A Framework for Low-Complexity Communication over Channels with Feedback

by

James Meng-Hsien Ooi

S.B., Massachusetts Institute of Technology (1992)
S.M., Massachusetts Institute of Technology (1993)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1998

Author: James Ooi
Department of Electrical Engineering and Computer Science
September 10, 1997

Certified by:
Gregory W. Wornell
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by:
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

A Framework for Low-Complexity Communication over Channels with Feedback

by
James Meng-Hsien Ooi

Submitted to the Department of Electrical Engineering and Computer Science
on September 10, 1997, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis develops a framework for low-complexity communication over channels with feedback. In this framework, which is referred to in the thesis as the compressed-error-cancellation framework, data are sent via a sequence of messages: the first message contains the original data; each subsequent message contains a source-coded description of the channel distortions introduced on the message preceding it. The usefulness and flexibility of the framework is demonstrated by applying it to a number of fundamental feedback communication problems.

The framework is first used for coding over known single-user channels. For discrete memoryless channels with complete, noiseless feedback (DMC$_f$'s), a coding scheme exploiting low-complexity lossless source coding algorithms is developed, and the associated encoder and decoder are shown to use a number of computations growing only linearly with the number of channel inputs used (linear complexity). The associated error exponent is shown to be optimal in an appropriate sense and implies that capacity is achievable. Simulations confirm the analytically predicted behavior. For the class of channels with memory known as discrete finite-state channels with complete, noiseless feedback (DFSC$_f$'s), the framework is used to develop linear-complexity coding schemes performing analogously in terms of rate and reliability to the schemes developed for DMC$_f$'s.

The framework is then used for coding over unknown DFSC$_f$'s. A linear-complexity universal communication scheme whose rate varies with the quality of the realized channel is developed and analyzed. The asymptotic rate and reliability characteristics of this universal scheme are shown to be similar to those of the schemes developed for known channels.

An extension of the compressed-error-cancellation framework is developed for discrete memoryless multiple-access channels with complete, noiseless feedback and leads to linear-complexity coding schemes achieving rates on the frontier of the feedback-free capacity region.

Finally, the compressed-error-cancellation framework is applied to the problem of coding for channels with noisy and partial feedback. The scheme developed for DMC$_f$'s is modified to incorporate Slepian-Wolf coded feedback, resulting in a linear-complexity, capacity-achieving coding scheme with partial, noiseless feedback. This modified scheme, of which the ARQ protocol is shown to be a special case, is then used as an outer code in a concatenated coding arrangement; with a forward error-correcting (FEC) code used as the inner code, a framework emerges for integrating FEC coding with feedback coding, leading to a broad class of coding schemes using various amounts of noiseless feedback. Preliminary investigations on partial-feedback multiple-access scenarios and noisy feedback scenarios are also discussed.

Thesis Supervisor: Gregory W. Wornell
Title: Associate Professor of Electrical Engineering

Acknowledgments

It is my privilege to mention here a number of individuals whose efforts have greatly enhanced this thesis as well as my professional and personal development in general. My thesis supervisor, Professor Gregory Wornell, has provided excellent technical and literary guidance that has greatly improved the content and presentation of this thesis; he has also provided me with much encouragement and given me many opportunities for professional development; I sincerely thank him for his efforts on my behalf. The other members of my thesis committee, Professors Amos Lapidoth and Robert Gallager, have inspired important developments in this thesis and given many thoughtful, constructive criticisms of it. My colleagues, Jeff Ludwig and Shawn Verbout, have had vast, positive influences on my thinking on both technical and personal issues; through their eyes, innumerable truths have been laid bare; their friendship has energized me every day, and I will treasure it always. My partner in life, Amy Yin, has been by my side, giving her utmost love and support throughout this experience; without her, the journey to this milestone would have been far more arduous, empty, and lonely.

Finally, let me mention that this thesis could not have been completed without the resources of the Digital Signal Processing Group in the MIT Research Laboratory of Electronics and the generous financial support given by the Defense Advanced Research Projects Agency, the Office of Naval Research, AT&T, Lucent Technologies, and the MIT Electrical Engineering and Computer Science Department.

Contents

List of Figures

Chapter 1

Introduction

Communication systems are charged with the task of sending data or information from one place to another[1]. A block diagram emphasizing certain elements essential to most communication systems is shown in Figure 1-1. In this figure, the source generates the raw information to be transmitted. This information may be speech or video signals, text, computer programs, or a variety of other data. The encoder takes the source data and processes it so that it is suitable for transmission over the channel. The channel may be, for example, a telephone link, a radio link, or an underwater acoustic link. It may deterministically and/or randomly distort the data transmitted over it. The decoder then processes the output of the channel to try to recover the raw information generated by the source.

The developments in this thesis are concerned only with the inner three blocks of Figure 1-1 — encoder, channel, and decoder. We assume that the source always generates independent binary digits. Because of this assumption, our results are most relevant for communication systems in which the true source is first transformed into such a binary source. An entire branch of information theory known as source coding theory, is devoted to this transformation. Yet we should always keep in mind that communication systems that require this initial transformation are a subset of all possible communication systems and may therefore exclude the best possible system, however "best" may be defined. Nevertheless, the assumption that the true source is initially transformed into a binary source is widely used, traditional, and most importantly allows great simplification of the design and analysis of communication systems.

Shannon [53] developed a powerful mathematical theory, now called information theory, for communication systems. The theory has two major branches, source coding theory, mentioned above, and channel coding theory. As we show in the following chapters, the two branches can hardly be considered separate, but strictly, the work herein lies within channel coding theory, because it concerns the inner three blocks mentioned above.

In channel coding theory, a channel consists of an input, an output, and a probabilistic relationship between the two. The channel is "used" over time by modulating its input — that is, the inputs are varied with time in a way that depends on the information to be sent. Discrete-time channels, to which we restrict our attention throughout the thesis, allow the input to the channel to change at discrete times; the output also changes at the same times. While many real channels allow modulation of the input in continuous time, these channels can often be converted to discrete-time channels.

[1] Sometimes storage media are considered communication systems that transfer data from one time to another.

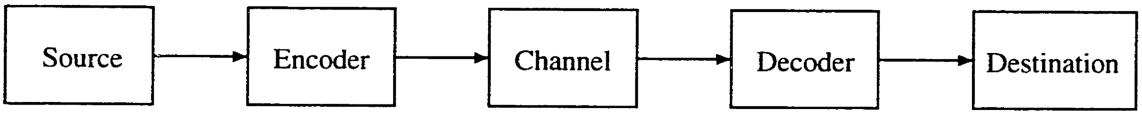


Figure 1-1: Block diagram of a canonical communication system.

Perhaps the most basic example of a noisy channel, a channel that randomly distorts its input, is the binary symmetric channel (BSC). A BSC allows inputs with values 0 or 1 and has outputs with values 0 or 1. An input of 0 becomes 1 with some probability ϵ and an input of 1 becomes a 0 with the same probability. The input passes through undistorted with probability $1 - \epsilon$. The inputs are "flipped" independently, making this an example of a memoryless channel.

Even a novice to channel coding theory can probably imagine that one could convey to another the value of a single binary digit (bit) across a binary symmetric channel very reliably by repeatedly sending that bit. That is, n 0's might be put in to the channel if the value of the bit to be conveyed is 0, and n 1's might be put in if the value to be conveyed is 1. To determine whether the value being conveyed is 0 or 1, one could observe the corresponding n outputs of the channel, count the number of 0's and 1's in the output sequence and estimate the value to be 0 if there are more 0's than 1's and estimate the value to be 1 otherwise. If the estimate is wrong, then a decoding error is said to occur. As n increases, the probability of decoding error decreases. But if we use this procedure to send each of many bits generated by a source, then as n increases, it takes a longer time to send the bits. Because we send one bit per n channel uses, we say the rate of the communication system is $1/n$, the rate being the number of bits we send per channel input we use. In this example, the rate goes to zero as the probability of error goes to zero.

Remarkably, Shannon [53] showed that for any channel in a particular class of channels — a broad class — there is a number C, known as the channel capacity, such that for any rate less than C, an encoder and decoder can be designed so that the probability of decoding error is arbitrarily small. Instead of the rate going to zero as the probability of error goes to zero, the rate can remain constant. Shannon also showed that the probability of error cannot go to zero if the rate is above C.

Shannon showed that this behavior could be achieved by using an appropriate codebook. A codebook is a set of sequences, each of which corresponds to a message. When the source generates a message, the corresponding codebook sequence, or codeword, is put into the channel. The decoder tries to identify the codeword after the channel distorts it and subsequently matches the codeword to a message. Shannon showed that one can construct a codebook with an arbitrarily small probability of decoding error. But the length of the codewords in the codebook, the blocklength, must increase as probability of decoding error decreases. It was subsequently shown (e.g., [23]) that the probability of decoding error decays exponentially with blocklength, but that the rate of decay depends on the rate of the codebook. As the rate approaches the channel capacity, the probability of decoding error decays more and more slowly, demanding longer blocklengths.

But encoding and decoding of codebooks with long blocklengths seem to require vast computational resources. If the codebook has rate R and blocklength n, then there are 2^{nR} codewords in the codebook. Unless the codebook has some special structure, such a codebook requires about $n2^{nR}$ memory elements to store, and the same number of computations to decode with minimum probability of error. As n becomes very large, the computational resources of any known computer are quickly exceeded.

Unfortunately, Shannon gave no constructions of specially structured, easily decodable codebooks for use on a given channel. In fact, he gave no codebook at all—he simply showed that an appropriate codebook exists. As a result, ever since Shannon put forth his celebrated coding theorem, researchers have struggled to find codebooks that attain the promised performance, but can also be decoded with reasonable amounts of computational resources.

Researchers have met with limited success. Justesen codes [32] and more recently Spielman codes [56] are two notable examples of codes with polynomial computational complexity[2] and exponentially decaying error probabilities. In fact, the encoding and decoding computational complexity of Spielman codes is linear in the blocklength. But neither Justesen codes nor Spielman codes give arbitrarily low error probabilities at rates arbitrarily near capacity. Convolutional codes with sequential decoding [44] form another class of low-complexity encoding/decoding schemes. The decoding process suffers from a random amount of computation, which in turn requires large amounts of storage to obtain exponentially decaying error probability. Using a framework he called concatenated coding, Forney [21] gave a class of codes with polynomial computational complexity and error probabilities decaying exponentially with blocklength at any rate below capacity. However, at rates near capacity, as we discuss in Section 2.6, the computation required by concatenated codes is still far too high for practical implementation. Thus, researchers are still searching for low-complexity encoding/decoding systems achieving rates near capacity with arbitrarily low probabilities of decoding error. This thesis develops a framework for designing such systems under special conditions, namely, when feedback is available.

1.1 Feedback in Communication

The availability of feedback in a communication system — i.e., a channel from receiver to transmitter through which the receiver passes the transmitter its observations — generally enables schemes for communicating over the forward channel to have lower computational complexity, higher reliability, higher capacity, or a combination of these advantages, in comparison to feedback-free communication schemes.

The research of Schalkwijk and Kailath [49], Horstein [30], Berlekamp [7], Yamamoto and Itoh [65], Burnashev [11], Kudryashov [34], Gaarder and Wolf [22], Cover and Leung [13], Veugen [61], and many others attests to these advantages. For example, while Shannon [52] showed that the capacity of memoryless channels is not increased by feedback, Gaarder and Wolf [22] showed that the capacity region of a memoryless multiple-access channel can be increased by feedback; in addition, it is straightforward to construct examples of channels with memory whose capacity is increased by feedback (see, e.g., Section 3.5.2). For the BSC with feedback (BSC$_f$), Horstein [30] developed a scheme with low complexity and bit error probability that decays exponentially with decoding delay at any rate below capacity. He also outlined an extension of his scheme for arbitrary discrete memoryless channels with noise-free feedback (DMC$_f$'s) [29]. Berlekamp [7] and Schalkwijk et al. [47, 50] also developed low-complexity strategies for the BSC$_f$ that, at certain rates, guarantee correction of the largest possible fraction of errors. For the Gaussian channel with average-power constraint, Schalkwijk et al. [46, 49] described a low-complexity scheme whose error probability decays with blocklength as an extremely fast double exponential ($2^{-2^{\alpha n}}$) at all rates below capacity. Even the peak-power limited scheme of Schalkwijk and Barron [48] has an

[2] We make the notion of computational complexity more precise in Section 2.6.

error probability that decreases exponentially with blocklength with an error exponent that is significantly larger than that for the corresponding feedback-free channel. Their scheme was adapted for use with arbitrary DMC$_f$'s by Yamamoto and Itoh [65] and provides similar reliability. Computational efficiency is poor, however, since high-rate random block codes are relied upon. More recently, Veugen [59, 60, 61] analyzed a low-complexity scheme based on repeated retransmission of symbols. With this scheme, a price is generally paid in terms of rate — i.e., the resulting schemes are not typically capacity-achieving.

But even though many communication links can be modeled as channels with feedback (see, e.g., [61] for a number of examples), system designers have shunned these low-complexity, high-rate, high-reliability techniques in favor of those that either avoid or minimize their use of feedback. Indeed, when feedback strategies are used, simple, low-rate feedback, automatic-repeat-request (ARQ) protocols or variants thereof are most often chosen. Such choices may sometimes be justified from a purely information-theoretic perspective : given unlimited computational resources for encoding and decoding, exploiting feedback on memoryless single-user channels is inherently inefficient in terms of bandwidth utilization — i.e., the bandwidth allocated for the feedback link is better used to directly increase the rate on the forward link.

When computational resources are limited, however, more sophisticated feedback strategies such as those mentioned above may have a place. Under computational constraints, the use of feedback can actually increase bandwidth efficiency on even memoryless two-way communication links. Specifically, given fixed computational resources, allocating some of the total available bandwidth for feedback and using a suitably designed coding scheme can increase throughput over feedback-free schemes at certain bit-error rates.

On a variety of increasingly important asymmetric two-way channels, this potential for throughput increase via feedback is even greater, because it may be possible to send many more bits per Hertz over the feedback path than over the forward path. Channels of this type arise rather naturally, for example, in the context of mobile-to-base communication in contemporary wireless networks.[3] In these systems, mobile-to-base transmission is often severely power-limited, while much greater power is available for base-to-mobile feedback. As a result, even a small amount of bandwidth can support high-rate feedback. Moreover, when the total available bandwidth is large, any reduction in capacity from reallocating mobile-to-base bandwidth for feedback is typically small, since power rather than bandwidth is the dominant limitation in this scenario.

But before feedback strategies can compete with feedback-free strategies, a framework for the design and understanding of feedback strategies is needed. As it stands, the feedback coding schemes developed in previous research are somewhat loosely related.

In this thesis, we put forth a framework for the design of low-complexity feedback coding strategies. The central notion underlying the framework, which we term the compressed-error-cancellation framework, is as follows. The transmitter sends a message (without forward error correction (FEC) coding) over the forward channel. Via the return path, the receiver feeds back what was received, so that the transmitter is able to determine what was received in error. The transmitter then uses source coding to form the smallest possible description of the errors and sends this description (again without FEC coding) over the forward channel to the receiver, which the receiver can use for the purposes of error correction. Because the channel introduces new errors into this compressed error description, an iterative process is required. In particular, the receiver

[3] Such networks may correspond to, for example, terrestrial cellular or satellite systems.

again feeds back what was received, and the transmitter again sends a compressed description of the new errors to the receiver. And so the process repeats.

