively. Let $W(\underline{e})$ denote the weight of an error pattern $\underline{e}$. Let D_{12} denote the event that n_1+n_2 symbols are transmitted over the memoryless channel and the transmit codeword is decoded correctly $W(\underline{e}_1 + \underline{e}_2) \leq r_e(n_1 + n_2)$. Here $+$ for error patterns denotes concatenation. Let U_{12} denote the event that the received codeword is decoded incorrectly. Let $Z^{12}(c, r_e^{12}$ denote a decoding sphere of dimensions n_1+n_2 of decoding radius r_e^{12} around a codeword c. Then U_{12} is the event $\underline{e}_1 + \underline{e}_2 \in \cup_{c\neq 0} Z^{12}(c, r_e^{12})$. Analogously, define D_{23} and U_{23} to be the events $\underline{e}_2 + \underline{e}_3 \in Z^{23}(0, r_e^{23})$ and $\underline{e}_2 + \underline{e}_3 \in \cup_{c\neq 0} Z^{23}(c, r_e^{23})$.

Approximating Transition Probabilities: Consider a HARQ protocol that starts in state s_0 and requests a feedback after transmitting $n_1 + n_2$ symbols. The feedback will be an ACK if event $D_{12} \cup U_{12}$ happens. Otherwise the feedback will be a NACK and the system will move to a new state s_1. All actions taken in state u have transition probabilities conditioned on the event $(D_{12} \cup U_{12})'$. A typical action u in state s_1 can be represented by integers $u \equiv n_1, n_3,$ which corresponds to requesting n_3 new symbols, discarding n_1 of the old symbols, and making a decoding attempt by combining the retained n_2 symbols with the new n_3 symbols. Note that, because of the MDS property of the RS codes, and the fact that the previous state was $s_0,$ the specific locations of the discarded n_1 symbols does not matter for the calculation of state transition probabilities. We are interested in the probabilities $P[D_{23}|(D_{12}\cup U_{12})']$ and $P[U_{23}|(D_{12}\cup U_{12})']$, which are the probabilities of correct decoding and undetected error respectively, when action u is taken in state s_1. Probability that a NACK is generated on action u is $P[(D_{23} \cup$

$U_{23})'|(D_{12} \cup U_{12})'] = 1 - P[D_{23}|(D_{12} \cup U_{12})'] + P[U_{23}|(D_{12} \cup U_{12})']$.

Consider $P[A|(D_{12} \cup U_{12})']$ for some event A. Note that D_{12} and U_{12} are mutually exclusive events. Therefore $P[(D_{12} \cup U_{12})'] = P[D'_{12}] - P[U_{12}]$ and we have

$$P[A|(D_{12} \cup U_{12})'] = \frac{P[A \cap D'_{12} \cap U'_{12}]}{P[D'_{12} \cap U'_{12}]} = \frac{P[A \cap D'_{12} \cap U'_{12}]}{P[D'_{12}] - P[U_{12}]} = \frac{P[A \cap D'_{12}] - P[A \cap U_{12}]}{P[D'_{12}] - P[U_{12}]}$$

Now if $P[D'_{12}] \gg P[U_{12}]$, *i.e.* $\frac{P[U_{12}]}{P[D'_{12}]} \ll 1$, we can neglect $P[U_{12}]$ in the denominator and obtain an approximation to $P[A|(D_{12} \cup U_{12})']$ as follows.

$$P[A|(D_{12} \cup U_{12})'] \approx P[A|D'_{12}] - \frac{P[A \cap U_{12}]}{P[D'_{12}]}. \tag{3.15}$$

The right hand side is a close lower bound on $P[A|(D_{12} \cup U_{12})']$ as the ratio of the error to the true value, $\frac{P[A|(D_{12} \cup U_{12})'] - P[A|D'_{12}] + \frac{P[A \cap U_{12}]}{P[D'_{12}]}}{P[A|(D_{12} \cup U_{12})']} = \frac{P[U_{12}]}{P[D'_{12}]}$ is small by assumption. The assumption is justified as Reed Solomon Codes are not “well packed” *i.e.* the number of codewords of given dimensions are much smaller than that promised by the Sphere Packing Bound[6]. The lower-bound in eq. 3.15 can be effectively used as an approximation to $P[A|(D_{12} \cup U_{12})']$ no matter how small the probability of the event A is. A coarser bound $P[A|(D_{12} \cup U_{12})'] \approx P[A|D'_{12}]$ can be obtained by neglecting $\frac{P[A \cap U_{12}]}{P[D'_{12}]} \leq \frac{P[U_{12}]}{P[D'_{12}]}$ in the numerator too. In that case, the absolute value of the error $|P[A|(D_{12} \cup U_{12})'] - P[A|D'_{12}]| \leq 2|\frac{P[U_{12}]}{P[D'_{12}]}|$ is small in absolute terms. But this bound does not guarantee that, relative to $P[A|(D_{12} \cup U_{12})']$ the error will be small.

Probability of Correct Decoding: Consider the computation of $P[D_{23}|D'_{12}]$ where the event D_{23}, as described earlier, denotes the probability of correct decoding when the last n_2 symbols from the $n_1 + n_2$ symbols received are combined with n_3 new requested symbols to form a codeword. Then $P[D_{23}|D'_{12}]$ is *exactly* given by the following expression.

$$\begin{aligned} & P[D_{23}|D'_{12}] \\ = \quad & \frac{P[D_{23} \cap D'_{12}]}{P[D'_{12}]} = \frac{\sum_{z_2=0}^{n_2} P[W(\underline{\mathrm{e}}_2) = z_2, W(\underline{\mathrm{e}}_1) + z_2 > r_{e_{12}}, W(\underline{\mathrm{e}}_3) + z_2 \leq r_{e_{23}}]}{\sum_{z_2=0}^{n_2} P[W(\underline{\mathrm{e}}_2) = z_2, W(\underline{\mathrm{e}}_1) + z_2 > r_{e_{12}}]} \end{aligned}$$

Where

$$\begin{aligned} & P[W(\underline{\mathrm{e}}_2) = z_2, W(\underline{\mathrm{e}}_1) + z_2 > r_{e_{12}}, W(\underline{\mathrm{e}}_3) + z_2 \leq r_{e_{23}}] \\ = \quad & \sum_{\substack{\{z_1, z_3 : 0 \leq z_1 \leq n_1, \\ 0 \leq z_3 \leq n_3, \\ z_1 + z_2 > r_{e_{12}}, \\ z_3 + z_2 \leq r_{e_{23}}\}}} \binom{n_2}{z_2}\binom{n_1}{z_1}\binom{n_3}{z_3}\left(\frac{p_e}{1-p_e}\right)^{z_1+z_2+z_3} (1-p_e)^{n_1+n_2+n_3}, \end{aligned} \tag{3.16}$$

and,

$$\begin{aligned} & P[W(\underline{\mathrm{e}}_2) = z_2, W(\underline{\mathrm{e}}_1) + z_2 > r_{e_{12}}] \\ = \quad & \sum_{\substack{\{z_1 : 0 \leq z_1 \leq n_1, \\ z_1 + z_2 > r_{e_{12}}\}}} \binom{n_2}{z_2}\binom{n_1}{z_1} p_e^{z_1+z_2}(1-p_e)^{n_1+n_2-z_1-z_2}. \end{aligned} \tag{3.17}$$

Probability of Undetected Error: To compute $P[U_{23}|D'_{12}]$ consider the following.

$$\begin{aligned} P[U_{23}|D'_{12}] = \quad & \tfrac{P[U_{23} \cap D'_{12}]}{P[D'_{12}]} = \left(\sum_{z_2=0}^{n_2} P[U_{23} \cap D'_{12} | W(\underline{\mathrm{e}}_2) = z_2] P[W(\underline{\mathrm{e}}_2) = z_2]\right) (P[D'_{12}])^{-1} \\ = \quad & \left(\sum_{z_2=0}^{n_2} P[U_{23} | W(\underline{\mathrm{e}}_2) = z_2] P[D'_{12} | W(\underline{\mathrm{e}}_2) = z_2] P[W(\underline{\mathrm{e}}_2) = z_2]\right) (P[D'_{12}])^{-1} \end{aligned}$$

Now

$$P[D'_{12}|W(\underline{\mathrm{e}}_2) = z_2] = P[W(\underline{\mathrm{e}}_1) + z_2 > r_{e_{12}}] = \sum_{z_1=0}^{n_1} \binom{n_1}{z_1} \chi_{\{z_1+z_2>r_{e_{12}}\}} p_e^{z_1}(1-p_e)^{n_1-z_1}.$$

$P[U_{23}|W(\underline{\mathrm{e}}_2) = z_2]$ an be computed as follows. For compactness, let $n_{23} \stackrel{def}{=} n_2 + n_3$.

Consider

$$\begin{aligned}
& P[U_{23} \cap W(\underline{e}_2) = z_2] \\
= \ & P[\{\underline{e}_1 + \underline{e}_2 \in \cup_{c \in C_{RS}^{23}, c \neq 0} Z^{23}(c, r_{e_{23})}\} \cap \{W(\underline{e}_2) = z_2\}] \\
= \ & \sum_{z_3=0}^{n_3} \sum_{s=0}^{r_{e_{23}}} \frac{\binom{n_2}{z_2}\binom{n_3}{z_3}}{\binom{n_{23}}{z_2+z_3}} \sum_{j=n_{23}-k+1}^{n_{23}} A_j(n_{23}, q, k) T(n_{23}, j, z_2+z_3, s) \left(\frac{p_e^{z_2+z_3}(1-p_e)^{n_{23}-z_2-z_3}}{(q-1)^{z_2+z_3}} \right).
\end{aligned}$$

Therefore

$$\begin{aligned}
& P[U_{23} | W(\underline{e}_2) = z_2] \\
= \ & P[\underline{e}_1 + \underline{e}_2 \in \cup_{c \in C_{RS}^{23}, c \neq 0} Z^{23}(c, r_{e_{23})} | W(\underline{e}_2) = z_2] \\
= \ & \frac{P[\{\underline{e}_1 + \underline{e}_2 \in \cup_{c \in C_{RS}^{23}, c \neq 0} Z^{23}(c, r_{e_{23})}\} \cap \{W(\underline{e}_2) = z_2\}]}{P[\{W(\underline{e}_2) = z_2\}]} \\
= \ & \sum_{z_3=0}^{n_3} \sum_{s=0}^{r_{e_{23}}} \frac{\binom{n_3}{z_3}}{\binom{n_{23}}{z_2+z_3}} \sum_{j=n_{23}-k+1}^{n_{23}} A_j(n_{23}, q, k) T(n_{23}, j, z_2+z_3, s) \frac{p_e^{z_3}(1-p_e)^{n_3-z_3}}{(q-1)^{z_2+z_3}}.
\end{aligned} \tag{3.18}$$

The expression uses the fact that the number of error patterns $\underline{e}_2 + \underline{e}_3$ of weight z_2 in first n_2 coordinates and z_3 in next n_3 coordinates , which are at a distance s from some nonzero codeword in C_{RS}^{23}, is given by eq. 3.14 as

$$\frac{\binom{n_2}{z_2}\binom{n_3}{z_3}}{\binom{n_{23}}{z_2+z_3}} \sum_{j=n_{23}-k+1}^{n_{23}} A_j(n_{23}, q, k) T(n_{23}, j, z_2+z_3, s).$$

To calculate a the second term in the approximation, namely $P[U_{12} \cap U_{23}]/P[D'_{12}]$, we need the *three-step* weight distribution function of the underlying mother code.

We have,

$$P[U_{12} \cap U_{23}] = P[\{\underline{e}_1 + \underline{e}_2 \in \cup_{c \neq 0} Z^{12}(c, r_{e_{12}})\} \cap \{\underline{e}_2 + \underline{e}_3 \in \cup_{c \neq 0} Z^{23}(c, r_{e_{23}})\}].$$

The expression for $P[U_{12} \cap U_{23}]$ is given by,

$$\sum_{\substack{z_1, z_2, z_3 \in B_z \\ j_1, j_2, j_3 \in B_J \\ s_1, s_2, s_3 \in B_S}} A^3_{j_1 j_2 j_3}(n_1, n_2, n_3, q, k) \prod_{i=1}^{3} T(n_i, j_i, z_i, s_i) \left(\frac{p_e}{q-1} \right)^{\sum_{i=1}^3 z_i} (1-p_e)^{\sum_{i=1}^3 (n_i - z_i)}$$

where $B_z \stackrel{def}{=} \{z_{1,}z_2, z_3 : 0 \leq z_i \leq n_i; z_1 + z_2 + z_3 \neq 0\}$ denotes the collection of error pattern distributions in the three sets of coordinates $n_1 n_2, n_3$. $B_J \stackrel{def}{=} \{j_1, j_{2,}j_3 : 0 \leq j_i \leq n_i; j_1 + j_2 + j_3 \geq n_1 + n_2 + n_3 - k + 1\}$ denotes the possible weights of non-zero codewords in the the coordinates. Finally $B_S \stackrel{def}{=} \{s_1, s_2, s_3 : 0 \leq s_i \leq n_i, s_1 + s_2 \leq r_{e_{12}}, s_2 + s_3 \leq r_{e_{23}}\}$ denotes the set of distances of error patterns from codewords which will result in the event $U_{12} \cap U_{23}$. Note that the constraints $s_1 + s_2 \leq r_{e_{12}}, s_2 + s_3 \leq r_{e_{23}}$ ensure that no error pattern is counted more than once.

Scheme	Throughput	Reliability	Avg. Feedback (NACKs)
$p_e = 0.1$			
CF-1	0.5690	6.9446	0.6611
CF-2	0.5603	6.9702	0.4722
CF-3	0.5180	7.4478	0.2089
CF-4	0.4928	7.7394	0.0778
CF-5	0.4685	7.1192	0.0248
CR-1	0.5917	7.0956	2.3210
CR-2	0.5807	7.4042	2.0775
CR-3	0.5643	8.0662	1.7867
CR-4	0.5312	8.6273	0.6324
CR-5	0.5078	9.1885	0.4484
T1-1	0.4658	7.7293	0.0734
T1-2	0.4602	7.1245	0.0226
T1-3	0.4319	9.2092	0.0290
T1-4	0.4174	8.6405	0.0087
T1-5	0.3990	8.1228	0.0024
T1-6	0.3797	10.1992	0.0033
T2-1	0.3881	8.3359	0.7179
T2-2	0.4463	7.2248	0.3788
T2-3	0.4037	9.3570	0.4154
T2-4	0.4504	8.4845	0.1841
T2-5	0.4699	7.7601	0.0684
T2-6	0.4622	7.1343	0.0222
T2-7	0.4356	9.2216	0.0282
T2-8	0.4188	8.6443	0.0086
T2-9	0.3995	8.1238	0.0024

Table 3.1: Performance of various schemes for symbol symmetric $GF(32)$ channel with $p_e = 0.1$

Scheme	Throughput	Reliability	Feedback
$p_e = 0.05$			
CF-1	0.6649	6.8446	0.6082
CF-2	0.6653	6.8581	1.0302
CF-3	0.6485	6.9565	0.4679
CF-4	0.6071	7.1893	0.1722
CF-5	0.6021	7.3797	0.1411
CF-6	0.5695	7.1622	0.0357
T1-1	0.5320	8.5284	0.1566
T1-2	0.5543	7.7346	0.0310
T1-3	0.5304	7.0479	0.0055
T1-4	0.4965	9.4175	0.0071
T2-1	0.5420	8.5916	0.1354
T2-2	0.5548	7.7479	0.0301
T2-3	0.5304	7.0503	0.0055

Table 3.2: Performance of various schemes for symbol symmetric $GF(32)$ channel with $p_e = 0.05$

Chapter 4

Progressive Unequal loss Protection in the absence of Feedback

4.1 Introduction

The high data rate, loss-tolerant and sometimes delay-sensitive nature of multimedia sources like images and video signals is in contrast with the delay-insensitive but loss-intolerant nature of data. Traditional transmission schemes and protocols developed for wireless transmission of data may be inefficient or overly conservative for the transmission of multimedia sources. As a large and increasing fraction of the network traffic comprises multimedia applications, their transmission over noisy channels and lossy networks merits special attention. Hence there has been much research in the last few years on devising "joint" source and channel coding schemes for transmission of specific sources over noisy channels and lossy networks.

Embeddedness (successive refinability) or scalability in bit rate is a desirable property for a source coder as it provides flexibility and the capability to progressively reconstruct the source. An embedded source coder allows the decoder to reconstruct the source at different bit rates from the prefixes of a single bit stream. *Progressive re-*

construction is possible as each additional bit (or a set of bits) improves the quality of reconstruction. Several highly competitive and low complexity algorithms for embedded image coding have been developed in the literature. Examples are Embedded Zerotrees of Wavelets (EZW) [56], the popular Set Partitioning In Hierarchical Trees (SPIHT) coder [52] as well as recent works in [49, 42].

The embeddedness property, which allows the user to transmit and receive the source progressively in the absence of transmission noise, typically makes the source coder sensitive to transmission noise. An error in an embedded bit stream may cause misinterpretation of the later bits, leading to error propagation and a possible loss in synchronization. The progressive property is lost as the bits following the error may not improve the quality of reconstruction; in fact, they might damage the reconstruction. Therefore, it is important to design good joint source-channel coding schemes for transmission of embedded source coders over noisy channels. In addition, it is desirable to retain the progressive reconstruction property in image transmission when the channel is noisy.

There is a growing body of recent work in transmitting progressively coded images over different kinds of noisy channels [57, 59, 12, 1, 39, 11, 46, 18]. They are based on equal or unequal error protection of the output of an embedded source coder and discuss ways to combat error propagation. These schemes, while making use of a progressive source coder, are designed for a fixed target transmission rate. They do not explicitly consider the performance of the scheme at intermediate rates or provide direct scalability to a higher transmission rate. Although, in some cases, operationally optimal progressive transmission is a by-product, either of the design or of the imposed constraints, *e.g.* [57, 12].

In this chapter, we consider the optimal design of a joint source-channel coder using an embedded source coder, with an emphasis on progressive transmission over memoryless bit error channels and packet erasure channels with no feedback.

First, we provide a formulation of optimal unequal protection for memoryless channels under a transmission budget constraint. We provide an algorithm which chooses an optimal unequal error protection policy from an arbitrary family of (block based) channel codes. Earlier attempts at this problem have used model based techniques, *e.g.* modeling the distortion-rate performance of the image coder by exponentials [1] or modeling the performance of the channel codes by curve fitting [39]. Here, the proposed algorithm is exact and does not require model based computation either for the source coder or for the channel code family chosen. It is also independent of the actual performance criterion used (average distortion, average Peak Signal-to-Noise Ratio (PSNR) or average useful source coding rate). The framework developed can be used for memoryless channels including memoryless bit error channels (*e.g.* BPSK transmission over AWGN channels with hard or soft demodulation) and memoryless packet erasure channels.

Second, we show how progressive transmission can be achieved while retaining optimality at intermediate transmission rates if the underlying family of channel codes is embedded (rate compatible). The Rate Compatible Punctured Convolutional (RCPC) codes [29] satisfy this criterion for bit error channels; punctured Reed-Solomon (RS) codes satisfy this property for erasure channels [67]. We do not consider the case when both bit errors and packet erasures are present in the channel. That situation is considered in [18].

Our studies show that the proposed schemes offer a performance – measured in average PSNR vs. bit rate – superior to any scheme using equal error protection. The amount of improvement depends on the transmission rate and a variety of other parameters, including the available choice of the error control codes and the channel statistics.

Further, the proposed scheme can be used for progressive encoding while guaranteeing optimality at a number of intermediate rates.

The organization of the chapter is as follows. Section 4.2 describes the basic set up. Section 4.3 discusses the performance criteria and the optimization problem for memoryless channels. It describes the solution of the optimization problem and presents the algorithm for unequal error and erasure protection. Section 4.4 discusses when and how optimal progressive transmission can be accomplished. Simulation results are presented in Section 4.5. Finally, Section 4.6 ends with concluding remarks.

4.2 The Transmission Scheme

Consider the transmission of the output of an embedded source coder over a noisy channel. A challenge in transmitting such codes is to minimize the damage caused by error propagation. A twofold strategy that can be employed is: 1) *prevention* of error and hence error propagation by forward error correction and 2) *detection* of possible post-decoding errors and discarding all the bits that may contribute to error propagation. In the case of packet erasure channels, the problem is to avoid uncorrectable erasures.

Consider an embedded source coder which simultaneously encodes N_s source samples. Its output, the source encoder bits, is packetized into *fixed-length* source-packets of, say, k_s bits each. As the source coder is embedded, the representation of the source at rates which are multiples of k_s/N_s can be obtained from a prefix of this stream of source-packets.

For error protection, we assume that we are provided with a finite family of block codes, each member of which has error correction and error detection capability, like those in [57]. These codes operate on source-packets of k_s bits and generate blocks of bits of different lengths which are subsequently transmitted over the channel. Typical examples of such families are concatenated RCPC-CRC codes (*e.g.* [29, 57]) or punctured RS codes with bounded distance decoding [67]. The punctured RS code family is also used as codes for a symbol erasure channel. These families provide a selection of code rates necessary for unequal error protection.

We use fixed-length source packetization but we allow source-packets to receive a variable number of error protection bits, *i.e.* to have a variable-length error correction. A three-fold motivation for doing this is as follows. (i) The rate compatible families of error and erasure correction codes can be implemented with a *single* channel encoder-decoder pair. Schemes using variable-length source-packet to fixed-length channel packets lose this advantage. (ii) The variable codeword lengths of the rate compatible families are usually multiples of a smaller fixed-length channel block, which can be used for synchronization. (iii) The influence of the size of the actual packet put on the channel over the logical 'packet' used for error control can be reduced by interleaving (*e.g.* [46]).

The transmission process proceeds as follows. Each source-packet output by the source coder is encoded with a potentially different channel code, chosen according to some *code assignment policy*. These channel coded bits are transmitted over the noisy channel. The receiver tries to recover the source-packets from the (noisy) received channel codewords. The channel decoder either correctly decodes a source-packet or detects an error and declares a *source-packet decoding failure*. In the case of a packet erasure channel, a source-packet decoding failure is declared if the source-packet cannot be recovered from the unerased received packets. We assume that the probability of undetected errors is zero. This assumption is true for erasure channels and can be approximated with high reliability for bit error channels. As discussed earlier, for embedded source coders it is often reasonable to assume that if a source-packet is decoded erroneously by the receiver, then the subsequent source-packets cannot improve the quality of the source. Hence, at any stage in transmission, the source is reconstructed only from the decoded bit stream up to the first source-packet that contains a detectable error or irrecoverable erasure. For some embedded source coders, it may be possible to separate the source bit stream either into critical and non-critical parts or into several independent substreams. In this chapter, we restrict our attention to the case where no such separation is available and the first error or erasure leads to error propagation. Alternatively, the proposed scheme may be applied to only the critical part of the bit stream or to each independent substream. We have not investigated that approach here.

In the next section, the performance of the proposed scheme is computed and optimized for a memoryless channel.

4.3 Optimal Unequal Protection for Memoryless Channel

Let us denote the family of error correction-detection channel codes by $\mathcal{C} = \{c_1, c_2, \dots c_J\}$. Let the code-rates of the channel codes be denoted by $r_c(c_i), i = 1, \dots, J$. Therefore, a codeword for a source-packet of length k_s bits, protected by code c_i, has length $k_s/r_c(c_i)$ bits. Let the probability of source-packet decoding failure for the given memoryless channel for the channel code $c_i \in \mathcal{C}$ be $P_e(c_i)$.

If the first i source-packets are available to the decoder, the source can be reconstructed to a rate ik_s/N_s bits per source sample, where N_s is the number of source samples. Let $r_s \stackrel{def}{=} k_s/N_s$ be the rate in bits per sample per source-packet for the source.

The unequal protection for the source-packets is described by specifying a *code allocation policy*. A code allocation policy π allocates channel code $c_\pi^i \in \mathcal{C}$ to the i^{th} source-packet out of the source coder. A policy π is described by the number of source-packets to be transmitted ($N(\pi)$) and by a sequence of channel codes $\{c_\pi^1, c_\pi^2, \ldots, c_\pi^{N(\pi)}\}$ to be used with the sequence of source-packets. $N(\pi)$ can also be thought of as the index of the terminating source-packet for the policy. The normalized transmission rate (in channel bits per source sample) for the policy π is given by

$$R_{T\pi} \stackrel{def}{=} \sum_{i=1}^{N(\pi)} \frac{r_s}{r_c(c_\pi^i)}. \tag{4.1}$$

4.3.1 Performance Criteria

Several single-parameter criteria can be used to measure the performance of a code-allocation policy. Consider the transmission of an image by the proposed scheme using a policy $\pi = \{c_\pi^1, c_\pi^2, \ldots, c_\pi^{N(\pi)}\}$. To compute the performance of the policy, let us introduce the following notation. For integers $k = 1, 2, \ldots, N(\pi)$ and $i = k - 1, k, k+1, \ldots, N(\pi)$, let $P_{i|k-1}(\pi)$ denote the conditional probability that exactly the first i source packets are decoded correctly given that the first $k-1$ packets are decoded correctly, while using the policy π. Then $P_{i|k-1}(\pi)$ can be computed as,

$$P_{i|k-1}(\pi) \stackrel{def}{=} \begin{cases} P_e(c_\pi^k) & i = k-1, \\ \prod_{j=k}^{i}(1 - P_e(c_\pi^j))P_e(c_\pi^{i+1}) & i = k, k+1, \ldots, N(\pi) - 1, \\ \prod_{j=k}^{N(\pi)}(1 - P_e(c_\pi^j)) & i = N(\pi). \end{cases} \tag{4.2}$$

Note that $\sum_{i=k-1}^{N(\pi)} P_{i|k-1}(\pi) = 1$ for $k = 1, 2 \ldots, N(\pi)$.

Let the operational distortion-rate performance of the source coder be given by $D(r)$ where r is the rate in bits per sample. Then, as the source is reconstructed only from the source-packets received prior to a source-packet decoding failure, the *expected distortion* at the receiver using a policy π (at transmission rate $R_T(\pi)$) is given by

$$\bar{D}_\pi \stackrel{def}{=} \sum_{i=0}^{N(\pi)} D(ir_s)P_{i|0}(\pi). \tag{4.3}$$

Similarly, let the PSNR-rate performance of the source coder for the source image be given by $PSNR(r)$ where $PSNR(r) \stackrel{def}{=} 10\log_{10}\frac{255^2}{D(r)}$ dB. Then the *expected PSNR* for the policy π is given by,

$$\overline{PSNR}_\pi \stackrel{def}{=} \sum_{i=0}^{N(\pi)} PSNR(ir_s)P_{i|0}(\pi). \tag{4.4}$$

Finally, we consider another performance criterion, namely the average number of source encoder bits per sample received before a source-packet decoding failure (which is the beginning of a possible error propagation). We call this criterion the *average useful source coding rate*. This criterion is motivated by the fact that the longer the error-free prefix is, the better would be the reconstruction of the source. For a policy π, the average number of source-packets received before a source-packet decoding failure is analogously written as,

$$V_\pi \stackrel{def}{=} \sum_{i=0}^{N(\pi)} iP_{i|0}(\pi). \tag{4.5}$$

Note that the average useful source coding rate is given by r_sV_π.

The channel code allocation problems for the joint source-channel coding scheme under the constraint of total transmission rate R bits per source sample, can be expressed in terms of the following optimization problems.

- **Problem A:** For minimization of the average distortion the problem is,

$$\min_\pi \bar{D}_\pi \text{ subject to } R_{T\pi} \leq R. \tag{4.6}$$

- **Problem B:** For maximization of the average PSNR, the problem is,

$$\max_{\pi} \overline{PSNR}_{\pi} \text{ subject to } R_{T\pi} \leq R. \tag{4.7}$$

- **Problem C:** Finally, as r_s is a constant, to maximize the average useful source coding rate, the problem is,

$$\max_{\pi} V_{\pi} \text{ subject to } R_{T\pi} \leq R. \tag{4.8}$$

Any of the above optimization criteria can be chosen to suit the application. The drawback of choosing to maximize the average PSNR or to minimize the average distortion is that the unequal error protection policy so obtained needs to be conveyed to the receiver somehow. This may require transmission of sensitive side information over the noisy channel.

There are some desirable properties that make the design criterion (4.5) (and hence Problem C) interesting and particularly useful:

1. The design criterion (4.5) does not involve the source statistics or the source-coder performance. The receiver can also carry out this optimization and hence the unequal protection policy can be available at the receiver without the need for transmission of any side information. Criterion (4.5) is also useful in situations where the source coder is not embedded but error propagation is still an issue. For example, in variable-length coded macroblocks with synchronization symbols, the error propagation within a macro-block can be prevented or maximally delayed by unequal error protection design based on maximizing the criterion (4.5).

2. As we shall see in the end of this section, its solution is considerably simpler than the other two criteria.

3. Finally, the optimal policies for optimization criterion (4.5) allow provably optimal progressive transmission at intermediate rates. We discuss this in detail in Section 4.4.

4.3.2 Solution to Optimization Problems

The cost functions (4.3),(4.4) and (4.5) are not additive, hence Problems A, B and C are not conventional rate allocation problems. But, it can be shown that the three problems can be solved exactly by a framework based on dynamic programming. The principal idea of the solution is to write the objective function in the absence of noise (distortion, PSNR or number of source-packets) as a sum of incremental rewards, which are accumulated as each source-packet is successfully decoded by the receiver. Let δ_i denote the incremental reward when the i^{th} source-packet is successfully received. Hence, if the task is to minimize the average distortion, δ_i is defined as

$$\delta_i \stackrel{def}{=} D((i-1)r_s) - D(ir_s),\ i = 1, 2, \ldots. \tag{4.9}$$

Similarly for average PSNR maximization, δ_i is defined as

$$\delta_i \stackrel{def}{=} PSNR(ir_s) - PSNR((i-1)r_s),\ i = 1, 2, \ldots. \tag{4.10}$$

And, for maximization of the average useful source coding rate, δ_i is defined as

$$\delta_i \stackrel{def}{=} 1,\ i = 1, 2, \ldots. \tag{4.11}$$

The objective functions in Eqs. (4.3), (4.4) and (4.5) are related to these incremental rewards as follows. For a code allocation policy $\pi = \{c_\pi^1, c_\pi^2, \ldots c_\pi^{N(\pi)}\}$ and for integers $k, 1 \le k \le N(\pi)$, define,

$$\Delta(k, \pi) \stackrel{def}{=} \sum_{i=k}^{N(\pi)} \left(\sum_{j=k}^{i} \delta_j \right) P_{i|k-1}(\pi). \tag{4.12}$$

From the values of δ_i defined in (4.9), (4.10) and (4.11), and using (4.3), (4.4) and (4.5) it can be verified that

$$\Delta(1,\pi) = \begin{cases} D(0) - \bar{D}_\pi, & \text{for Problem A,} \\ \overline{PSNR}_\pi - PSNR(0), & \text{for Problem B,} \\ \bar{V}_\pi, & \text{for Problem C.} \end{cases} \tag{4.13}$$

Hence Problems A, B and C defined in Eqs. (4.6), (4.7) and (4.8) reduce to the following problem:

$$\max_\pi \Delta(k,\pi) \text{ subject to } R_T(k,\pi) \overset{def}{=} \sum_{i=k}^{N(\pi)} \frac{r_s}{r_c(c_\pi^i)} \leq R, \tag{4.14}$$

for $k = 1$.

Now, from Eq. (4.2) it can be seen that, for $k = 1, 2, \ldots, N(\pi)$ and $i = k, k + 1, \ldots, N(\pi)$, the following holds.

$$P_{i|k-1}(\pi) = (1 - P_e(c_\pi^k))P_{i|k}(\pi). \tag{4.15}$$

From Eqs. (4.12) and (4.15) notice that $\Delta(k,\pi)$ satisfies the following recursion.

$$\Delta(k,\pi) = \begin{cases} (1 - P_e(c_\pi^{N(\pi)}))\delta_{N(\pi)}, & \text{for } k = N(\pi), \\ (1 - P_e(c_\pi^k))(\delta_k + \Delta(k+1,\pi)), & \text{for } k = 1, 2, \ldots, N(\pi) - 1. \end{cases} \tag{4.16}$$

Also, clearly, $R_T(k,\pi) = \frac{r_s}{r_c(c_\pi^k)} + R_T(k+1,\pi)$. Notice that, for a policy π, $\Delta(k,\pi)$ and $R_T(k,\pi)$ do not depend on $c_\pi^1, c_\pi^2, \ldots, c_\pi^{k-1}$. Hence the solution to the maximization problem in Eq. (4.14) needs to be specified only over a subsequence of channel codes, namely, $c_\pi^k, c_\pi^{k+1}, \ldots$.

Equation (4.16) leads to the following dynamic programming result for solving (4.14).

Proposition 2 *Let* $\{c_*^k, c_*^{k+1}, c_*^{k+2}, \ldots, c_*^{N^*(k,R)}\}$ *be the solution for the maximization problem in (4.14). That is, it is the subsequence of channel codes achieving the maximum in (4.14) for starting source-packet index* k *and rate constraint* R*. Let* $r_{min} \stackrel{def}{=} \min_{c \in C} \frac{r_s}{r_c(c)}$ *, then the following results hold.*

1. *For notational convenience, let* $\Delta^*(k, R)$ *denote the optimal value of the objective function in (4.14) (the total reward). Then,* $\Delta^*(k, R)$ *satisfies the dynamic programming equation,*

$$\Delta^*(k,R) = \begin{cases} 0, & \textit{if } R < r_{min}, \\ \max_{c \in \mathcal{C}} (1 - P_e(c))(\delta_k + \Delta^*(k+1, R - \frac{r_s}{r_c(c)})), & otherwise \end{cases} \tag{4.17}$$

2. *The channel code* c_*^k*, is the channel code achieving the maximum in (4.17).*

3. *The subsequence* $\{c_*^{k+1}, c_*^{k+2}, \ldots, c_*^{N^*(k,R)}\}$ *solves (4.14) for starting source-packet index* $k+1$ *and rate constraint* $R - \frac{r_s}{r_c(c_*^k)}$*.*

4. *Finally, the terminating source-packet index is found by,*

$$\begin{aligned} N^*(k,R) &= N^*(k+1, R - \frac{r_s}{r_c(c_*^k)}) \textit{ if } \Delta^*(k+1, R - \frac{r_s}{r_c(c_*^k)}) > 0 \\ &= k \textit{ otherwise.} \end{aligned}$$

The proof is straightforward and is omitted here. It is based on the recursion in Eq. (4.16) and on the observation that for any policy π, $\Delta(k+1, \pi)$ and $R_T(k+1, \pi)$ do not depend on the k^{th} channel code c_π^k.

Algorithm for arbitrary δ_i

For arbitrary incremental rewards δ_i, the statement of Proposition 2 can be written as an algorithm as follows.

Algorithm 1 (optimal unequal error protection) $\Delta^*(k,r)$ *is computed as a recursive function call.*

$$\begin{aligned} \Delta^*(k,r) &:= 0 \textit{ if } r < r_{min} \\ &:= \max_{c\in\mathcal{C}}(1-P_e(c))(\delta_k + \Delta^*(k+1, r - \frac{r_s}{r_c(c)})). \end{aligned} \tag{4.18}$$

The channel code achieving the maximum in (4.18) is used for encoding the k^{th} *source-packet.*

Notice that the channel code obtained by the algorithm for the k^{th} source-packet depends on all the δ_i as well as the target transmission rate.

The computation of $\Delta^*(1,R)$ depends on the computation of $\Delta^*(2,r)$ for a finite number of values of r, all of which are strictly smaller than R. The computation of $\Delta^*(k,r)$ in turn, depends on the computation of $\Delta^*(k+1,r')$ for even smaller values of r'. The recursion terminates by returning a value of 0 when k is sufficiently large so that the target transmission rate falls below r_{min}. It may appear that the number of calls to the recursion grows exponentially. But computation can reduced by storing the computed values $\Delta^*(k,r)$ in the memory. Figure 4.1 illustrates how the values of $\Delta^*(k,r)$ can be computed using a time varying trellis.

Algorithm for average useful source coding rate

It is easy to see that, if $\delta_i =$ constant $\forall i$, then the optimization of (4.14) does not depend on the starting source-packet index k. Hence, for such a case, we have

$$\Delta^*(k,R) = \Delta^*(1,R) \;\; \forall R \text{ for } k = 1,2,\ldots. \tag{4.19}$$

Further, in such a case, the channel code obtained by the algorithm for the k^{th} source-packet depends only on the target transmission rate. Hence, for the design criterion in (4.5), *i.e.* solution of (4.8), Algorithm 1 can be rewritten as follows.

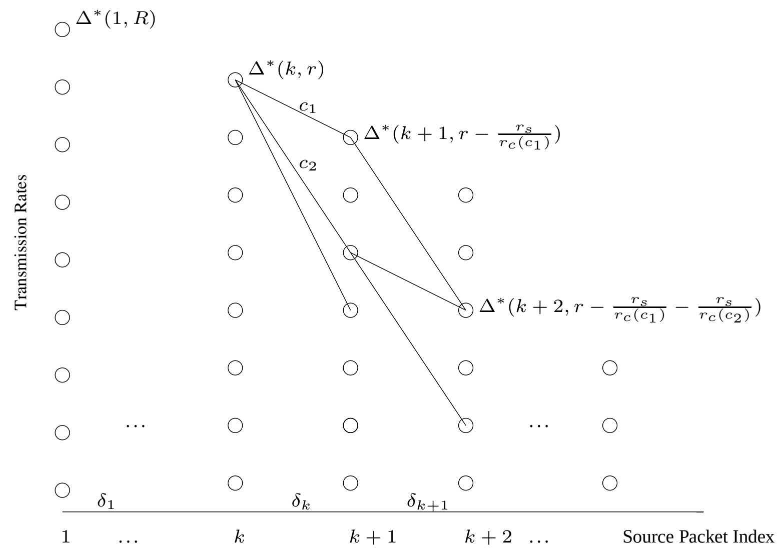

Figure 4.1: Trellis for maximizing the performance for arbitrary δ_i.

Algorithm 2 (maximization of average useful source coding rate) *If the incremental rewards are constant,* i.e. $\delta_i = 1 \ \forall i$, *then* $\Delta^*(1, r)$ *is computed as a recursive function call.*

$$\begin{aligned}\Delta^*(1, r) &:= 0 \textit{ if } r < r_{min} \\ &:= \max_{c \in \mathcal{C}} (1 - P_e(c))(1 + \Delta^*(1, r - \frac{r_s}{r_c(c)})). \end{aligned} \tag{4.20}$$

Figure 4.2 illustrates the trellis used for the computation of $\Delta^*(1, r)$ for the maximization of average useful source coding rate.

4.3.3 Complexity

Algorithm 1 for arbitrary sequence of nonnegative incremental rewards δ_i has a complexity (number of calls to the recursive function in which maximization in (4.18) needs to be performed) proportional to R^2. This can be seen as follows. Let ρ be the smallest grain of rate-increment per sample in the code family, *i.e.* $\rho = \frac{1}{N_s} GCD(\{\frac{k_s}{r_c(c)}, c \in C\})$.

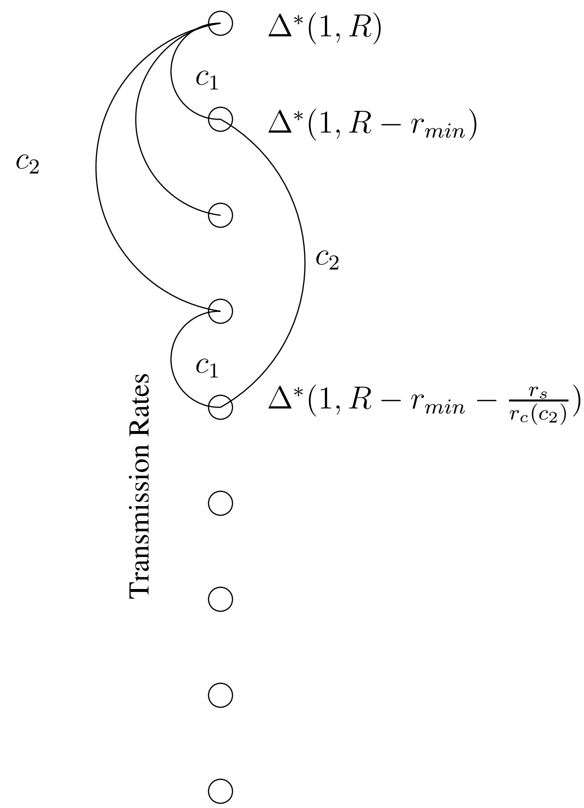

Figure 4.2: Trellis for maximizing the average useful source coding rate.

Then all achievable transmission rates, *i.e.* those in the collection $G(R) \stackrel{def}{=} \{R_T(\pi) : R_T(\pi) \leq R\}$ must be multiples of ρ. $|G(R)|$ grows linearly with R. Now from Algorithm 1, the computation of $\Delta^*(1, r)$ for all values of $r \in G(R)$, requires computation of $\Delta^*(2, r)$ for all values of $r \in G(R - r_{min})$, which, in turn, requires the computation of $\Delta^*(3, r)$ for all values of $r \in G(R - 2r_{min})$ and so on. Hence the total number of function calls needed to compute $\Delta^*(1, r)$ for all $r \in G(R)$ is upper bounded by $|G(R)| + |G(R - r_{min})| + |G(R - 2r_{min})| + \ldots + |G(R - (\lfloor R/r_{min} \rfloor - 1)r_{min})|$. This is an arithmetic progression, upper bounded by αR^2, for some α. For Algorithm 2, the number of function calls needed to compute $\pi^*(1, R)$ vary linearly with R. $|G(R)|$ computations of $\Delta^*(1, r)$ are sufficient to compute $\Delta^*(1, R)$.

4.4 Progressive Transmission

Good embedded source coders, like SPIHT [52], by design have very good performance at all rates. It is desirable to perform joint source-channel coding for these coders in such a way that in addition to having the best end-to-end performance for a given target transmission rate, the coder also achieves good performance at intermediate rates.

We shall say that two policies π_1 and π_2 with transmission rates R_1 and R_2, $R_2 > R_1$, allow progressive transmission, if the output at rate R_2 can be obtained by appending $(R_2 - R_1)$ bits per source sample to the bit stream at rate R_1. Or, conversely, if the bit stream for target rate R_1 can be obtained as a prefix in the bit stream for target rate R_2.

We suggest the use of rate compatible channel codes to achieve progressive transmission. Rate compatible codes are a family of channel codes in which the codewords of a low rate code can be obtained by adding some extra parity bits to the codeword of a high rate code. Popular examples of such codes are rate compatible punctured convolutional RCPC codes [29]. These codes combined with an outer error detection code like CRC encoding fixed-length source-packets provide good error correction and detection capabilities and have been used in the literature [57, 12, 39]. Similarly RS codes and their punctured versions are used for erasure channels.

In this section we shall assume that the channel code family $\mathcal{C}$ is rate compatible. Consider two policies $\pi_1 = \{c_{\pi_1}^1, c_{\pi_1}^2, \ldots, c_{\pi_1}^{N(\pi_1)}\}$ and $\pi_2 = \{c_{\pi_2}^1, c_{\pi_2}^2, \ldots, c_{\pi_2}^{N(\pi_2)}\}$ designed by some scheme for target rates R_1 and $R_2, R_2 > R_1$. Then, we have the following simple proposition.

Proposition 3 *If the channel code family is rate compatible, progressive transmission at rates R_2 and R_1, for $R_2 > R_1$, is possible using the two policies π_2 and π_1, if and* *only if,* $N(\pi_2) \geq N(\pi_1)$ *and*

$$\frac{r_s}{r_c(c^i_{\pi_1})} \leq \frac{r_s}{r_c(c^i_{\pi_2})} \textit{ for } 1 \leq i \leq N(\pi_1). \tag{4.21}$$

Proof: The proof is rather simple. If condition (4.21) is satisfied, progressive transmission is accomplished by first transmitting the bit stream corresponding to policy π_1 followed by the extra parity check bits needed to obtain the lower rate codes for policy π_2, *i.e.* $\frac{r_s}{r_c(c^i_{\pi_2})} - \frac{r_s}{r_c(c^i_{\pi_1})}$ bits per source sample for packet i, $i = 1, 2, \ldots$. Clearly, this cannot be done if (4.21) is not satisfied. Figure 4.3 illustrates the sequence of transmission for two policies. $\square$

If we require the transmission to be *optimally* progressive, then the policies $\pi^*(1, R)$ obtained by solving (4.6), (4.7) or (4.8) for different values of R must allow progressive transmission. Therefore, we must verify that those policies satisfy the conditions in Proposition 3.

Now let us consider the optimization criterion of (4.5). Let

$$\pi^*(1, R) = \{c^1, c^2, \ldots, c^{N(\pi^*(1,R))}\}$$

be the optimal policy solving (4.8) for rate R. Then by the result in Proposition 2, the subsequence, $\{c^2, \ldots, c^{N(\pi^*(1,R))}\}$ solves the corresponding version of (4.14) for the starting index 2 and rate constraint $R - \frac{r_s}{r_c(c^1)}$. Now, if δ_i's are constant, then as we have discussed earlier, the optimization (4.14) does not depend on the starting index k. Hence a policy which assigns c^2 to the *first* source-packet, c^3 to the second source-packet and similarly assigns $c^{N(\pi^*(1,R))}$ to the $N(\pi^*(1, R)) - 1^{st}$ source-packet is the optimal policy $\pi^*(1, R - \frac{r_s}{r_c(c^1)})$ for starting index 1 and rate constraint $R - \frac{r_s}{r_c(c^1)}$.

A simple interchange argument can be used to show that, for arbitrary rewards sequences δ_i and for all transmission rates R, if π is an optimal policy then, $P_e(c^i_\pi) \leq P_e(c^j_\pi)$ for $1 \leq i \leq j \leq N(\pi)$. Consequently, we must have the property that $r_s/r_c(c^i_\pi) \geq$

$r_s/r_c(c_\pi^j)$ for $1 \leq i \leq j \leq N(\pi)$ for an optimal policy π *i.e.* the optimal policies are *code rate increasing*. Therefore, we have,

$$\frac{r_s}{r_c(c^i_{\pi^*(1,R)})} \geq \frac{r_s}{r_c(c^{i+1}_{\pi^*(1,R)})} = \frac{r_s}{r_c(c^i_{\pi^*(1,R-\frac{r_s}{r_c(c^1)})})} \text{ for } i = 1, 2, \ldots, N(\pi^*(1,R)) - 1. \tag{4.22}$$

This implies that the conditions in Proposition 3 are satisfied. Hence we get the following result.

Proposition 4 *For a channel code family consisting of rate compatible codes, let* $\pi^*(1, R)$ *be the optimal policy solving (4.8) for target rate* R*. Then* $\pi^*(1, R)$ *and* $\pi^*(1, R - \frac{r_s}{r_c(c^1_{\pi^*(1,R)})})$ *allow optimal progressive transmission at rates* R *and* $R - \frac{r_s}{r_c(c^1_{\pi^*(1,R)})}$.

Proof: Proof is already outlined before the statement of Proposition 4.

This proposition can now be applied to the optimal policy at rate $R - \frac{r_s}{r_c(c^1_{\pi^*(1,R)})}$, to obtain another lower intermediate transmission rate where the optimal policy can be executed. In the same manner, a sequence of intermediate transmission rates can be obtained, at which provably optimal progressive transmission is possible. The sequence of bits transmitted follows the scheme discussed in Figure 4.3. First, bits corresponding to the policy for a low target rate are transmitted. Then the extra parity check bits and new source-packets needed to achieve a higher target rate are transmitted. Figure 4.4 sketches the inverse code rate profile of an optimal policy consisting of five source-packets. In this figure, the labels $1, 2, 3, 4, 5$ indicate the order in which bits corresponding to the source-packets are transmitted.

The resulting bit stream has some interesting properties. It is a stream in which the bits for a single channel codeword are not necessarily contiguous. This deferred transmission of redundancy creates the possibility that a source-packet decoding failure at one target rate can be overcome and the source-packet recovered if the target rate is increased and more bits for that packet are received. In that sense, the bit stream is always progressive.

It can be shown by counterexamples that, even for the criterion of average useful source coding rate, optimal progressive transmission may not be possible at *arbitrary* rate pairs R_1 and R_2. For other performance criteria, at this point, not much can be said about optimal progressive transmission without making assumptions on the arbitrary incremental rewards δ_i.

4.5 Simulation Results

The schemes presented in the chapter assume that the designer is provided with the source coder and a family of channel codes. The design requires only the knowledge of source-packet decoding failure probabilities for the given family of channel codes over the given channel. The design is independent of the actual decoding techniques used, *e.g.* for memoryless bit error channels, it is possible to use the proposed scheme both with hard or soft demodulation at the receiver.

Simulations were conducted on the 512×512 gray-scale Lenna image compressed with the SPIHT algorithm with arithmetic coding. The source bit stream was divided into source-packets of length 32 bytes.

For binary symmetric channels, the channel code families were chosen to be concatenated codes of RCPC codes as inner codes and a 2-byte CRC for outer error detection code. We present results for three different channel code families. These families are are RCPC codes derived from different mother codes and used with different decoders.

Code family A is a collection of RCPC codes derived from a 64-state, rate 1/3 convolutional mother code taken from [29]. When used along with list-viterbi decoding with a search depth of 100 paths, these codes form a high performance channel code family similar to that in [57].

Code family B is a relatively weaker RCPC code family derived from a 16-state, rate 1/4 mother code taken from [29]. It is used with list-viterbi decoding with a search depth of 10. When the codes of code family B are used without list decoding (search depth = 1), we get the family C of channel codes. The parameters P_e of the code families were obtained by simulation. The results of using the proposed algorithm for unequal error protection of the image Lenna for a binary symmetric channel with bit error rate of 0.01 are presented in Figures 4.5 through 4.10.

Figures 4.5, 4.7 and 4.9 show the average PSNR performance of the scheme optimizing the PSNR for channel BER 0.01 for the code families A, B and C, respectively. The figures also show the performance of Equal Error Protection (EEP) schemes using the channel codes from the same family. The code rates in the legends do not include the (fixed) code rate of the outer CRC code.

For clarity, Figures 4.6, 4.8 and 4.10 depict the difference in the average PSNR of the optimized scheme and that of different EEP schemes, against the total transmission rate. Figures 4.6, 4.8 and 4.10 also include the difference in PSNR of the scheme maximizing the expected PSNR and the scheme maximizing the average useful source coding rate.

The first conclusion that can be drawn from Figures 4.6, 4.8 and 4.10 is that the loss of EEP schemes over the optimized schemes is positive. That is, as expected, the optimized schemes always perform as good as or better than any equal error protection scheme from the same family, for all transmission rates.

The second key observation is that the improvement of the optimal scheme over any fixed EEP scheme depends on the transmission rate. For example, in Figure 4.8, the loss of the EEP scheme with code rate $4/7$ varies from 0.4 dB to less than 0.2 dB to as high as 0.6 dB depending on the transmission rate. A low code rate EEP scheme which performs well (close to optimal) at high transmission rates is overprotective at low transmission rates. A higher code rate EEP scheme may be efficient at low transmission rate but as the transmission rate is increased, the average PSNR may saturate as the probability of source-packet decoding failure somewhere in the image increases with the target transmission rate. Note that it is not possible to "switch" between two EEP schemes at the crossover points during a progressive transmission. The two policies may not satisfy the conditions in Proposition 3. The performance loss of the scheme maximizing the average useful source coding rate also appears to depend on the transmission rate. But the loss is smaller than that for any EEP scheme and hence, the scheme maximizing the average source coding rate will also perform better than any EEP scheme at all transmission rates.

Third, the unequal error protection scheme is more effective when the available channel-code family is weak. If the code family is strong, *e.g.* the high performance codes in [57], then for a significant portion of the range of transmission rates of interest, the performance of a single channel code is fairly close to the optimal. In such cases the benefits of unequal error protection are marginal. For the system designed with code family A and for channel BER 0.01 the expected PSNR values for the image Lenna at 0.25, 0.5 and 1.0 bits per pixel are 32.30, 35.28, and 38.28 dB respectively. These figures are approximately 0.3 dB higher than the corresponding PSNR results in [57]. When the channel code family is weak, any EEP scheme performs closest to the optimal only for a short range of transmission rates. At other rates, its performance may be substantially suboptimal compared to the UEP scheme.

The same technique can be applied to memoryless packet erasure channels. For simulations, we considered transmission over a memoryless packet erasure channel of packet size 8 bytes. We use a (255, 32) RS code over $GF(2^8)$ as the mother code of the family of erasure correcting codes. Eight consecutive 1-byte symbols of $GF(2^8)$ are arranged in one packet to yield a mother Packet Erasure Correcting (PEC) code of parameters (31,4). The family consists of $(n, 4)$ PEC codes, for $4 \leq n \leq 31$, obtained as punctured versions of the mother code. An $(n, 4)$ PEC code is capable of correcting up to $n - 4$ packet erasures. This code family, though less efficient than RS codewords for byte erasures, is chosen primarily to keep the number of codes in the family small.

Figures 4.11 and 4.12 plot analogous results for this packet erasure channel with a packet loss rate of 20%. Again, no single EEP policy performs closest to the optimal at all transmission rates. Depending on the target rate, gains up to 0.5 dB can be obtained over any EEP scheme chosen from the family.

4.6 Conclusion

In this chapter we consider joint source-channel coding of images compressed with embedded source coders for transmission over memoryless noisy channels. The emphasis is on retaining the progressive nature of the transmission. A framework for optimal transmission over memoryless error and packet erasure channels is developed. An algorithm is developed for assigning optimal unequal error or erasure protection for a given memoryless bit error or packet erasure channel. We also show how progressive transmission can be achieved with rate compatible families of channel codes. The optimization criterion of maximizing the average useful source coding rate is shown to have the possibility of optimal progressive transmission at a number of intermediate transmission rates.

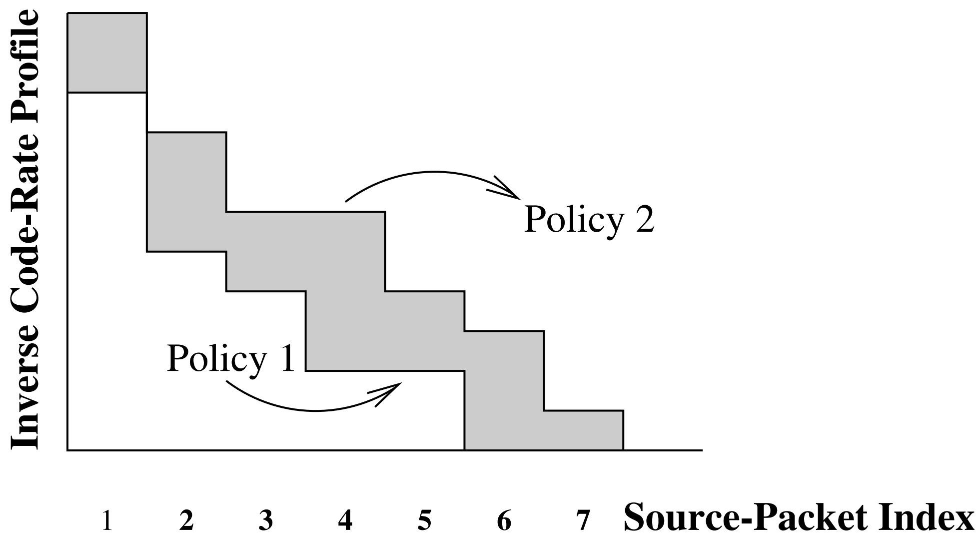

Figure 4.3: Progressive transmission with two policies; shaded area is transmitted second

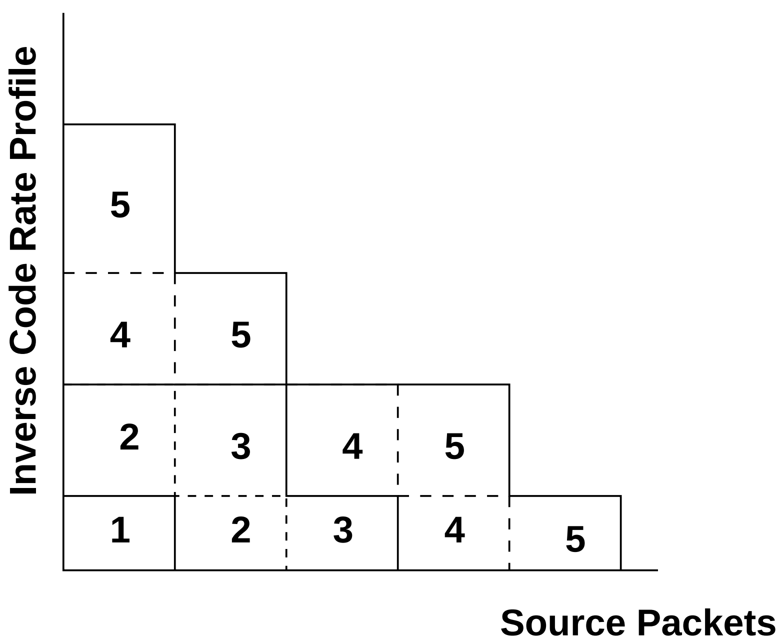

Figure 4.4: Optimal progressive transmission of five source-packets; the numbers indicate the sequence in which bits are transmitted

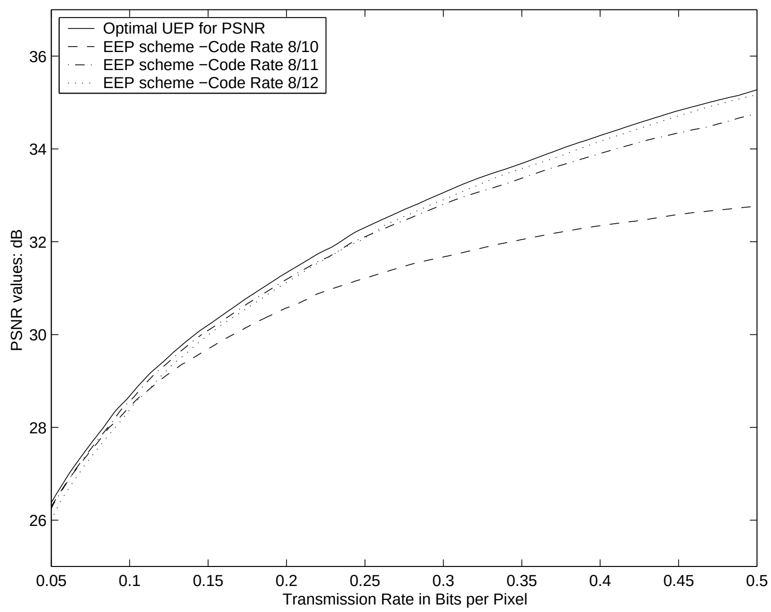

Figure 4.5: Average PSNR performance of unequal error protection over memoryless channels for the image Lenna. Code family A, BER = 0.01.

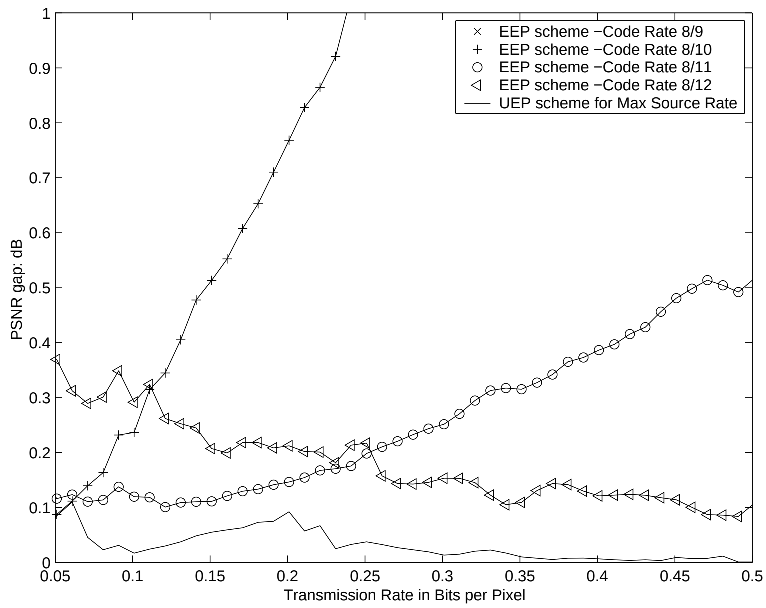

Figure 4.6: The loss of PSNR in EEP schemes and optimal UEP scheme maximizing average useful source coding rate compared to the optimal UEP scheme maximizing PSNR for the image Lenna. Code family A. BER =0.01.