We should note that an idea of this type was used by Ahlswede in his constructive proof of the coding theorem for DMC$_f$'s [1], though his investigation was rather limited in scope. In this thesis, we build extensively on this idea, showing how it can be applied to develop low-complexity, high-rate, high-reliability coding schemes for a wide variety of channels with feedback, including DMC$_f$'s, discrete finite-state channels (DFSC$_f$'s), unknown channels, multiple-access channels, and channels with noisy and partial feedback.

1.2 Overview

The thesis is organized as follows:

In Chapter 2, we introduce the compressed-error-cancellation framework in the context of developing a low-complexity, capacity-achieving, high-reliability coding scheme for arbitrary DMC$_f$'s. By exploiting low-complexity lossless source-coding algorithms, we develop a coding scheme requiring a number of computations that grows linearly with the total number of channel inputs used (linear complexity). The error exponent for the scheme is also optimal in a certain sense. We explore variations of the scheme and also address certain practical considerations such as delays due to computation and feedback delays. We also demonstrate that the scheme operates as predicted by simulating it on a digital computer.

In Chapter 3 we consider channels with memory, which arise in a number of practical applications. We show how the compressed-error-cancellation framework can be used to develop linear-complexity, high-rate, high-reliability coding schemes for DFSC$_f$'s, which constitute a very general class of channels with memory.

In Chapter 4, we consider an even more complex problem: communicating over a channel that is unknown to some extent. In practice, this problem arises when, for example, the channel varies with time in an unknown way. While the transmission of training data is a typical solution to this problem, these information-free transmissions can substantially reduce the rate and error exponent of a coding scheme. We explore in Chapter 4 how the compressed-error-cancellation framework can be used to accomplish universal communication — variable-rate communication over unknown channels with feedback. We see that the framework leads to linear-complexity schemes with some very attractive asymptotic properties.

In Chapter 5, we consider multiple-access channels, which are becoming increasingly important in communication within a network of users. When several transmitters send information to a single receiver, a common scenario, then these transmitters are said to communicate over a multiple-access channel. In Chapter 5, we extend the compressed-error-cancellation framework to cope with multiple-access channels with feedback. We show that rate pairs on the frontier of the two-user, feedback-free, multiple-access capacity region can be achieved with linear complexity.

The schemes we develop in Chapters 2–5 all require complete noiseless feedback. But one of the obstacles preventing feedback communication systems from seeing more widespread use is that complete noiseless feedback is often not available in practice. Often the feedback channel is noisy. It may also have insufficient capacity to allow the receiver to feed back is complete observation. In Chapter 6, we show that the compressed-error-cancellation framework is useful for coding for channels with partial and noisy feedback.

Concluding remarks are given in Chapter 7.

1.3 Summary of Notation and Abbreviations

The main notational convention of which we must be aware at this point is that if X is a discrete random variable, then p_X automatically denotes its pmf, and $E[X]$ automatically denotes its expected value. Standard adjustments to this notation are used for conditional and joint pmfs and expectations.

Remaining notational conventions and abbreviations are introduced as needed throughout the thesis (often in footnotes). For reference, we summarize here some important conventions and abbreviations, which should be assumed to hold unless otherwise stated :

Abbreviations:

Bernoulli-ϵ	$\rightarrow$ Bernoulli with probability of equaling one being ϵ
BSC	$\rightarrow$ binary symmetric channel
BSC$_\text{f}$	$\rightarrow$ BSC with complete noiseless feedback
BSC$_\text{pf}$	$\rightarrow$ BSC with partial noiseless feedback
cdf	$\rightarrow$ cumulative distribution function
CSWCF	$\rightarrow$ concatenated Slepian-Wolf coded feedback
DFSC	$\rightarrow$ discrete finite-state channel
DFSC$_\text{f}$	$\rightarrow$ DFSC with complete noiseless feedback
DFSC$_\text{pf}$	$\rightarrow$ DFSC with partial noiseless feedback
DMC	$\rightarrow$ discrete memoryless channel
DMC$_\text{f}$	$\rightarrow$ DMC with complete noiseless feedback
DMC$_\text{pf}$	$\rightarrow$ DMC with partial noiseless feedback
DMMAC	$\rightarrow$ discrete memoryless multiple-access channel
DMMAC$_\text{f}$	$\rightarrow$ DMMAC with complete noiseless feedback
DMMAC$_\text{pf}$	$\rightarrow$ DMMAC with partial noiseless feedback
FEC	$\rightarrow$ forward error-correcting
i.i.d.	$\rightarrow$ independent and identically distributed
MABC	$\rightarrow$ multiple-access broadcast channel
pmf	$\rightarrow$ probability mass function
SWCF	$\rightarrow$ Slepian-Wolf coded feedback
UFSC	$\rightarrow$ unknown DFSC
UFSC$_\text{f}$	$\rightarrow$ UFSC with complete noiseless feedback
UFSC$_\text{pf}$	$\rightarrow$ UFSC with partial noiseless feedback

Notational Conventions:

$\square \rightarrow$ marks the end of a proof (1.1)

$\nabla \rightarrow$ marks the end of a proof of a lemma introduced within a larger proof (1.2)

$x[\cdot] \rightarrow$ square brackets have no general meaning; they are used as
an alternative or in addition to super- and subscripts. (1.3)

$o_n(g(n)) \rightarrow$ a function in the set $\{f(n)\ :\ \lim_{n\to\infty} f(n)/g(n) = 0\}$ (1.4)

$O_n(g(n)) \rightarrow$ a function in the set $\{f(n)\ :\ \liminf_{n\to\infty} f(n)/g(n) \geq 0$ and

$\limsup_{n\to\infty} f(n)/g(n) < \infty\}$ (1.5)

$\Theta_n(g(n)) \to$ a function in the set $\{f(n)$: $\liminf_{n\to\infty} f(n)/g(n) > 0$ and $\limsup_{n\to\infty} f(n)/g(n) < \infty\}$ (1.6)

$\mathbb{Z} \to$ the set of integers (1.7)

$\mathbb{R} \to$ the set of real numbers (1.8)

$\mathbb{N} \to$ the set of natural numbers ($\{0, 1, 2, \cdots\}$) (1.9)

$|\mathcal{A}| \to$ cardinality of the set $\mathcal{A}$ (1.10)

$\mathrm{co}\mathcal{A} \to$ convex hull of the subset $\mathcal{A}$ of Euclidean space (1.11)

$\mathcal{A}^n \to$ n-fold Cartesian product of $\mathcal{A}$ with itself (1.12)

$\mathcal{A}^\dagger \to$ the set of variable-length tuples with elements in $\mathcal{A}$:

$\mathcal{A}^\dagger = \cup_{n=1}^{\infty} \mathcal{A}^n$ (1.13)

$a_s^t \to (a_s, \cdots, a_t)$ (1.14)

$a_s^\infty \to (a_s, a_{s+1}, \cdots)$ (1.15)

$a^n \to a_1^n$ (1.16)

$\ell(a) \to$ length of variable-length tuple a:

$\ell(a) = n$ if $a \in \mathcal{A}^n$ (1.17)

$$0_M.a^n \to \sum_{i=1} a_i M^{-i} \qquad (1.18)$$

$x_{[n]} \to$ nth M-ary expansion digit of $x \in [0, 1]$, where M is determined from context. When two expansions exist, the one ending with zeros is taken. (1.19)

$x_{[s]}^{[t]} \to (x_{[s]}, \cdots, x_{[t]})$ (1.20)

$x^{[t]} \to x_{[1]}^{[t]}$ (1.21)

$x \uparrow y \to$ x converges to y from below (1.22)

$\lceil x \rceil \to$ ceiling of x:

$\lceil x \rceil = \min\{z : z \in \mathbb{Z}, z \geq x\}$ (1.23)

$\log x \to$ base-2 logarithm of x (1.24)

$\ln x \to$ natural logarithm of x (1.25)

$\exp_2\{x\} \to 2^x$ (1.26)

$\exp\{x\} \to e^x$ (1.27)

$\Pr\{\mathcal{A}\} \to$ probability of an event $\mathcal{A}$ (1.28)

$\Pr\{\mathcal{A}|\mathcal{B}\} \to$ probability of an event $\mathcal{A}$ conditioned on the event $\mathcal{B}$ (1.29)

$p_X \to$ probability mass function (pmf) for X: $p_X(x) = \Pr\{X = x\}$ (1.30)

$p_{X|Y} \to$ pmf for X conditioned on Y: $p_{X|Y}(x|y) = \Pr\{X = x|Y = y\}$ (1.31)

$E[X] \to$ expected value of the random variable X:

$$E[X] = \sum_x x p_X(x) \qquad (1.32)$$

$\mathrm{var}(X) \to$ variance of the random variable X:

$$\mathrm{var}(X) = \sum_x (x - E[X])^2 p_X(x) \qquad (1.33)$$

$\mathrm{std}(X) \to$ standard deviation of the random variable X:

$$\mathrm{std}(X) = \sqrt{\mathrm{var}(X)} \tag{1.34}$$

$H_2(\epsilon)$ → binary entropy function:

$$H_2(\epsilon) = -\epsilon \log \epsilon - (1-\epsilon)\log(1-\epsilon) \tag{1.35}$$

$H(X)$ → entropy of random variable X:

$$H(X) = -\sum_x p_X(x) \log p_X(x) \tag{1.36}$$

$H(X,Y)$ → joint entropy of random variables X and Y:

$$H(X,Y) = -\sum_{x,y} p_{X,Y}(x,y) \log p_{X,Y}(x,y) \tag{1.37}$$

$H(X|Y)$ → conditional entropy of X given Y:

$$H(X|Y) = H(X,Y) - H(Y) \tag{1.38}$$

$I(X;Y)$ → mutual information between X and Y:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{1.39}$$

$H_\infty(X^\infty)$ → entropy rate of random process X^∞:

$$H_\infty(X^\infty) = \lim_{n\to\infty} \frac{1}{n} H(X^n) \tag{1.40}$$

$H_\infty(X^\infty, Y^\infty)$ → joint entropy rate of random processes X^∞ and Y^∞:

$$H_\infty(X^\infty, Y^\infty) = \lim_{n\to\infty} \frac{1}{n} H(X^n, Y^n) \tag{1.41}$$

$H_\infty(X^\infty|Y^\infty)$ → conditional entropy rate X^∞ conditioned on Y^∞:

$$H_\infty(X^\infty|Y^\infty) = H_\infty(X^\infty, Y^\infty) - H(Y^\infty) \tag{1.42}$$

$D_2(\epsilon \,\|\, \alpha)$ → Kullback-Leibler distance between Bernoulli pmfs with parameters ϵ and α:

$$D_2(\epsilon \,\|\, \alpha) = \epsilon \log \frac{\epsilon}{\alpha} + (1-\epsilon)\log\frac{1-\epsilon}{1-\alpha} \tag{1.43}$$

$D(p \,\|\, q)$ → Kullback-Leibler distance between pmfs p and q:

$$D(p \,\|\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{1.44}$$

Chapter 2

Discrete Memoryless Channels: An Introduction to the Framework

2.1 Introduction

In this chapter, we develop the compressed-error-cancellation framework for coding for feedback channels, which supports the next four chapters. We convey the central ideas that constitute the framework via the following three progressively more complicated illustrations.

The first is an illustration of the method used by a man, Tex (Tx), to tell his father, Rex (Rx), who is losing his faculties, his new address:

Tx.1: "My new address is four twenty-seven Elmwood Avenue, Maplewood, New Jersey, oh seven oh four one."

Rx.1: "Your new address is forty-seven Maplewood Avenue, Elmwood, New Jersey, oh seven oh four one?"

Tx.2: "No, transpose Elmwood and Maplewood and change forty-seven to four twenty-seven."

Rx.2: "Transpose Elmwood and Maplewood and change forty-seven to twenty-seven?"

Tx.3: "Not twenty-seven — FOUR TWENTY-SEVEN."

Rx.3: "Not twenty-seven — four twenty-seven?"

Tx.4: "Right."

Rx.4: "Ok. Goodbye, son."

In this illustration, Rex distorts Tex's statements and tells Tex (feeds back) the distorted interpretation. To guide Rex to an undistorted picture of his original message, Tex makes four progressively shorter statements with the following property: Tex's ith statement paired with what Rex interprets his $(i-1)$th statement to be is sufficient to determine what Tex's $(i-1)$th statement truly is. For example, item Tx.2 above can be recovered from items Rx.2 and Tx.3. That is, Tex immediately tells Rex, in a compact way, only what errors Rex has made in interpreting his previous statement.

An efficient adaptation of this method for communicating over a noisy channel is suggested by the following second illustration involving a BSC$_f$ with crossover probability ϵ. With $H_2(\epsilon)$ denoting the entropy of a Bernoulli-ϵ random variable, the transmitter (Tx) and receiver (Rx) (and the channel (Ch)) participate in an iterative coding scheme as follows (also see Figure 2-1 for a graphical representation of the scheme):

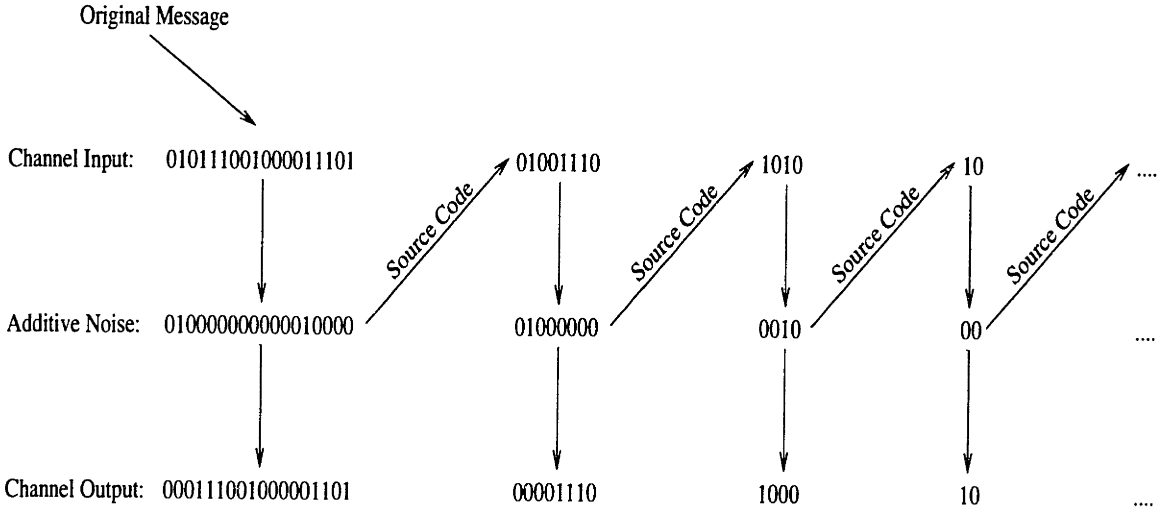


Figure 2-1: Graphical representation of the iterative coding scheme for a BSCf.