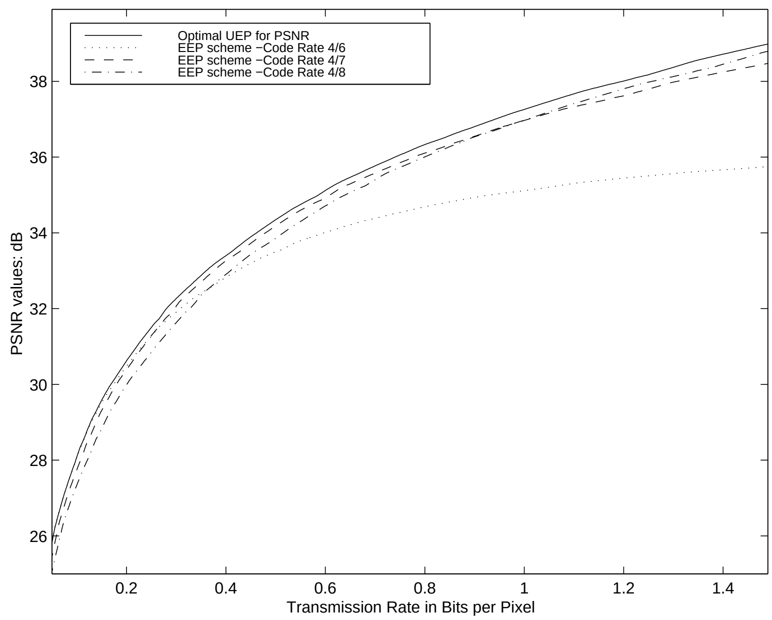

Figure 4.7: Average PSNR performance of unequal error protection over memoryless channels for the image Lenna. Code family B, BER = 0.01.

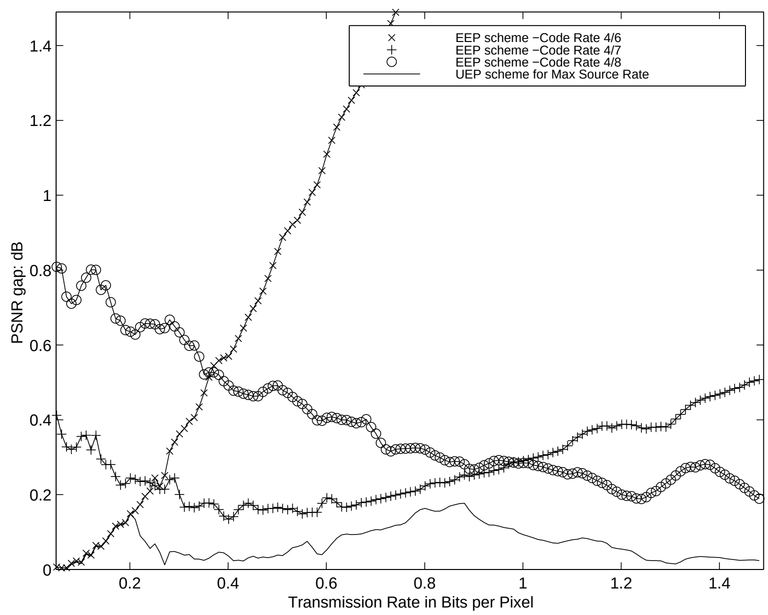

Figure 4.8: The loss of PSNR in EEP schemes and optimal UEP scheme maximizing average useful source coding rate compared to the optimal UEP scheme maximizing PSNR for the image Lenna. Code family B, BER =0.01.

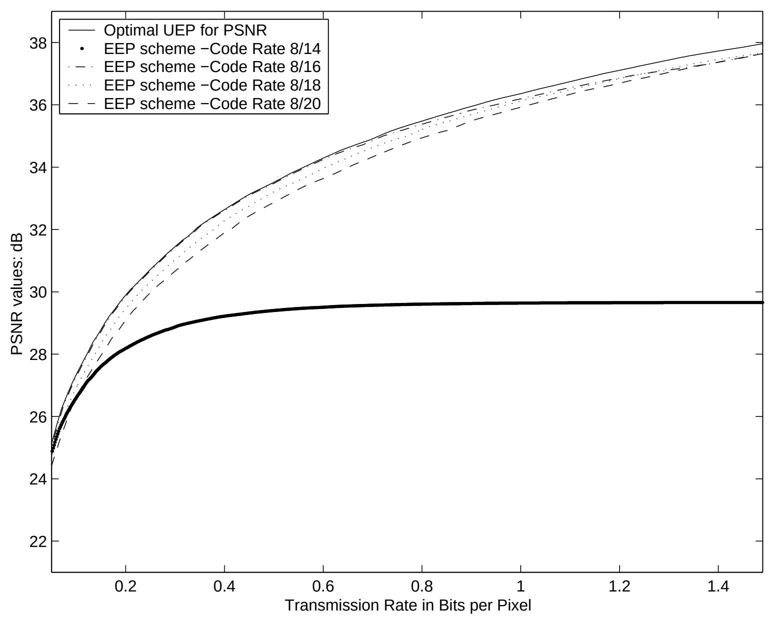

Figure 4.9: Average PSNR Performance of unequal error protection for memoryless channels for the image Lenna. Code family C, BER = 0.01.

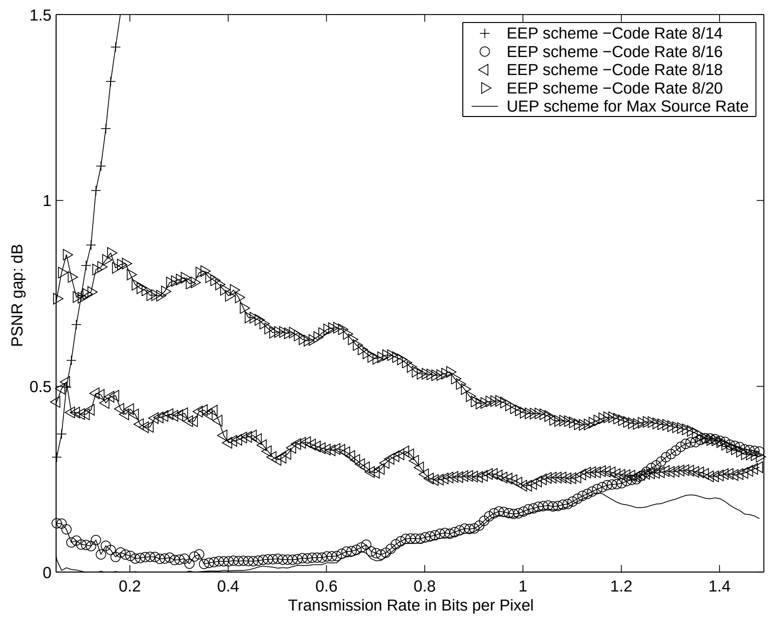

Figure 4.10: The loss of PSNR in EEP schemes and optimal UEP scheme maximizing average useful source coding rate compared to optimal UEP scheme maximizing PSNR, for the image Lenna. Code family C, BER =0.01.

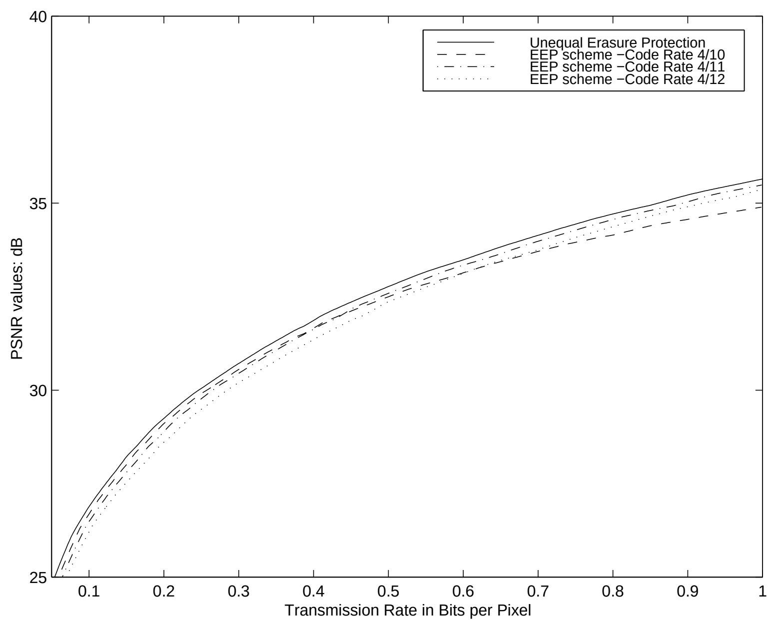

Figure 4.11: Average PSNR performance of EEP and the optimal UEP scheme for the Lenna image for memoryless packet erasure channels: packet size 8 bytes, erasure rate 20%.

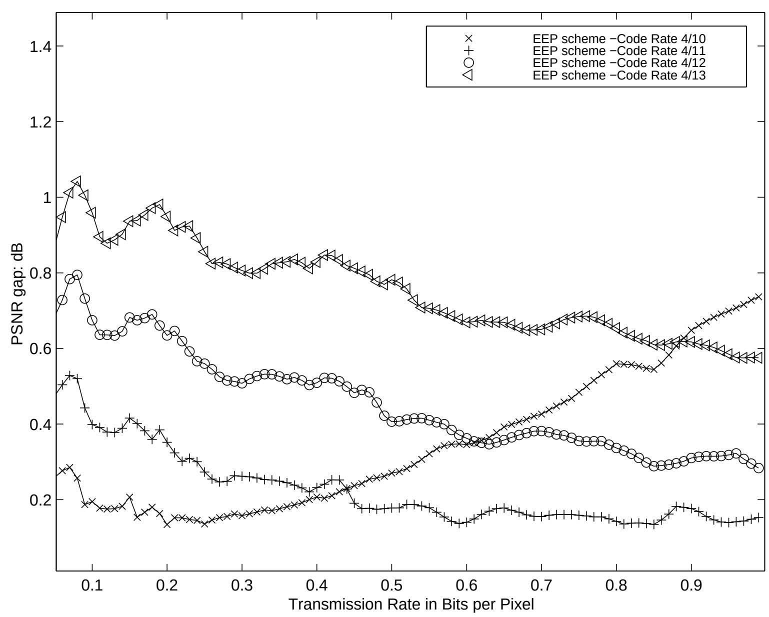

Figure 4.12: Average PSNR gain of the optimal UEP scheme over equal erasure protection schemes: memoryless erasure channels: packet size 8 bytes, erasure rate 20%.

Chapter 5

Progressive Image Transmission over Compound Packet Erasure Channels

5.1 Introduction

Embedded image coders like SPIHT [52] allow the user to reconstruct the image at different qualities from the prefixes of a single bit stream. Such image coders are useful in progressive reconstruction of the image, where the quality of the reconstructed image improves as more bits are added and decoded. Progressive reconstruction capability is desirable in many applications, *e.g.* fast browsing of image databases and multicasting to different users with varying channel usages. It is of interest to retain the progressive reconstruction property when such an image coder is used for transmission over a noisy or lossy channel such as a congested packet network or a wireless link in deep fade.

In this chapter, we undertake the design of a system for *progressive* image transmission over a lossy packet network with unknown packet-loss characteristics in the absence of any network layer loss recovery mechanism and feedback channel (*e.g.* transmission using User Datagram Protocol (UDP) or raw transmission of packets over ATM in unreliable mode). We select a high performance embedded image coder like SPIHT as the source coder. The objective is to design recovery from packet loss or erasures at the application level by the use of erasure correction codes while maintaining high average performance at each transmission budget. There are three issues that we hope to tackle.

- **Design of unequal erasure protection:** While using an embedded source coder like SPIHT, an irrecoverable loss of a source packet at the beginning of the stream is potentially more damaging than a loss near the end. This is so because a loss or corruption of the bits at the beginning of the bitstream can render all the subsequent bits of the source-coder useless. Hence there is a hierarchy of importance of the source-bits and a potential need for unequal erasure protection.

- **Combat against an unknown channel:** We consider transmission over a lossy network whose packet loss rate varies from session to session. We model such a network with unknown packet loss rate as a compound channel made of memoryless packet erasure channels. The determination of the optimal tradeoff between source coding and erasure protection is of interest in this situation.

- **Better Progressive Transmission:** To quantify the notion of progressivity of a joint source-channel coder, we must consider its performance at a given intermediate transmission budget compared to the performance of a joint source-channel coder optimized for that budget. Similar to the corresponding property in the source-coders like SPIHT, we would like the joint source-channel coder to have a performance that is close to optimal at all the intermediate transmission budgets. This requires not only the allocation of protection but also the scheduling of the source and the protection bits in the transmit bitstream.

There is a large body of work in the literature which addresses robust transmission of images over noisy or lossy channels. In the context of bit error channels, some of the techniques involve effective use of strong equal error protection ([57, 59]), unequal error protection ([63, 39, 1], [11] and Chapter 4), the use of feedback ([12, 40]), and better decoding schemes. For packet erasure channels, the techniques have been to construct robust packetized source coders for graceful degradation against packet loss [51], use of forward error correction [46] and multiple description coding (*e.g.* [53, 61]). The work in [18] provides a technique for progressive image transmission over a channel which has both bit errors and packet erasures. A new technique for combating packet erasures using erasure correction codes has been developed recently in [46]. They use unequal erasure protection using fixed block length Reed Solomon (RS) codes with variable number of source symbols in each codeword. They also use an interleaver to decorrelate the symbol erasures within a RS codeword. The general channel model in [46] can also be used for a compound channel discussed in this work.

Most of these coders are designed to maximize the performance at a given transmission budget. While some of them indeed use a high performance embedded source coder like SPIHT, they do not explicitly consider the performance at intermediate transmission budgets.

In this work we propose an algorithm which attempts to address the three issues discussed earlier simultaneously. The proposed algorithm uses a variable block length, fixed source-length family of erasure correction codes obtained by puncturing a low code rate mother RS code for the unequal erasure protection. It is a greedy non-iterative suboptimal algorithm that obtains an allocation of unequal erasure protection for a higher transmission budget from an allocation designed for a lower transmission budget. It does this in such a way that the channel symbol-stream output by the coder for the lower budget is a prefix of that for the higher budget. It results in a bitstream with deferred transmission of redundancy - that is, the channel symbols in a codeword are not necessarily contiguous in the bitstream. Also, this allows the possibility that an erasure that is irrecoverable in the beginning becomes correctable as the transmission rate is increased. By design, it yields a progressive stream which has a good performance at a number of intermediate transmission budgets.

An interleaving structure similar to [46] is constructed in order to match the length of "packet" used for erasure protection to the actual packet length used in the network. The algorithm together with the interleaver yield the transmit bitstream for the network.

Simulation results show that for compound channels such an unequal erasure protection scheme outperforms equal erasure protection schemes at all transmission rates.

The structure of the chapter is as follows. Section 5.2 defines the compound channel. In Section 5.3 the transmission scheme is described. In Section 5.4 the performance measure is computed and the optimization problem is set up. Section 5.5 describes the algorithm. Simulation results are presented in Section 5.6. Section 5.7 discusses the structure of an interleaver that can be used with the output of the algorithm to yield transmission schemes which use a larger packet size. Section 5.8 is the concluding section.

5.2 Compound Packet Erasure Channels

We assume that the bitstream generated by the application is transmitted in fixed length packets over the network. In the presence of network congestion, some of these packets may be lost. If we assume that the fixed length packets arrive at the receiver (the decoder in the application) in the same order and that the location of the lost packets is known, then, the application sees the end-to-end equivalent channel as a packet erasure channel. The packet erasures seen by the application may be independent or correlated, and the net packet loss rate may vary from session to session depending on the network congestion. The situations of unknown packet loss rate, mismatched packet loss rate or slow variability of the packet loss rate can be approximately modeled as a compound packet erasure channel.

A compound packet erasure channel is a channel whose packet erasure rate is an unknown random variable with a known probability distribution. It is described by a set of states $s \in \mathcal{S}$, with associated probability mass function f^s. In each state $s \in \mathcal{S}$ the *channel is memoryless* with an associated packet erasure rate $e(s)$. The state is chosen at the beginning of the transmission session according to probabilities f^s and it is assumed that conditioned on the state, during the entire transmission session, the packet erasures are independent and identically distributed.

5.3 Transmission Scheme

It is necessary to employ an embedded source-coder to achieve progressive transmission. Often the output of an efficient embedded source coder like SPIHT is a very sensitive bitstream in which bits coming later in the bitstream can only be used if all the previous bits are available. Any loss of source bits early in the stream can render all the subsequent source bits useless for image reconstruction. The main design challenge while using an embedded source-coder over a packet erasure channel is to avoid or else delay any irrecoverable loss of the source bits in the source bitstream. We accomplish this by the use of erasure correcting codes. The scheme works as follows.

Consider an embedded source coder which simultaneously encodes N_S source samples. Its output, the source-encoder-bits, is packetized into *fixed-length* source-packets of l_s bits each. As the source coder is embedded, the representation of the source at rates which are multiples of l_s/N_s, can be obtained from a prefix of the stream of source-packets output by the source coder.

Let l denote the length in bits of the packets used over the channel. We assume that l divides l_s. Let $k_0 = l_s/l$ denote the number of packets that fit into a source-packet. For erasure protection, we use a family of (n, k_0) packet erasure correcting (PEC) codes obtained by puncturing a mother Reed Solomon code (See Section 5.3.1) for different blocklengths n. A codeword of an (n, k_0) PEC codeword is n packets long. Because the RS codes are Maximum Distance Separable (MDS [67]), the performance of the code family does not depend on the puncturing tables used to generate the family. This family of punctured codes provides a selection of different code-rates, necessary for unequal erasure protection.

The transmission proceeds as follows. Each source-packet output by the source coder is encoded with a potentially different channel code, chosen from the family of codes according to some code-assignment policy. The joint source-channel coder generates a single stream of packets and transmits over the lossy network. Some of these packets are lost or dropped by the network. The receiver tries to recover the source-packets by forming the corresponding (partially erased) codewords of the PEC code. The receiver declares a *source-packet decoding failure* if the source-packet cannot be recovered from the unerased received packets. It is often reasonable to assume that, when using an embedded source coder like SPIHT, if a source-packet cannot be decoded successfully at the receiver, then the subsequent source-packets cannot improve the quality of the source. Hence, at any stage in the transmission (*i.e.* at any transmission budget), the source is reconstructed *only from the decoded bitstream up to the first source-packet that contains irrecoverable erasure*.

We use fixed-length source-packetization but we allow source-packets to receive a variable number of parity check packets, *i.e.* to have a variable-length erasure protection. The unit of erasure correction remains a fixed length packet (l bits). If this "logical" packet size l is different from the "true" transmission packet - (*i.e.* the length of the packet which is dropped by the network, or the packet whose erasures are modeled as a compound channel), the effect of this difference can be minimized by using a progressive interleaver, which is described in Section 5.7.

5.3.1 Packet Erasure Correcting Codes

Consider a compound packet erasure channel which erases packets of length $l = bm$ bits for some integers b and m. Consider RS codes over $GF(2^m)$. Each symbol in the RS code is m bits long. Then a (nb, k_0b) RS code, when transmitted uninterleaved, can correct $nb - k_0b$ symbol erasures, and hence $n - k_0$ packet erasures. Therefore a (nb, k_0b) RS code is a (n, k_0) PEC code. A PEC code of the same performance can also be obtained from b copies of a (n, k_0) RS code over $GF(2^m)$.

We assume that the channel code family consist of (nb, k_0b) RS codes for a fixed k_0 and different "blocklengths" n. Hence the source-packet size is $l_s \stackrel{def}{=} k_0mb$ bits (= k_0 packets). The maximum value of n is $\lfloor \frac{2^m}{b} \rfloor$. Note that the family, considered as a PEC code family, is rate compatible. Let us denote the bank of erasure protection codes by $\mathcal{C} = \{c_1, c_2, \ldots c_J\}$. If c is an (n, k_0)-PEC code in the family then let $\eta(c) = n$ denote the block length in number of packets for c.

Now consider a compound packet erasure channel with packet erasure probability $e(s)$ in state s. Then the probability of source-packet decoding failure for a (n, k_0) PEC-code c is computed as,

$$P_e^s(c) = \sum_{i=n-k_0+1}^{n} \binom{n}{i} e(s)^i (1 - e(s))^{n-i} \, \forall s \in \mathcal{S}.$$

In fact for $k_0 > 1$, the (n, k_0) PEC code is less efficient than an interleaved $(nb, k_0 b)$ punctured RS code. But this somewhat artificial construction of PEC code is chosen just to make the point that the proposed algorithm does not depend on the size of the Galois Field symbol or the relative size of the true channel packet and the logical channel packet used for code allocation.

5.4 Performance Criterion

If the first i source-packets are available, the source can be reconstructed to a rate il_s/N_S bits per source sample, where N_S is the number of source samples. Let $r_s \stackrel{def}{=} l_s/N_S$ be the rate in bits per sample per source-packet for the source.

The unequal protection for the source-packets is described by specifying a *code allocation policy*. A code allocation policy π allocates channel code $c_\pi^i \in \mathcal{C}$ to the i^{th} source-packet out of the source coder. A policy π is described by the number of source-packets transmitted ($N(\pi)$) and by specifying a sequence of channel codes $\{c_\pi^1, c_\pi^2, \ldots, c_\pi^{N(\pi)}\}$ for the sequence of source-packets. The normalized transmission rate (in channel bits per source sample) for the policy π is given by $\frac{M_T(\pi)l}{N_s}$, where $M_T(\pi)$ is the total number of packets used by policy π. It is computed as,

$$M_T(\pi) \stackrel{def}{=} \sum_{i=1}^{N(\pi)} \eta(c_\pi^i). \tag{5.1}$$

Several non-equivalent single-parameter criteria can be used to measure the performance of a code allocation policy (*e.g.* expected squared error distortion, expected Peak Signal to Noise Ratio (PSNR), or expected useful source-coding rate [11]). Without loss of generality we select the expected value of PSNR (measured in dB) as the performance criterion.

Consider the transmission of the image over the compound channel using a code allocation policy π. To compute the performance of a policy π over a compound channel, define, for each state $s \in \mathcal{S}$ of the channel, real numbers $\beta^0(\pi, s) \stackrel{def}{=} 1$ and

$$\beta^i(\pi, s) \stackrel{def}{=} \prod_{j=1}^{i}(1 - P_e^s(c_\pi^j)) \text{ for } i = 1, 2, \ldots N(\pi).$$

$\beta^i(\pi, s)$ represents the probability that the first i source-packets are successfully decoded by the receiver given that the compound channel is in state s and policy π is executed.

Let the operational PSNR-rate performance of the source coder for the source image is given by $PSNR(r)$ where r is the rate in bits per sample. Then as the the source is reconstructed only from the source-packets received prior to a source-packet decoding failure, the *expected PSNR* for the policy π is given by,

$$\begin{aligned}\overline{PSNR}_\pi \stackrel{def}{=} & \sum_{s \in S} f^s \Big(PSNR(0) P_e^s(c_\pi^1) \\ + & \sum_{i=1}^{N(\pi)-1} PSNR(ir_s) P_e^s(c_\pi^{i+1}) \beta^i(\pi, s) \\ + & PSNR(N(\pi) r_s) \beta^{N(\pi)}(\pi, s) \Big). \end{aligned} \quad (5.2)$$

The code allocation problem for the joint source-channel coding scheme under the constraint of total transmission budget of R bits per source sample, can be written as follows.

$$\max_\pi \overline{PSNR}_\pi \text{ subject to } M_{T\pi} \leq M, \quad (5.3)$$

Here $M = \lfloor RN_s/l \rfloor$ is the equivalent constraint on the number of packets.

Under the transmission scheme, equation (5.2) can be converted to the following more convenient form. The principal idea is to write the objective function in the absence of loss, as a sum of incremental rewards, which are accumulated as each source-packet is successfully decoded by the receiver. Let δ_i denote the incremental reward when the i^{th} source-packet is successfully received. For average PSNR maximization,

$$\delta_i \stackrel{def}{=} PSNR(ir_s) - PSNR((i-1)r_s),\ i = 1, 2, \ldots. \tag{5.4}$$

Now, for a code allocation policy $\pi = \{c_\pi^1, c_\pi^2, \ldots c_\pi^{N(\pi)}\}$ and for integers $k, 1 \le k \le N(\pi)$, and for all channel states $s \in \mathcal{S}$ define,

$$\Delta^s(k,\pi) \stackrel{def}{=} \sum_{i=k}^{N(\pi)-1} \left(\sum_{j=k}^{i} \delta_j\right) \prod_{j=k}^{i} (1 - P_e^s(c_\pi^j)) P_e^s(c_\pi^{i+1}) + \prod_{j=k}^{N(\pi)} (1 - P_e^s(c_\pi^j)) \left(\sum_{j=k}^{N(\pi)} \delta_j\right). \tag{5.5}$$

Then the problem (5.3) reduces to solving the problem given by

$$\max_\pi \sum_{s \in \mathcal{S}} f^s\left(\Delta^s(k,\pi)\right) \text{ subject to } M_T(k,\pi) \stackrel{def}{=} \sum_{i=k}^{N(\pi)} \eta(c_\pi^i) \le M, \tag{5.6}$$

for $k = 1$.

5.5 Progressive Unequal Erasure Protection

Let the best policy designed by the algorithm for problem (5.6) for packet-constraint M be denoted by $\pi^*(M)$. Notice that just specifying the policy π for transmission does not completely describe the bitstream generated by the joint source-channel coder. It is also necessary to describe the order in which the packets corresponding to the codewords in the policy are transmitted. Though, for a compound channel, the performance of a policy at its transmission budget is not affected by the order of the packets, the performance of the system at intermediate budgets is definitely controlled by the order of the packets.

Consider two code allocation policies π_1 and π_2 with $M_T(\pi_1) < M_T(\pi_2)$. The necessary and sufficient condition for two policies π_1 and π_2 to allow progressive transmission is that for each source-packet i, the PEC codeword for π_1 be a punctured version of that for π_2, *i.e.* $\eta(c_{\pi_1}^i) \le \eta(c_{\pi_2}^i)$ [11]. Now, in order to obtain the performance of π_1 at the budget $M_T(\pi_1)$, first all the packets necessary for executing policy π_1 are transmitted. Next, by transmitting only the extra parity check packets needed to execute policy π_2, the performance of π_2 can be obtained at the budget $M_T(\pi_2)$ in the same stream. This way progressive scheduling of the packets is accomplished. In Chapter 4 Figure 4.3 shows progressive transmission using two policies π_1 and π_2. Note that the generated bitstream is such that, all the bits corresponding to a PEC codeword are not contiguous.

The proposed algorithm generates the best policy for packet-constraint M from the best generated policies $\pi^*(j)$ for $j < M$, in such a way that the resulting policies are embedded by design. Consider an intermediate stage in image transmission. After transmitting the packets corresponding to any policy π, the next transmission can consist of (i) transmitting additional parity-check packets for source-packets transmitted earlier or, (ii) transmitting packets for the new $(N(\pi) + 1)^{st}$ source-packet. Hence we can restrict our search of the best policy for packet-constraint M, to a union of (i) all policies which can be obtained by adding one packet to policy $\pi^*(M-1)$ and, (ii) all policies of packet-constraint M obtained by adding one *source-packet* to one of the policies $\pi^*(j)$ for $j < M$.

Consider the change in average total reward as a policy is changed by replacing a single channel code. It can be computed as follows. Let π be a code allocation policy. Let $g(\pi, i)$ denote the increase in the total reward, when an additional parity check packet corresponding to the i^{th} source-packet is transmitted. Let π' denote the new policy so obtained. Let c be the channel code with parameters $(\eta(c^i_\pi) + 1, k_0)$. Then $g(\pi, i)$ is given by,

$$\begin{aligned} g(\pi, i) &\stackrel{def}{=} \sum_{s \in \mathcal{S}} f^s \left(\Delta^s(1, \pi') - \Delta^s(1, \pi)\right) \\ &= \sum_{s \in \mathcal{S}} f^s \beta^{i-1}(\pi, s)(P^s_e(c^i_\pi) - P^s_e(c))(\delta_i + \Delta^s(i+1, \pi)). \end{aligned} \tag{5.7}$$

Here $\Delta^s(k,\pi) \stackrel{def}{=} 0$ if $k > N(\pi)$.

Similarly, let π'' be a policy obtained by adding an additional *source-packet* encoded with some channel code $c \in \mathcal{C}$. Let the change in objective function be denoted by $h(\pi, c)$. Then $h(\pi, c)$ is computed as ,

$$h(\pi, c) \stackrel{def}{=} \sum_{s\in\mathcal{S}} f^s \left(\Delta^s(1,\pi'') - \Delta^s(1,\pi)\right) = \sum_{s\in\mathcal{S}} f^s \beta^{N(\pi)}(\pi, s)(1 - P_e^s(c))\delta_{N(\pi)+1}. \tag{5.8}$$

From these two results, the following greedy and suboptimal but progressive unequal erasure protection (PUXP) can be derived for computation of $\pi^*(M_0)$ for some final packet constraint M_0.

Algorithm 3 (Progressive Unequal Erasure Protection (PUXP))

1. *Initialization: For some* $c_0 \in \mathcal{C}$, *Set* $\pi^*(\eta(c_0)) = \{c_0\}$ *and* $M = \eta(c_0) + 1$.

2. *Given designed policy* $\pi^*(M-1)$, *compute*

$$G(M) \stackrel{def}{=} \max_{i=1,N(\pi^*(M-1))} g(\pi^*(M-1), i) \tag{5.9}$$

3. *Given designed policies* $\pi^*(j)$ *for* $\eta(c_0) \le j < M$, *compute*

$$H(M) \stackrel{def}{=} \max_{c\in C} h(\pi^*(M - \eta(c)), c) \tag{5.10}$$

4. *If* $G(M) > H(M)$ *and* i *is the source-packet index achieving the maximum in (5.9) then the policy* $\pi^*(M)$ *is obtained by adding the extra parity check packet to the code* $c^i_{\pi^*(M-1)}$.

5. *If* $G(M) < H(M)$ *and* c *is the code achieving the maximum in (5.10) then the policy* $\pi^*(M)$ *is obtained by adding the extra code word for the next source-packet to policy* $\pi^*(M - \eta(c))$.

6. If $M = M_0$ stop, else increment M.

The initialization step can be any arbitrary policy. The good policies can be "grown" from any initial policy. The policy at each transmission rate is obtained by adding extra parity packets to a lower rate policy. Hence the performance at any target transmission rate can be obtained by progressive transmission through a sequence of policies at lower transmission rates. The algorithm therefore, generates the code allocation as well as specifies the scheduling of the packets in the packet stream.

5.6 Results

As an illustration, Figures 5.1-5.3 refer to the simulation results for a compound packet erasure channel with a packet length of 8 bytes. The channel is a 7-state model with packet loss rate vector $[0.0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4]$. The probability vector for these states is chosen as
$[0.05, 0.05, 0.15, 0.15, 0.25, 0.20, 0.15]$. Though the mean packet loss rate is near 0.2, with probability 0.40 the packet loss rate is lower than 0.2 and with probability 0.35 it is higher.

Simulations were conducted on this channel for transmission of 512×512 grayscale Lenna compressed with the SPIHT coder with arithmetic coding. The channel code family is $(n, 4)$ PEC codes for 8-byte packets, derived from $(255, 32)$ RS code over $GF(2^8)$. We assume that the packets arrive in sequence. Though the sequence number information was not encoded and is not reflected in the rate, we assume that the location of the lost packets is known. (A fixed size sequence number scales the Transmission rate axis by a fixed factor.)

Figure 5.1 compares the mean PSNR in dB for the given channel for the PUXP scheme obtained by Algorithm 3 with that of Equal Erasure Protection (EEP) schemes derived from the same family of channel codes. Figure 5.2 provides the gain of PUXP scheme over EEP schemes for the same compound channel. Notice that the EEP schemes have a performance loss which varies with the transmission rates. The gain of PUXP is consistently above 0.4 dB for all EEP schemes and can be more, depending on the transmission rate considered. Also, no single EEP scheme is closest to the PUXP scheme at all transmission rates.

Figure 5.3 plots the inverse code-rate profile (the block length n of the $(n, 4)$ PEC code used for a 4-packet long source-packet), for different transmission rates of 1.0, 0.75, 0.5 and 0.25 bpp. Clearly, the profiles look very different from EEP schemes. Also, they satisfy the conditions of progressive transmission by design.

5.7 Progressive Interleaving for Packet Erasure Channels

The previous sections assume that the "true" packets (*i.e.* those whose loss is independent and identically distributed in each state of the compound channel,) are of same length as the logical packets used as units in erasure correction codes. Quite often this may not be true, *e.g.* when RS codes over $GF(2^8)$, with 8-bit long symbols are to be used with ATM packets of length 48 bytes. It is necessary to devise a scheme to pack the logical packets into the true packets without losing the benefits of progressive transmission. This can be accomplished by the use of interleavers.

Interleavers are used to convert a channel with memory into a channel with no apparent memory. In the context of image communication, an interleaver was used in [59] in conjunction with a product code consisting of RCPC-CRC codes and RS codes for transmission of images over fading channels. There the interleaver was employed to break correlated burst of symbol erasures. The ingeneous packetization and erasure-correction scheme in [46] can be interpreted as an interleaver to break a packet erasure channel with large packet size into another packet erasure channel with a smaller packet size suitable for use with the chosen RS codes.

In this section we consider how the progressive scheme designed for one packet size - which is typically determined by the code family - can be used over networks with larger "true" packet size.

Consider a memoryless packet erasure channel with "true" packet size of L bits. Also consider a (n, k_0) packet erasure correcting code derived from a RS code for a packet size of l bits, as discussed in section 5.3.1. For clarity, let us call the packet erasure channel an *L-packet erasure channel* and the packets of length L as *L-packets*. A codeword of the PEC code consists of n *packets* of length l bits. Let us assume that L/l is an integer. Then the memoryless L-packet erasure channel with L-packet erasure rate e is in effect, a packet erasure channel with correlated erasures and the mean packet erasure rate e.

One can make the following key observations. (i) Over all packet erasure channels of mean erasure rate e, the (n, k) PEC code has the least probability of failure if the packet erasures are independent. (ii) Suppose the packets are fit into L-packets and transmitted over a memoryless L-packet erasure channel. Then the erasures in two packets are independent if and only if they belong to different L-packets. (iii) Hence, the PEC code will perform the best over this channel, if each packet of its codeword belongs to a different L-packet. (iv) It does not matter how far apart the L-packets are so long as they are different.

Even in a compound L-packet erasure channel, the distance between two L-packets does not affect the performance of the PEC code if all its packets are in different L-packets. Therefore, given the output of a progressive transmission scheme designed for memoryless or compound packet erasure channels, a simple low-delay progressive interleaver can be designed using the following strategy.

- From the sequence of packets output by the progressive scheme, start filling a L-packet while observing that no two packets from the same PEC codeword (or equivalently, those corresponding to same source-packet) are put in the same L-packet.

- Maintain a list of partially filled L-packets and the indices of codewords whose packets occupy them. Put a packet into the earliest eligible L-packet. If none of the unfilled L-packets are eligible, put it in a new L-packet. Update the list of partially filled L-packets.

The filled L-packets are transmitted over the network sequentially. The number of unfilled packets to be maintained is indicative of the "distance" between transmission over memoryless sub-packet channel and interleaved memoryless L-packet channel. As outlined in Figure 4.3 the sequence of packets output by the progressive transmission schemes is such that, the packets belonging to same codeword are not necessarily contiguous. Hence they are already partially interleaved. This helps in reducing the number of unfilled L-packets during progressive interleaving.

As an illustration, progressive transmission scheme was designed for compound erasure channel with packet size 8 bytes, (*i.e.* the system depicted in Figures 5.1,5.3). The output was transmitted over a compound erasure channel with L-packet-size 48 bytes, using the interleaver suggested above. Figure 5.4 shows the number of unfilled L-packets to be maintained for transmitting image Lenna. It turns out that, though the

PEC-codewords are sometimes as long as 16 packets, (*e.g.* Figure 5.3), typically the number of unfilled L-packets remains below five. Hence the interleaved compound L-packet erasure channel closely approximates the compound packet erasure channel for which the joint source-channel coder was designed.

5.8 Conclusion

We design a progressive unequal erasure protection schemes for compound packet erasures channels where the packet loss is memoryless but the loss rate is unknown random variable with known statistics. The algorithm PUXP attempts to achieve good performance simultaneously for a number of transmission rates. It does so by performing both code allocation and scheduling of the packet stream. It is shown that such a scheme works well for all transmission rates compared to any EEP scheme.

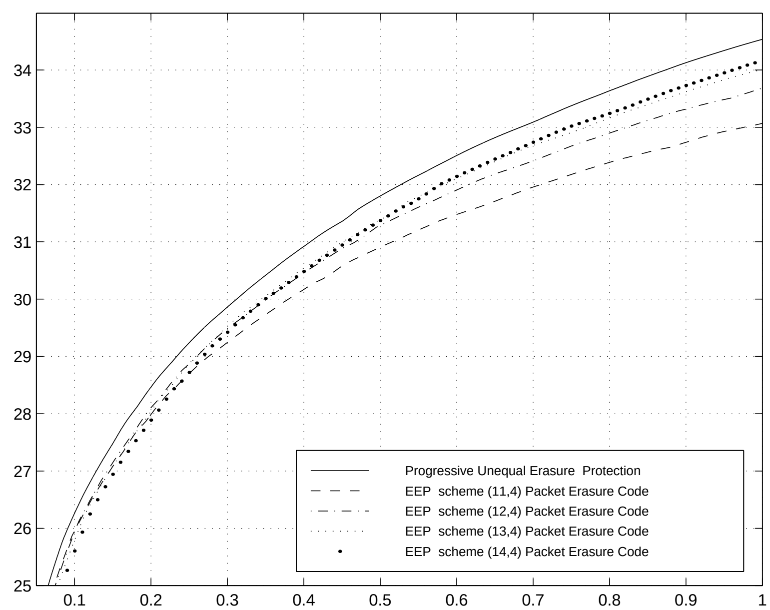

Figure 5.1: Average PSNR performance for image Lenna for Compound Erasure Channel: Packet Size 8 bytes

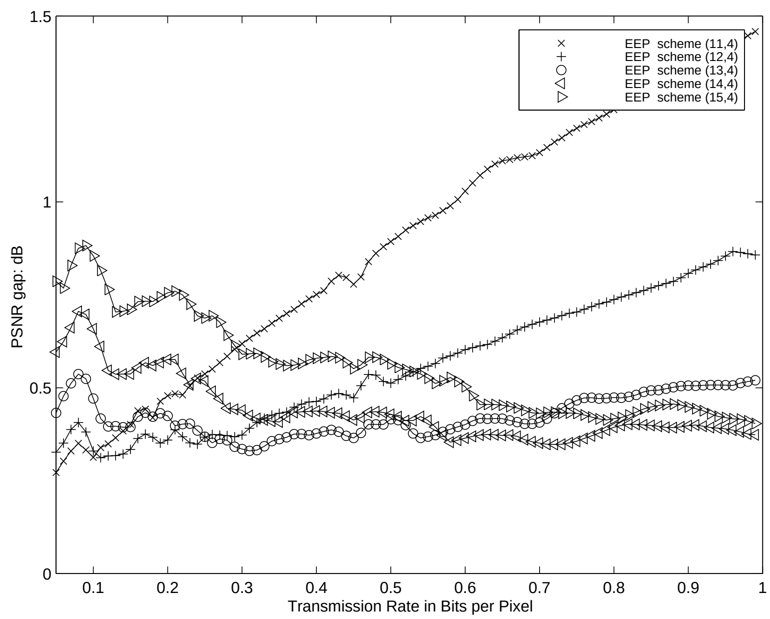

Figure 5.2: Average PSNR gain over Equal Erasure Protection Schemes for image Lenna for Compound Erasure Channel: Packet Size 8 bytes

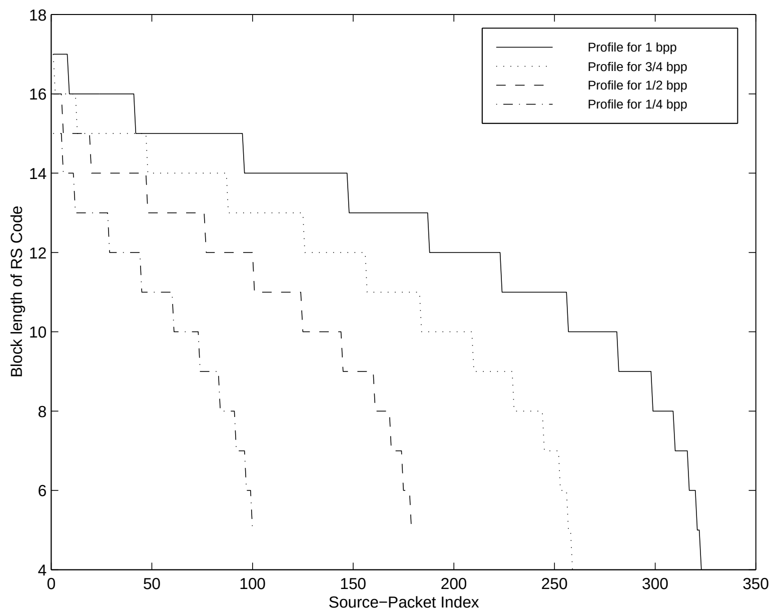

Figure 5.3: Inverse Code Rate Profile for the policy designed for Lenna by PUXP, for total rates 0.25, 0.5, 0.75 and 1.00 bpp. Compound packet erasure channel, Packet Size 8 bytes

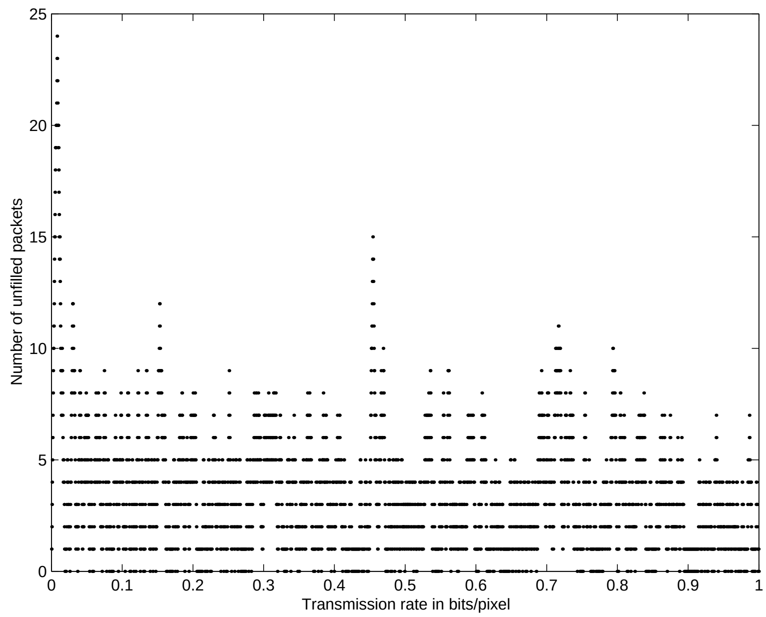

Figure 5.4: Progressive Interleaving: Number of unfilled 48-byte packets as a function of target rate. Sub-packet size = 8 bytes

Chapter 6

Source-Channel Decoding with Optimal Use of ACK/NACK Feedback

6.1 Reverting to First Principles

Decision Feedback (ACK/NACK) has been used extensively in communication situations where there is a feedback channel available from the receiver to the transmitter. Link layer protocols based on Automatic repeat query (ARQ) and combination of ARQ and Forward Error Correction (FEC), also called Hybrid ARQ, are used for data communication in a wireless environment. Feedback and retransmission is also used at the transport layer for end-to-end error recovery, *e.g.* in the TCP/IP protocol. Conventionally these protocols are designed for reliable transmission of data. The ACK/NACK generation is accomplished by an error-detection mechanism such as cyclic redundancy check (CRC) or bounded distance decoding. Protocols designed for data transmission attempt to trade the probability of undetected bit errors with the average code-rate or throughput.

A more meaningful performance measure for digital transmission of multimedia sources such as images, video and audio, is a distortion metric such as squared error. In this chapter we investigate how a *distortion metric can be incorporated into the design* of a transmission system for a loss tolerant source, which uses an ARQ or Hybrid ARQ protocol over a noisy channel allowing and ACK/NACK feedback.

In the earlier chapters we focused on the use of the feedback channel primarily for design of smart error control techniques useful for progressive transmission of the source. In this chapter we reverts *to first principles in its formulation and design methodology*. This first principles approach involves viewing source-encoding as *quantization* followed by *index assignment* and decoding as reproduction of the source from the received, possibly corrupted, information. This approach has been at the focus of joint source-channel coding research since its beginning. All the previous work, which includes smart source encoding (*e.g.* [23]), smart index-assignment (*e.g.* [34]) and smart decoding (*e.g.* [30]), concerns transmission of loss tolerant sources in the absence of a feedback channel.

This chapter is part theoretical and part experimental investigation of the effective use of ACK/NACK feedback, primarily on the receiver side when the objective is to obtain the best trade-off between the transmission rate and the distortion at the receiver.

In Sections 6.2, through 6.6, we formulate the problem of design of joint source channel coding in the presence of ACK/NACK feedback in its generality, from the first principles. In sections 6.7 through 6.12 we solve the decoder design problem for a pure ARQ system over a memoryless channel with packet combining at the receiver. In chapter 7 we show an interesting property of the decoder structure in a slightly more general scenario.

6.2 General Formulation for a System with ACK/NACK Feedback

A general point-to-point discrete time communication system for transmission of a loss tolerant source using ACK/NACK feedback can be described by the block diagram in Figure 6.1.

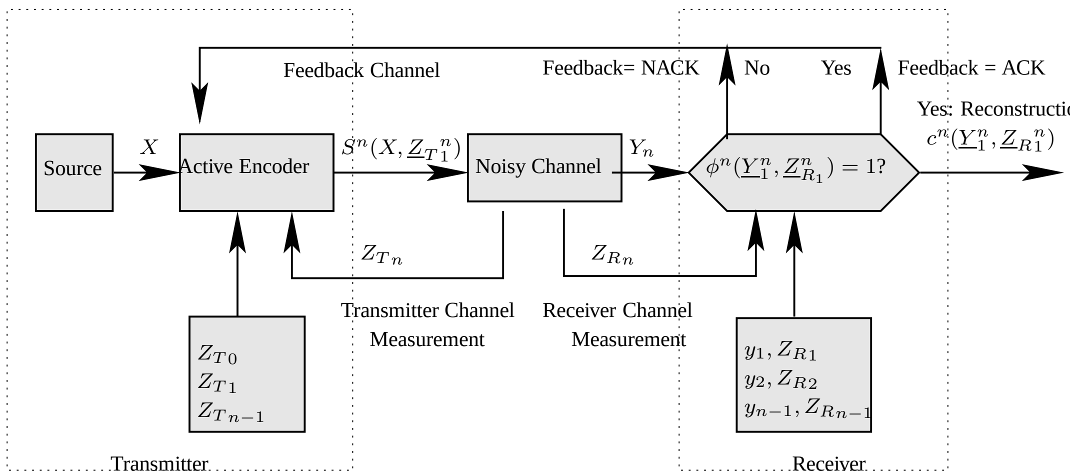

Figure 6.1: General JSCC system with ACK/NACK feedback at n^{th} step in transmission

The source is a random vector of fixed dimension and known statistics taking values in finite dimensional real valued space denoted by $\mathcal{X} \subset \mathcal{R}^k$. The encoding and transmission of this vector from the transmitter to the receiver takes place in several steps. In each step some channel symbols are transmitted from the transmitter to the receiver over the noisy channel. A general discrete time noisy channel takes channels symbols from input alphabet $\mathcal{I}$ and generates received symbols from output alphabet $\mathcal{Y}$.

A feedback $F \in \{ACK, NACK\}$ is transmitted from the receiver to the transmitter over the feedback channel at the end of every step. The transmission for the $n+1^{th}$ step may take place only if the feedback after n^{th} step was a NACK.

The random variables Z_{Tn} and Z_{Rn}, taking values in alphabet $\mathcal{Z}$, represent the "transmitter channel measurement" and the "receiver channel measurement" respectively, which may be available at the two ends as additional information about the channel. This information can be assumed to be uncorrelated with the source X. Note that, for analysis, the receiver channel measurement Z_{Rn} *can be omitted without loss of generality* as it can be included with the received noisy symbols Y_n as a combined received information.

The transmitter can be described mathematically by an *encoding rule* $\mathbf{S}$ which is a sequence of integers $l_n \geq 0,\ n = 1, 2, \ldots$, and a sequence of encoder maps $S^n : \mathcal{X} \times \mathcal{Z}^n \rightarrow \mathcal{I}^{l_n}, n\ = 1, 2, \ldots$. On receiving NACK at $n - 1^{th}$ step, the transmitter sends l_n symbols given by computing $S^n(X, Z_{T1}, Z_{T2}, \ldots, Z_{Tn})$ over the noisy channel at the n^{th} step. This vector of channel symbols (also called *channel codeword, transmit codeword or n^{th} step codeword*) is corrupted by the channel and is received as random vector Y_n taking values in $\mathcal{Y}^{l_n}$. We will say that Y_n is *received codeword* at the n^{th} step. For simplicity, with no loss of generality, we shall assume that $l_1 = l_2 = \ldots = l_n = \ldots = L$, *i.e.* exactly L symbols are transmitted over the noisy channel between two feedbacks. L is the *packet length* or the *codeword length*.

The receiver is described by the *feedback generation rule* and the *reproduction rule* . The feedback generation rule $\boldsymbol{\phi}$, is a sequence of *feedback generations maps* $\phi^n : \mathcal{Y}^{nL} \rightarrow \{0, 1\}, n = 1, 2, \ldots$. At the n^{th} step, let the realizations of the received codewords be $y_1, y_2, \ldots, y_n$ for $y_i \in \mathcal{Y}^L$. Then an ACK is transmitted over the feedback channel if $\phi^n(y_1, y_2, \ldots, y_n) = 1$. A NACK is transmitted if $\phi^n(y_1, y_2, \ldots, y_n) = 0$. For mathematical convenience we also define the constant "function" ϕ^0 which is either 0 or 1.

We assume here and the rest of the thesis that the feedbacks ACK/NACK is instantaneous and error free.[1]

The *reproduction rule* $\mathbf{c}$ is a sequence of *reproduction maps* $c^n : \mathcal{Y}^{nL} \rightarrow \mathcal{C} \subset \mathcal{X}$ for $n = 1, 2, \ldots$. $\mathcal{C}$ is the *reproduction codebook*. For mathematical convenience, define constant function $c^0 \in \mathcal{C}$. If an ACK is generated at the n^{th} step, *i.e.* if $\phi^n(y_1, y_2, \ldots y_n) = 1$, then the source is reconstructed as $c^n(y_1, y_2, \ldots, y_n)$. It is not necessary for $\mathcal{C}$ to be discrete.

We will be using the shorthand notation $\underline{y}_1^n$ and $\underline{Y}_1^n$ for denoting the sequences $y_1, y_2, \ldots, y_n$ and random vectors $Y_1, Y_2, \ldots, Y_n$, respectively. Similarly, $\underline{Z}_{T1}^{\ n}$ denotes the sequence $Z_{T1}, Z_{T2}, \ldots, Z_{Tn}$.

The noisy channel is assumed to be independent of the source vector. The channel can be described by (i) the joint distributions of transmitter channel measurement $F^n_{Z_T}(\underline{Z}_{T1}^{\ n}), n = 1, 2, \ldots$ (ii) transition probabilities, which are conditional probability density functions of vectors $f_{\underline{Y}_1^n|\underline{Z}_{T1}^{\ n}, \underline{I}_1^n}(\underline{y}_1^n|\underline{z}_1^n, \underline{i}_1^n), n = 1, 2, \ldots$ for $y_n \in \mathcal{Y}^L$ and $i_n \in \mathcal{I}^L$, satisfying appropriate consistency conditions on marginal distributions.

6.3 Performance Measurement

The simplest performance measures for loss tolerant systems are the distortion and the transmission rate. The transmission rate is the average channel usage per source sam-

[1]Though this assumption is limiting, it is made to simplify our investigation of design of feedback based JSCC systems and evaluation of their relative merits over systems not using feedback, without getting sidetracked. Some effect of delay can be mitigated by the use of buffers at the transmitter and the receiver along with “selective-repeat” strategy. As ACK/NACK feedback requires very low data rate on the feedback channel, it can be protected by strong error correction and can be reasonably assumed to be error-free.

ple.[2] Distortion measures the separation between the original source vector X and its reproduction at the receiver $\hat{X}$. We shall assume squared error distortion measure throughout, *i.e.* $d(X, \hat{X}) = \|X - \hat{X}\|^2$.

As discussed earlier, the receiver channel measurement need not be explicitly mentioned and will be omitted in the rest of the discussion. Note that the source is reproduced at the n^{th} step only if current step generated ACK and previous $n-1$ steps resulted in NACK - *i.e.* $\phi^i(\underline{y}_1^i) = 0$ for $i = 1, 2, \ldots n-1$ and $\phi^n(\underline{y}_1^n) = 1$. Define the function $\psi^n(\underline{y}_1^n) \stackrel{def}{=} \prod_{i=0}^{n-1}(1 - \phi^i(\underline{y}_1^i))\phi^n(\underline{y}_1^n)$. It is straightforward to show that the average distortion for a given transmitter, receiver and channel can be computed as,

$$D(\boldsymbol{\phi}, \mathbf{c}) = E\left[\sum_{n=0}^{\infty} d(X, c^n(\underline{Y}_1^n))\psi^n(\underline{Y}_1^n)\right]. \tag{6.1}$$

For clarity let us write down the expectation calculations explicitly.

$$D(\mathbf{S}, \boldsymbol{\phi}, \mathbf{c}) = \int_{\mathcal{X}} f_X(x) \left(\sum_{n=0}^{\infty} \left(\int_{\mathcal{Y}^{nL}} d(x, c^n(\underline{y}_1^n))\psi^n(\underline{y}_1^n) f_{\underline{Y}_1^n|X,S}(\underline{y}_1^n|x) d\underline{y}_1^n \right) \right) dx \tag{6.2}$$

where

$$f_{\underline{Y}_1^n|X,S}(\underline{y}_1^n|x) \stackrel{def}{=}$$
$$\int_{\mathcal{Z}^n} f_{Z_T}^n(\underline{z}_{T1}^{\ n}) f_{\underline{Y}_1^n|\underline{Z}_{T1}^{\ n}, \underline{I}_1^n}(\underline{y}_1^n|\underline{z}_{T1}^{\ n}, S_1(\underline{z}_{T1}^{\ 1}, x), S_2(\underline{z}_{T1}^{\ 2}, x), \ldots S_n(\underline{z}_{T1}^{\ n}, x)) d\underline{z}_{T1}^{\ n} \tag{6.3}$$

define the "effective" transition probabilities as seen by the receiver.

[2] Transmission rate is expressed in channel symbols per source sample. Transmission rate should not to be confused with the channel baud rate in symbols per second - which is a property of the modulation-demodulation system, or the channel coding rate or channel throughput, which is dimensionless. For a fixed (time invariant) quantizer channel coding rate or throughput is inversely proportional to the transmission rate.

Similarly, the expected transmission rate, which is proportional to expected value of stopping time, is given by

$$R(\mathbf{S}, \boldsymbol{\phi}) = E\left[\sum_{n=0}^{\infty} nL\psi^n(\underline{Y}_1^n)\right]. \tag{6.4}$$

6.4 Classification of the Transmitters

The transmitter, or more specifically, each encoder map S^n can be conceived as a composition of two maps, namely a *quantizer* $Q^n : \mathcal{X} \rightarrow \mathcal{N}$ and an *index-assignment* $b^n : \mathcal{N} \rightarrow \mathcal{I}^L$. The quantizer divides the source space $\mathcal{X}$ into a finite number of partitions and the index-assignment map assigns a unique vector of channel symbols to each partition. The index assignment may include explicit or implicit redundancy for the purpose of error control coding. The transmitter can be classified into three categories based on how the quantizer and the index assignment map change at each step.

1. **Active Encoder (Embedded source coding/multiple description based source coding + Hybrid ARQ) :** We say that the encoder at the transmitter is an *"active encoder"* (Figure 6.2), if both the quantizer and the index assignment are time varying, *i.e.* are allowed to vary at each step in transmission. A quantizer changing with n can be thought of as an embedded source coding because the partition of $\mathcal{X}$ after n^{th} step is a refinement of the partition obtained up to step $n-1$. It can also be conceived as Multiple Descriptions as the individual quantizers $Q^n, n = 1, 2, 3....$ are different descriptions of the source transmitted at different times. Clearly, this kind of encoding allows the *source distortion to diminish to arbitrarily small value*.

2. **Incremental Redundancy Transmission or general Hybrid ARQ:** When the quantizer map is fixed (*i.e.* time-invariant) and only the index assignment map

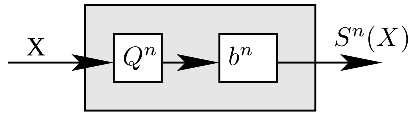

Figure 6.2: Active encoder at n^{th} step

varies with each step, the encoder implements incremental redundancy transmission or Type III hybrid ARQ (Figure 6.3). A protocol analogous to this was considered in Chapter 2. The advantage of this configuration is that the source coding can be separated from the transmission protocol. On the other hand, the drawback over the more general encoder is that the *distortion at the receiver is limited by the quantizer induced distortion* and it cannot be driven to zero no matter how well the channel behaves or how efficient the error control scheme is.

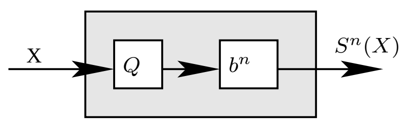

Figure 6.3: System with incremental redundancy transmission *e.g.* using RCPC codes

3. **Passive Encoder/Pure Retransmission Encoder/ Type I Hybrid ARQ:** The simplest system using ACK/NACK feedback is one in which the source coding and the index assignment are time invariant. On receiving a NACK, the transmitter retransmits a copy of the same codeword. In such a case we say that the encoder (or the transmitter) is *"passive"*. This is attractive because it is simple. But it does not make use of the feedback channel in the best possible way at the transmitter side.[3]

[3]We use the term "passive encoder" because the term Type I HARQ also puts restrictions on the receiver side.

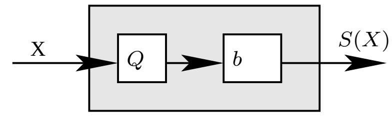

Figure 6.4: Passive Encoder for any step

6.5 Decoder Structure

The receiver or the decoder can be classified analogously based on degrees of freedom, complexity and memory usage. Note that the decoder consists of the feedback generation rule and the reproduction rule. The simplest form of decoder, the *Type I Hybrid ARQ* decoder, uses only the current observation for generating a feedback *i.e.* the feedback generation map ϕ^n does not depend of $\underline{y}_1^{n-1}$. Type I Hybrid ARQ decoder has low computational and memory requirements but it does not make use of the full potential of ACK/NACK feedback.

The more general decoder, at n^{th} step, can use all the received codewords up to the step n in generating ACK/NACK feedback. Its general structure is show in Figure 6.5. If the encoder is active, such a decoder is said to be doing *code-combining* and If the encoder is passive, the decoder is said to be doing *packet-combining*. We will be using the term code-combining decoder to denote both decoders.

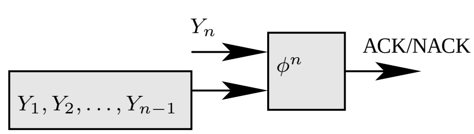

Figure 6.5: Code Combining or Packet Combining

Clearly, a code-combining decoder is more complex and has larger and variable memory requirements. The memory requirements can be reduced if the decoder, instead of storing all the received codewords, can store only a “state” estimated from the past received codewords. The decoder structure with this decomposition is depicted in Figure 6.6. Note that the “state” need not be a sufficient statistic. It may be used only to impose additional structure on the feedback generation maps.