Tx.1: Sends N uncoded data bits over channel.
Ch.1: Adds (modulo-2) N samples of Bernoulli-ϵ noise.
Rx.1: Feeds its N noisy observations back to Tx.
Tx.2: (a) Finds N samples of noise added by channel.
(b) Compresses noise into $NH(\epsilon)$ new data bits.
(c) Sends these data bits uncoded over the channel.
Ch.2: Adds (modulo-2) $NH(\epsilon)$ samples of Bernoulli-ϵ noise.
Rx.2: Feeds its $NH(\epsilon)$ noisy observations back to Tx.
Tx.3: (a) Finds $NH(\epsilon)$ samples of noise added by channel.
(b) Compresses noise into $NH(\epsilon)^2$ new data bits.
(c) Sends these data bits uncoded over the channel.

⋮

If the compression (source coding) steps are assumed lossless, then, like the first illustration, the data sent in item Tx.3 along with the data received in item Rx.2 is sufficient to determine the data sent in item Tx.2. More generally, at each iteration, a message is transmitted over the forward channel without additional redundancy bits for (forward) error correction. The message contains enough information to cancel the errors in the previous block.

If the process were to continue indefinitely, the number of channel inputs used to send the N bits would be

$$N + NH_2(\epsilon) + NH_2(\epsilon)^2 + NH_2(\epsilon)^3 + \cdots = \frac{N}{1 - H_2(\epsilon)},$$

which corresponds to a rate of $1 - H_2(\epsilon)$, the capacity of the BSCf. In practice, the process is terminated after a finite number of iterations by sending a final block using a different coding scheme

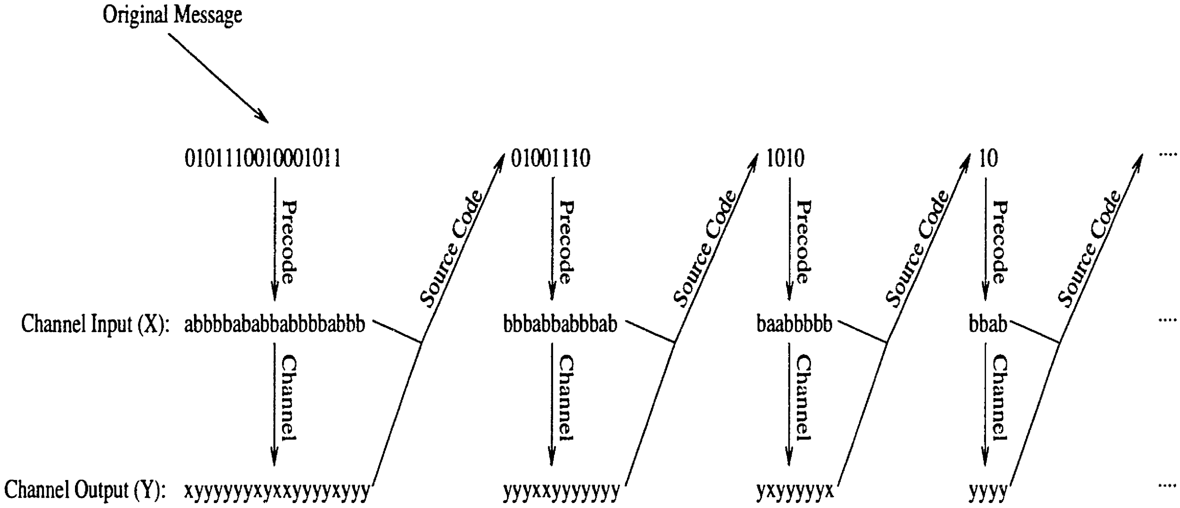


Figure 2-2: Graphical representation of the iterative coding scheme for a DMC_f with input alphabet $\{\text{a}, \text{b}\}$ and output alphabet $\{\text{x}, \text{y}\}$; the capacity-achieving distribution is such that $H(X) < 1$.

known as the *termination coding scheme*. If the receiver decodes this final block correctly, then it can recursively process the received data to reconstruct the original N-bit message.

The coding scheme illustrated above can be generalized to a coding scheme for an arbitrary DMC_f, which gives us the following third illustration. Let $q_{Y|X}$ be the channel transition function of a given DMC_f, and let q_X be its associated capacity-achieving input distribution. For notational convenience, let X and Y be random variables such that $p_{X,Y}(x,y) = q_X(x)q_{Y|X}(y|x)$. Then consider a coding scheme in which the transmitter, receiver, and channel act as follows (also see Figure 2-2 for a graphical representation):

Tx.1: (a) Precodes N message bits into $N_1 = N/H(X)$ channel inputs[1] $X_1^{N_1}$ that look i.i.d. according to q_X.
(b) Sends $X_1^{N_1}$ over channel.

Ch.1: Corrupts $X_1^{N_1}$ according to $q_{Y|X}$.

Rx.1: Feeds corrupted data $Y_1^{N_1}$ back to Tx.

Tx.2: (a) Using $Y_1^{N_1}$, compresses $X_1^{N_1}$ into $N_1 H(X|Y)$ new data bits.
(b) Precodes new data bits into $N_2 = N_1 H(X|Y)/H(X)$ channel inputs $X_{N_1+1}^{N_1+N_2}$, which look i.i.d. according to q_X.
(c) Sends $X_{N_1+1}^{N_1+N_2}$ over channel.

Ch.1: Corrupts $X_{N_1+1}^{N_1+N_2}$ according to $q_{Y|X}$.

Rx.2: Feeds corrupted data $Y_{N_1+1}^{N_1+N_2}$ back to Tx.

[1] As a notational convenience to consolidate lists of variables in this paper, we adopt the shorthand a_m^n for $(a_m, a_{m+1}, \ldots, a_n)$, and, in turn, the shorthand a^n for a_1^n. As related notation, we use $\mathcal{A}^n$ to denote the n-fold Cartesian product of a set $\mathcal{A}$ with itself, where n may be infinite. This notation holds only for sub- and super-scripted variables that have not otherwise been specifically defined.

```
function feedback_enc(message, it)
   if (it > B)
      return append(synch_sequence, termination_enc(message));
   else
      e = precode(message);
      return append(e, feedback_enc(cond_source_enc(e, channel(e)), it+1));
   end
end
```

Figure 2-3: Pseudocode for compressed-error-cancellation coding.

Tx.3: (a) Using $Y_{N_1+1}^{N_1+N_2}$, compresses $X_{N_1+1}^{N_1+N_2}$ into $N_2 H(X|Y)$ new data bits.

(b) Precodes new data bits into $N_3 = N_2 H(X|Y)/H(X)$ channel inputs $X_{N_1+N_2+1}^{N_1+N_2+N_3}$, which look i.i.d. according to q_X.

(c) Sends $X_{N_1+N_2+1}^{N_1+N_2+N_3}$ over channel.

⋮

If we assume both precoding and source coding to be invertible, then items Tx.2, Rx.2, and Tx.3 again have the same relationship as in the previous two illustrations, and more generally, data transmitted at Tx.$(i+1)$ with data received at Rx.i are sufficient to determine the data transmitted at Tx.i.

When the process iterates indefinitely, the number of channel inputs used is

$$\frac{N}{H(X)} + \frac{N}{H(X)}\left(\frac{H(X|Y)}{H(X)}\right) + \frac{N}{H(X)}\left(\frac{H(X|Y)}{H(X)}\right)^2 + \cdots = \frac{N}{H(X) - H(X|Y)},$$

giving an average rate equal to the channel capacity. Again, in practice, we terminate the process after a finite number of iterations by sending a final block of data over the channel using a termination coder.

The notions in these three illustrations are compactly summarized in the even more general description of the framework given by the pseudocode in Figure 2-3, which also highlights the framework's recursive nature. As the pseudocode suggests, the encoder for a coding scheme based on the compressed-error-cancellation framework comprises three key components : a precoder, a source coder, and a termination coder. (A fourth component, the synchronization subsystem, is needed when the precoder and source coder outputs have variable length.)

The precoder is needed for two reasons: 1) because the input alphabet of a given channel may not be the binary alphabet $\{0, 1\}$ and 2) because the optimal way in which the input symbols should be used may be complex. Neither of these conditions holds for the BSC, which is why the second illustration above uses no precoder. But for a channel such as a DMC, for example, the frequency with which each symbol should be used in order to maximize transmission rate is given by the capacity-achieving input distribution q_X. The purpose of the precoder in this case is then to transform a sequence of Bernoulli-$\frac{1}{2}$ random variables (bits) into a sequence that uses the channel inputs with the optimal frequency, e.g., a sequence that is i.i.d. according to q_X. For more general channels, the precoder transforms a sequence of bits into a sequence of channel inputs that is appropriately distributed.

The source coder is needed to form a compact description of the corruption introduced by the channel. That is, on a given iteration, the precoder produces a sequence of channel inputs that are sent over the channel. After seeing the corresponding channel outputs, the receiver has some remaining uncertainty about the input sequence. The source coder produces data that is sufficient to resolve this uncertainty.

The termination coder is needed when the recursive process reaches its "base" case. At this point, a final sequence of bits remains to be transmitted reliably. The termination coder fills this need.

When we put these three subsystems together in the way suggested by the pseudocode, the resulting system is both easy to understand and elegant. Intuitively, we can see that it is also efficient in terms of rate. But what justifies its designation as a framework for *low-complexity* coding? Is it somehow very practical, too? The answer is "yes", and "yes" for an important reason. This framework shows the problem of channel coding with feedback to be strongly related to the problem of source coding. As a result, very general and computationally efficient implementations of codes based on the compressed-error-cancellation framework are obtained by exploiting the rich theory and efficient algorithms of lossless source coding [14]. In particular, by using variants of arithmetic coding [63] and Lempel-Ziv coding [67], we can construct low-complexity feedback coding schemes for a wide range of channels. Moreover, the Slepian-Wolf source coding theorem [55] has some important interpretations in the context of our channel coding framework (see Section 6.6.2).

The remainder of this thesis is about the application of this framework to some of the fundamental scenarios typically addressed in information theory. The remainder of this chapter focuses on a precise development and analysis of coding schemes for DMC$_f$'s. Using the coding scheme we develop for DMC$_f$'s as a foundation, subsequent chapters explore coding schemes that are applicable to different models of both the forward and feedback channels.

The DMC$_f$ provides a natural and useful starting point for a number of reasons. First, the DMC is perhaps the simplest and most fundamental channel in information theory. It is, for example, the channel emphasized by Shannon in his development of the coding theorem [53]. Shannon also showed that the presence of feedback does not increase the capacity of DMC's [52], which further simplifies our discussion. Second, the DMC is also quite a general channel model, requiring only that the channel input and output alphabets be finite sets and that the channel output at a particular time be statistically dependent only on the channel input at the same time. As a result, a wide variety of commonly used channels can be well modeled as DMC's, including, with appropriate quantization, channels with continuous valued inputs and outputs such as the deep space channel. Third, the assumption that the forward channel is a DMC coupled with the assumption that a complete, noiseless feedback[2] channel is available admits the simplest description and analysis of a coding scheme based on the compressed-error-cancellation framework. While a complete, noiseless feedback channel may often be a somewhat unrealistic model for a given feedback channel, it may provide a reasonable model in certain situations, such as those discussed in Chapter 1 in which the receiver has much greater transmit power available than does the transmitter.

Before turning our attention to the precise development of coding schemes for DMC$_f$'s, it is useful to first make our notions of a DMC and a DMC$_f$ precise. A DMC is described by a triple $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$. The finite set $\mathcal{X}$ is the range of allowable channel inputs, the finite set $\mathcal{Y}$ is the range of possible channel outputs, and $q_{Y|X}$ is the conditional probability mass function (pmf) of the channel

[2] Complete, noiseless feedback is also known as *information* feedback.

output at time k given the channel input at time k for any integer $k > 0$. The meaning of this tuple is defined more formally as follows:

Definition 2.1.1 Let $\{X_k\}_{k=1}^{\infty}$ and $\{Y_k\}_{k=1}^{\infty}$ be random processes. Then $\{Y_k\}_{k=1}^{\infty}$ is the *output process* resulting from *passing* $\{X_k\}_{k=1}^{\infty}$ *through a DMC* $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$ (or just $q_{Y|X}$) *without feedback* if for all $n > 0$,

$$p_{Y^n|X^n}(y^n|x^n) = \prod_{k=1}^{n} q_{Y|X}(y_k|x_k) \tag{2.1}$$

for all $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$.

A DMC$_\text{f}$ is a DMC with a feedback channel and is also described by a triple that specifies the DMC. The feedback channel is noiseless, delayless, and has sufficient capacity to feed back the receiver's complete observation—i.e., at time k, the transmitter knows the value of the channel outputs from time 1 to time $k-1$.

We emphasize that the probability law (2.1) does *not* hold when feedback is available. For example, consider the case in which the forward channel is a BSC with crossover probability ϵ. Suppose that the first channel input is 0 or 1 with equal probability and that all subsequent inputs are equal to the previous channel output. Then (2.1) would say that given an input sequence, say $x^n = (0, 1, 1, 0, 1, 1, 1, 0, 0)$. every output sequence has positive probability. But it is clear that only output sequences of the form $(1, 1, 0, 1, 1, 1, 0, 0, \cdot)$ are possible; that is, only two sequences have positive probability given such an input.

So what is a DMC$_\text{f}$? A definition that captures the important features of these channels and the way they are typically used is as follows:

Definition 2.1.2 Let M be a random variable taking values in $\mathcal{M}$. For all $m \in \mathcal{M}$, let $\{f_{m,i}\}_{i=1}^{\infty}$ be a sequence of functions, where $f_{m,i} : \mathcal{Y}^{i-1} \to \mathcal{X}$ maps a sequence of $i-1$ channel outputs to a single channel input, and $f_{m,1}$ takes a constant value in $\mathcal{X}$. Then $\{Y_k\}$ is the *output process resulting from passing M through a DMC$_f$* $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$ (or just $q_{Y|X}$) *via* $\{\{f_{m,i}\}_{i=1}^{\infty}\}_{m \in \mathcal{M}}$ if for all $n > 0$,

$$p_{Y^n|M}(y^n|m) = \prod_{k=1}^{n} q_{Y|X}(y_k|f_{m,k}(y^{k-1})) \tag{2.2}$$

for all $y^n \in \mathcal{Y}^n$ and all $m \in \mathcal{M}$.

We now develop in detail a coding scheme for DMC$_\text{f}$'s. We begin with a formulation of variable-length coding in Section 2.2, and follow with a description and analysis of a coding scheme for DMC$_\text{f}$'s in Sections 2.3–2.6. We discuss some variations on the scheme in Section 2.7 and end with a discussion of remaining practical and theoretical issues in Section 2.8.

2.2 Variable-Length Codes and Their Properties: Rate, Reliability, and Complexity

A variable-length *code* is a 4-tuple $(N, \varepsilon, \{\chi_i\}_i, \Delta)$, where N is the number of message bits to be transmitted, $\varepsilon : \{0, 1\}^N \times \mathcal{Y}^{\infty} \to \mathcal{X}^{\infty}$ is the encoding function, $\{\chi_i : \mathcal{Y}^i \to \{0, 1\}\}_{i=1}^{\infty}$ is a sequence of stopping functions, and[3] $\Delta : \mathcal{Y}^\dagger \to \{0,1\}^N$ is the decoding function.

The first argument to the encoding function ε represents the message to be encoded, while the second argument represents the stream of feedback coming back to the transmitter. Because of the causal nature of the feedback, ε is restricted to have the form

$$\varepsilon(w^N, y^\infty) = (\tilde{\varepsilon}_1(w^N), \tilde{\varepsilon}_2(w^N, y^1), \tilde{\varepsilon}_3(w^N, y^2), \dots), \tag{2.3}$$

where the range of $\tilde{\varepsilon}_i$ is $\mathcal{X}$ for all i.