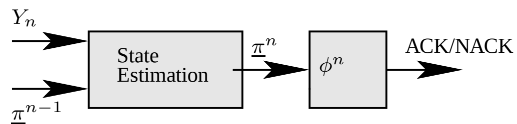

Figure 6.6: Code Combining or Packet Combining with State Estimation

6.6 Decoder Design

Having described these concepts about the transmitter and receiver sides, we embark on a topic that forms the Sections 6.7 and Chapter 7, namely the design of the decoder. We focus on the decoder (the feedback generation rules and the reproduction rules) in the rest of the thesis. It can be argued that the design of the decoder must precede the design of the encoder. We shall see that, systems which use ACK/NACK feedback are primarily receiver driven. Even in scenarios involving a passive transmitter, by letting the feedback generation maps change, the receiver can exercise a lot of control over the end-to-end performance of the system.

Nevertheless, design of the transmitter side remains and interesting and important issue that we do not address in this thesis.

Notice that the decoder performance (eqs. (6.1) and (6.4) as functions of $(\boldsymbol{\phi}, \mathbf{c})$) and depends on the encoding rule only through the effective transition probabilities (eq. 6.3) We shall assume in the rest of the chapter that the effective transition probabilities given by eq. (6.3) are known at the receiver.

In Sections 6.7 through 6.13 we restrict our attention to the design of an optimal decoder for *a passive encoder* and *a memoryless noisy channel,* when the number of steps is allowed to be unbounded. The optimality criterion is the tradeoff between end-to-end distortion and transmission rate. We obtain the optimal design and also propose some suboptimal but competitive, computationally simpler decoder designs.

In the next chapter, Chapter 7 we focus on decoder design when the encoder is active but predesigned, under the constraint that the maximum number of steps is bounded. We draw parallels between source-channel coding with ACK/NACK feedback and Pruned Tree Structured Vector Quantization. We also analyze the decoder structure and show that the optimal feedback generation rules are embedded in a special sense. This property of embeddedness has applications in progressive transmission.

6.7 Packet Combining for Joint Source-Channel ARQ over Memoryless Channels

In Sections 6.7 through 6.13 we restrict our attention to a passive transmitter scenario, where, on receiving a NACK, *the transmitter can only do a retransmission of the codeword* earlier transmitted (*i.e.* the scenario of Figure 6.4). On the other hand, we look at an active-receiver system in which the receiver retains all the (noisy) copies of the received codeword and can use them for generation of the next feedback or reproduction of the source. This is analogous to *packet combining* or *diversity combining* in the context of data transmission [67]. Clearly, as the encoder is passive, any transmitter channel measurement is not used, and we shall assume in these sections that the *transmitter channel measurement is absent*.

We show that the task of designing a source-channel feedback generation rule for packet combining based ARQ can be mapped to a classical sequential decision problem [24]. Consequently we obtain a dynamic programming based solution for the optimal feedback generation rule and reproduction rule so as to minimize a Lagrangian sum of rate and distortion. We shall see that the distortion metric plays an important role, not only in the source reproduction, but also in the feedback generation. As the optimal solution is computationally complex, we also suggest simpler alternatives for feedback generation. Results indicate that they also outperform schemes not incorporating the distortion metric.

6.8 Transmission Scheme and Notation

The transmission protocol we consider is most generally described as Type I Hybrid ARQ with packet combining (*e.g.* [67]) at the receiver. As earlier, consider the transmission of a k dimensional random source-vector X taking values in $\mathcal{X} \subset \mathcal{R}^k$, over a memoryless noisy channel with discrete input alphabet $\mathcal{I}$, possibly continuous valued output alphabet $\mathcal{Y}$ and known transition probabilities. The source-vector is quantized by a fixed, pre-designed k dimensional vector quantizer (VQ) with M cells.

Each VQ cell is assigned an L dimensional channel-codeword (or *"packet"*) by a fixed, pre-designed channel coding scheme. As the encoder is assume to be passive, let $S : \mathcal{X} \rightarrow \mathcal{I}^L$, denote the (fixed) map for the codeword assignment. Note that the map includes quantization, index-assignment and channel coding, if any. Therefore, for a realization x of random vector X, $S(x)$ denotes the codeword to be transmitted over the channel. $S(x)$ takes M possible values denoted by S_i, $i = 1, 2, \ldots, M$, in $\mathcal{I}^L$. The transmission proceeds as follows. Codeword $S(X)$ is transmitted and a feedback of ACK/NACK is requested. On receiving NACK, a *copy of* $S(X)$ is retransmitted. This is continued until an ACK is received. At the end of n^{th} retransmission, the receiver uses all the available noisy copies Y_i of $S(X)$ to generate the ACK/NACK feedback. As the channel is assumed to be memoryless, Y_i for $i = 1, 2, \ldots$, are statistically independent given the codeword $S(X)$.

As the encoding rule is fixed, we shall drop the symbol $\mathbf{S}$ from the expressions of distortion $D(\mathbf{S}, \boldsymbol{\phi}, \mathbf{c})$ and rate $R(\mathbf{S}, \boldsymbol{\phi})$ in equations (6.1) and (6.4) for the subsequent sections.

6.9 Decoder Design Problem

For the general system described in Section 6.2, the quantizer, the assigned channel codewords as well as the decoder structure determine the average rate and distortion. For a fixed quantizer and channel codeword assignment, the general design problem is the minimization of a Lagrangian sum of the expected distortion $D(\boldsymbol{\phi}, \mathbf{c})$ and average rate $R(\boldsymbol{\phi})$ with respect to $\boldsymbol{\phi}$ and $\mathbf{c}$. Mathematically, for a non-negative Lagrangian penalty λ, the problem can be written as,

$$\min_{\boldsymbol{\phi}, \mathbf{c}} E\left[\sum_{n=1}^{\infty} \left(d(X, c^n(\underline{Y}_1^n)) + \lambda n L\right) \psi^n(\underline{Y}_1^n)\right]. \tag{6.5}$$

Let π_i^0 be the probability that the source vector lies in the i^{th} cell, *i.e.* $\pi_i^0 = \Pr(S(X) = S_i), i = 1, 2, \ldots, M$. Let $\underline{\pi}^0(\underline{y}_1^n) \stackrel{def}{=} \{\pi_i^0(\underline{y}_1^n), i = 1, 2, \ldots, M\}$.We restrict our attention to the squared error distortion measure, *i.e.* $d(X, c) = \|X - c\|^2 = (X - c)^T(X - c)$. Also let s_i denote the centroid of the i^{th} VQ cell, *i.e.* $s_i = E[X|S(X) = S_i]$. Then, under general conditions, we can write $D(\boldsymbol{\phi}, \mathbf{c})$ as,

$$D(\boldsymbol{\phi}, \mathbf{c}) = \underbrace{\sum_{i=1}^{M} \pi_i^0 E\left[\, \|X - s_i\|^2 \middle| S(X) = S_i\right]}_{D_s} +$$

$$\underbrace{\sum_{i=1}^{M} \pi_i^0 E\left[\sum_{n=1}^{\infty} \|s_i - c^n(\underline{Y}_1^n)\|^2 \psi^n(\underline{Y}_1^n) \middle| S(X) = S_i\right]}_{D_c(\boldsymbol{\phi}, \mathbf{c})} \tag{6.6}$$

where D_s, the distortion due to the vector quantizer, is a term independent of $\boldsymbol{\phi}$ and $\mathbf{c}$. Therefore the design problem reduces to the following.

$$\min_{\boldsymbol{\phi}, \mathbf{c}} J(\boldsymbol{\phi}, \mathbf{c}, \underline{\pi}^0, \lambda) \text{ where } J(\boldsymbol{\phi}, \mathbf{c}, \underline{\pi}^0, \lambda) \stackrel{def}{=} D_c(\boldsymbol{\phi}, \mathbf{c}) + \lambda R(\boldsymbol{\phi}). \tag{6.7}$$

For reasons soon to become clear, we have explicitly shown the dependence of the objective function on the prior probability vector $\underline{\pi}^0$.

6.10 Sequential Decision Problem

An examination of the expression for the objective function $J(\boldsymbol{\phi}, \mathbf{c}, \lambda)$ reveals that, $J(\boldsymbol{\phi}, \mathbf{c}, \underline{\pi}^0, \lambda)$ is the Bayesian risk in a classical sequential decision problem [24]. The corresponding terminology is as follows. The collection of VQ cells indices, $\{i = 1, 2, \ldots, M\}$ is the *parameter space*. π_i^0 is the *a priori* probability of parameter i used for computation of the Bayesian risk. Y_i's are the observation random variables which are conditionally independent and identically distributed, given the parameters. The set of reproduction vectors $\mathcal{C} \subset \mathcal{X}$ is the *action space*. The feedback generation rule $\boldsymbol{\phi}$ represents the *stopping rule*. A NACK feedback corresponds to a request for another observation. The reproduction vector map $c^n : \mathcal{Y}^{nN} \rightarrow \mathcal{C}$ is the *terminal decision rule*. The *loss function*, or penalty for taking an action $c \in \mathcal{C}$ when the parameter is i, is given by the squared error $\|s_i - c\|^2$. The increase in rate at a given step, λN, is the *cost of the incremental observation*.

Given this translation, the optimal joint source-channel decoder is the solution to the sequential decision problem given by eq. (6.7). The solution provides a feedback generation rule which explicitly considers the tradeoff between distortion and rate, and makes use of the available source statistics.

Notice that there is flexibility in choosing the reproduction vectors, *i.e.* the elements of reproduction codebook $\mathcal{C}$. If they are chosen as the the centroids of the source-encoder maps, *i.e.* if $\mathcal{C} = \{s_i, i = 1, 2, \ldots, M\}$, then the problem is a *M-ary sequential detection problem with Bayes penalties* $C_{i,j} = \|s_i - s_j\|^2$. This problem has been studied in the context of signal detection (*e.g.* [3]). The non-sequential analog in the context of joint source-channel coding has also been studied (*e.g.* [22]). (ii) A finite but densely populated codebook can also be used for reproduction. [23] consider such table-lookup codebooks for reproduction vectors in the non-sequential case. It can be seen that any Maximum A posteriori estimate of the source will lie in the convex closure of the centroids s_i of the source-encoder cells . Therefore, most generally, the set of reproduction vectors, the action space, should be the set of convex combinations of the centroids s_i. For our simulations we used the collection of all convex combinations of source-encoder centroids s_i as the reproduction codebook $\mathcal{C}$. This set includes the Minimum Mean Squared Error (MMSE) estimate of the source-encoder centroids.

6.11 Optimal Sequential Design

Let $\pi_i^n(\underline{y}_1^n)$ denote the posterior probability of codeword S_i given the observations $\underline{y}_1^n$. That is, $\pi_i^n(\underline{y}_1^n) \stackrel{def}{=} \Pr(S(X) = S_i|\underline{y}_1^n)$ for $i = 1, 2, \ldots, M$. Let $\underline{\pi}^n(\underline{y}_1^n) \stackrel{def}{=} \{\pi_i^n(\underline{y}_1^n), i = 1, 2, \ldots, M\}$. Let $f(y_n|S_i)$ denote transition probabilities for the codewords computed from the transition probabilities for the channel. Then for a given observation vector $\underline{y}_1^n$, the following relationship exists between $\underline{\pi}^n(\underline{y}_1^n), \underline{\pi}^{n-1}(\underline{y}_1^{n-1})$ and y_n,

$$\pi_i^n(\underline{y}_1^n) = \frac{\pi_i^{n-1}(\underline{y}_1^{n-1}) f(y_n|S_i)}{\sum_{j=1}^{M} \pi_j^{n-1}(\underline{y}_1^{n-1}) f(y_n|S_j)}. \tag{6.8}$$

Let this function, which is independent of time index n, be denoted by $H(\underline{\pi}, y)$. Then $\underline{\pi}^n(\underline{y}_1^n) = H(\underline{\pi}^{n-1}(\underline{y}_1^{n-1}), y_n)$.

Let Γ denote the simplex of all probability distributions over transmit codewords S_i, *i.e.* $\Gamma \stackrel{def}{=} \{a_1, a_2, , \ldots a_M : 1 \geq a_i \geq 0, \sum_{i=1}^{M} a_i = 1\}$. All posterior probability distributions $\underline{\pi}^n$ belong to Γ. Define function $\rho : \Gamma \rightarrow [0, \infty)$ as $\rho(\underline{\pi}) \stackrel{def}{=} \min_{c \in C} \sum_{i=1}^{M} \|s_i - c\|^2 \pi_i$.

We get the following main result from the theory of sequential decisions.

Proposition 5 *For every $\lambda \geq 0$, there exists a unique cost-to-go function $V(\cdot, \lambda) : \Gamma \rightarrow \mathcal{R}$ which satisfies the following dynamic programming equation for all $\underline{\pi} \in \Gamma$.*

$$V(\underline{\pi}, \lambda) = min\left(\lambda L + E[V(H(\underline{\pi}, Y), \lambda)|\underline{\pi}], \rho(\underline{\pi})\right). \tag{6.9}$$

Let $A(\underline{\pi}, \lambda) \stackrel{def}{=} E[V(H(\underline{\pi}, Y), \lambda)|\underline{\pi}] = \sum_{i=1}^{M} \pi_i E[V(H(\underline{\pi}, Y), \lambda)|S_i]$. Then we have the following result.

Proposition 6 *Consider the feedback generation rule $\boldsymbol{\phi}^*$ and the reproduction rule $\mathbf{c}^*$, given as,*

- $\phi^{*n}(\underline{y}_1^n) = 1$ i.e. *send ACK if* $\rho(\pi_i^n(\underline{y}_1^n)) \leq \lambda L + A(\underline{\pi}^n(\underline{y}_1^n), \lambda)$. *Else* $\phi^{*n}(\underline{y}_1^n) = 0$ i.e. *send NACK.*

- *Whenever* $\phi^{*n}(\underline{y}_1^n) = 1$ *the reproduction rule is,* $c^{*n}(\underline{y}_1^n) = arg\min_{c \in C} \sum_{i=1}^{M} \|s_i - c\|^2 \pi_i^n(\underline{y}_1^n)$.

Then $\boldsymbol{\phi}^*$ and $\mathbf{c}^*$ are optimal, that is, they solve problem (6.7).

Note that the optimal reproduction rule and optimal feedback generation rule, for each λ, are time invariant functions of $\underline{\pi}$.

The outline of proofs for Propositions 5 and 6 is presented in the following sequence of facts.

1. For any feedback generation rule $\boldsymbol{\phi}$, the optimal reproduction rule depends on $\underline{y}_1^n$ through the posterior probabilities $\underline{\pi}^n(\underline{y}_1^n)$. The optimal reproduction rule is given by $c^{*n}(\underline{y}_1^n) = arg\min_{c\in C}\sum_{i=1}^{M}\|s_i - c\|^2\pi_i^n(\underline{y}_1^n)$

2. For any $\underline{\pi}\in\Gamma$, let $V_0(\underline{\pi},\lambda)\stackrel{def}{=}\rho(\underline{\pi})$ and

$$V_T(\underline{\pi},\lambda)\stackrel{def}{=}\inf_{\boldsymbol{\phi},\phi^T(\underline{y}_1^T)=1\forall\underline{y}_1^T} J(\boldsymbol{\phi},\mathbf{c}^*,\underline{\pi},\lambda)\text{ for }T=1,2,\ldots. \tag{6.10}$$

$V_T(\underline{\pi},\lambda)$ is the minimum Bayesian risk over all feedback generation rules which are forced to send ACK at step T, when the prior probability is some $\underline{\pi}\in\Gamma$. Then, the following decomposition holds for a memoryless channel.

$$\begin{aligned} & V_T(\underline{\pi},\lambda) \\ &= \min\left(\lambda L + E[V_{T-1}(H(\underline{\pi},Y),\lambda)|\underline{\pi}],\rho(\underline{\pi})\right) \\ &= \min(\lambda L + \sum_{i=1}^{M}\pi_i\int_{\mathcal{Y}^L} f_{Y|S_i}(y)V_{T-1}(H(\underline{\pi},y),\lambda)dy,\rho(\underline{\pi})). \end{aligned} \tag{6.11}$$

3. $V_T(\underline{\pi},\lambda)\geq V_{T+1}(\underline{\pi},\lambda)\geq V_{T+2}(\underline{\pi},\lambda)\geq\ldots$. Therefore $V_T(\underline{\pi},\lambda)$, as $T\to\infty$ converges pointwise to a function that can be shown to be $V(\underline{\pi},\lambda)$ satisfying eq. 6.9.

The structure of the optimal decoder obtained in Proposition 6 is shown in Figure 6.7.

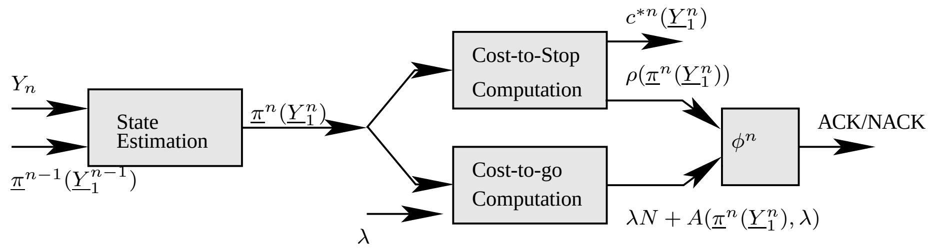

Figure 6.7: Feedback Generation with State Estimation

6.12 Suboptimal Schemes

The general solution obtained in Proposition 6 is exceedingly complex. The complexity can be localized in two distinct blocks in Figure 6.7.

Complexity of state estimation The optimal decoder, *i.e.* the optimal feedback generation rule as well as the optimal reproduction rule are computed from the state which is the posterior probability distribution over transmit codewords. The state space is the M dimensional probability simplex, where M is the number of possible input vectors. For even moderately long size of the vectors, and moderate source coding rate, M can be prohibitively large. As state-estimation has to be done at all steps during the transmission, it is a big contributor to implementation complexity.

Complexity of Design and Implementation of optimal feedback generation rule The feedback generation rule ϕ^* compares the conditional expected channel distortion given current observations given by $\rho(\pi_i^n(\underline{y}_1^n))$ with the cost of sending a NACK, that is $\lambda L + A(\underline{\pi}^n(\underline{y}_1^n), \lambda)$. This requires the knowledge of the functions $V(\underline{\pi}, \lambda)$ and $A(\underline{\pi}, \lambda)$ *for all* posterior probability distributions $\underline{\pi} \in \Gamma$. It turns out that the determination of these functions is highly nontrivial. The general solution in Proposition 6 has only been characterized in a very few cases such as binary sequential hypothesis testing [24] and only approximate methods have been developed for the case of M-ary detection, ***i.e.*** the case of $C = \{s_i, i = 1, 2, \ldots, M\}$ and 1-0 penalty (***e.g.*** [3]). M-SPRT uses expressions similar to Wald's approximations to approximate $V(\pi, \lambda)$ and $A(\pi, \lambda)$[3, 24].

It is still beneficial to consider suboptimal schemes which consider distortion metric explicitly. We propose and consider the following suboptimal schemes.

1. Distortion based feedback generation rule

2. Finite horizon optimal rules

3. Finite lookahead rules

6.12.1 Scheme DIST: Distortion based Feedback Generation Rule

Notice that ϕ^* in Proposition 6 compares the conditional expected channel distortion given current observations, given by $\rho(\pi_i^n(\underline{y}_1^n))$, to $\lambda L + A(\underline{\pi}^n(\underline{y}_1^n), \lambda)$, which varies with $\underline{\pi}$. The function $A(\underline{\pi}, \lambda)$ is a monotonically increasing function of λ, for every prior $\underline{\pi} \in \Gamma$.

Proposition 7 *For* $\lambda_1 \geq \lambda_2$, $A(\pi, \lambda_1) \geq A(\pi, \lambda_2)$*, for all* π*.*

Proof Outline: Let $V_T(\underline{\pi}, \lambda)$ be defined as in eq. (6.10) for T = 0,1,2,$\ldots$. Define $A_T(\underline{\pi}, \lambda) \stackrel{def}{=} E[V_{T-1}(H(\underline{\pi}, Y), \lambda)|\underline{\pi}]$ for T = 1,2,$\ldots$. $V_0(\underline{\pi}, \lambda)$ is independent of λ and hence $A_1(\underline{\pi}, \lambda)$ is monotonically increasing with λ. Assume $A_T(\underline{\pi}, \lambda)$ is monotonically increasing function of λ. Then as $V_T(\underline{\pi}, \lambda)$ is a minimum of two monotonically increasing functions, it is monotonically increasing. Consequently, $A_{T+1}(\underline{\pi}, \lambda)$, which is an expectation over monotonically increasing functions is monotonically increasing. Again, it can be shown that $A_T(\underline{\pi}, \lambda)$ converges to $A(\pi, \lambda_1)$ and hence $A(\pi, \lambda_1)$ is monotonically increasing.

Consider the behavior of $\boldsymbol{\phi}^*$ and $\mathbf{c}^*$ for different values of the Lagrangian rate penalty λ. It is easy to see that the reproduction rule $\mathbf{c}^*$ remains unchanged. On the other hand, increasing λ results in greater rate penalty and hence a smaller rate.

This implies that the decision to send NACK will be taken more infrequently as λ increases. Hence the behavior of $A(\underline{\pi}, \lambda)$ is similar to distortion, as larger rate penalty λ leads to larger distortion.

We propose the use of distortion itself to determine the feedback generation rule. To get the first suboptimal feedback generation rule $\hat{\boldsymbol{\phi}}$, we replace the function $\lambda L + \mathcal{A}(\underline{\pi}, \lambda)$, which varies with $\underline{\pi}$, with a function $\delta(\lambda)$ which is independent of $\underline{\pi}$. Hence the proposed feedback generation rule $\hat{\boldsymbol{\phi}}$ is as follows: Set $\hat{\phi}^n(\underline{y}_1^n) = 1$ *i.e.* send ACK if $\rho(\pi_i^n(\underline{y}_1^n)) \leq \delta$, else set $\hat{\phi}^n(\underline{y}_1^n) = 0$ *i.e.* send NACK. The reproduction rule is same as the optimal *i.e.* $\hat{c}^n(\underline{y}_1^n) = c^{*n}(\underline{y}_1^n) = arg\min_{c \in C} \sum_{i=1}^{M} \|s_i - c\|^2 \pi_i^n(\underline{y}_1^n)$. Varying δ from small to large values captures the rate-distortion tradeoff/ throughput-reliability tradeoff in ARQ with packet combining. Note that, like the optimal rules, scheme A also results in time invariant feedback generation rules. Large δ result in high throughput and small δ result in low distortion. It turns out that for sequential detection of 1-bit equiprobably quantized symmetrical sources, $A(\underline{\pi}, \lambda)$ is indeed independent of $\underline{\pi}$ and hence for this special case, the proposed scheme coincides with the optimal solution.

6.12.2 Scheme FINHZN: Finite Horizon Optimal Rules

An T*-horizon optimal* feedback generation rule is obtained by minimizing $J(\boldsymbol{\phi}, \mathbf{c}, \lambda)$ over only those feedback generations rules for which, $\phi^T(\underline{y}_1^T) = 1$ for all $\underline{y}_1^T$ for some fixed integer T. That is, such a feedback generation rule is the solution of the optimization problem in eq. (6.10). These are straightforward to design as the feedback generation maps are computed explicitly instead of being governed by an implicit formula. These result in time varying feedback generation rules, but have the advantage of bounded delay and bounded memory requirements.

6.12.3 Scheme FINLKHD: Finite Lookahead Rules

A class of time-invariant suboptimal rules, called T*-step lookahead rules* is obtained by executing at each step, the T-horizon optimal feedback rule designed for next T steps. For large enough T, such a rule can be expected to approximate the optimal feedback generation rule.

Schemes DIST, FINHZN and FINLKHD together, will be referred to as *distortion-aware feedback generation rules* or simply *distortion-aware* schemes.

6.13 CRC Based and BER based Systems for Comparison

In this entire section, which presents the illustrative simulation results for comparison with conventional schemes, *we shall assume that the channel input is binary*, such as the one obtained by Binary PSK modulation. Therefore we will be referring to channel input symbols as *bits*. Consequently, we shall assume that the codewords belong to $\{0,1\}^L$.

The features of the distortion-aware schemes proposed in the previous section (Section 6.12) are the following.

1. Distortion metric plays a significant part in the feedback generation.

2. Channel statistics and source statistics are used, both for reproduction rule as well as feedback generation.

3. Independent of the source statistics, the sensitivity of the bits to channel errors, measured from their contribution to distortion, may still be different for different bits. The distortion-aware schemes, therefore ascribe, possibly unequal importance to the transmit symbols.

4. There is a direct way of controlling the tradeoff between quality and rate.

These four features of the distortion-aware schemes, the fallouts of the analysis of the optimal solutions of the Lagrangian formulation are also the features which distinguish the proposed approaches from the conventional tandem protocol designs. Conventional approach to generating ACK/NACK feedback has been through the use of error detection at the receiver. A NACK is generated if there are detectable but uncorrectable errors in the received sequence of channel symbols. The detection is accomplished by adding redundancy and using error detection codes such as CRC.

Scheme CRC-Baseline: Baseline CRC Based system: Figure 6.8 describes the decoder for a baseline packet combining system based on CRC. The main features of the baseline system are (i) Maximum Likelihood (ML) estimation of transmit bits, (ii) check of integrity of the bits by error-detection and (iii) reproduction of the source by inverse quantization.

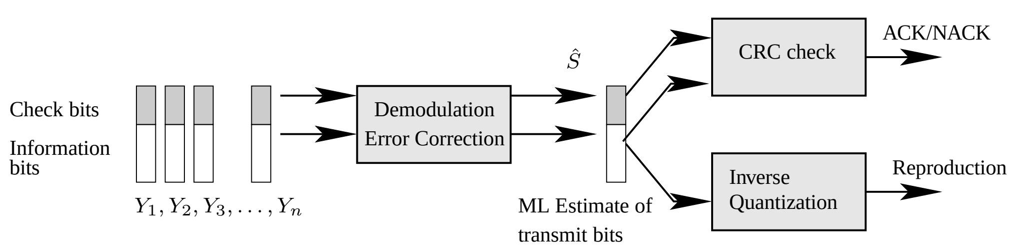

Figure 6.8: Receiver for Baseline CRC based system

Scheme CRC-MMSE: CRC Based system with Pseudo-MMSE decoding: The distortion-aware schemes expect to improve upon the baseline CRC based system by use of (1) different feedback generation rule, (2) reproduction by MMSE estimation of the source as opposed to inverse quantization. For assessment of gains due to these two separate factors, we can conceive another CRC based system which uses CRC for feedback generation but uses MMSE estimation of the source for reproduction. In order to keep the reproduction rule identical to the proposed schemes, we must use only the information bits, *i.e.* the bits in the received symbols, excluding the CRC bits, for MMSE estimation. As CRC bits are ignored for reproduction, we dub this system as CRC Based system with Pseudo-MMSE decoding. It is shown in Figure 6.9.

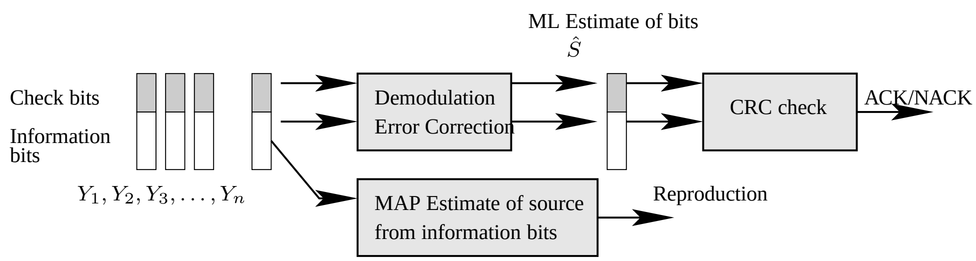

Figure 6.9: Receiver for CRC based system with Pseudo-MMSE decoding

Scheme CRC-List: CRC Based system with List Decoding: Some control over throughput-reliability tradeoff can be obtained in a CRC based system with the help of list-decoding. The CRC based system with List Decoding is shown in Figure 6.10. In list decoding, instead of generating a single ML estimate of the transmit bits, a finite list of most likely candidate estimates is generated. If any of the candidates satisfies the CRC, an ACK is generated and that candidate is used for reproduction by inverse quantization. If no candidate satisfies the CRC, a NACK is generated. Clearly, by varying the size of list, throughput can be traded for reliability.

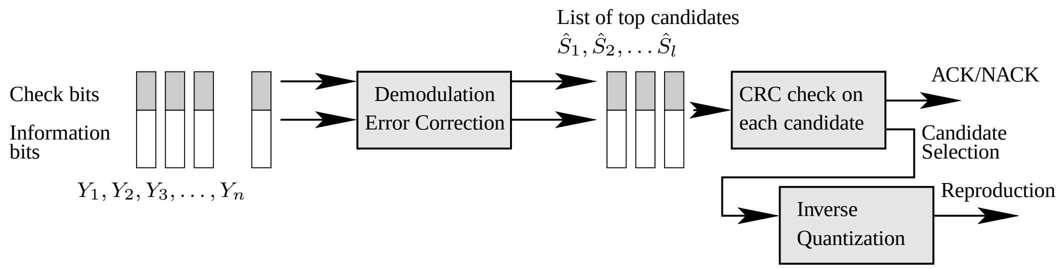

Figure 6.10: Receiver for CRC based system with List decoding

Also, we can also conceive a **CRC Based system with List and Pseudo-MMSE decoding (Scheme CRC-List-MMSE)** where list decoding is used for feedback generation but MMSE decoding from the information bits alone is used for reproduction.

6.13.1 Zero Redundancy BER based Techniques

In addition to comparison against the CRC based systems, which represent the conventional error-detection based techniques, we would also like the performance gain/loss of the distortion-aware schemes, which attempt to minimize distortion and use source statistics, over *optimized* techniques designed to minimize Bit Error Rate (BER) for a given throughput.

For such a comparison, we can conceive Zero Redundancy BER Based feedback generation rules, which are obtained as suboptimal solutions (analogous to Schemes FINHZN and FINLKHD) to a modification of the sequential decision problem (6.7) where the action space $\mathcal{C}$ is the collection of source-encoder indices or codewords $\{0,1\}^L$, and the loss function is *bit-wise Hamming distance* between the true parameter (transmitted source-encoder index) and the action. Thus in this case, the objective is to minimize, for different values of Lagrange Multiplier λ,

$$E\left[\sum_{n=1}^{\infty}\left(Ham(S(X), c^n(\underline{Y}_1^n)) + \lambda nL\right)\psi^n(\underline{Y}_1^n)\right], \tag{6.12}$$

where $Ham : \{0,1\}^L \times \{0,1\}^L \rightarrow \mathcal{N}_+$ is the Hamming Distance between two binary vectors.