In describing the performance characteristics of such a code on a given DMCf $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$, we let the random variables $W_1, \cdots, W_N$ represent the N independent and identically distributed (i.i.d.) equally likely message bits, i.e., the message W^N is a discrete random variable that is uniformly distributed over the set

$$\mathcal{W} = \{0,1\}^N. \tag{2.4}$$

We then let the process $\{Y_k\}$ be the output process resulting from passing W^N through the DMCf $q_{Y|X}$ via $\{\{\tilde{\varepsilon}_i(m,\cdot)\}_{i=1}^\infty\}_{m \in \{0,1\}^N}$.

To define the rate of the code, we must define its transmission length, which is the point at which the receiver stops data acquisition. The stopping functions χ_i are the mechanism by which the receiver determines from its observations when to stop data acquisition. By simulating the receiver using the feedback link, the transmitter can also determine when to correspondingly terminate transmission. In particular, at each time k, the function χ_k maps the data observed up to that time, Y^k, into a decision as to whether to terminate transmission; the value one is the signal to terminate, while the value zero is the signal to continue, i.e., $\chi_i : \mathcal{Y}^i \to \{0,1\}$. In terms of this notation, the transmission length is given by the random variable

$$L^* = \min_k \{k \;:\; \chi_k(Y^k) = 1\}. \tag{2.5}$$

The rate of the code $(N, \varepsilon, \{\chi_i\}_{i=0}^\infty, \Delta)$ is then defined to be $N/E[L^*]$.

The decoder makes an estimate $\hat{W}^N$ of the message sent using its decoding function $\Delta : \mathcal{Y}^\dagger \to \mathcal{W}$, i.e.,

$$\hat{W}^N = \Delta(Y^{L^*}), \tag{2.6}$$

and the associated error probability of the code is $\Pr\{\hat{W}^N \neq W^N\}$.

A *coding scheme* is a function mapping two parameters p_1 and p_2 to a variable-length code. A coding scheme c achieves a rate R if for any $\delta > 0$ and any $\epsilon > 0$, there are parameters p_1 and p_2 such that $c(p_1, p_2)$ has probability of error less than δ and rate greater than $R - \epsilon$. A coding scheme is (feedback-free) feedback-capacity-achieving if it achieves a rate equal to the (feedback-free) feedback capacity of the channel.

We say a coding scheme c attains an error-exponent function $F : I \to \mathbb{R}$ if for each rate $r \in I$, where $I \subset \mathbb{R}$, there is a sequence of parameter pairs $\{(p_{1,n}(r), p_{2,n}(r))\}_n$ such that the resulting sequence of codes $\{c(p_{1,n}(r), p_{2,n}(r))\}_n$ is such that the corresponding rate sequence $\{R_n\}_n$ and

[3] In the description of variable-length tuples, it is convenient to use $\mathcal{A}^\dagger$ to denote the set of all tuples whose elements are in the set $\mathcal{A}$, i.e., $\mathcal{A}^\dagger = \cup_{n=1}^\infty \mathcal{A}^n$.

the probability of error sequence $\{P_n\}_n$ obey

$$\lim_{n\to\infty} R_n \geq r \tag{2.7}$$

$$\lim_{n\to\infty} -\frac{R_n \log P_n}{N_n} \geq F(r), \tag{2.8}$$

where N_n is the number of message bits for the code $c(p_{1,n}(r), p_{2,n}(r))$. A useful interpretation is that if an error exponent $F(r)$ is attained, then for any $\epsilon > 0$, a sequence of codes with rates greater than $r - \epsilon$ can be generated such that their associated error probabilities decay exponentially with the expected transmission length with an associated rate of decay greater than $F(r) - \epsilon$ — i.e., the error probability is reduced by at least two when expected transmission length increases by $1/(F(r) - \epsilon)$.

While rate and reliability are two important properties of codes, they say little about the computational resources required by encoding and decoding algorithms. Hence, in addition to assessing the rate and reliabilities of the various coding schemes we introduce, we also characterize their computational requirements by describing their time and space complexities. We devote the remainder of this section to making our notions of time and space complexity more precise.

We define time complexity and space complexity to be the asymptotic order of growth of the time cost and space cost used to run an algorithm on a random-access machine under the *uniform-cost* criterion. A random-access machine can be thought of as the usual computer with memory registers that can perform some basic operations on the contents of its registers, including reading and writing to its registers. Each register is assumed to be capable of holding an arbitrarily large integer. Under the uniform-cost criterion, basic operations on registers are assumed to take constant time independent of the size of the integers in the registers; space cost per register is a constant independent of the number held in the register.

We break slightly from the usual definition of space complexity and use a more conservative definition. Normally, input for an algorithm is assumed to be given on a read-only input tape, and the output is assumed tc be written onto a write-once output tape. The size of the input tape and output tape is then usually not considered in the space complexity. In other words, only memory that can be both read from and written to is considered in space complexity. But because communication is by its nature dynamic — i.e., we are not interested in sending one message over and over — the notion of a read-only input tape and a write-only output tape seems inappropriate. Hence, the ultimate aim being to characterize the amount of memory that we must have on hand to implement the entire coding scheme, we feel it is appropriate to count the storage required to store inputs and outputs, and we do so.

Note that order of growth is generally described relative to the size of the input to the algorithm but can be described relative to the size of the output, which we do in certain cases. The reason for these choices of time and space complexity is the combination of intuitive simplicity and the reasonable degree to which it models computation on contemporary digital computers.

It is important to keep in mind that there are many alternative notions of computational complexity. A number of possible notions are outlined in [54]. No single notion seems to have become dominant in the channel coding literature. We choose the above measure as for its tractability and meaningfulness. Under this measure, for example, summing n numbers has $O_n(n)$ complexity, the fast Fourier Transform has $O_n(n \log n)$ time complexity, and other intuitively satisfying results are obtained. Yet this count may not accurately reflect the time required to carry out the algorithm on a contemporary digital computer if, for example, in the summing of n numbers, n is truly enormous. Indeed, a more useful measure might be a count of the number of bitwise operations required to carry out the algorithm. But this measurement is difficult to obtain, requiring us to delve into the bitwise details of our algorithms. Because it is difficult to make precise statements about the number of bitwise computations, we avoid doing so.

Nevertheless, in recognition of the importance of what the bitwise computation model intends to capture, which is the amount of time required by the algorithm to run on a contemporary digital computer with finite-length registers, we adopt the following approach: We state the time and space complexity of our algorithms using the uniform-cost criterion and then comment on issues one might encounter when implementing the algorithms on a finite-length-register contemporary digital computer, since this is what is of immediate practical importance. We see that the complexity given under the uniform-cost criterion should give a fairly accurate characterization of the computational resources required on such computers for reasonably sized problems.

2.3 A Feedback Coding Scheme for DMC$_\text{f}$'s

Now that we have defined the elements that constitute a variable-length feedback code, we can describe a coding scheme c_{DMC} for a given a DMC$_\text{f}$ $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$ based on the compressed-error-cancellation framework. As stated in the previous section, a coding scheme is a function taking two parameters as input and returning a code. Our scheme c_{DMC} takes as its two parameters a message length N and a termination coder parameter ν, which together determine the rate and probability of error for the code, as we show later. Then $c_{\text{DMC}}(N, \nu)$ is a variable-length feedback code $(N, \varepsilon, \{\chi_i\}_{i=1}^{\infty}, \Delta)$, which we now describe, with primary emphasis on the encoder ε — the corresponding definitions of the decoder and stopping functions are implied by definition of the encoder. Note that for notational cleanliness, we do not make explicit the dependence of the last three elements of the 4-tuple on ν, N, or the DMC$_\text{f}$ parameters. Also for notational purposes, we let q_X be the capacity-achieving input distribution for the channel, and let X and Y be random variables that are jointly distributed according to $p_{X,Y}(x, y) = q_X(x)q_{Y|X}(y|x)$, for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$.

2.3.1 Encoding

To define ε, we define the sequence of channel inputs $\varepsilon(W^N, Y^\infty)$ that is transmitted when the message to be sent to the receiver is W^N and when the sequence Y^∞ is fed back.

To describe this sequence of transmitted channel inputs, we make use of the following subsystems: precoders $\{\pi_n\}_{n=1}^{\infty}$, where π_n precodes n bits; lossless source coders $\{\sigma_n\}_{n=1}^{\infty}$, where σ_n conditionally source codes n channel inputs; and a termination encoder $e_{\kappa_N,\nu}^{\text{term}}$, which encodes a block of κ_N bits, where κ_N is given further below as a function of N, at some rate and probability of error determined by the parameter ν. At this stage, we focus on the basic structure of the encoding process, deferring a precise description of the functions $\{\pi_n\}_{n=1}^{\infty}$, $\{\sigma_n\}_{n=1}^{\infty}$, and $e_{\kappa_N,\nu}^{\text{term}}$ to Sections 2.3.1.1–2.3.1.3.

We begin with the simplifying assumption that we have available a length-t_N sequence $\emptyset^{t_N}[N]$, where[4] $t_N = o_N(N)$, that is perfectly detectable — no false alarms and no missed detections —

[4] $o_N(\cdot)$ is the usual order notation with a subscript explicitly denoting the limiting variable, i.e., if $f(N) = o_N(g(N))$,

after passing through the DMC. This assumption is removed in Section 2.5.

The sequence of transmitted channel inputs is generated as follows. On the first iteration, the raw message bits W^N are precoded and transmitted. Each subsequent iteration generates new bits to be precoded and transmitted that can be used by the decoder to correct errors on the previous iteration. If the source coding and precoding subsystems are designed appropriately, then the transmission on a given iteration is about $H(X|Y)/H(X) < 1$ times as long as the transmission on the previous iteration on average. After B_N iterations, some finai bits remain to be transmitted and are sent with protection from the termination coder. As we show later, if sufficiently few final bits remain, then overall complexity of the scheme can be kept small compared to N. We can ensure that the expected number of final bits is small enough for sufficiently large N by choosing B_N as the following function of N:

$$B_N = \lceil \log^2 N \rceil \tag{2.9}$$

The transmissions on each iteration, which are blocks of varying lengths, are denoted by ε'_0, ε'_1, $\ldots, \varepsilon'_{B_N - 1}$, respectively, and are generated according to the following recursive procedure:

1. Let $i = 0$ and initialize the block Ψ_i of data bits to be precoded, the length L_i^σ of this block, and the time index $A_i + 1$ of the beginning of the ith block:

$$\Psi_i = W^N, \qquad A_i = 0, \qquad L_i^\sigma = N$$

2. Compute the variable-length tuple ε'_i that is the precoded version of Ψ_i, and compute its length L_i:[5]

$$\varepsilon'_i = \pi_{L_i^\sigma}(\Psi_i) \tag{2.10}$$
$$L_i = \ell(\varepsilon'_i). \tag{2.11}$$

then

$$\lim_{N\to\infty} \frac{f(N)}{g(N)} = 0.$$

We also use $O_N(\cdot)$ and $\Theta_N(\cdot)$ so that $f(N)=O_N(g(N))$ means that

$$\liminf_{N\to\infty} \frac{f(N)}{g(N)} \geq 0$$
$$\limsup_{N\to\infty} \frac{f(N)}{g(N)} < \infty,$$

and $f(N)=\Theta_N(g(N))$ means that

$$\liminf_{N\to\infty} \frac{f(N)}{g(N)} > 0$$
$$\limsup_{N\to\infty} \frac{f(N)}{g(N)} < \infty.$$

The extra subscript is convenient in expressions where dependencies between variables are implicit.

[5] With $\mathbb{N}$ denoting the set of natural numbers $\{1, 2, 3, \ldots\}$, we let $\ell : \mathcal{A}^\dagger \to \mathbb{N}$ for a given set $\mathcal{A}$ denote the function whose value is the length of its argument; for example, if $a \in \mathcal{A}^n$, then $\ell(a) = n$.

3. Increment i and compute the new time index A_i:

$$A_i = A_{i-1} + L_{i-1}. \tag{2.12}$$

4. Compute the variable-length tuple Σ_i that results from source coding the previously transmitted block according to the input distribution given the feedback information, prefix Σ_i by an encoding of the lengths associated with the $(i-1)$th block to form Ψ_i, and compute the length L_i^σ of Ψ_i:

$$\Sigma_i = \sigma_{L_{i-1}}\left(\varepsilon'_{i-1}, Y_{A_{i-1}+1}^{A_i}\right) \tag{2.13}$$
$$\Psi_i = r((\phi(L_{i-1}), \phi(L_{i-1}^\sigma), \Sigma_i), i) \tag{2.14}$$
$$L_i^\sigma = \ell(\Psi_i). \tag{2.15}$$

In (2.14), the mapping $\phi : \{0\} \cup \mathbb{N} \rightarrow \{0,1\}^\dagger$ returns a sequence of length $2\lceil \log n \rceil + 2$ corresponding to a binary representation of its integer argument n with each bit repeated once followed by a terminating string $(0,1)$;[6] The invertible mapping[7] $r : \{0,1\}^\dagger \times \mathbb{N} \rightarrow \{0,1\}^\dagger$ is introduced to ensure that Ψ_i is uniformly distributed over the set $\{0,1\}^{L_i^\sigma}$. To do so, r adds (modulo-2) a pseudorandom Bernoulli-$\frac{1}{2}$ sequence to its first argument, using its second argument as a "seed". As a result, the output of r is indistinguishable from an i.i.d. Bernoulli-$\frac{1}{2}$ source, and r is reversible in the sense that the tuple d can be recovered from $r(d, s)$ and s.

5. If $i = B_N$, stop; otherwise, go to Step 2.

After directly transmitting the blocks ε'_i for $i = 0, 1, \ldots, B_N - 1$ on successive iterations, a special encoding is used for the data comprising the final source coded data Σ_B together with the block lengths L_{B-1} and L_{B-1}^σ needed for decoding. In particular, these data are transmitted via the invertible (i.e., uniquely decodable) sequence

$$G^\infty = (\rho(\Sigma_{B_N}), \phi(\lceil A_{B_N}/t_N \rceil t_N - A_{B_N}), \phi(L_{B_N-1}), \phi(L_{B_N-1}^\sigma), 0, 1, 0, 1, 0, 1, 0, 1, \ldots). \tag{2.16}$$

In (2.16), the mapping $\rho : \{0,1\}^\dagger \rightarrow \{0,1\}^\dagger$ repeats each input bit and adds the terminating string $(0,1)$.[8] It is worth emphasizing that although G^∞ is infinite in length, the transmitter stops transmission when the receiver stops data acquisition, since the transmitter knows the receiver's actions via the feedback.

Since no subsequent data are transmitted to correct the errors occurring in the transmission of G^∞, we must use termination coding to protect G^∞. Accordingly, the complete encoding function

[6] For example, $\phi(6) = (1,1,1,1,0,0,0,1)$, since the number 6 has binary representation $(1,1,0)$.

[7] We have observed that in practice, the randomizing function r in (2.14) is unnecessary; it is, however, convenient for analysis.

[8] So that, for example, $\rho((1,0,1)) = (1,1,0,0,1,1,0,1)$.