6.13.2 Results

To highlight the differences between the distortion-aware techniques and the described conventional CRC-based schemes and Zero redundancy BER based schemes, we consider transmission of synthetic random sources quantized by tree structured vector quantizers over a memoryless noisy channel. We present here the simulation results for memoryless unit variance Gaussian source.

The channel

The channel was chosen to be a binary input, ternary output discrete memoryless channel obtained by quantizing the output of BPSK transmission over an AWGN channel into three regions, $(-\infty, -t_0], (-t_0, t_0)$, and $[t_0, \infty)$. For each signal-to-noise ratio (SNR) of the AWGN channel, the threshold t_0 was numerically obtained so as to maximize the information theoretic capacity of the resulting discrete channel. This channel is useful for simulation as it captures the features of both hard decoding and soft decoding. Also, for the design described, which requires numerical computation of expectations, it helps that the set of all possible channel outputs be finite.

The schematic of quantization of the AWGN channel and the corresponding discrete channel is depicted in Figure 6.11.

Comparing CRC based Schemes

Figures 6.12 through 6.15 present results for comparison of distortion-aware schemes with CRC based schemes. Transmission of a unit variance Gaussian source quantized

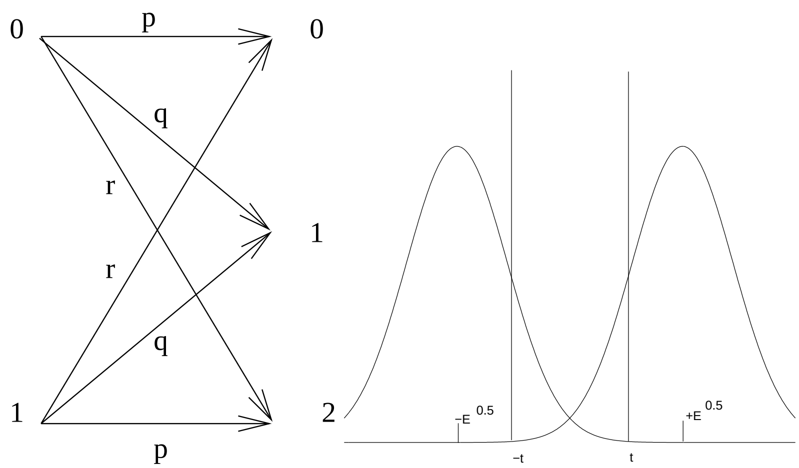

Figure 6.11: Discrete 2-input 3 output channel is obtained as BPSK over quantizing AWGN channel

SNR dB	-1	0	1	2	3	4	5
p	0.7716	0.819	0.864	0.9049	0.9387	0.9644	0.9819
q	0.1905	0.1530	0.1164	0.0828	0.05414	0.03181	0.01636
r	0.03782	0.02768	0.01908	0.01220	0.007121	0.0037	0.001675

Table 6.1: Transition probabilities of the derived discrete channel for different AWGN SNR's.

by a 16-level Tree Structured scalar quantizer [26] over a noisy channel was considered. The 16 levels were mapped into $L = 4$ bits using natural binary indexing. These 4 bit long codewords were transmitted across the chosen discrete memoryless noisy channels using schemes DIST, FINHZN with T=3 and T=4, 1-step FINLKHD, as well as CRC Based Schemes.

Figure 6.12 and 6.13 show the results for the channel with equivalent AWGN channel with SNR 0 dB and 3dB respectively. End-to-end total SNR is the mean squared error per sample expressed in dBs. The points on the curve DIST are obtained by simulation as the threshold δ is varied from large values to small values. Similarly the points for 1-step FINLKHD were obtained by simulation as Lagrange multiplier λ was varied from large values to small values. For simulations we used 1,200,000 samples of unit variance Gaussian source. For each sample, the channel was used nearly 20 times.

The points for FINHZN T=3 and T=4, were obtained by numerical calculation. They are operational rate-distortion performance curves obtained by pruning a depth-T Pruned TSVQ [15] with $3^L = 81$ children per node. The relationship between Pruned TSVQ and the design problem is explained in Chapter 7.

Three simple CRC based transmitters, ones with 1 bit, 2 bit and 3 bit CRC's applied to each 4 bit packet, are used for comparison. The decoders are CRC-List and CRC-List-MMSE for different list sizes. Scheme CRC-List with list size 1, is the baseline CRC based system. CRC-List-MMSE with list size of 1, is the CRC-MMSE scheme. Results were obtained for list sizes of $1, 2, 4, \ldots, 2^n$ for a transmitter which uses n-bit CRC. The number's next to points for CRC-List represent the list size used.

The plots also show results for Fixed Horizon schemes which are in fact schemes with *repetition coding and no feedback*. In such a scheme codeword is repeated a fixed number of times. The decoder performs a MMSE estimation of the source from the received copies. The performance at the highest transmission rate, achieved by a T-horizon FINHZN scheme is equal to that of a fixed horizon scheme transmitting T copies.

The feature immediately noticeable about the plots is the high flexibility offered by the distortion-aware schemes. The Lagrangian approach yields a continuum of operating points for each of the distortion-aware schemes. The CRC based schemes on the other hand, provide limited flexibility, operating at discrete set of points.

Secondly, though the distortion-aware schemes are suboptimal, they consistently outperform the CRC based schemes for a wide range of transmission rates. For the

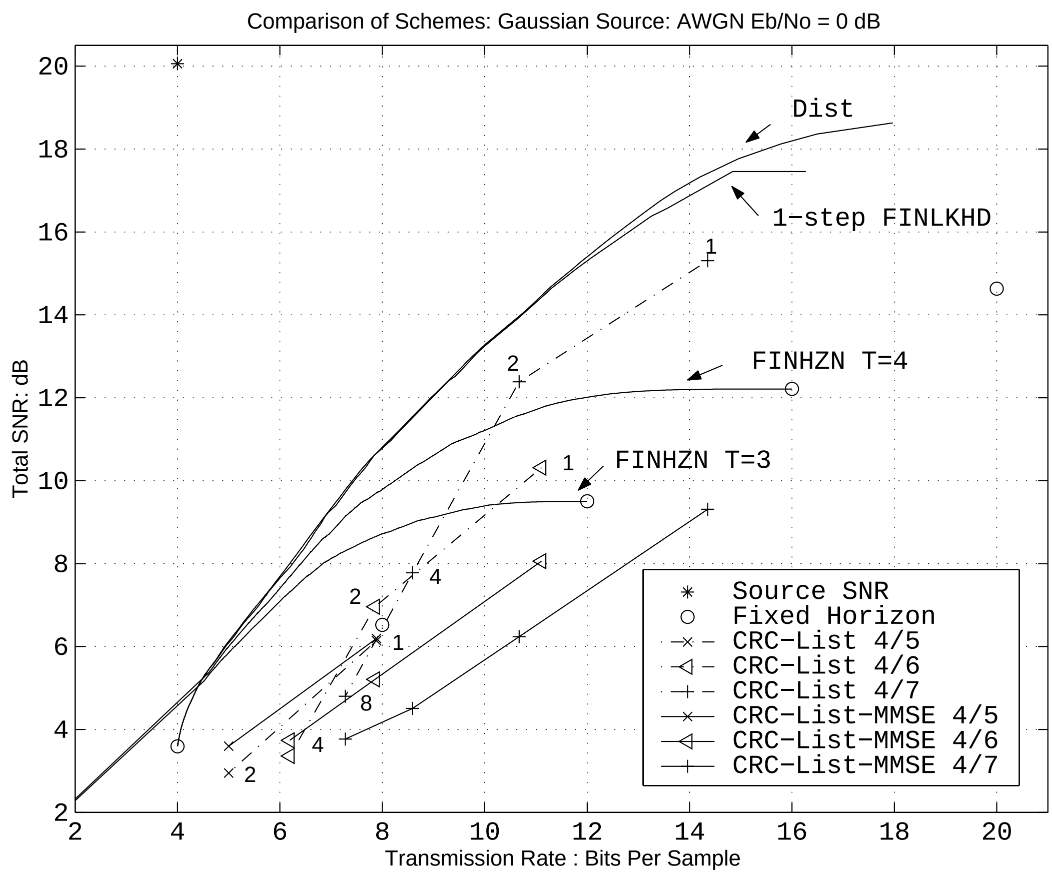

Figure 6.12: Performance (Total SNR vs. Trans. Rate) of Various Schemes of Scalar IID Gaussian source quantized with 4 bit TSVQ over noisy channel (equiv. AWGN SNR = 0dB)

more noisy channel, namely the one corresponding to AWGN-0dB, the gains of DIST and 1-step FINLKHD are nearly 2 dB at almost all transmission rates. The gains of distortion aware schemed for the 3 dB channel are lower, they still outperform all CRC based schemes except one. CRC-List with coderate 4/7, which adds 3 bit CRC to every 4 bits, with list size 2 outperforms the distortion-aware schemes. Note that distortion-aware schemes in the plots have no redundancy added.

Another interesting observation is that for high redundancy CRC - such as CRC-

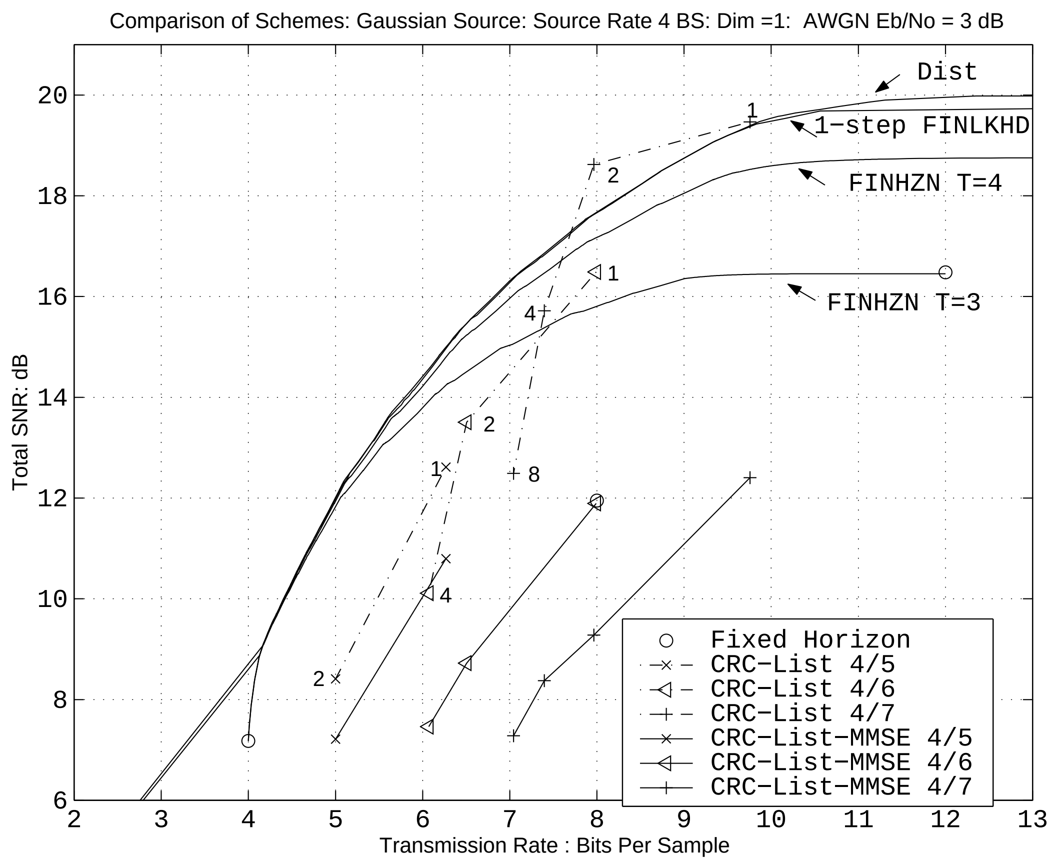

Figure 6.13: Performance (Total SNR vs. Trans. Rate) of Various Schemes of Scalar IID Gaussian source quantized with 4 bit TSVQ over noisy channel (equiv. AWGN SNR = 3dB)

List with coderate 4/7, CRC-List outperforms CRC-List-MMSE. This observation can be explained from the fact that the extra diversity provided by redundant bits more than compensates for the suboptimality of ML decoding over MMSE decoding. This is not the case for high coderate (*i.e.* low redundancy) CRC schemes (Figure 6.14). High coderate CRC Based schemes used are of rates 4/5, 8/10 and 16/19, which are 1 bit CRC added to 4 bits, 2 bit CRC added to 8 bits, and 3 bit CRC added to 16 bits respectively. For these coderates, CRC-List-MMSE generally perform better than CRC-List.

Fourth noticeable feature is that the schemes DIST and and 1-step FINLKHD perform nearly identically.

The source coder used in Figures 6.12 through 6.15, is a 4 bit TSVQ with average distortion $D_s \sim 0.0097$ which is nearly 20 dB. The source distortion becomes dominating factor for higher transmission rates. Figure 6.15 plots only the channel induced distortion D_c, expressed in dB, for these schemes for the 0dB channel. As channel distortion can be driven arbitrarily close to zero, the channel-induced SNR for schemes DIST and 1-step FINLKHD does not saturate, unlike the curves in Figure 6.12.

The curves for DIST and 1-step FINLKHD are nearly linear, implying that the distortion drops exponentially with transmission rate. Also, they are at a sharper slope than Fixed Horizon schemes. This shows that the gain in SNR of DIST and 1-step FINLKHD over schemes not using feedback increases with transmission rate.

FINHZN schemes are efficient at low transmission rates, but their performance curves saturate as the rate approaches the corresponding fixed horizon schemes.

Zero Redundancy BER based Schemes

As discussed earlier, we would also like to isolate the contribution of distortion metric in the feedback generation rule as opposed to Hamming Distance metric. Towards this end, we consider comparison with zero-redundancy BER based schemes.

Figures 6.16 and 6.17 present the curves for average rate vs. total SNR as λ is varied from small to large, for various schemes for channels obtained from AWGN channels with SNR 0 dB and 3 dB, respectively. In all the curves, including the zero redundancy BER based schemes, *the reproduction rule is chosen to be the MMSE estimate of the source.* The rate distortion performance of the distortion aware schemes DIST, FINHZN with T=3 and 1-step FINLKHD is compared against zero redundancy packet-

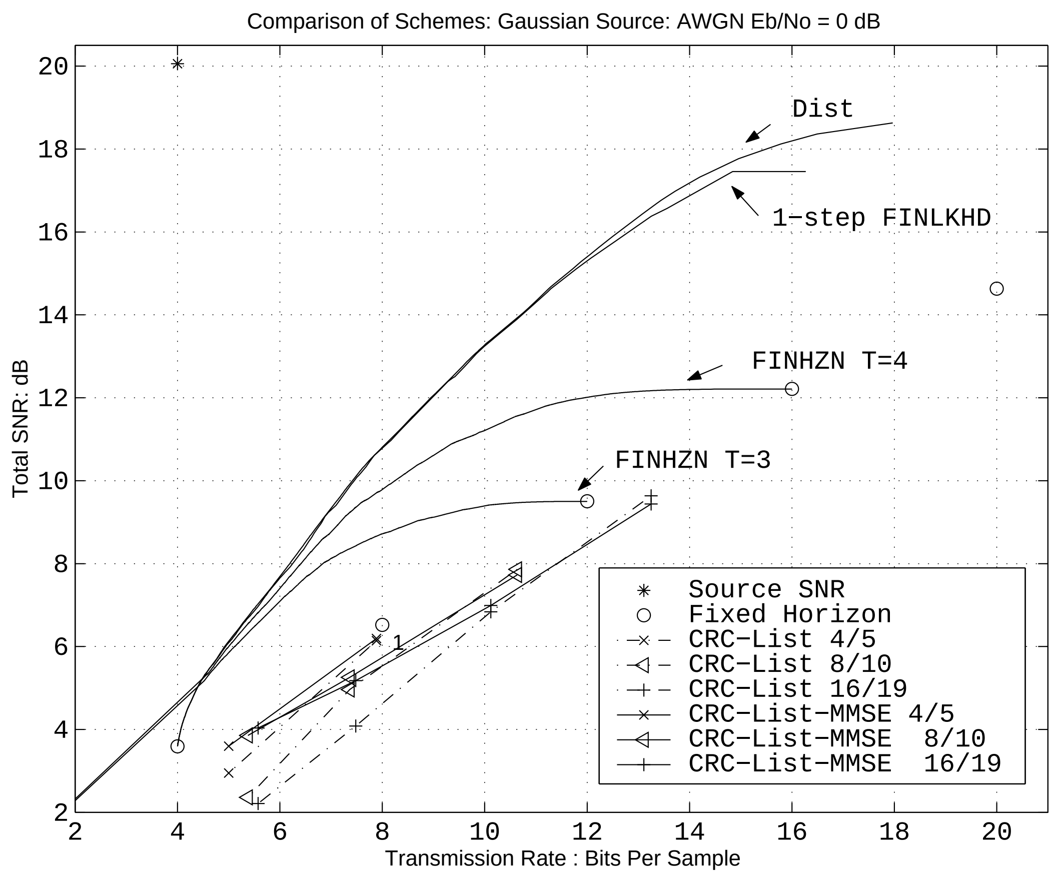

Figure 6.14: Performance (Total SNR vs. Trans. Rate) of High Rate CRC based Schemes, IID Gaussian source, dim = 1, TSVQ 4 bit/sample, equiv. AWGN SNR = 0dB

combining feedback generation rules 1) BER based FINHZN with T=3, 2) BER based 1-step FINLKHD.

Although, the BER based zero redundancy schemes, behave like the conventional CRC, that is, they treat all the source-encoder bits equally, *they make use of source statistics for ACK/NACK generation.* The only difference between the BER based schemes and distortion-aware schemes is the distortion metric.

From the figures, it is evident that the channel-distortion/rate performance of the

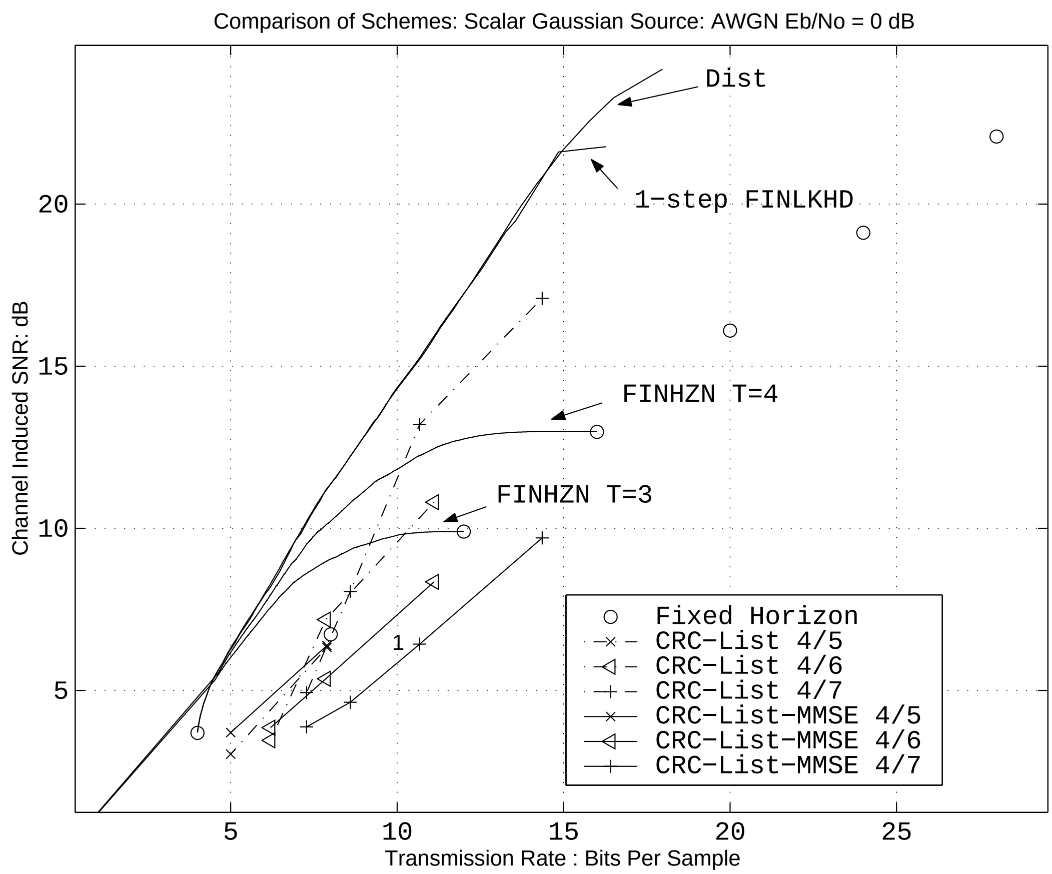

Figure 6.15: Channel Distortion for Various Schemes, IID Gaussian source, dim = 1, TSVQ 4 bit/sample, equiv. AWGN SNR = 0dB

distortion-aware schemes is almost always superior to the BER based schemes. But the most interesting feature is that, at high transmission rates, the BER based zero redundancy schemes, seem to catch up with the corresponding distortion aware schemes. The distortion aware schemes, show high gains in the high-throughput *i.e.* low transmission rate region. The highest performance improvement is about 2 dB in both the cases. Another advantage of the curves for the distortion-aware schemes is their high positive slope at low rate region, compared to the zero-redundancy BER based schemes. This has implications in progressive transmission, where a rapid improvement in source quality as a function of bit rate is desirable.

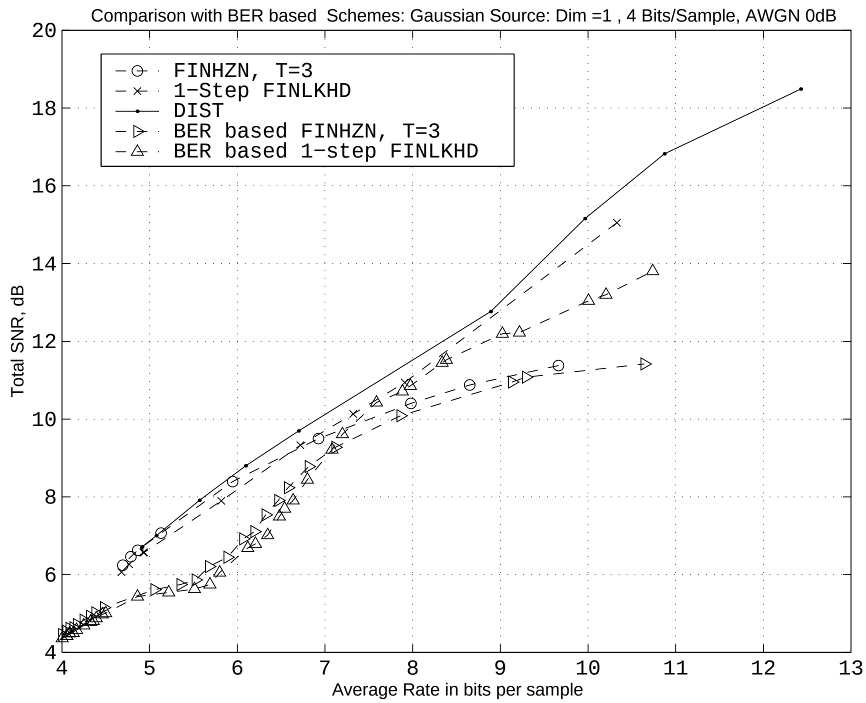

Figure 6.16: Performance Comparison with Zero Redundancy BER based schemes. Gaussian Source, TSSQ with 4 bits/sample. AWGN Channel SNR =0 dB.

6.14 Conclusion

In this chapter we have addressed the problem of joint source channel coding with ACK/NACK feedback from first principles. We have identified the different components and classified the transmitter and the receiver side according to the degree of freedom allowed in the use of the ACK/NACK feedback. As every system without feedback is a special case of the one with feedback, and a tandem system a special case of a joint source -channel coding system, the design of a communication system from first principles can be construed as rather naive. Nevertheless, there are significant insights to

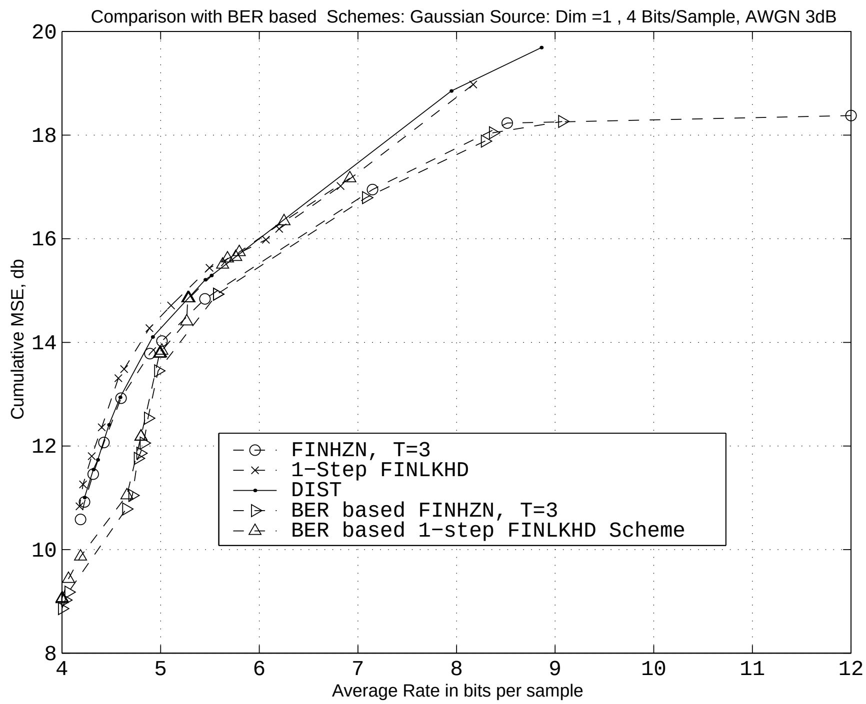

Figure 6.17: Performance Comparison with Zero Redundancy BER based schemes. Gaussian Source, TSSQ with 4 bits/sample. AWGN Channel SNR =3 dB.

be gained from this approach. As a special and simplified case,. we have considered decoder design for a passive encoder system in which the transmitter transmits copies of the same codeword over the noisy channel. We have obtained optimal design by dynamic programming techniques, which yielded the optimal reproduction rule and the optimal feedback generation rule. We have proposed some reduced-complexity suboptimal feedback generation rules, which take into consideration the source statistics and the distortion metric and hence are called distortion-aware. Distortion-aware schemes, in addition to outperforming conventional CRC based and BER based schemes, also offer a lot of flexibility in choosing the operating transmission rate and allow easy switching from lower transmission rate to higher transmission rate. The three fronts on which distortion-aware schemes are superior to conventional schemes are 1) use of source-statistics *i.e.* exploiting residual redundancy in the source-encoder, 2) use of distortion measure - which is more meaningful for transmission of loss-tolerant sources, 3) flexible selection of operating points.

The next chapter extends the ideas in this chapter to an active source-encoder and explores the structure of the optimal solution in more detail. It also establishes the close link between the ARQ design problem and Pruned TSVQ, and shows how progressive transmission can be accomplished for such a system while retaining optimality.

Chapter 7

Pruned Tree Structured Quantization in Noise and Feedback

In this chapter, we take our program of designing optimal joint-source-channel coding for channels with feedback one step further. We devise optimal decoding schemes where a *progressively transmitted embedded source coder* suffers channel noise and at each step in progressive transmission there is a feedback from the receiver to the transmitter. In this framework, the transmitter is active, that is, on receiving a NACK it does not retransmit the codeword transmitted earlier but instead transmits new information. We restrict our attention to the finite horizon case, where the transmission is not allowed to continue beyond a fixed number of steps, say, T. Again we focus on the receiver side and investigate the structure of the optimal decoder and feedback generation mechanism here. The tools we use will be as earlier, based on Lagrangian formulation and sequential analysis.

In the absence of channel coding the progressive coding is typically performed using a tree structured quantizer. The tree structured quantizer is capable of coding in several stages, each stage provides a refinement of the previous stage. If some form of variable length coding is available, then an effective way of obtaining a collection of quantizers from a single tree structured quantizer is by the method of pruning [15]. The resulting collection of quantizers is called pruned tree structured vector quantizers (PTSVQ). There is an elegant theory associated with pruning. Pruned tree structured quantizers first appeared in the context of decision trees where Breiman et al [8] presented an algorithm for pruning. It was later generalized to other contexts, such as tree structured quantization, regression trees, quantization of noisy sources and variable order Markov modeling [15, 26].

In this chapter we show the close link between PTSVQ and transmission using an embedded source-coder over a channel with ACK/NACK feedback. Consequently we generalize the concept to carry out joint source-channel PTSVQ, or PTSVQ in the presence of noise and feedback. In addition to establishing the close link, we show the existence of a "feedback-threshold" function which reveals the simple structure behind the optimal feedback generation rules for all Lagrangian penalties.

7.1 Pruned Tree Structured Vector Quantizers

An T-stage TSVQ is a collection of T vector quantizers, one associated with each stage, such that, every VQ cell of i^{th} stage is obtained by partitioning some cell, (its "parent") at $i-1^{th}$ stage, for $i = 1, 2, \ldots, T$. The quantizer at 0^{th} stage consists of one cell. The parent-child relationship between cells gives a full balanced tree of VQ cells. Without loss of generality, we shall assume that the tree is binary, ***i.e.*** each cell is either a leaf or has exactly two children.

Let the collection of cells in a TSVQ denoted by Z_0, be denoted by $\widehat{Z}_0$. A pruned TSVQ, Z', is obtained from a full TSVQ by selecting a subset $\widehat{Z} \subset \widehat{Z}_0$ of the cells, with the property that a cell $t \in \widehat{Z}$ if and only if its parent cell $parent(t) \in \widehat{Z}$. We say that $Z \preceq Z_0$. The relation $\preceq$ is naturally extended as a partial order for comparing two pruned TSVQs - or just pruned trees. For two pruned trees Z and Z', $Z' \preceq Z$ if $\widehat{Z}' \subset \widehat{Z}$. A cell in a PTSVQ is called a *leaf* if it has no children, else, it is called an *interior node*. Figure 7.1 illustrate a pruned tree obtained from a full tree.

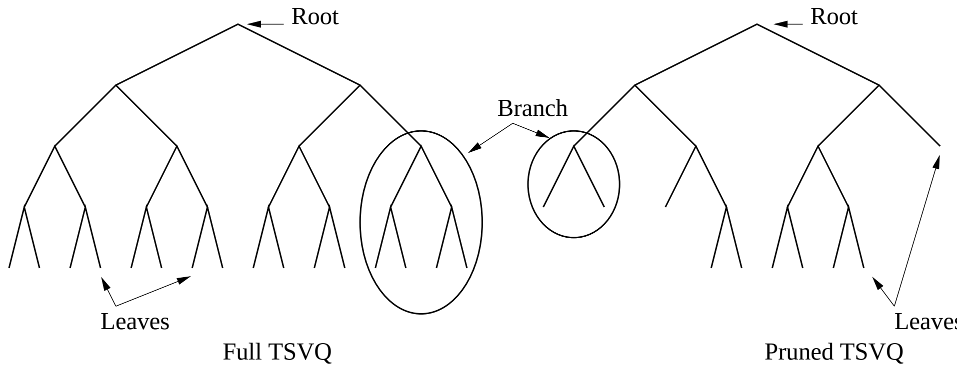

Figure 7.1: TSVQ and Pruned TSVQ

The encoding and decoding of a PTSVQ is analogous to that of a full TSVQ. A source vector is quantized in stages, till a leaf cell that contains the vector, is found. The path from the root node to the leaf is used for encoding the vector, and a representative vector, the "centroid" of the leaf cell is used for reproduction.

For a given source with known statistics, a rate and an average distortion can be associated with every PTSVQ. The rate is measured as either (i) the expected length of the path from the root to the leaf or (ii) the expected entropy of such a path. We shall assume the former definition of the rate. Let the distortion-rate pair for a PSTVQ $Z \preceq Z_0$ be denoted by $(\underline{\mathrm{D}}(Z), \underline{\mathrm{R}}(Z))$. Then the collection of optimal rate distortion pairs, namely those on the lower convex hull of the set $\{(\underline{\mathrm{D}}(Z), \underline{\mathrm{R}}(Z)) : Z \preceq Z_0\}$ has the following interesting property [15].

Theorem 1 PTSVQ property: *The collection of points on the lower convex hull of the set $\{(\underline{\mathrm{D}}(Z), \underline{\mathrm{R}}(Z)) : Z \preceq Z_0\}$ can be obtained by repeatedly pruning a single tree.*

In other words there is a sequence of PTSVQs, $\ldots Z_k \preceq Z_{k-1} \preceq \ldots \preceq Z_2 \preceq Z_1 \preceq Z_0$, *which traces the convex hull.*

This elegant result also leads to the generalized algorithm due to Breiman- Friedman-Olshen and Sloane (BFOS) for obtaining the points on the convex hull [15, 26].

In the following sections we consider progressive transmission of a TSVQ encoded source in the presence of channel noise and ACK/NACK and obtain a PTSVQ like property for the optimal decoding schemes. In that sense, the following sections present a generalization of PTSVQ.

7.2 Extending the Interpretation of ACK/NACK

In Chapter 6 we considered how to carry out joint source-channel decoding when the transmitter does retransmission of the codeword. There, the selection of feedback generation map could be used to control the throughput-reliability or rate-distortion tradeoff. Here we consider a slightly general case in which on receipt of a NACK the transmitter proceeds with the transmission of new information.

Clearly, this contains as a special case, the case of retransmission of the same codeword. In this chapter we obtain the optimal decoding schemes for this case. This chapter widens the interpretation of ACK/NACK feedback. Conventionally ACK/NACK feedback was used for indicating if the transmit codewords was decoded with acceptable reliability or not. The conventional interpretation turns out to be narrow in the light of the possibility of the transmitter transmitting new information on receiving a NACK feedback. When we develop the decoding scheme, we shall see that the NACK feedback serves a dual purpose. (i)First, it is used for indicating that the previous transmission was corrupted beyond recovery by the channel noise. (ii) Second, it is used for controlling the rate distortion performance of the joint source-channel coder! A NACK may be sent when the previous transmission was noiseless, but it is favorable for rate-distortion tradeoff that further information about the same source-vector be sent. In other words *NACK is used as a permission to continue transmission of new or old information about the same source vector.*

This new interpretation essentially says that NACK feedback can be used for rate control. We shall see that the decoder structure in fact has a property like that of the PTSVQ, namely the optimal decoding schemes at different rate distortion tradeoffs are embedded.

This is still not the most general transmission scheme conceivable as the transmitter is still not active. It transmits a fixed sequence of codewords for a given source vector and stops when an ACK is received.

7.3 Transmission Set-up and Notation

As earlier, consider the transmission of a k dimensional random source-vector X taking values in $\mathcal{X} \subset \mathcal{R}^k$, over a noisy channel with discrete input alphabet $\mathcal{I}$ and possibly continuous valued output alphabet $\mathcal{Y}$. X is quantized by a TSVQ with depth T which generates a channel codeword $S_n(X) \in \mathcal{I}^L$ for each stage $n = 1, 2, \ldots, T$. We assume that the TSVQ and the codeword allocation is predesigned and fixed. If codeword $S_k(X)$ is transmitted, a noisy version of the codeword $Y_n \in \mathcal{Y}^L$ is received. We need not assume that the channel is memoryless. We shall just assume that the statistics of the source, *i.e.* the distribution of X and that of the channel, *i.e.* the joint distributions of $Y_1, Y_2, \ldots, Y_T$ are known for each value of X. For simplicity, we shall assume that conditional probability densities of the kind $f(y_{i_1}|y_{i_2}, y_{i_3}, \ldots y_{i_k}, x)$ can be computed for all values of $y_{i_k} \in \mathcal{Y}$ and $x \in \mathcal{X}$.

The transmission proceeds as follows. First the codeword $S_1(X)$ is transmitted and a feedback of ACK/NACK is requested. On receiving NACK, which is taken as a “permission to continue transmission”, $S_2(X)$ is transmitted. This way, codewords $S_1(X), S_2(X), \ldots, S_n(X), \ldots$ are transmitted one by one until either an ACK is received or $S_T(X)$ has been transmitted.

Similar to Chapter 6 the *feedback generation rule* $\boldsymbol{\phi}$ at the receiver, is specified by a sequence of *feedback generations maps* $\phi^n : \mathcal{Y}^{nL} \to \{0, 1\}, n = 1, 2, \ldots$. At the n^{th} step, let the received realizations of the noisy copies be $y_1, y_2, \ldots, y_n$ for $y_i \in \mathcal{Y}^L$. Then an ACK is transmitted if $\phi^n(y_1, y_2, \ldots, y_n) = 1$. A NACK is transmitted if $\phi^n(y_1, y_2, \ldots, y_n) = 0$. The *reproduction rule* $\mathbf{c}$ at the receiver is specified by a sequence of reproduction maps $c^n : \mathcal{Y}^{nL} \to \mathcal{C} \subset \mathcal{X}$. $\mathcal{C}$ is the *reproduction codebook*. If an ACK is generated at the n^{th} step, *i.e.* if $\phi^n(y_1, y_2, \ldots y_n) = 1$,, then the source is reconstructed as $c^n(y_1, y_2, \ldots, y_n)$. It is not necessary for $\mathcal{C}$ to be discrete. Again, let $\underline{y}_1^n$ and $\underline{Y}_1^n$ be the shorthand for denoting the sequences $y_1, y_2, \ldots, y_n$ and random vectors $Y_1, Y_2, \ldots, Y_n$, respectively.

Let $\psi^n(\underline{y}_1^n) = \prod_{i=1}^{n-1}(1 - \phi^i(\underline{y}_1^i))\phi^n(\underline{y}_1^n)$. Then $\psi^n(\underline{y}_1^n) = 1$ for all those sequences $\underline{y}_1^n$ which generate a ACK only at the n^{th} step and not earlier. In this chapter we consider only a finite stage TSVQ hence we require that $\phi^T(\underline{Y}_1^T) = 1$ always. In other words this implies $E\left[\sum_{n=1}^T \psi^n(\underline{Y}_1^n)\right] = 1$.

The average rate per source vector, that is the expected number of channels symbols put on the channel before stopping (*i.e.* before an ACK is received) is given by,

$$R(\boldsymbol{\phi}) = E\left[\sum_{n=1}^{T} nL\psi^n(\underline{Y}_1^n)\right]. \tag{7.1}$$

Let $d(\cdot, \cdot)$ denote the squared error distortion measure. Then for given $\boldsymbol{\phi}$ and $\mathbf{c}$, the expected distortion is computed as,

$$D(\boldsymbol{\phi}, \mathbf{c}) = E\left[\sum_{n=1}^{T} d(X, c^n(\underline{Y}_1^n))\psi^n(\underline{Y}_1^n)\right]. \tag{7.2}$$

Note that, although specifying the collection of maps $\phi^n, n = 1, 2, \ldots, T$ is not the same as specifying the collection $\psi^n, n = 1, 2, \ldots, T$, the performance measures $D(\boldsymbol{\phi}, \mathbf{c})$ and $R(\boldsymbol{\phi})$ depend only on $\psi^n, n = 1, 2, \ldots, T$.

For a non-negative multiplier $\lambda \geq 0$, define,

$$J(\boldsymbol{\phi}, \mathbf{c}, \lambda) \stackrel{def}{=} D(\boldsymbol{\phi}, \mathbf{c}) + \lambda R(\boldsymbol{\phi}) \tag{7.3}$$

Then the problem of decoder design can be expressed as,

$$\min_{\boldsymbol{\phi}, \mathbf{c}} J(\boldsymbol{\phi}, \mathbf{c}, \lambda) = \min_{\boldsymbol{\phi}, \mathbf{c}} E\left[\sum_{n=1}^{T} \left(d(X, c^n(\underline{Y}_1^n)) + \lambda n L\right) \psi^n(\underline{Y}_1^n)\right], \tag{7.4}$$

This problem, like the special case in Chapter 6, is a Bayesian sequential decision problem. We shall refer to the Lagrangian sum of distortion and transmission rate as "Bayesian risk" or simply "risk".

7.3.1 PTSVQ as Bayesian Sequential Decisions over Noiseless Channel

It is straightforward to see that there is close relationship between PTSVQ and transmission of a TSVQ over a noiseless channel with ACK/NACK feedback. In fact, over a discrete output noiseless channel, there is a one to one relationship between all possible pruned trees of a tree, and all possible feedback generation rules (specified in terms of ψ^n's, as opposed to ϕ's). Any pruning of a full tree can be represented in terms of some feedback generation rule $\boldsymbol{\phi}$ and vice versa. Over a noiseless channel, a sequence of received codewords, $\underline{y}_1^n$, (which is the same as the sequence of transmit codewords,) is equivalent to a path from the root node to a node at depth n. If $\phi^n(\underline{y}_1^n) = 0$, a NACK is transmitted, then the node corresponding to $\underline{y}_1^n$ is an interior node. $\psi^n(\underline{y}_1^n) = 1$ then $\underline{y}_1^n$ corresponds to a leaf in the pruned tree.

Figure 7.2 illustrates a binary TSVQ transmitted over a noiseless binary channel, with the values of some feedback generation rule ϕ and the equivalent PTSVQ.

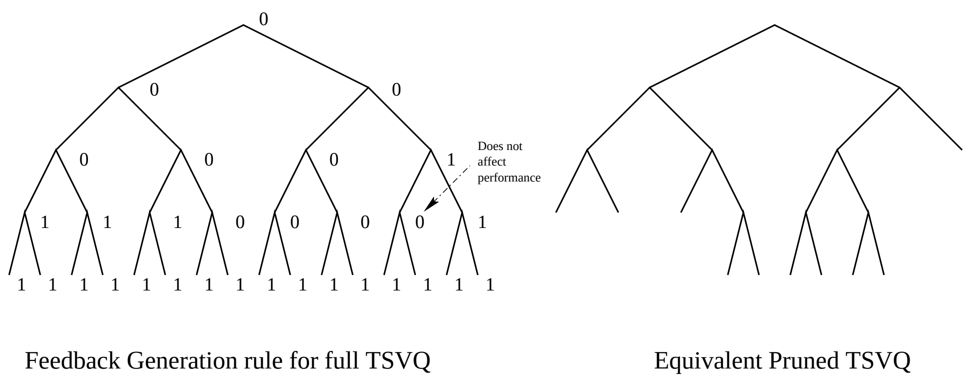

Figure 7.2: Feedback Generation Rule over a Full TSVQ and Equivalent Pruned TSVQ

7.4 Decoder Design

The optimal decoder design is obtained by the solution to the sequential design problem given by equation (4.14). The two main results consist of design of the optimal reproduction rule $\mathbf{c}$ and the optimal feedback generation rule ϕ. These results can be obtained by a dynamic programming argument [24, 28].

First we state the optimal reproduction rule.

Theorem 2 Optimal Reproduction Rule: *Let $c^{*n}(\underline{Y}_1^n)$ be a Bayes estimate of the source based on a fixed number of received codewords $\underline{Y}_1^n$. Then for every feedback* *generation rule* ϕ *and* λ*,* $J(\boldsymbol{\phi}, \mathbf{c}, \lambda)$ *is minimized with respect to* $\mathbf{c}$*, by the functions,* $c^{*n}(\underline{Y}_1^n)$ *for* $n = 0, 1, 2, \ldots, T$*. Here,*

$$\begin{aligned} c^{*0} &= arg \min_{c \in \mathcal{C}} E\left[d(X, c)\right], \text{ and} \\ c^{*n}(\underline{y}_1^n) &= arg \min_{c \in \mathcal{C}} E\left[d(X, c) | \underline{Y}_1^n = \underline{y}_1^n\right]. \end{aligned} \tag{7.5}$$

The proof is straightforward and can be found in [24, 28].

It can be seen that the optimal reproduction rule turns out to be independent of the feedback generation rule $\boldsymbol{\phi}$ and λ. This implies that one can always use the same reproduction rule for all methods of generating feedback and for all penalties λ on the rate. We shall assume in the subsequent portion of the chapter that $\mathbf{c}^*$ is the reproduction rule.

To obtain the feedback rule $\boldsymbol{\phi}^*$ which minimizes $J(\boldsymbol{\phi}, \mathbf{c}^*, \lambda)$ for a fixed λ, define

$$\begin{aligned} \rho_n(\underline{Y}_1^n) &\stackrel{def}{=} E\left[d(X, c^{*n}(\underline{Y}_1^n)) | \underline{Y}_1^n\right] \text{ and} \\ U_n(\underline{Y}_1^n, \lambda) &\stackrel{def}{=} \rho_n(\underline{Y}_1^n) + \lambda n L \end{aligned} \tag{7.6}$$

We have assumed that at each step, the possible number of codewords transmitted is finite and also that the total number of steps in transmission is finite. Under these conditions it is straightforward to show that $\rho_n(\underline{Y}_1^n)$ and consequently $J(\boldsymbol{\phi}, \mathbf{c}^*, \lambda)$ is bounded for each λ.

$U_n(\underline{Y}_1^n, \lambda)$ is the conditional risk of stopping, *i.e.* sending ACK, at the n^{th} step, having received $\underline{Y}_1^n$.

Let us define a feedback generation rule $\boldsymbol{\phi}^*$ as follows. Suppose $T-1$ noisy codewords $\underline{Y}_1^{T-1}$ have already been received. Then if an ACK is to be sent at that point, then the conditional risk is $U_{T-1}(\underline{Y}_1^{T-1})$. While, if a NACK is sent then another noisy copy will have to be received, and in that case the conditional risk is $E[U_T(\underline{Y}_1^T)|\underline{Y}_1^{T-1}]$. Thus at $T-1^{st}$ step, for a point $\underline{y}_1^{T-1} \in \mathcal{Y}^{L(T-1)}$ the risk is minimized if we define

$\phi^{*T-1}(\underline{y}_1^{T-1})$ as,

$$\phi^{*T-1}(\underline{y}_1^{T-1}) \stackrel{def}{=} \begin{cases} 1 \text{ if } U_{T-1}(\underline{y}_1^{T-1}, \lambda) \leq E[U_T(\underline{Y}_1^T, \lambda)|\underline{y}_1^{T-1}] \\ 0 \text{ otherwise.} \end{cases} \tag{7.7}$$

Clearly, at $T-1^{st}$ step, the minimum posterior risk given received codewords $\underline{y}_1^{T-1}$ is given by,

$$G_{T-1}(\underline{y}_1^{T-1}, \lambda) \stackrel{def}{=} min[U_{T-1}(\underline{y}_1^{T-1}, \lambda), E[U_T(\underline{Y}_1^T, \lambda)|\underline{y}_1^{T-1}]] \tag{7.8}$$

This cost is obtained by using the feedback generation map $\phi^{*T-1}(\underline{y}_1^{T-1})$.

Similarly, inductively define "minimum" posterior risk $G_n(\underline{y}_1^n)$ as follows.

$$\begin{aligned} G_T(\underline{y}_1^T, \lambda) &\stackrel{def}{=} U_T(\underline{y}_1^T, \lambda) \\ G_{n-1}(\underline{y}_1^{n-1}, \lambda) &\stackrel{def}{=} min[U_{n-1}(\underline{y}_1^{n-1}, \lambda), E[G_n(\underline{Y}_1^n, \lambda)|\underline{y}_1^{n-1}]] \text{ for } n = 2, 3, \ldots, T, \\ G_0 &\stackrel{def}{=} min[U_0, E[G_1(\underline{Y}_1^1, \lambda)]] \end{aligned} \tag{7.9}$$

Hence we can define the feedback generation maps $\phi^{*n}(\underline{y}_1^n)$, inductively.

$$\begin{aligned} \phi^{*T}(\underline{y}_1^T) &\stackrel{def}{=} 1 \text{ and} \\ \phi^{*n-1}(\underline{y}_1^{n-1}) &\stackrel{def}{=} \begin{cases} 1 \text{ if } U_{n-1}(\underline{y}_1^{n-1}, \lambda) \leq E[G_n(\underline{Y}_1^n, \lambda)|\underline{y}_1^{n-1}] \\ 0 \text{ otherwise, for } n = 1, 2, \ldots T. \end{cases} \end{aligned} \tag{7.10}$$

The collections of maps $\phi^{*n}(\underline{y}_1^n)$ for $n = 0, 1, \ldots T$, defined above are an optimal feedback generation rule. Note that G_n, and hence ϕ^{*n} vary with λ. We have the following result [24, 28].

Theorem 3 Optimal Feedback Generation Rule: *The collection of maps* $\boldsymbol{\phi}^* \stackrel{def}{=} \{\phi^{*n}, n = 0, 1, \ldots T\}$ *is the optimal feedback generation rule for a given* λ*,* i.e. *, for any other feedback generation rule* $\boldsymbol{\phi}$*, following holds.*

$$J(\boldsymbol{\phi}^*, \mathbf{c}^*, \lambda) \leq J(\boldsymbol{\phi}, \mathbf{c}^*, \lambda) \tag{7.11}$$

We have obtained the optimal feedback generation rules and optimal reproduction rules for the joint source-channel coding problem with ACK/NACK feedback using standard techniques from sequential analysis. We now embark on further investigation and show that, the optimal feedback generation rules and optimal reproduction rules have a property analogous to PTSVQ.

7.5 Embedded Optimal Policies

We would like to examine the property in Theorem 1 in terms of the interpretation of PTSVQ as sequential decisions with ACK/NACK feedback

Consider the noiseless channel case and refer to the relationship between a PTSVQ and feedback generation rule described in Section 7.3.1. Let $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ be the corresponding feedback generation rules for two pruned trees Z_1 and Z_2 respectively. It can be verified easily that, $Z_1 \preceq Z_2$ if and only if $\phi_1^n \geq \phi_2^n$ for $n = 0, 1, 2, \ldots, T$.

Therefore the PTSVQ property (Theorem 1) can be restated in terms of the feedback generation rules. The main result of this section is the generalization of Theorem 1 to noisy channels.

Theorem 4 *Let $\lambda_1 \geq \lambda_2 \geq 0$. Let $\boldsymbol{\phi}_1^*$ and $\boldsymbol{\phi}_2^*$ denote the optimal feedback generation rules for the problem (4.14), given by eq. (7.10) corresponding to λ_1 and λ_2 respectively. Then $\phi_1^{*n}(\underline{y}_1^n) \geq \phi_2^{*n}(\underline{y}_1^n)$ for all $\underline{y}_1^n \in \mathcal{Y}^{nL}$ for $n = 0, 1, \ldots T$. In other words $\phi_1^{*n}(\underline{y}_1^n) = 0 \implies \phi_2^{*n}(\underline{y}_1^n) = 0$ and $\phi_2^{*n}(\underline{y}_1^n) = 1 \implies \phi_1^{*n}(\underline{y}_1^n) = 1$.*

In order to prove Theorem 4, we need to show a small result. Define maps $W_n(\underline{y}_1^n, \lambda) \stackrel{def}{=} G_n(\underline{y}_1^n, \lambda) - L\lambda n$ for $n = 0, 1, 2, \ldots, T$. It can be easily verified by induction that, the equation (7.10) can be reformulated as,

$$\begin{aligned}\phi^{*T}(\underline{y}_1^T) &= 1 \text{ and} \\ \phi^{*n-1}(\underline{y}_1^n) &= \begin{cases} 1 \text{ if } \rho_{n-1}(\underline{y}_1^{n-1}) \leq \lambda L + E[W_n(\underline{Y}_1^n, \lambda)|\underline{y}_1^{n-1}] \\ 0 \text{ otherwise, for } n = 1, 2, \ldots T. \end{cases}\end{aligned} \tag{7.12}$$

Also,

$$\begin{aligned}W_T(\underline{y}_1^T, \lambda) &= \rho_T(\underline{y}_1^T) \\ W_{n-1}(\underline{y}_1^{n-1}, \lambda) &= min[\rho_{n-1}(\underline{y}_1^{n-1}, \lambda), \lambda L + E[W_n(\underline{Y}_1^n, \lambda)|\underline{y}_1^{n-1}]] \text{ for } n = 2, 3, \ldots, T, \\ W_0 &= min[\rho_0, \lambda L + E[W_1(\underline{Y}_1^1, \lambda)]]\end{aligned} \tag{7.13}$$

We need the following lemma.

Lemma 1 *The functions $W_n(\underline{y}_1^n, \lambda)$ for given received codewords, $\underline{y}_1^n$, is a continuous and monotonically increasing function of λ, for $n = 0, 1, \ldots, T$.*

Proof: By induction. Clearly, $W_T(\underline{y}_1^T, \lambda)$ is independent of λ and hence is a monotonically increasing and continuous function of λ for all $\underline{y}_1^n$. Assume that $W_m(\underline{y}_1^m, \lambda)$ is a monotonically increasing continuous function of λ, for $m = n+1, \ldots T$. Let $\lambda_1 \geq \lambda_2 \geq 0$. Then, we have, for monotonicity,

$$\begin{aligned}W_n(\underline{y}_1^n, \lambda_1) &= min[\rho_n(\underline{y}_1^n), \lambda_1 L + E[W_{n+1}(\underline{Y}_1^{n+1}, \lambda_1)|\underline{y}_1^n]] \\ &\geq min[\rho_n(\underline{y}_1^n), \lambda_2 L + E[W_{n+1}(\underline{Y}_1^{n+1}, \lambda_2)|\underline{y}_1^n]] \\ &= W_n(\underline{y}_1^n, \lambda_2).\end{aligned} \tag{7.14}$$

Analogously, $W_n(\underline{y}_1^n, \lambda_1)$ is the minimum of a two continuous functions.

Proof of Theorem 4: From equation (7.12) and Lemma 1, if $\lambda_1 \geq \lambda_2 \geq 0$, then $\rho_n(\underline{y}_1^n) \leq \lambda_2 L + E[W_{n+1}(\underline{Y}_1^{n+1} n, \lambda_2)|\underline{y}_1^n] \leq \lambda_1 L + E[W_{n+1}(\underline{Y}_1^{n+1} n, \lambda_1)|\underline{y}_1^n]$. This means that $\phi_2^{*n}(\underline{y}_1^n) = 1 \implies \phi_1^{*n}(\underline{y}_1^n) = 1$. Hence proved.

Theorem 4 shows that optimal feedback generation rules for different λ are embedded. This property is very useful for progressive transmission. Consider the transmission of a single source vector using the transmission scheme discussed here. Suppose the transmission starts with the decoder using an optimal feedback generation rule for certain λ. Theorem 4 implies that, at any step in transmission, the decoder can switch to an optimal feedback generation rule for a lower λ, *i.e.* a higher rate, *without losing optimality*. It never happens that, for a set of received codewords, the feedback generation rule designed for a higher rate sends an ACK and one designed for lower rate sends a NACK.

The collection of optimal feedback generation rules can be characterized further as follows.

7.6 The Feedback-Threshold function

For the remaining part of the chapter, let $\boldsymbol{\phi}_\lambda^*$, (and ϕ_λ^{*n}) denote the optimal feedback generation rule (and respectively, optimal feedback generation maps) for the Lagrangian rate penalty λ. For any sequence of received codewords $\underline{y}_1^n \in \mathcal{Y}^{nL}$,, consider the set $B(\underline{y}_1^n) \stackrel{def}{=} \{\lambda : \phi_\lambda^{*n}(\underline{y}_1^n) = 1\}$. From eq. (7.12), this set is the same as $\{\lambda : \rho_n(\underline{y}_1^n) \leq \lambda L + E[W_{n+1}(\underline{Y}_1^{n+1}, \lambda)|\underline{y}_1^n]\}$. As the function $\lambda L + E[W_{n+1}(\underline{Y}_1^{n+1}, \lambda)|\underline{y}_1^n]$ is continuous as a function of λ, and $B(\underline{y}_1^n)$ is the inverse image of $[\rho_n(\underline{y}_1^n), \infty)$ under that function, $B(\underline{y}_1^n)$ is a closed set. From Theorem 4, $B(\underline{y}_1^n)$ is of the form, $[\lambda_0, \infty)$, for some number $\lambda_0 \geq 0$, which depends on $\underline{y}_1^n$. Define a function $\Lambda^{*n}(\underline{y}_1^n) : \mathcal{Y}^{nL} \to [0, \infty)$ as,

$$\Lambda^{*n}(\underline{y}_1^n) \stackrel{def}{=} \inf\{\lambda \in B(\underline{y}_1^n)\}. \tag{7.15}$$

Then clearly, the random variable $\Lambda^{*n}(\underline{Y}_1^n)$, has the property that $\phi_\lambda^{*n}(\underline{Y}_1^n) = 1$ if and only if $\Lambda^{*n}(\underline{Y}_1^n) \leq \lambda$ for all $\lambda \geq 0$. Hence we have the following interesting result.

Theorem 5 *The optimal feedback generation rules* ϕ_λ^* *satisfy,*

$$\phi_\lambda^{*n}(\underline{Y}_1^n) = u(\lambda - \Lambda^{*n}(\underline{Y}_1^n)) \text{ (a.e.) for } n = 0, 1, \ldots T, \text{ and } \lambda \geq 0, \tag{7.16}$$

where u *is the unit step function,* i.e. $u(\lambda) = 1$ *if* $\lambda \geq 0$ *and* 0 *otherwise.*

Proof: The proof is just outlined above. Note that eq. (7.16) is true only for the optimal feedback generation rules obtained from eq. (7.10) (or eq. (7.12)).

We shall refer to the functions $\Lambda^{*n}(\underline{y}_1^n)$ as the *Feedback-Threshold functions or maps.*

The result is interesting because it "reduces" the task of designing a different feedback generation rule for every λ to constructing a single collection of maps Λ^{*n} from which all optimal feedback generation rules can be obtained.

7.7 Characterization of Feedback-Threshold Function

To characterize $\Lambda^{*n}(\underline{Y}_1^n)$ further , consider the following definitions. Define, for any feedback generation map $\boldsymbol{\phi}$ with $\phi^T(\underline{Y}_1^T) = 1$, the function $D^T(\boldsymbol{\phi}, \underline{Y}_1^T) \stackrel{def}{=} \rho^T(\underline{Y}_1^T)$. And define recursively,

$$\Delta D^n(\boldsymbol{\phi}, \underline{Y}_1^n) \stackrel{def}{=} \rho^n(\underline{Y}_1^n) - E\left[D^{n+1}(\boldsymbol{\phi}, \underline{Y}_1^{n+1})|\underline{Y}_1^n\right] \text{ for } n = 0, 1, \ldots T-1, \tag{7.17}$$

$$D^n(\boldsymbol{\phi}, \underline{Y}_1^n) \stackrel{def}{=} \rho^n(\underline{Y}_1^n) - (1 - \phi^n(\underline{Y}_1^n))\Delta D^n(\boldsymbol{\phi}, \underline{Y}_1^n) \text{ for } n = 0, 1, \ldots T-1. \tag{7.18}$$

Analogously define

$$R^T(\boldsymbol{\phi}, \underline{Y}_1^T) \stackrel{def}{=} TL \tag{7.19}$$

$$\Delta R^n(\boldsymbol{\phi}, \underline{Y}_1^n) \stackrel{def}{=} E\left[R^{n+1}(\boldsymbol{\phi}, \underline{Y}_1^{n+1})|\underline{Y}_1^n\right] - nL \text{ for } n = 0, 1, \ldots T-1, \tag{7.20}$$

$$R^n(\boldsymbol{\phi}, \underline{Y}_1^n) \stackrel{def}{=} nL + (1 - \phi^n(\underline{Y}_1^n))\Delta R^n(\boldsymbol{\phi}, \underline{Y}_1^n) \text{ for } n = 0, 1, \ldots T-1. \tag{7.21}$$

Notice that $D^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ and $R^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ depend only on $\phi^i(\underline{Y}_1^i), i = n, n+1, \ldots, T$. Also $\Delta D^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ and $\Delta R^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ do not depend on the value of $\phi^n(\underline{Y}_1^n)$ but only on $\phi^i(\underline{Y}_1^i), i = n+1, \ldots, T$.

It is straightforward to verify that $D^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ equals $D(\boldsymbol{\phi}, \mathbf{c}^*)$ for $n = 0$, where $D(\boldsymbol{\phi}, \mathbf{c})$ is defined in eq. (7.2) and $\mathbf{c}^*$ is defined in eq. (7.5). Similarly $R^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ equals $R(\phi)$ in eq. (7.1). Also $\Delta R^n(\boldsymbol{\phi}, \underline{Y}_1^n) \geq L > 0$ for any n.