ε is given by

$$\varepsilon(W^N, Y^\infty) = (\varepsilon'_0, \varepsilon'_1, \ldots, \varepsilon'_{B_N-1}, \Phi^F, \emptyset^{t_N}[N], \\ e^{\text{term}}_{\kappa_N,\nu}(G_1^{\kappa_N}), e^{\text{term}}_{\kappa_N,\nu}(G_{\kappa_N+1}^{2\kappa_N}), e^{\text{term}}_{\kappa_N,\nu}(G_{2\kappa_N+1}^{3\kappa_N}), \ldots). \tag{2.17}$$

The synchronization sequence $\emptyset^{t_N}[N]$ is used to enable correct parsing of the incoming stream by the receiver, as we discuss in Section 2.3.2. The sequence Φ^F is a fairly arbitrary length-F "filler" sequence to be ignored, where $F = \lceil A_{B_N}/t_N \rceil t_N - A_{B_N} < t_N$; it serves only to ensure that $\emptyset^{t_N}[N]$ is transmitted at an integer multiple of t_N. Note that we have simplified our notation in (2.17) by suppressing the (potential) dependence of the termination encoder $e^{\text{term}}_{\kappa_N,\nu}$, which may itself be a feedback coder, on the appropriate subsequences of Y^∞.

Let us now precisely define the precoding, source coding, and termination coding subsystems.

2.3.1.1 Precoding Subsystem

To effect a transformation from a sequence of n i.i.d. Bernoulli-$\frac{1}{2}$ random variables (bits) into a sequence that is approximately i.i.d. according to q_X, the precoder $\pi_n : \{0,1\}^n \to \mathcal{X}^\dagger$ uses two key ideas: 1) that a sequence of variables taking values in a discrete set with cardinality M can be mapped to a real number in $[0,1)$ via its M-ary expansion; and 2) that a real random variable with some desired distribution can be created by applying the inverse cumulative distribution function (cdf) associated with the desired distribution to a uniformly distributed random variable.

The precoding then takes place in three steps: 1) The data bits to be precoded are mapped to a real number $S \in [0,1)$; 2) an appropriate inverse cdf function $F_{\tilde{X}}^{-1}$ is applied to this real number to form another real number $U \in [0,1)$; 3) an appropriate number of M-ary expansion digits of U are taken to be the output of the precoder. These three steps are similar to those taken by a decoder for an arithmetic source coder.

To define π_n precisely, we define the sequence of channel inputs $\pi_n(D^n)$ that correspond to precoding the n data bits D^n. We first transform D^n to a real number $S \in [0,1)$ according to

$$S = 0_2.D^n + 2^{-n}Z, \tag{2.18}$$

where

$$0_K.a^j = 0_K.a_1a_2\cdots a_j = \sum_{i=1}^{j} a_iK^{-i}, \tag{2.19}$$

and Z is a random variable that is uniformly distributed over $[0,1)$. The sole purpose of Z is so S is uniformly distributed over $[0,1)$ when D^n is uniformly distributed over $\{0,1\}^n$.

Next, assuming $\mathcal{X} = \{0,1,\cdots,M-1\}$ (which sacrifices no generality), we define the cdf $F_{\tilde{X}}$ whose inverse is used for the transformation. Let $\{\tilde{X}_k\}_{k=1}^\infty$ be an i.i.d. process with marginal pmf q_X. We then map this process onto the unit interval $[0,1)$ by letting $\tilde{X}$ be a random variable defined by $\tilde{X} = 0_M.\tilde{X}_1\tilde{X}_2\cdots$, and we let $F_{\tilde{X}}$ be the cdf for $\tilde{X}$.

With the base of all expansions in this section taken to be M, the precoder π_n is defined by[9]

$$T_n(u) = u^{[l_n(u)]} \tag{2.20}$$

$$\pi_n(D^n) = T_n(F_{\tilde{X}}^{-1}(S)), \tag{2.21}$$

where the expansion in (2.20) is M-ary, and $l_n : [0,1) \to \mathbb{N}$ is defined as follows to ensure that the output of the precoder stops after enough digits of the M-ary expansion of $F_{\tilde{X}}^{-1}(S)$ have been put out to uniquely determine the first n bits of S. That is, l_n is defined by

$$l_n(u) = \min\{k \ : \ F_{\tilde{X}}([0_M.u^{[k]}, 0_M.u^{[k]} + M^{-k})) \subseteq [i2^{-n}, (i+1)2^{-n})\}$$
$$\text{for } u \in F_{\tilde{X}}^{-1}([i2^{-n}, (i+1)2^{-n})) \text{ for } i = 0, \cdots, 2^n - 1. \tag{2.22}$$

This definition of π_n implies that the precoder is a lossless coder, i.e., if $d^n \in \{0,1\}^n$, then $\pi_n(d^n)$ is a variable-length tuple whose elements are in $\mathcal{X}$; from n and $\pi_n(d^n)$, we can determine d^n. No knowledge of Z is required by the decoder.

This definition also implies that the distribution of the output of the precoder approximates the desired channel input distribution in the following sense: If the input to the precoder D^n is uniformly distributed over $\{0,1\}^n$, then the elements of $\pi_n(D^n)$ form a random process that can be generated by taking an i.i.d. process with marginal pmf q_X and truncating it according to a stopping rule [25].

2.3.1.2 Source Coder

We now define the source coding function $\sigma_n : \mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}^\dagger$. This function compresses its first argument, a sequence $x^n \in \mathcal{X}^n$ representing the channel inputs, using a statistical model for its dependence on its second argument, a sequence $y^n \in \mathcal{Y}^n$ representing the corresponding channel outputs.

Since x^n is generated by the precoder, whose output is approximately an i.i.d. process, and y^n results from passing x^n through the DMC effectively without feedback, we choose the following statistical model for use in the source coder. Let $\{\hat{X}_i\}$ denote an i.i.d. process with marginal pmf q_X, and let $\{\hat{Y}_i\}_{i=1}^{\infty}$ denote the channel output process resulting from passing $\{\hat{X}_i\}$ through the DMC $q_{Y|X}$ without feedback. The source coder then assumes that the probability of x^n given y^n is simply

$$p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n). \tag{2.23}$$

Source coding is accomplished with this model via a Shannon-Fano strategy [14]. That is, with $\tilde{X} = 0_M.\hat{X}^n$,

$$\sigma_n(x^n, y^n) = u^{[l]} \tag{2.24}$$

[9] As additional notation, for any given number $a \in [0,1)$, we use $a_{[i]}$ to denote its ith K-ary (base-K) expansion digit, i.e., $a = \sum_{i=1}^{\infty} a_{[i]} K^{-i}$ with $a_{[i]} \in \{0, 1, \ldots, K-1\}$. When using this notation, the base K of the expansion is either stated explicitly or clear from context. If a has two K-ary expansions, then $a_{[i]}$ specifically denotes the ith digit of the expansion that ends in an infinite sequence of zeros. Also, $a_{[s]}^{[t]}$, where $s < t$ denotes the tuple $(a_{[s]}, \cdots, a_{[t]})$, and $a_{[1]}^{[t]}$ is abbreviated by $a^{[t]}$.

where the expansion in above is binary, and

$$u = F_{\tilde{X}|\hat{Y}^n}(0_M.x^n|y^n) + p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n)/2 \tag{2.25}$$

$$l = \lceil -\log p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n) \rceil + 1, \tag{2.26}$$

and $F_{\tilde{X}|\hat{Y}^n}$ is the cdf for $\tilde{X}$ conditioned on $\hat{Y}^n$. The source coder is lossless in the sense that x^n can be recovered from y^n with $\sigma_n(x^n, y^n)$.

Note that the statistical model used in the source coder is inaccurate: Modeling the precoder output as n samples of the i.i.d. process $\{\hat{X}_k\}_{k=1}^{\infty}$ is not strictly correct, because n itself is dependent on the values of $\hat{X}^n$. Nevertheless, we use this source coder in our overall scheme and show that these inaccuracies are inconsequential.

2.3.1.3 Termination Coding Subsystem

We now describe the termination encoder $e^{\text{term}}_{\kappa_N,\nu}$ that we use in the code $c_{\text{DMC}}(N,\nu)$. The termination encoder, as we mentioned earlier, protects κ_N bits with a rate and probability of error determined by its parameter ν. Let us first discuss how many bits κ_N the termination coder should be designed to protect.

For the overall scheme to have high reliability, the final data bits should be encoded within a single κ_N-bit block, on average. For this reason, we choose κ_N according to

$$\kappa_N = \left\lceil N^{1/4} \right\rceil . \tag{2.27}$$

To show that κ_N is sufficiently large, we let B_G denote the number of κ_N-bit blocks required to send the first $\tilde{L}$ bits of G^{∞}, where

$$\tilde{L} = 2\ell(\Sigma_{B_N}) + 2\lceil \log t \rceil + 2\left\lceil \log L_{B_N - 1} \right\rceil + 2\left\lceil \log L^{\sigma}_{B_N - 1} \right\rceil + 10 \tag{2.28}$$

represents the number of final data bits that are sent by the termination coder up to and including the first appearance of the string (0, 1, 0, 1 in G^{∞}. Then

$$B_G = \left\lceil \frac{\tilde{L}}{\kappa_N} \right\rceil \tag{2.29}$$

$$\leq \frac{\tilde{L}}{\kappa_N} + 1, \tag{2.30}$$

where the inequality follows from the simple ceiling function property $\lceil x \rceil \leq x + 1$. Taking expectations of the right-hand side of (2.30),

$$E[B_G] \leq 1 + \frac{E[\tilde{L}] + 2}{\kappa_N}. \tag{2.31}$$

We show in Appendix A.5 that with the precoder and source coder designs given in the previous two sections,

$$E[B_G]{=}1 + o_N(1), \tag{2.32}$$

which shows that our choice of κ_N gives the desired behavior.

While any of a wide range of codes can be used as the termination coding scheme that protects the final bits, we choose what we call a *modified Schalkwijk-Barron* (mSB) coding scheme, so named because it is based on a coder developed by Yamamoto and Itoh [65] in a paper bearing this name in the title. The mSB coder is itself a highly reliable feedback coder, which, as we show in Section 2.4, gives the overall scheme high reliability.

The mSB coder that we use sends a block of κ_N bits as follows: Let x and x' be two elements of $\mathcal{X}$ defined by

$$(x, x') = \arg\max_{(x,x')} D(q_{Y|X}(\cdot|x) \ \ || \ \ q_{Y|X}(\cdot|x')), \tag{2.33}$$

where $D(\cdot \ || \ \cdot)$ denotes the Kullback-Leibler distance between two pmfs.[10] Assuming the channel has positive capacity, it can be shown easily that the probability of error associated with maximum-likelihood decoding of the two-codeword, length-κ_N codebook consisting of the words $a^{\kappa_N}[0] \triangleq (x, \cdots, x)$ and $a^{\kappa_N}[1] \triangleq (x', \cdots, x')$ is below $2^{-\alpha\kappa_N}$ for some $\alpha > 0$.

Each of the κ_N bits is sent via this two-codeword codebook, i.e., the sequence $a^{\kappa_N}[0]$ is sent for a 0 and $a^{\kappa_N}[1]$ is sent for a 1. After the κ_N bits are sent via this procedure, the transmitter determines whether the receiver has decoded any of the bits incorrectly. Via the union bound, the probability P_{in} that any of these bits are decoded incorrectly can be shown to be $o_{\kappa_N}(1)$. If any errors occur, then the transmitter sends the length-ν sequence $w^\nu \triangleq (x', \cdots, x')$. Otherwise the length-$\nu$ sequence $c^\nu \triangleq (x, \cdots, x)$ is sent; if the receiver successfully determines that w^ν (and not c^ν) was sent, the process is repeated from the beginning. In other words, $e^{\text{term}}_{\kappa_N,\nu}$ is the solution to the following equation:

$$e^{\text{term}}_{\kappa_N,\nu}(b^{\kappa_N}, y^\infty) = (e^{\text{in}}_{\kappa_N}(b^{\kappa_N}), \rho(y^\infty)) \tag{2.34}$$

$$\rho(y^\infty) = \begin{cases} c^\nu & \text{if } d^{\text{in}}(y^{\eta_{\text{in}}}) = b^{\kappa_N}, \\ w^\nu & \text{if } d^{\text{in}}(y^{\eta_{\text{in}}}) \neq b^{\kappa_N} \text{ and } d^{cw}(y^{\eta_{\text{in}}+\nu}_{\eta_{\text{in}}+1}) = 0, \\ (w^\nu, e^{\text{term}}_{\kappa_N,\nu}(b^{\kappa_N}, y^\infty_{\eta_{\text{in}}+\nu+1})) & \text{otherwise,} \end{cases} \tag{2.35}$$

where

$$e^{\text{in}}_{\kappa_N}(b^{\kappa_N}) = (a^{\kappa_N}[b_1], \cdots, a^{\kappa_N}[b_{\kappa_N}]) \tag{2.36}$$

[10] The Kullback-Leibler distance, also called the information divergence, between two pmfs p and q is given by [14]

$$D(p \,||\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}.$$

and $d^{\text{in}}_{\kappa_N} : \mathcal{X}^{\kappa_N{}^2} \rightarrow \{0,1\}^{\kappa_N}$ is the corresponding decoder for $e^{\text{in}}_{\kappa_N}$, and d^{cw} decides whether w^ν or c^ν was sent, returning 0 to indicate that c^ν was sent. Note that d^{cw} is not a maximum-likelihood detector, because it does not treat errors equally. It tries to minimize the probability of mistaking w^ν for c^ν subject to making the probability of the reverse error $o_\nu(1)$; see [65] for details.

2.3.2 Decoding

Let us now outline the processing that takes place in the associated decoder for the encoder (2.17). This high-level description implicitly defines the stopping functions $\{\chi_i\}_i$ and decoder Δ.

The receiver has two operating modes, "normal mode," in which it starts, and "termination-decoding mode." In normal mode, the receiver saves the data from the incoming bitstream directly to memory without processing it while watching for the occurrence of the sequence $\emptyset^{t_N}[N]$. When this sequence is detected, it enters termination-decoding mode and begins decoding incoming data using the decoder corresponding to the termination code. It concatenates the decoded messages into a long decoded string, appending each newly decoded message to the end of the string. After decoding each block of κ_N bits and appending it to the string, the receiver searches for the termination sequence $(0,1,0,1)$ anywhere in the full string and stops data acquisition when it detects this sequence.

The receiver then proceeds to decode enough of G^∞ to recover Σ_{B_N}, F, L_{B_N-1}, and $L^\sigma_{B_N-1}$. Starting with this information, it decodes the data set acquired in normal mode according to the following recursive procedure:

1. Subtract F from the location at which $\emptyset^{t_N}[N]$ was detected to determine A_{B_N}. Let $i = B_N$.

2. Let $A_{i-1} = A_i - L_{i-1}$. Use the received data $Y^{A_i}_{A_{i-1}+1}$ and L_{i-1} to construct the (*a posteriori*) source coding pmf that was used to generate Σ_i, and invert the source coded block Σ_i according to this pmf to obtain ε'_{i-1}.

3. Use L^σ_{i-1} to invert the precoded block ε'_{i-1} to obtain Ψ_{i-1}.

4. Decrement i. If $i \geq 1$, extract Σ_i, L_{i-1}, and L^σ_{i-1} from Ψ_i using the seed i to invert the effect of r, and go to Step 2; otherwise, stop because $\Psi_0 = W^N$, and the message has been determined.