Equations (7.18) and (7.21) isolate the dependence of $D^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ and $R^n(\boldsymbol{\phi}, \underline{Y}_1^n)$ on the function $\phi^n(\underline{Y}_1^n)$. Also, by definition, $(1 - \phi^n(\underline{Y}_1^n)) \geq 0$. Therefore the following lemma about separation of minimizations holds.

Lemma 2 *For any Lagrange Multiplier $\lambda \geq 0$, and for $n = 0, 1, \ldots T-1$,*

$$\min_{\phi^i(\underline{y}_1^i), i=n,n+1,\ldots,T} D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda R^n(\boldsymbol{\phi}, \underline{y}_1^n) =$$

$$\rho^n(\underline{y}_1^n) + \lambda n L + \min_{\phi^n(\underline{y}_1^n)} \left((1 - \phi^n(\underline{y}_1^n)) \left(\min_{\phi^i(\underline{y}_1^i), i=n+1,\ldots,T} \{ -\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n) \} \right) \right)$$

Consequently, to minimize $D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda R^n(\boldsymbol{\phi}, \underline{y}_1^n)$, we must set $\phi^n(\underline{y}_1^n) = 0$ if and only if $\min_{\phi^i(\underline{y}_1^i), i=n+1,\ldots,T} (-\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n)) < 0$.

With these results, we are equipped to show the main result of this section.

Theorem 6

$$\Lambda^{*n}(\underline{y}_1^n) = \sup_{\phi^i(\underline{y}_1^i), i=n,n+1,\ldots,T} \frac{\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n)}{\Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n)} \tag{7.22}$$

Proof: First, note from the recursive definition (eq. (7.17) and (7.20)) that if functions $\phi^i(\underline{y}_1^i), i = n+1, \ldots, T$ minimize $D^{n+1}(\boldsymbol{\phi}, \underline{y}_1^{n+1}) + \lambda R^{n+1}(\boldsymbol{\phi}, \underline{y}_1^{n+1})$ for all $\underline{y}_1^{n+1}$, then they minimize $-\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n)$. Second, define

$$\hat{\lambda}^n(\underline{y}_1^n) \stackrel{def}{=} \sup_{\phi^i(\underline{y}_1^i), i=n,n+1,\ldots,T} \frac{\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n)}{\Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n)}$$

We shall show that $\Lambda^{*n}(\underline{y}_1^n) = \hat{\lambda}^n(\underline{y}_1^n)$, *i.e.* if $\lambda \geq \hat{\lambda}^n(\underline{y}_1^n)$ then setting $\phi_\lambda^{*n}(\underline{y}_1^n) = 1$ and if $\lambda < \hat{\lambda}^n(\underline{y}_1^n)$ then $\phi_\lambda^{*n}(\underline{y}_1^n) = 0$, is optimal.

Case 1: For any $\lambda > \hat{\lambda}^n(\underline{y}_1^n)$ and any $\boldsymbol{\phi}$, we have ,

$$\begin{aligned} & -\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n) \\ = \quad & \left(-\frac{\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n)}{\Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n)} + \lambda \right) \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n) \\ > 0 \quad & \text{as } \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n) > 0. \end{aligned}$$

Therefore, by Lemma 2, $\phi_\lambda^{*n}(\underline{y}_1^n) = 1$.

Case 2: Similarly, if $\lambda < \hat{\lambda}^n(\underline{y}_1^n)$ then, by definition of supremum, there is a feedback generation rule $\boldsymbol{\phi}'$ such that, $\lambda < \frac{\Delta D^n(\boldsymbol{\phi}', \underline{y}_1^n)}{\Delta R^n(\boldsymbol{\phi}', \underline{y}_1^n)} \leq \hat{\lambda}^n(\underline{y}_1^n)$. consequently, $-\Delta D^n(\boldsymbol{\phi}', \underline{y}_1^n) + \lambda \Delta R^n(\boldsymbol{\phi}', \underline{y}_1^n) < 0$. By Lemma 2 we must set $\phi_\lambda^{*n}(\underline{y}_1^n) = 0$.

Case 3: If $\lambda = \hat{\lambda}^n(\underline{y}_1^n)$, for any $\boldsymbol{\phi}$, $-\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \lambda \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n) \geq 0$. Therefore, we can safely set $\phi_\lambda^{*n}(\underline{y}_1^n) = 1$ without any penalty. Therefore, we can set $\phi_\lambda^{*n}(\underline{y}_1^n) = u(\lambda - \hat{\lambda}(\underline{y}_1^n))$. Hence Theorem 6 holds.

Case 3 leads us to more explicit characterization of feedback-threshold functions $\Lambda^{*n}(\underline{Y}_1^n)$.

Lemma 3 *Consider the design of optimal feedback generation rule for* $\hat{\lambda}^n(\underline{y}_1^n)$, i.e. *the solution of minimization problem*

$$\min_{\phi^i(\underline{y}_1^i), i=n+1,\ldots,T} \left\{ -\Delta D^n(\boldsymbol{\phi}, \underline{y}_1^n) + \hat{\lambda}(\underline{y}_1^n) \Delta R^n(\boldsymbol{\phi}, \underline{y}_1^n) \right\}. \tag{7.23}$$

Then $\boldsymbol{\phi}^*_{\hat{\lambda}^n(\underline{y}_1^n)}$ *is a solution to the above minimization if and only if*

$$-\Delta D^n(\boldsymbol{\phi}^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n) + \hat{\lambda}(\underline{y}_1^n) \Delta R^n(\boldsymbol{\phi}^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n) = 0.$$

Proof: Establishing sufficiency is straightforward as, for any feedback generation rule, and hence for $\boldsymbol{\phi}^*_{\hat{\lambda}^n(\underline{y}_1^n)}$, by definition of $\hat{\lambda}^n(\underline{y}_1^n)$ we must have,

$$\begin{aligned} & \frac{\Delta D^n(\boldsymbol{\phi}^*_{\hat{\lambda}(\underline{y}_1^n)}, \underline{y}_1^n)}{\Delta R^n(\boldsymbol{\phi}^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n)} \leq \hat{\lambda}^(\underline{y}_1^n) \\ \Longrightarrow \quad & -\Delta D^n(\boldsymbol{\phi}^*_{\hat{\lambda}(\underline{y}_1^n)}, \underline{y}_1^n) + \hat{\lambda}(\underline{y}_1^n) \Delta R(\boldsymbol{\phi}^*_{\hat{\lambda}(\underline{y}_1^n)}, \underline{y}_1^n) \geq 0. \end{aligned} \tag{7.24}$$

We show the necessity as follows. By definition of the minimum,

$$\begin{aligned} & -\Delta D^n(\phi^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n) + \hat{\lambda}(\underline{y}_1^n)\Delta R(\phi^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n) \\ \leq \quad & -\Delta D^n(\phi, \underline{y}_1^n) + \hat{\lambda}^n(\underline{y}_1^n)\Delta R^n(\phi, \underline{y}_1^n) \text{ for all } \phi \\ \Longrightarrow \quad & 0 \leq -\Delta D^n(\phi^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n) + \hat{\lambda}^n(\underline{y}_1^n)\Delta R(\phi^*_{\hat{\lambda}^n(\underline{y}_1^n)}, \underline{y}_1^n) \\ \leq \quad & \Delta R^n(\phi, \underline{y}_1^n)(-\frac{\Delta D^n(\phi, \underline{y}_1^n)}{\Delta R^n(\phi, \underline{y}_1^n)} + \hat{\lambda}^n(\underline{y}_1^n)) \end{aligned}$$

As $\Delta R^n(\phi, \underline{y}_1^n)$ is bounded above by TL and by definition of supremum the second term in right hand side can be made arbitrarily small, we must have the left hand side equal to zero. Therefore Lemma 3 is established.

Finally, we establish the uniqueness of $\Lambda^{*n}(\underline{Y}_1^n)$.

Theorem 7 *For some non-negative λ, if*

$$\min_{\phi^i(\underline{y}_1^i), i=n+1,\ldots,T} \{-\Delta D^n(\phi, \underline{y}_1^n) + \lambda \Delta R^n(\phi, \underline{y}_1^n)\} = 0 \tag{7.25}$$

then $\lambda = \hat{\lambda}^n(\underline{y}_1^n)$. Therefore equation (7.25) is necessary and sufficient condition for computation of $\hat{\lambda}^n(\underline{y}_1^n)$ and hence that of $\Lambda^{*n}(\underline{y}_1^n)$.

Proof: We have already seen in Lemma 3 that $\hat{\lambda}^n(\underline{y}_1^n)$ satisfies equation (7.25). It is straightforward to check that, for $a, b \geq 0$ and $c, d > 0$, if λ_1 and λ_2 are such that $-a + \lambda_1 b = 0 \leq -c + \lambda_1 d$ and $-c + \lambda_2 d = 0 \leq -a + \lambda_2 b$, then $\lambda_1 = \lambda_2$. Therefore no other λ can satisfy equation (7.25). Hence proved.

7.8 Progressive Transmission and Receiver Driven Rate Control

The embeddedness of the optimal policies and the existence of $\Lambda^*(Y_1^n)$ is a very useful property that can come in handy in a variety of application scenarios. Note that, by extending the definition of NACK to mean a permission to continue transmission, ACK/NACK can be actively used for receiver driven rate control. As the optimal policies are embedded, the progressive transmission, say that of an image, can be accomplished without losing optimality at the terminal and at intermediate transmission budgets. The quality of the received image can be successively improved as new bits are received. The optimal feedback generation rules reveals a very simple structure in the form of Theorem 5. If the feedback-threshold function $\Lambda^*(Y_1^n)$ is known or if it can be approximated, then the ACK/NACK generation for a range of operating points can be accomplished at once. Secondly, the receiver can switch from operating at a low average -transmission rate to a higher average transmission rate, in the middle of a transmission, without losing optimality of the rate-distortion tradeoff. The rate control technique can be potentially useful in the following situations.

Delay-limited Reconstruction: In interactive applications such as video conferencing, a quick reconstruction at low transmission budget for the foreground, and slow but detailed and error free reconstruction of the background might be used, provided such a separation is available. Controlling the ACK/NACK of the appropriate packets may allow a trade off between reconstruction speed and quality.

Bandwidth/Data Rate-limited Reconstruction: While receiving statistically multiplexed streams of variable rate at a receiver, the transmission rates of one or more of the streams can be controlled using appropriate ACK/NACK feedback.

Computation-Limited and Buffer-Size limited Reconstruction: Similarly, for a multitasking environment such as a server at a base station, the CPU usage and memory allocated to an incoming stream over a noisy channel can be variable. Based on the current processing capability, some amount of control can be exercised by appropriate operating point selection in the feedback generation rules.

Tolerance-Limited Reconstruction: In digital encoding of video, the intra-coded frames generate a lot more data than predictive or "inter" coded frames. The predictive frames can be thought of as incremental information. In a low noise environment, fewer intra-coded frames can be transmitted while in a noisy environment they need to be more frequent. The switching between the two for best rate-distortion performance can be accomplished by the use of ACK/NACK feedback.

7.9 Conclusions

In this chapter we addressed a slightly more general problem of transmission of loss tolerant sources over noisy channels in the presence of ACK/NACK feedback. We extend the interpretation of NACK to mean "a permission to continue transmission", which permits the transmitter to transmit additional redundancy, or even completely new information on receiving NACK. We continued the first principles approach from the last chapter to establish optimal feedback generation rules and optimal reproduction rule, for an embedded encoder which transmits new information about the source at each transmission. We showed the close link between the transmission of such a source with ACK/NACK feedback and the PTSVQ. We also showed that the PTSVQ property holds for continuous valued output. Hence we obtain that the optimal feedback generation policies are embedded. Then we investigated the structure of the embedded policies further and showed that the optimal policies for all Lagrange multipliers λ have a simple form in terms of λ and a feedback-threshold function of the received codewords. We also investigated the structure of the feedback-threshold function further and obtained a necessary and sufficient condition for computing its value at each sequence of received codewords.

We have not provided an explicit algorithm for computation of the feedback-threshold function. But such an algorithm can be conceived. When the observations (received codewords) take discrete values, the pruning algorithm of PTSVQ [15] is useful. It can be shown that the value of the feedback-threshold function at a node, is equal to the slope of a subtree at that node, just before it gets pruned. For continuous valued observations, an iterative successive approximation algorithm based on eq. (7.25) may be found. We have not addressed the algorithm design in this thesis.

Chapter 8

Conclusions and Future Work

8.1 The Theme

In the thesis we consider transmission schemes of loss-tolerant sources, mainly images and synthetic sources over noisy and lossy channels. We propose solution and design schemes for a collection of problems which are closely related and at the same time require different methodology/approaches.

The common threads in the thesis are:

- **Joint Source-Channel Rate Scalability and Optimized Progressive Transmission:** The existence of rate-distortion curves, that is, the possibility of constructing approximate reconstructions makes the problem of compression and transmission of loss tolerant multimedia sources different from that for data. Rate Scalable source coders offer the flexibility of selecting the rate from a single bitstream. In absence of noise or loss they allow progressive transmission of the source where the source is constructed with increasing quality at the receiver as the receiver gets more and more bits. The emphasis of the thesis, in particular that of Chapters 2, 4, 5, 6, and 7 was on extending this property in the presence of noise and loss. Chapters 2, 6, and 7 accomplish this with the help of a feedback channel. In Chapters 4, 5 we provide an approach to carry our progressive transmission in the absence of feedback. We achieve operational optimality of the joint source-channel coder by unequal error protection of a rate scalable source coder. On the other hand, progressive character is obtained by a *rate compatible* channel code family, and by *scheduling* of source and parity bits for operational optimality at a number of rates. The scheduling generates a single stream of bits, whose prefixes carry optimally allocated source and channel bits for the corresponding bit budget. In this way the proposed systems achieve a "Joint Source-Channel Rate Scalability ' in the absence of feedback'.

The optimality of rate allocations in feedback based schemes is discussed in Chapter 2. The combination of HARQ protocol and rate-scalable source coder automatically carries out optimal allocation of source-bits and channel bits during the transmission. This allocation is also "automatically" adaptive, when the channel is a time-varying (finite-state) channel.

Chapters 6, and 7 establish a optimized rate-scalability or progressivity of a totally different kind. There the receiver can control the operating point on the rate distortion curve by selecting the feedback appropriately. We establish that the operating points can be switched from lower rate to higher rate in the middle of the transmission of a single source-vector. In this way, operationally optimal feedback generation policies are shown to be rate-scalable.

- **Feedback, No Feedback and Limited Feedback:** The thesis, essentially for the first time, (with the exception of independent work of [37]) makes use of feedback channel from a joint-source-channel coding perspective. The use of a simple ACK/NACK feedback can significantly improve the performance of a joint source-channel coding system. This was demonstrated by designed illustrative image transmission systems which achieve up to 1.2 dB improvement in PSNR for the chosen binary symmetric channels and up to 2 dB improvement in PSNR for the chosen Gilbert-Elliot channels, compared to state of the art high performance joint source-channel coding systems which use pure FEC.

We restrict our attention to systems which use the feedback channel sparingly, and in Chapters 2 and 3 provide explicit algorithms to control the use of feedback which yields optimal tradeoff between the parameters of interest, namely throughput vs.. complexity. That allows us to compare the benefit of using feedback with that of not using feedback as mentioned above.

In Chapters 6, and 7 we again undertake the investigation of joint source-channel schemes which use feedback. In Chapters 2 and 3 feedback is mainly used for error control. The automatic adaptive source-channel rate allocation is a bonus. In Chapters 6, and 7 we take the first principles approach and devise feedback generation schemes and decoders which explicitly use the distortion metric. We characterize the optimal schemes and also provide a number of suboptimal but efficient solutions. Simulation results show that this can yield large gains over BER-based feedback generation schemes which treat the source-coder bits equally.

The main message of this investigation is that feedback is useful, and can be exploited in a controlled fashion to yield significant gains in joint source-channel coding systems.

- **Sequential Nature of the Solutions:**

  Though the optimization problems encountered in Chapters 2 and 3 are different from those in Chapters 4 , 5, and Chapters 6, and 7, the solutions are related in some sense. The problems in Chapters 2 and 3 are solved by Controlled Markov Chain (Markov Decision Processes) approach. The solution of problems in Chapters 4 can be thought of as controlled Markov Chains in the absence of state observations. Solution to the problems, in Chapters 6, and 7 where only the decoder is involved, are also shown to be problems in sequential decision theory which implies design in the absence of observation. This way there is an underlying unity in the techniques presented.

8.2 Future Research Directions

A number of interesting questions can be asked based on the work presented in the thesis, which merit further investigation.

- **Image Transmission: Tradeoff of Rate-Scalability and Robustness with flexible selection of image coders, under small feedback:** We argued in Chapter 2 that a combination of a completely embedded source coder and an optimized hybrid ARQ protocol is the best combination for maximizing end to end image quality. When ACK/NACK feedbacks are used for every packet, the quality is maximized for all transmission budgets. On the other hand, in Chapter 4 we designed schemes for progressive transmission of a fixed embedded source-coder in the absence of feedback, under the assumption that the only the longest correctly decoded prefix of the source bitstream is useful for reconstruction. This assumption is true for efficient rate scalable source coders like SPIHT. But the source coders can be modified to increase robustness by giving up some of the efficiency in rate scalability *e.g.* the idea presented in [51]. This is done by generating several independent bitstreams instead of a single bitstream. In general, more independent streams result in a loss in the distortion-rate performance and less efficient rate scalability. If the use of feedback channel is severely limited, (say 2 or 3 uses of the feedback channel for entire transmission), the situation is somewhere in between the cases of Chapters 2 and 4. It is interesting to investigate how the selection of source-coders, unequal error protection, and feedback channel be combined to obtain best end-to-end performance for a fixed transmission budget and rate scalable transmission schemes which are efficient at intermediate transmission budget.

- **HARQ protocols on Time Varying Channels: Tradeoff of Delay, Throughput and Feedback usage, under interleaving:** In the absence of feedback channel, the way to combat time-variability of the channels is to use interleavers and burst error correction codes. When the feedback channel is available, appropriately designed HARQ protocols work well. If the feedback usage is severely constrained a combination of the two approaches is needed. What combination gives the best tradeoff of performance parameters, such as delay, throughput and feedback usage, is of interest. Also, exact analysis/design techniques for constrained HARQ protocols over time varying channels need to be investigated.

- **On-the-Fly HARQ Protocol Design and Adaptive Negotiation for Time Varying Channels :** It can be argued that fast changes in the channel are best handled by interleavers/burst error correction, moderately slowly varying channels are handled by HARQ protocol design as discussed in Chapter 2 and 3. If the channel variation is drastic but slow a change in the protocol used might be beneficial. An interesting question is how to carry out quick protocol design and smooth negotiation of the protocol between the transmitter and the receiver, so that, channel changes can be tracked by protocols which yield high throughput for the current channel conditions.

- **Packet Length selection:** The algorithms developed in the thesis, and a number of works presented by other researcher in literature, assume fixed, known, pre-specified packet (block, frame, word) lengths for source/channel coding or transmission. We have seen that, sometimes the complexity and the performance of a number of schemes crucially depends on the packet lengths chosen. A systematic way of selection of packet lengths suitable for any particular application/ transmission scheme , in itself merits investigation.

- **Combined Source-channel Encoder design in the presence of feedback:** As described in the classification of schemes with feedback in 6, Chapters 2 and 3 are active-encoder active-decoder systems which are active only for error-protection purposes. On the other hand, the systems described in Chapters 6 and 7 are passive-encoder, active-decoder systems, which are "true" joint source-channel systems, as the decoders cannot be decomposed into the steps of error-correction followed by source-reconstruction. As we mentioned in Chapter 6, it is of considerable interest to design active-encoder joint source-channel coding systems, in which the source-channel encoder (quantizer + index assignment) is aware or the channel statistics as well as of the fact that a feedback channel is available.

- **Ultimate Goal: Delay-Complexity-Memory constrained communication:** An ultimate goal is to design a communication system, in which a collection of distributed sensors, encode correlated sources, in a scalable or non-scalable fashion, using single or multiple descriptions, communicate to a destination on a network with lossy links, using forward error correction, interleaving, hybrid or pure ARQ or more general feedback based protocols, single or multiple routes, single or multiple transmit or received antennas, to yield the best reproduction of the source or best extracted useful information, in a given finite time, with a given limited complexity and memory.[1]

8.3 In Closing

It is exciting to be living in these revolutionary times.

[1] As in many textbooks, this problem has been left as an exercise to the reader.

BIBLIOGRAPHY

[1] S. Appadwedula, D. Jones, K. Ramchandran, and I. Kozintsev. Joint source channel matching for a wireless communication link. In *Proceedings of ICC-98*. IEEE, 1998.

[2] P. Bahl and B. Girod, editors. *IEEE Commun. Mag.,Special Section on Wireless Video*. IEEE, June 1998.

[3] C. W. Baum and V. V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Trans. Info. Theory*, 40(6):1994–2007, November 1994.

[4] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[5] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[6] R.E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.

[7] J. C. Brailean, T. Sikora, and T. Miki, editors. *Signal Processing: Image Commun*, volume 14, May 1999. Special Issue on Error Resilience.

[8] L. Breiman, J.H Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

[9] V. Chande and N. Farvardin. A dynamic programming approach to constrained feedback hybrid ARQ design. In *IEEE Symposium on Information Theory*, page 286, August 1998.

[10] V. Chande and N. Farvardin. A joint source-channel coding scheme for robust image transmission. In *Proc. Data Compression Conference (DCC)*, page Poster, Snowbird, Utah, March 1998.

[11] V. Chande and N. Farvardin. Joint source-channel coding for progressive transmission of embedded source-coders. In *Proc. Data Compression Conference (DCC)*, pages 52–61, Snowbird, Utah, March 1999.

[12] V. Chande, H. Jafarkhani, and N. Farvardin. Joint source-channel coding of images for channels with feedback. In *Proceedings, IEEE Workshop on Information Theory, San Diego*, pages 50–51, Feb. 1998.

[13] D. Chase. Code combining–a maximum likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Trans. Comm.*, COM-33:385–393, May 1985.

[14] Q. Chen and T. Fischer. Robust quantization for image coding and noisy digital transmission. In *Proc. Data Compression Conference (DCC)*, pages 3–12, 1996.

[15] P. A. Chou, T. A. Lookabaugh, and R. M. Gray. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. Info. Theory*, 35(2):299–315, March 1989.

[16] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra. Fec and pseudo-arq for receiver-driven layered multicast of audio and video. Technical Report MSR-TR-99-86, Microsoft Research, 1999.

[17] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra. Error control for receiver-driven layered multicast of audio and vide. *IEEE Transactions on Multimedia*, 3(1), 2001.

[18] P. Cosman, J. 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*, To appear.

[19] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley Inc., 1991.

[20] R.E.Van Dyck and D.J. Miller. Transport of wireless video using separate, concatenated, and joint source-channel coding. *Proceedings of the IEEE*, 87, 1999.

[21] W. H. R. Equitz and T. M. Cover. Successive refinement of information. *IEEE Trans. Info. Theory*, 37:269–274, March 1991.

[22] N. Farvardin. A study of vector quantization for noisy channels. *IEEE Trans. Info. Theory*, 36(4):799–809, July 1990.

[23] N. Farvardin and V. Vaishampayan. On the performance and complexity of channel optimized vector quantizers. *IEEE Trans. Info. Theory*, 37(1):155–160, January 1991.

[24] T. S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York, 1967.

[25] J. Garcia-Frias and J. D. Villasenor. Joint turbo decoding and estimation of hidden markov sources. *IEEE Journ. Sel. Areas in Comm.*, pages 1671–1679, September 2001.

[26] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, 1991.

[27] H. Gharavi and L. Hanzo, editors. *Proceedings of the IEEE*, volume 87, October 1999. Special Issue on Video Transmission for Mobile Multimedia Applications.

[28] M. Ghosh, N. Mukhopadhyay, and P. K. Sen. *Sequential Estimation*. John Wiley and Sons, 1997.

[29] J. Hagenauer. Rate compatible punctured convolutional (RCPC) codes and their applications. *IEEE Trans. Comm.*, 36(4):389–399, April 1988.

[30] J. Hagenauer. Source controlled channel decoding. *IEEE Trans. Comm.*, 43(9):2449–2457, Sept. 1995.

[31] L. Hanzo, P. J. Cherriman, and J. Streit. *Wireless Video Communications: Second to Third Generation Systems and Beyond*. IEEE Series on Digital and Mobile Communications. IEEE Press, 2001.

[32] G. David Forney Jr. Exponential error bounds for erasure, list and decision feedback schemes. *IEEE Trans. Info. Theory*, pages –, Mar. 1968.

[33] Navin Kashyap and David L. Neuhoff. Variable-rate codes for synchronization with timing. In *Proceedings, Int. Symp. on Info. Theory*, 2000.

[34] P. Knagenhjelm and E. Agrell. The hadamard transform - a tool for index assignment. *IEEE Trans. Info. Theory*, 42(4):1139–1151, July 1996.

[35] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[36] S. Lin and P. S. Yu. A hybrid-ARQ system with parity retransmission for error control of satellite channels. *IEEE Trans. Comm.*, COM-30:1701–1719, July 1982.

[37] H. Liu and M. E. Zarki. Performance of h.263 video transmission over wireless channels using hybrid arq. *IEEE Journ. Sel. Areas in Comm.*, 15(9):1775–1786, 1997.

[38] D. L. Lu and J. F. Chang. Analysis of ARQ protocols by signal flow graphs. *IEEE Trans. Comm.*, COM-37(3):245–251, March 1989.

[39] J. Lu, A. Nosratinia, and B. Aazhang. Progressive source channel coding for images over bursty error channels. In *Proceedings, IEEE Intl. Conf. Image Proc. (ICIP)*. IEEE, 1998.

[40] J. Lu, A. Nosratinia, and B. Aazhang. Progressive joint source-channel coding in feedback channels. In *Proc. Data Compression Conference (DCC)*, pages 140–148, 1999.

[41] X. Lu, E. Erkip, and Y. Wang. Power efficient multimedia communication over wireless channels. *IEEE-JSAC special issue on Recent Advances in Wireless Multimedia*, 2003.

[42] H. S. Malavar. Fast progressive wavelet coding. In *Proc. Data Compression Conference (DCC)*, pages 336–343, Snowbird, Utah, March 1999.

[43] D. Mandelbaum. An adaptive-feedback coding scheme using incremental redundancy. *IEEE Trans. Info. Theory*, pages 388–389, May 1974.

[44] M. W. Marcellin, M. J. Gormish, Ali Bilgin, and M. P. Boliek. An overview of JPEG-2000. In *Proceedings, Data Compression Conference*, 2000.

[45] F. J. McWilliams and N.J.A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977.

[46] A. Mohr, E. Riskin, and R. Ladner. Graceful degradation over packet erasure channels through forward error correction. In *Proc. Data Compression Conference (DCC)*, pages 92–101, Snowbird, Utah, March 1999.

[47] J. P. M.Shalkwijk. A class of simple and optimal strategies for block coding on the binary symmetric channels with noiseless feedback. *IEEE Trans. Info. Theory*, 17:283–287, May 1973.

[48] J. Ooi and G. Wornell. Fast iterative coding for feedback channels. In *Proceedings, IEEE Symp. on Info. Theory*, page 133, 1997.

[49] E. Ordentlich, M. J. Weinberger, and G. Seroussi. A low complexity modelling approach for embedded coding of wavelet coefficients. In *Proc. Data Compression Conference (DCC)*, pages 408–417, Snowbird, Utah, March 1998.

[50] M. Rabbani and R. Joshi. An overview of the JPEG-2000 still image compression standard. *Signal Processing-Image Communication*, 17:3–48, 2002.

[51] J. Rogers and P. Cosman. Robust wavelet zerotree image compression with fixed-length packetization. In *Proc. Data Compression Conference (DCC)*, Snowbird, Utah, 1998.

[52] A. Said and W. A. Pearlman. A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, 6(3):243–250, Jun. 1996.

[53] S. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt. Multiple description wavelet based image coding. In *Proc. Intl. Conf. Image. Proc.*, Chicago, IL, Oct. 1998.

[54] Sergio D. Servetto and Klara Nahrstedt. Video streaming over the public internet: Multiple description codes and adaptive transport protocols. In *Proceedings ICIP*, pages 85–89, 1999.

[55] N. Seshadri and C. Sundberg. List viterbi decoding algorithm with applications. *IEEE Tras. Comm.*, 42:313–323, Feb./Mar.Apr. 1994.

[56] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Proc.*, 41:3445–3462, December 1993.

[57] P. Sherwood and K. Zeger. Progressive image coding on noisy channels. In *Proc. Data Compression Conference (DCC)*, pages 72–81, March 1997.

[58] P. G. Sherwood, X. Tian, and K. Zeger. Channel code blocklength and rate optimization for progressive image transmission. In *Wireless Communications and Networking Conference (WCNC)*, pages 978–982, New Orleans, Louisiana, September 1999.

[59] P. G. Sherwood and K. Zeger. Error protection for progressive image transmission over memoryless and fading channels. *IEEE Trans. Comm.*, pages 1555–1559, December 1998.

[60] M. Srinivasan. Iterative decoding of multiple description. In *Proceedings, Data Compression Conference*, pages 463–472, 1999.

[61] M. Srinivasan and R. Chellappa. Multiple description subband coding. In *Proc. Intl. Conf. Image. Proc.*, Oct. 1998.

[62] M.T. Sun and A. R. Reibman, editors. *Compressed Video over Networks*. Marcel Dekker, 2000.

[63] N. Tanabe and N. Farvardin. Subband image coding using entropy coded quantization over noisy channels. *IEEE Journal Sel. Areas. Comm*, 10:926–943, June 1992.

[64] A.S. Tannenbaum. *Computer Networks*. Prentice-Hall, 1996.

[65] George I. Turin. Signal design for sequential detection systems with feedback. *IEEE Trans. Info. Theory*, 11:401–408, July 1965.

[66] H. Wang and N. Moayeri. Finite-state markov channel- a useful model for radio communication channel. *IEEE Trans. Veh. Technol.*, 44:163–171, February 1995.

[67] S. Wicker. *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.

[68] S. B. Wicker and M. D. Bartz. Type-II hybrid ARQ protocols using punctured MDS codes. *IEEE Trans. Comm.*, April 1994.

[69] Y.Wang and Q. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86:974–997, May 1998.