2.4 Reliability

The reliability of a coding scheme, also known as its error exponent, gives the asymptotic relationship among the coding scheme's rate, probability of error, and blocklength. In this section, we prove at a high level, leaving most of the details to Appendix A, the following theorem regarding the reliability for the scheme:

Theorem 2.4.1 Let c_{DMC} be a coding scheme mapping N and ν to the corresponding code consisting of the encoder described in Section 2.3.1 with its implied associated stopping functions and decoder. Then, in the sense of Section 2.2, c_{DMC} attains the error exponent function E_{CEC} defined on the interval $(0, I(X;Y))$ by

$$E_{\text{CEC}}(r) = (1 - r/I(X;Y))E_{cw}, \tag{2.37}$$

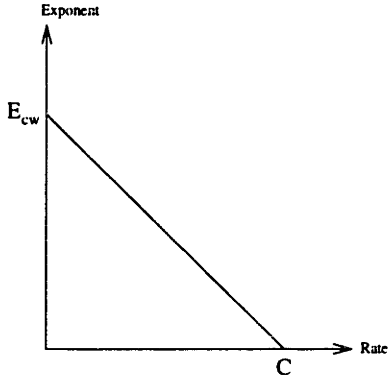


Figure 2-4: Plot of E_{CEC}.

where

$$E_{cw} = \max_{x \in \mathcal{X}, x' \in \mathcal{X}} D(q_{Y|X}(\cdot|x) \parallel q_{Y|X}(\cdot|x')). \tag{2.38}$$

Remarks: This theorem implies that the probability of error for the scheme decays exponentially with N or faster at any rate below capacity. Furthermore, Burnashev [11] has shown that E_{CEC} is an *upper* bound to the error exponent of any variable-length feedback transmission scheme for a DMC$_{\text{f}}$. This scheme therefore attains the largest possible error exponent at all rates. It is also useful to realize that the error exponent E_{CEC}, which is plotted in Figure 2-4, is the same as that of the mSB scheme introduced in [65]. Finally, note that E_{cw} may be infinite (e.g., if an output symbol has positive probability under one channel input but zero probability under another as in a "Z-channel") in which case the error exponent becomes degenerate and somewhat meaningless.

Proof: This theorem follows directly from three key properties of the coding scheme, which we highlight to begin the proof. We call these properties "Subsystem Properties," because each focuses on a key aspect of one of the subsystems. The reader should be aware, though, that some of the properties do depend on multiple subsystems and the way they interconnect. The main point is that if we choose a different design for the precoder, source coder, termination coder or synchronization subsystem, then as long as these properties hold, then Theorem 2.4.1 also holds.

Subsystem Property 1 If D^n is uniformly distributed over $\{0,1\}^n$, then there exists a constant $0 < C_\pi < \infty$ that is independent of n such that

$$E[\ell(\pi_n(D^n))] \leq (n + C_\pi)/H(X). \tag{2.39}$$

Subsystem Property 2 The precoders $\{\pi_n\}_{n=1}^{\infty}$ and the source coders $\{\sigma_n : \mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}^\dagger\}_{n=1}^{\infty}$ are such that there exists a a function λ such that

$$E[L_{i+1}^{\sigma}] \leq E[L_i]\, H(X|Y) + \lambda(E[L_i]), \quad \text{for } i = 0, \cdots, B_N - 1, \tag{2.40}$$

where λ also has the properties that $\lambda(x)=o_x(x)$ and λ is non-negative, monotonically increasing, and concave ($\cap$) over $[1,\infty)$.

Subsystem Property 3 The termination coding scheme c^{term} takes two parameters κ and ν and returns a code. For any $\alpha > 0$, there exists a sequence of parameters $\{(\kappa_n, \nu_n(\alpha))\}_{n=1}^{\infty}$ such that $c^{\text{term}}(\kappa_n, \nu_n(\alpha))$, whose corresponding encoder is denoted $e^{\text{term}}_{\kappa_n,\nu_n(\alpha)}$, encodes $\kappa_n = \lceil n^{1/4} \rceil$ message bits into an average number of channel inputs $\eta_n = \alpha n + o_n(n)$ and has error probability $P_{\text{e,term},n}$ bounded according to

$$P_{\text{e,term},n} < \exp_2\left\{-\eta_n\left(E_{cw} - o_n(1)\right)\right\}, \tag{2.41}$$

where $\exp_2(x) \triangleq 2^x$.

Subsystem Property 1 is proven in Appendix A.1, Subsystem Property 2 is proven in Appendix A.2, and Subsystem Property 3 is proven in Appendix A.3.

Using these key properties, we can now prove the theorem as follows: Let $r < I(X;Y)$ be given, and let us construct a sequence of codes with corresponding sequence of rates and error probabilities satisfying (2.7) and (2.8).

Intuition from the third illustration in Section 2.1 suggests that the expected transmission length $E[L^*]$ of a code sending N message bits and using a termination coder that puts out a sequence of average length η, satisfies

$$E[L^*] \approx \frac{N}{I(X;Y)} + \eta. \tag{2.42}$$

This equation in turn suggests that a sequence of codes with rate converging to r is $\{c_{\text{DMC}}(n, \nu_n)\}_{n=1}^{\infty}$, where ν_n is a termination code parameter giving the termination code a corresponding expected length of $\eta_n(r) + o_n(n)$, where

$$\eta_n(r) = \frac{n}{r}\left(1 - \frac{r}{I(X;Y)}\right). \tag{2.43}$$

That an appropriate sequence of parameters $\{\nu_n\}$ exists is guaranteed by Subsystem Property 3. Let us examine this sequence of codes $\{c_{\text{DMC}}(n, \nu_n)\}_{n=1}^{\infty}$ more closely to verify that it behaves as desired.

To prove that the sequence of rates corresponding to this sequence of codes satisfies (2.7), we first develop a bound on the the expected transmission length $E[L^*]$ of the code $c_{\text{DMC}}(N, \nu_N)$ as follows: Using the notation of Section 2.3, first consider the termination-coded transmission of the sequence G^{∞} defined in (2.16). If the receiver fails to detect the sequence $(0,1,0,1)$ when it first appears in the transmission G^{∞}, then it is detected in a subsequent block because this coded sequence is repeated indefinitely thereafter [cf. (2.16)]. Moreover, for each of these blocks the probability of a missed detection is also less than $P_{\text{e,term},N}$, the probability of error associated with the termination coder used by $c_{\text{DMC}}(N, \nu_N)$. Thus, the expected length of the transmission starting with the length-t_N transmission of $\emptyset^{t_N}[N]$ until termination is less than

$$\mu_{\mathrm{I}} \triangleq t_N + \left(E[B_G] + \frac{P_{\mathrm{e,term},N}}{(1 - P_{\mathrm{e,term},N})}\right)(\eta_N(r) + o_N(N)). \tag{2.44}$$

Furthermore, the expected length of the transmission before $\emptyset^{t_N}[N]$ in (2.17) is

$$\mu_{\mathrm{II}} \triangleq \sum_{i=0}^{B_N-1} E[L_i]. \tag{2.45}$$

Hence, the total expected length of the transmission is bounded according to

$$E[L^*] < \mu_{\mathrm{I}} + \mu_{\mathrm{II}}. \tag{2.46}$$

The following lemma, which is proven in Appendix A.4, uses Subsystem Properties 1 and 2 to upper bound μ_{II}:

Lemma 2.4.1

$$\mu_{\mathrm{II}} \leq \left(\frac{N}{H(X) - H(X|Y)}\right) + o_N(N) \tag{2.47}$$

And the next lemma, which is proven in Appendix A.5, uses Subsystem Property 3 to upper bound μ_{I}:

Lemma 2.4.2

$$\mu_{\mathrm{I}} < t_N + (1 + o_N(1))(\eta_N(r) + o_N(N)) \tag{2.48}$$

Since $t_N/N = o_N(1)$, by substituting (2.48) and (2.47) into (2.46), we get that

$$E[L^*]/N < \frac{1}{I(X;Y)} + (1 + o_N(1))\,\frac{\eta_N(r)}{N} + o_N(1). \tag{2.49}$$

This inequality with (2.43) implies that the rate R_N of $c_{\mathrm{DMC}}(N, \nu_N)$ satisfies $R_N > r + o_N(1)$.

The last step in showing that E_{CEC} is attainable is to find the probability of error P_N corresponding to $c_{\mathrm{DMC}}(N, \nu_N)$. With the decoder described above, the invertibility of the source coder and precoder—together with the perfect detectability of $\emptyset^{t_N}[N]$—mean that decoding errors in the overall scheme occur only if one of the B_G blocks that is termination coded is decoded incorrectly. Since P_N equals the probability of such an event, it can be union bounded above according to

$$P_N \leq E[B_G]\, P_{\mathrm{e,term},N}. \tag{2.50}$$

Inequality (2.41) with (2.32) gives an upper bound on right-hand side of (2.50). Substituting (2.43) into this upper bound, taking the log and multiplying by $-R_N/N$ yields

$$-\frac{R_N \log P_N}{N} \geq \frac{R_N}{r}\left(1 - \frac{r}{I(X;Y) + o_N(1)}\right)\left(E_{cw} - o_N(1)\right), \tag{2.51}$$

$$\geq E_{\text{CEC}}(r) - o_N(1), \tag{2.52}$$

where (2.52) follows from the fact that $R_N > r + o_N(1)$. Since these results hold for $c_{\text{DMC}}(N, \nu_N)$ for arbitrary N, the theorem is proved. □

2.5 Removing the Perfect-Detection Assumption

In the previous section, we assume that a perfectly detectable sequence $\emptyset^{t_N}[N]$ exists. Since, in general, no such sequence exists when the forward channel is a DMC, we must modify the coding scheme before it can work with a forward channel that is a DMC. In this section, we outline modifications to our basic iterative coding scheme that allow us to remove the assumption that $\emptyset^{t_N}[N]$ is perfectly detectable.

To develop the modified scheme, we first construct $\emptyset^{t_N}[N]$ out of legitimate channel inputs from $\mathcal{X}$. Let a be an element of $\mathcal{X}$ defined by

$$a = \arg\max_{x \in \mathcal{X}} D(q_{Y|X}(\cdot|x) \parallel p_Y) \tag{2.53}$$

and then let $\emptyset^{t_N}[N] = (a, \cdots, a)$. The encoder uses the sequence in the same way, communicating the time index A_{B_N} defined in (2.12) to the decoder by sending $\emptyset^{t_N}[N]$ at time $A_{B_N} + F + 1$ and later sending bits describing F. The decoder tests for the presence of this sequence in each new incoming block of t_N samples, using a minimum-probability-of-error detector δ_{t_N} of the form

$$\delta_{t_N}(y^{t_N}) = \begin{cases} 1 & \text{if } \prod_{i=1}^{t_N} q_{Y|X}(y_i|a) \geq \prod_{i=1}^{t_N} p_Y(y_i) \\ 0 & \text{otherwise} \end{cases}. \tag{2.54}$$

There is now the possibility that δ_{t_N} returns a 0 when $\emptyset^{t_N}[N]$ is sent over the channel (missed detection) or that δ_{t_N} returns a 1 when $\emptyset^{t_N}[N]$ is not sent over the channel (false alarm). As t_N increases, the probability of either sort of error can be shown to be less than $\exp_2\{-\beta t_N\}$ for some $\beta > 0$.

We now encounter a dilemma in choosing t_N: If we accept that $W^N \neq \hat{W}^N$ whenever a false alarm or missed detection occurs, then we need to choose t_N proportional to N to maintain probability of error decaying exponentially in N. But choosing t_N proportional to N causes the rate to decrease by an asymptotically non-negligible amount. On the other hand, if we choose $t_N = o_N(N)$, then the probability of error does not decrease exponentially in N.

The solution is to choose $t_N = o_N(N)$ but to use feedback to detect missed detections and false alarms, allowing the encoder to take corrective action. We therefore choose t_N according to

$$t_N = \left\lceil \sqrt{N} \right\rceil, \tag{2.55}$$

which ensures that the probability of a false alarm or missed detection occurring at any time during the transmission decays to 0 as $N \to \infty$. The scheme is then modified as follows.

As in the idealized case, the receiver has two modes of operation: normal mode and termination-decoding mode. In normal mode, at every time k that is an integer multiple of t_N, the receiver tests the most recent t_N channel outputs to see whether $\delta_{t_N}(Y_{k-t_N+1}^k) = 1$. If this condition holds true, then the receiver enters termination-decoding mode; this is the only way the receiver can go from normal mode into termination-decoding mode. Once in termination-decoding mode, the receiver decodes each incoming block to find the message coded in that block. Concatenating these messages, the receiver stops receiving when it is in termination-decoding mode and finds the sequence $(0,1,0,1)$ somewhere in the concatenated message.

The transmitter knows exactly what the receiver is doing via the feedback. Hence, it can exploit the fact that the receiver always enters termination-decoding mode on detection of $\emptyset^{t_N}[N]$ by sending the receiver a message regarding its detection of $\emptyset^{t_N}[N]$. In particular, if a false alarm occurs, then the sequence $(1,0,W_1,W_1,W_2,W_2,\cdots,W_N,W_N,0,1,0,1,0,1,\cdots)$ is transmitted in blocks of κ_N bits using the termination coder. The first two elements of the sequence, $(1,0)$, inform the receiver that a false alarm has occurred and that the remainder of the sequence is to be regarded as the original message. Note that even if some of these κ_N-bits blocks are decoded incorrectly, the receiver eventually sees the sequence $(0,1,0,1)$ and stops data acquisition.

In the case of a missed detection—that is, when $\emptyset^{t_N}[N]$ is transmitted but not detected by the receiver—the transmitter resends $\emptyset^{t_N}[N]$ until it is detected by the receiver. After detection, the transmitter sends $(\phi(C_{\text{MD}}),G^{\infty})$ coded in κ_N-bit blocks using the termination coder. The sequence $\phi(C_{\text{MD}})$ encodes the number C_{MD} of missed detections that occurred. From this information, the receiver can correctly ascertain the value of $A_{B_N}+1$.

In Figure 2-5, a flow chart giving an outline of how the scheme makes use of the synchronization subsystem is shown.

2.5.0.1 Reliability of the Modified Scheme

It is proven in Appendix A.7 that Theorem 2.4.1 continues to hold for the modified scheme when a synchronization subsystem can be designed with the following property:

Subsystem Property 4 With $\{t_N\}_{N=1}^{\infty}$ a sequence satisfying $t_N = o_N(1)$, the sequence of detector-sequence pairs $\{(\delta_{t_N},\emptyset^{t_N}[N])\}_{N=1}^{\infty}$ is such that the false-alarm and missed-detection probabilities P_{FA,t_N} and P_{MD,t_N}, respectively, associated with each use of δ_{t_N} by the receiver, satisfies

$$P_{\text{FA},t_N} < o_N(N^{-3}) \tag{2.56}$$

$$P_{\text{MD},t_N} < M_{\text{MD}} + o_N(1) \tag{2.57}$$

for some constant $M_{\text{MD}} < 1$.

In Appendix A.6, Subsystem Property 4 is shown to hold for the synchronization subsystem design given by (2.53)–(2.55).

2.6 Complexity

To this point, we have claimed that the compressed-error-cancellation framework leads to low-complexity coding schemes. In what follows, we analyze the complexity of the coding scheme we designed for DMC$_f$'s and argue that it has time and space complexity that is linear in the length of the number of channel inputs used.

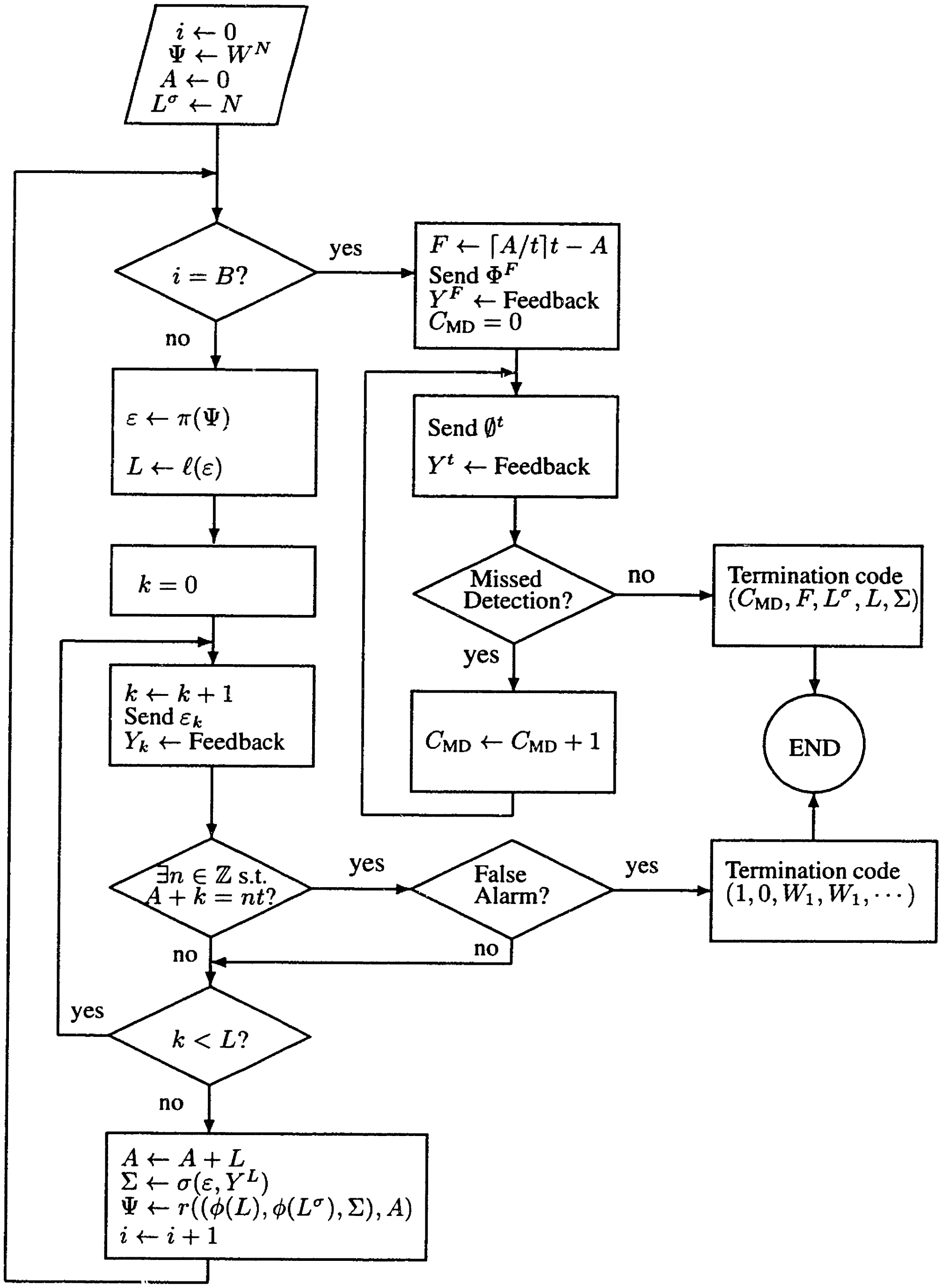

$i \leftarrow 0$
$\Psi \leftarrow W^N$
$A \leftarrow 0$
$L^\sigma \leftarrow N$
$i = B?$
yes
$F \leftarrow \lceil A/t \rceil t - A$
Send Φ^F
$Y^F \leftarrow$ Feedback
$C_{\text{MD}} = 0$
no
$\varepsilon \leftarrow \pi(\Psi)$
$L \leftarrow \ell(\varepsilon)$
Send $\emptyset^t$
$Y^t \leftarrow$ Feedback
$k = 0$
Missed Detection?
no
Termination code
$(C_{\text{MD}}, F, L^\sigma, L, \Sigma)$
yes
$k \leftarrow k + 1$
Send ε_k
$Y_k \leftarrow$ Feedback
$C_{\text{MD}} \leftarrow C_{\text{MD}} + 1$
END
$\exists n \in \mathbb{Z}$ s.t.
$A + k = nt?$
yes
False Alarm?
yes
Termination code
$(1, 0, W_1, W_1, \cdots)$
no
no
yes
$k < L?$
no
$A \leftarrow A + L$
$\Sigma \leftarrow \sigma(\varepsilon, Y^L)$
$\Psi \leftarrow r((\phi(L), \phi(L^\sigma), \Sigma), A)$
$i \leftarrow i + 1$

Figure 2-5: Encoding modified for imperfect detection of $\emptyset^{t_N}[N]$. The notation "$Y_k \leftarrow$ Feedback" indicates that one sample of feedback is retrieved and stored in Y_k. Similarly, "$Y^a \leftarrow$ Feedback" indicates that a samples are retrieved and stored in Y^a.

2.6.1 Time and Space Complexity for the Transmitter and Receiver

To assess the time and space complexities of the encoder and decoder, we first assess the time and space complexities of each of the four constituent subsystems: precoding, source coding, synchronization, and termination coding.

Precoding and postcoding (precoding inversion) operate nearly identically to arithmetic source coding decoders and encoders, respectively, for i.i.d. sources. The dominant computation required for arithmetic source coding and decoding is the computation of the relevant cdf. Because the cdf can be computed using a recursive algorithm [14], it can easily be seen that arithmetic coding can be performed with time cost that is proportional to the sum of the number of inputs and outputs under the uniform-cost criterion. Space cost, excluding buffering of input and output, can be easily seen to be a constant, so the total space cost including buffering requirements is proportional to the sum of the number of inputs and outputs. Thus, the total time cost and total space cost associated with precoding in the transmitter and postcoding in the receiver are linear in L^*.

When carrying out precoding and postcoding on a digital computer with finite-length registers, that the cdf must be computed to arbitrary precision is a concern. In [63], a k-bit integer arithmetic implementation of an arithmetic source coding encoder and decoder for an i.i.d. source is given. The implementation requires appropriate quantization of the marginal pmf for the source. The quantization error in the characterization of the marginal pmf decreases exponentially with the number of bits allowed in the registers and presents no practical difficulties. The implementations of the encoder and decoder both require a number of k-bit integer operations that grows linearly with the sum of the number of inputs to and outputs from each. We have adapted this implementation to precoding and postcoding and achieved the same behavior on a digital computer with finite-length registers. Thus, the linear time complexity and linear space complexity seems to accurately reflect the computation required when implementing the procedures on a digital computer.

The source encoding and decoding subsystems are based on Shannon-Fano source coding, which can also be carried out using the arithmetic source coding algorithm. Use of arithmetic coding again results in the time costs of the source encoder and decoder both being linear in the length of their inputs plus outputs. Space cost is again constant. Therefore, the time complexities and space complexities associated with source encoding and decoding are also $O_{L^*}(L^*)$. As for implementation on a digital computer with finite-length registers, the same comments apply as for precoding and postcoding.

The synchronization subsystem requires that the transmitter send the synchronization sequence and that the decoder test for the sequence every t_N samples using δ_{t_N}. Sending the sequence clearly takes a number of operations linear in t_N and requires t_N buffer registers to store the sequence. Each use of δ_{t_N} clearly costs time linear in t_N under the uniform-cost criterion. Space cost is clearly $O_N(t_N)$. Since δ_{t_N} is used fewer than L^*/t times, the total time complexity attributable to the synchronization subsystem in the receiver is linear in L^*. Each time δ_{t_N} is used, it can reuse its registers, so the total space cost is only $O_N(t_N)$. The transmitter must also perform each of these hypothesis tests to determine the state of the receiver, so it shares the same complexity.

On a digital computer with finite-length registers, we must worry about "underflow", i.e., we must worry about the probabilities becoming so small that they are rounded to zero. To avoid underflow, we compute the negative log-likelihoods, which increase linearly in t_N, instead of the likelihoods, which decrease exponentially in t_N. The only new problem seems to be the possibility of "overflow", i.e., that the negative-log-likelihood becomes so large that it is called infinity by the computer. But, clearly, extremely large values of t_N are required before this can happen, and it is a simple matter to allocate a sufficiently large register to hold this log probability. For example, a 1024-bit register is large enough to prevent overflow for any reasonable size of t_N. We should also note that rounding of the intermediate sums to a certain number of significant digits is allowable, because the negative-log-likelihood for the correct hypothesis should be smaller than the incorrect hypothesis by an amount proportional to t_N. Hence, the characterization of δ_{t_N} as costing $O_N(t_N)$ in time under the uniform-cost criterion is a reasonable estimate of the time required to perform the computation cn a digital computer with finite-length registers for reasonable values of t_N. For extraordinarily large values of t_N, even if $O_N(\log t_N)$ registers are allocated to hold the running negative-log-likelihood sum, most of the time, an addition only affects the least significant register, so it appears that $O_N(t_N)$ is still an accurate characterization of the time cost even for such enormous values of t_N.

The number of computations required for mSB decoding depends on the specific inner code used, but is at most

$$O_N(\kappa_N{}^2) + O_\nu(\nu) \tag{2.58}$$

under the uniform-cost criterion for each time the inner codeword and corresponding length-ν verification message are sent. The first term accounts for decoding of the inner code at the receiver (which must also be replicated at the transmitter). The second term accounts for the computation required for the transmitter to send c^ν or w^ν and for the receiver to distinguish the two sequences. For the important case in which $\nu \propto N$, we can write $\kappa_N = O_\nu(\nu^{1/4})$, and the two terms in (2.58) can be combined into $O_\nu(\nu)$, i.e., the time and space cost of a single use of the mSB coder is nearly proportional to the number of channel inputs used by the coder. Since the total number of channel uses due to mSB coding is less than L^*, the total computation due to mSB decoding must also be $O_{L^*}(L^*)$. The time complexity of mSB encoding is less than for decoding, so it can also be shown to be $O_{L^*}(L^*)$. Space complexity for the mSB encoder and decoder is $O_\nu(\nu)$ under both cost criteria, since c^ν and w^ν and the inner codebook, all of which are composed of discrete-valued sequences, must be stored.

For implementation of the mSB decoder on a digital computer with finite-length registers, the same comments made regarding the synchronization detector hold for maximum-likelihood decoding of the inner codebook and verification sequences. That is, the complexity characterization under the uniform-cost criterion appears to be valid.

Summarizing, each subsystem has time and space complexity that is $O_{L^*}(L^*)$ under the uniform-cost criterion, and therefore so does the overall scheme. It follows that at a fixed rate, the expected number of computations and amount of storage required by the transmitter and receiver is $O_N(N)$. Furthermore, this characterization of the complexity seems to reflect the behavior in practice were these algorithms to be implemented on standard digital computers.

2.6.2 Uniform Versus Non-Uniform Complexity

In comparing this scheme to other low-complexity coding schemes, we see that concatenated codes [21] with linear complexity can also be designed. Concatenated codes encode data first with an outer code and then with an inner code. The inner code is usually a relatively short block code that is used to obtain relatively small error probabilities on the channel. The outer code is then used to correct errors that escape the inner code, driving the error probability even lower. Forney showed that this structure allows error probabilities that decay exponentially with total blocklength with polynomial computational complexity. It has recently become possible to obtain linear computational complexity with exponentially decaying error probability by using Spielman's linear-complexity codes [56]. Specifically, this performance is obtained by using Spielman's codes as outer codes and using a randomly selected inner code. It may be tempting then to conclude that feedback offers no advantages in terms of complexity.

But there is a very important distinction between the linear complexity of the feedback scheme we have just introduced and the linear complexity of such a concatenated code. A concatenated coding scheme whose inner code is decoded via exhaustive-search maximum-likelihood decoding requires more computation per message bit as its operating rate increases. That is, at a particular value of the rate R, the computation per message bit is independent of the number of message bits. But the computation per message bit depends heavily on the rate and increases rapidly and without bound as the rate approaches capacity. While the problem can be mitigated by using special ·nner codes that can be decoded with more computationally efficient decoders, no such capacity-achieving codes and corresponding decoders appear to be known.

On the other hand, it is straightforward to verify that the feedback scheme we have introduced does not behave in this way. The rate of ιhe feedback scheme can be made to increase to capacity by letting $\nu/N \to 0$ and $N \to \infty$. Computations per input sample need not grow without bound as these two limits are approached. There must therefore exist a positive number U *independent* of the rate R and the number of message bits N such that the average number of computations per channel input required for encoding and decoding is less than U for any R below the channel capacity.

We say that our feedback scheme has *uniform* linear complexity, while the above concatenated scheme is an example of a scheme with *non-uniform* linear complexity. The difference has important consequences in terms of what rates are actually achievable in practice.

2.7 Variations

The coding scheme we have developed thus far in this chapter is only one of many possible schemes based on the compressed-error-cancellation framework. A number of minor variations on the coding scheme are possible. For example, a synchronization sequence could be transmitted after every iteration, so that block boundaries are known sooner. We could use a different feedback or feedback-free code as the termination code. Or we could let the number of iterations be random, using the termination coder when the length of the message drops below a certain threshold. Such variations could lead to different performance characteristics and might be appropriate for certain situations.

One variation that is particularly important for certain analytical purposes that we encounter in the remainder of this chapter and in Chapter 5, is a scheme that uses lossy (but effectively lossless) precoders and source coders with fixed-length inputs and outputs. We describe this scheme in the following section.

2.7.1 Scheme Using a Fixed-Length Precoder and Source Coder

We construct a scheme that uses the fixed-length precoder $\pi_{\delta,n} : \{0,1\}^n \to \mathcal{X}^{f_\delta(n)}$, where $f_\delta(n) = \lceil n/(H(X) - \delta)\rceil$ and the fixed-length source coder $\sigma_{\delta,n} : \mathcal{X}_1 \to \{0,1\}^{g_\delta(n)}$, where $g_\delta(n) = \lceil n(H(X|Y) + \delta)\rceil$. The constant δ is a design parameter. These two subsystems are described in Appendix A.9. The encoder $e^{\text{FL}}_{\delta,N}$ then works essentially as the single-user coder described in Section 2.2 but has fixed length:

Initialization:

$$l_0^\sigma = N, l_0 = \lceil l_0^\sigma/(H(X)-\delta)\rceil, A_0 = 0 \tag{2.59}$$

$$\Sigma_0 = w^N, \epsilon_0 = \pi_{\delta,l_0^\sigma}(r(\Sigma_0, 0)) \tag{2.60}$$

for $i = 1, \cdots, B_N$:

$$l_i^\sigma = \lceil (H(X|Y)+\delta)l_{i-1}\rceil \tag{2.61}$$

$$l_i = \lceil l_i^\sigma/(H(X)-\delta)\rceil \tag{2.62}$$

$$A_i = A_{i-1} + l_{i-1} \tag{2.63}$$

$$\Sigma_i = \sigma_{\delta,l_{i-1}}(\epsilon_{i-1}, y_{A_{i-1}+1}^{A_i}) \tag{2.64}$$

$$\epsilon_i = \pi_{\delta,l_i^\sigma}(r(\Sigma_i, i)) \tag{2.65}$$

Number of channel inputs:

$$l = A_{B_N} + \ell(\Sigma_{B_N})N^{1/4} \tag{2.66}$$

Encoding function:

$$e^{\text{FL}}_{\delta,N}(w^N, y^l) = (\epsilon_0, \cdots, \epsilon_{B_N-1}, e^{\text{rep}}_{\lceil N^{1/4}\rceil}(\Sigma_{B_N})) \tag{2.67}$$

where $r : \{0,1\}^\dagger \to \{0,1\}^\dagger$ is the same binary randomization function used in Section 2.3, and B_N is chosen as

$$B_N = \frac{3\log N}{4(\log(H(X)-\delta) - \log(H(X|Y)+\delta))} \tag{2.68}$$

so that $l_{B_N}^\sigma \approx N^{1/4}$. Then $e^{\text{rep}}_{\lceil N^{1/4}\rceil}$ encodes its data by using a two-codeword codebook of length $\lceil N^{1/4}\rceil$ to encode each of the $l_{B_N}^\sigma$ bits. It can easily be shown that the rate of this coding scheme is $I(X;Y) - 2\delta + o_N(1)$, and the error decreases exponentially with $N^{1/4}$. These performance characteristics hold for any $\delta > 0$.

This fixed-length code has relatively slow decay of the probability of error with N. To construct a variable-length code with better error properties, we simply use it as an inner code with an mSB scheme, which would then attain the error exponent $(1 - r/(I(X;Y) - 2\delta))E_{cw}$, where E_{cw} is as given in (2.38).

This scheme has a number of advantages: it is easier to describe because it requires no synchronization subsystem, it has essentially the same performance as the scheme given in Section 2.3, and the lengths of the transmissions on each iteration are fixed. These make it advantageous for certain techniques that are more easily analyzed with fixed-length iterations.

The scheme also has a number of disadvantages: As will become apparent in Chapter 4, universal communication is not possible because of the fixed-length nature of the scheme; the scheme is not robust to channel modeling errors; and, for finite-state channels, it is not clear how to generalize the techniques used to prove that the fixed-length precoders and source coders work. Nevertheless, we see in later parts of this thesis that this version of the scheme is useful in some circumstances.

2.8 Discussion

Thus far, we have concerned ourselves with certain fundamental theoretical aspects of the feedback coding scheme. A host of other interesting issues remains. We discuss some of these issues in this section.

2.8.1 Structuring Computation

We have assumed in the foregoing analyses that computation takes no time. This assumption is fairly common in the channel coding literature. For example, when computing error exponents, which can be viewed as relating decoding delay and probability of error, delay due to computation is generally ignored.

It may seem almost contradictory to assume that computation takes no time but to concern ourselves with computational complexity. However, there are cases in which complexity may be a concern even though we can effectively ignore the time delays associated with computation within the coding scheme. Such cases include, for example, the case in which a processor is so powerful that all needed computations require negligible time, but incur a financial or energy cost; another case is that in which coding is performed for a recording channel, in which the goal is simply to fit the most data onto the medium.

But often we have a computer that does a fixed number of computations per unit of time, and low-complexity encoding and decoding algorithms are required to ensure that the computer keeps pace with the encoding and decoding of a long stream of messages. Note that in this situation, any code with a total number of encoding or decoding operations growing faster than $O_N(N)$ would require the blocklength N to be limited to some finite number which would in turn require the probability of error to be above some positive constant. Only a scheme using $O_N(N)$ computations is capable of truly providing arbitrarily low error rates.

Such a setup with a computer providing constant computations per unit time may cause delays during which computation is performed. Prior research generally has ignored such delays, perhaps because such delays do not create serious problems as long as the computer does not fall behind in servicing the computational demands. For example, for FEC codes in which there is an infinite stream of data to be sent, one can easily structure the transmission so that computation for encoding or decoding of one message takes place during transmission of another message.

But complications may arise in feedback schemes. In particular, in the framework we have described in this chapter, if we assume that the precoders and source coders require the full block of input before they can compute their outputs[11], a computational delay arises between iterations that is proportional to the length of the transmission on the previous iteration. If we send information-free filler during the delay, then the rate is reduced, because we waste a number of channel inputs proportional to N. Even with such a delay, our scheme is computationally more desirable than any scheme whose complexity grows faster than N in the following sense: For any particular transmission rate, we can at least find a computer providing a sufficiently large number of computations per

[11] This is not actually the case for the subsystems described in this chapter, but it is the case for those described in Chapters 3 and 4.

second to allow arbitrarily small probability of error and arbitrarily large blocklengths. However, the computer must become more powerful as the rate increases, suggesting that the scheme behaves with a sort of non-uniform linear complexity.

Fortunately, we can structure the computation so that computation is performed while useful, information-bearing data rather than information-free filler, is being sent. This allows us to find a particular sufficiently powerful computer with which any rate below capacity can be achieved with arbitrarily small probability of error.

To demonstrate the technique, which we call *interleaving*, we use the variation in Section 2.7 that uses a fixed-length precoder and source coder, sending a $2N$-bit message as follows:

- Precode the first N bits into $N/H(X)$ inputs.
- Precode the second N bits into $N/H(X)$ inputs.
- Send the first length-$N/H(X)$ block of precoded data.
- Send the second length-$N/H(X)$ block of precoded data. While these data are being sent, use the feedback about the first block of precoded data to source code the first $N/H(X)$ transmissions into $NH(X|Y)/H(X)$ bits and also precode these bits into $NH(X|Y)/H^2(X)$ inputs.
- Send these $NH(X|Y)/H^2(X)$ inputs. While they are being transmitted, source code the second $N/H(X)$ transmissions into $NH(X|Y)/H(X)$ bits and also precode these bits into $NH(X|Y)/H^2(X)$ inputs.
- Send these $NH(X|Y)/H^2(X)$ inputs. While they are being transmitted, ...

After approximately $2N/I(X;Y)$ samples have been sent, we send the final coded block of data and the verification message. This technique allows us to use the computer and channel efficiently. We can see that the computer must be fast enough to process n channel inputs in the time required to send $nH(X|Y)/H(X)$ channel inputs.

Empirical evidence indicates that the technique works even when using the variable-length precoding and source coding subsystems. For very long blocklengths, we would expect this behavior because the lengths of the outputs from these variable-length subsystems cluster sharply around the expected value of the length. But even for relatively short message lengths, such as 100 bits, the technique seems to work. However, analysis seems to be difficult, although some preliminary arguments for why the technique should work are given in Appendix A.10.

2.8.2 Variability of Transmission Length

Because our scheme produces variable-length transmissions, buffer overflows are possible. However, for large blocklengths, preliminary experiments suggest that variations in transmission length due to the varying lengths of the precoder and source coder outputs are modest; for example, in an experiment in which $E[A_{B_N} + F] \approx 204500$, the maximum value of $A_{B_N} + F$ in 100 trials was about 206336, and the sample standard deviation of L^* was about 794. Again, this behavior is not surprising — for long blocks, most of the relevant sequences are "typical" and are compressed to "typical" lengths, which are close to the corresponding expected length.

2.8.3 Number of Iterations

The number of iterations B_N need not be chosen strictly according to (2.9). A variety of choices of B_N allow the asymptotic theoretical results to follow. In practice, we find that for a particular channel and value of N, the length of the transmission on the ith iteration tends toward some positive constant as i increases. A good choice of B_N is the value of i at which this positive constant is first reached. As a general guideline, B_N should increase as N increases and decrease as the quality of the channel increases.

2.8.4 Feedback Delay

When communicating over large distances, a significant delay in feedback may be present. That is, the transmitter at time k may only know Y^{k-d}, where d is some fixed delay. The primary effect of this delay is that at the beginning of an iteration, the transmitter may not yet have enough feedback to start the iteration. In this case, the transmitter may send information-free filler data. This wastes at most $B_N d$ samples, which is negligible as $N \to \infty$. If feedback delays are large compared to the desired blocklength, however, one could use a multiplexing strategy whereby one sends another message during the periods that would otherwise be idle.

2.8.5 Inaccurate Channel Modeling

The variable-length version of the scheme we have described has some attractive robustness properties that make it well suited for use on channels that are modeled inaccurately. As an example, reliable communication over the $\mathrm{BSC_f}$ is possible even when the parameter ϵ is a noisy estimate of the true crossover probability. The price paid for this mismatch is in terms of rate: the smaller the mismatch, the closer the achievable rate is to capacity. When the iterative coding scheme is appropriately designed, the resulting degradation can be quite graceful. Note that this robustness contrasts sharply with the behavior of more traditional FEC codes. In particular, with pure FEC coding the probability of error can go sharply to 1 when the channel signal-to-noise ratio exceeds a code-dependent threshold.

To understand how the graceful degradation property can be obtained with iterative coding, it is important to understand the factors contributing to the rate gap. One component is due to the fact that the source coder is unable to operate at the appropriate entropy rate, if the source coder is not accurately tuned to the pmf of the true channel. In fact, the source coding rate that is achieved increases compared to the true source entropy by an amount given by the Kullback-Leibler distance between the true channel and the assumed channel model [14]. A second component to the rate gap in the iterative coding scheme under mismatch conditions occurs because the capacity-achieving input distribution of the channel model may not be the capacity-achieving distribution of the true channel.

Achieving this graceful degradation property requires that the final block of data be mSB coded in such a way that it can be decoded correctly with high probability even when there is channel model mismatch. When we know the channel has some minimum quality (e.g., in the case of the $\mathrm{BSC_f}$, if we know that the crossover probability ϵ is always less than $\epsilon_0 < .5$), we can termination code for this worst-quality case without sacrificing overall rate.

We show in Chapter 4 that with appropriate modification of the subsystems, a system that incurs no rate loss due to channel mismatch can be designed — in fact, the system needs no channel model at all.

2.8.6 Simulation

To verify that our scheme behaves as predicted by our analysis and that the asymptotic performance is approachable in practice, we implemented the scheme and simulated its performance on a digital computer with finite-length registers.

To simultaneously demonstrate that the coding scheme is viable on continuous-valued channels, we applied it to the Gaussian channel. To use the scheme on a continuous-valued channel, the channel inputs and outputs must be quantized and the corresponding channel transition probabilities determined. In a preliminary experiment, we used the fifteen input symbols $\{-7, -6, -5, \cdots, 5, 6, 7\}$ and chose an approximately Gaussian input distribution with variance 4.0 as an input to a discrete-time channel with zero-mean additive white Gaussian noise of variance 4.0. We then quantized the output to the twenty-one symbols $\{-10, -9, -8, \cdots, 8, 9, 10\}$. We simulated the Gaussian channel to empirically calculate the channel transition probabilities for this quantized channel, and used this channel model in our coder. With $N = 10^6$, and $\nu = 1000$, our coder achieved a rate[12] of 0.483. The probability of error can be determined to be less than $1 - F(\sqrt{\nu})$, where F is the cdf for a unit-variance, zero-mean Gaussian random variable, which is upper bounded [41] by $\exp\{-\nu/2\}/\sqrt{2\pi\nu}$. Comparing this performance with the capacity of the discrete-time Gaussian channel with a 0 dB signal-to-noise ratio, which is 0.50 bits per channel input, our scheme appears very promising for use with continuous-valued channels and has certain advantages over the scheme of Schalkwijk et al. [49]. Namely, our scheme can be easily adapted to achieve rates near capacity on non-Gaussian channels such as fading channels and also allows quantized feedback.

We note that the characterization of the overall time complexity as linear in N seems to accurately reflect the time taken to run the algorithm on the digital computer. Indeed, the fact that a 10^6-bit message could be encoded and decoded is evidence that the computation per bit is reasonably low even though the message length is large.

2.8.7 Summary and Future Directions

We developed a linear-complexity coding scheme giving exponentially decaying error probabilities at any rate below the capacity of any DMC$_\text{f}$. The error exponent achieved by the scheme is the highest possible. We explored variations of the scheme and also implemented the scheme on a computer to show it is practically viable.

A particularly interesting direction for future research includes the development of a "sequential" scheme based on the compressed-error-cancellation framework. A sequential scheme is one in which each of an infinite sequence of bits is transmitted and decoded individually rather than as a block. Horstein's scheme [30] is an example of such a scheme. A sequential scheme is useful in practice when the delay caused by long blocklengths is undesirable. For example, when transmitting speech in a two-way conversation, only very short delays are tolerable.

[12] The theoretical capacity of this DMC approximation to the Gaussian channel is 0.493 bits/input. In our simulation, this rate was not approached more closely in part because our implementation of the source coder and precoder used 32- rather than 64-bit integers, which did not allow for sufficiently accurate characterization of the channel transition pmf. This shortcoming can easily be remedied with a more sophisticated implementation.

While we have not been able to develop a sequential scheme based on the compressed-error-cancellation framework, we believe that Horstein's scheme may be related to the scheme developed in this chapter. The relationship between these two schemes may be worth further exploration and may reveal a corresponding sequential version of the compressed-error-cancellation framework.

Additional interesting directions for future research that are not addressed in the remainder of this thesis include finding the pmfs or higher moments of L_i, and L_i^σ for all i. Also, one could attempt to analyze the scheme without the randomizing function r. Extensions to channels with countable and uncountable input and output alphabets may also be of interest.

Chapter 3

Channels with Memory

3.1 Introduction

While DMC's provide accurate models for some channels used in practice, they are not accurate models of channels with memory. A channel has memory if the output of the channel at a particular time k depends statistically on the input or output of the channel at a time other than k.

An important example of a channel with memory is a wireless electromagnetic channel relying on non-line-of-sight propagation. In practice, such channels may arise, for example, in high-frequency-band communication and cellular telephony. The memory in such channels arises because the transmitted signal arrives to the receiver via multiple paths, each of a different length, at several different times [44]. Such "multipath propagation" gives rise to a phenomenon referred to as fading, in which the transmitted signal is attenuated or boosted because of the destructive or constructive interference of the multiple signals. The amount of the fading may depend on certain elements of the physical environment such as location of clouds, or temperature, which may vary randomly but also smoothly with time so that the amount of attenuation is a random process with memory.

Many researchers have studied fading channels and other channels with memory (see, for example, [44] and the references therein). A number of techniques have been proposed for fading channels without feedback, and some have been used with success to communicate over such channels. An example of such a technique is interleaving, in which the symbols of the channel are scrambled to make it appear memoryless. In contrast, the use of feedback on channels with memory has been comparatively less explored.

In this chapter, we apply the compressed-error cancellation framework to coding for channels with memory. In particular, we restrict our attention to channels whose statistical parameters are known *a priori* to both transmitter and receiver. In Chapter 4, we extend our results to the important case in which neither transmitter nor receiver has prior knowledge of the channel parameters.

A useful and rather general model of channels with memory, and the one we use, is the discrete finite-state channel (DFSC) [23]. A DFSC is described by the 4-tuple $(\mathcal{B}, \mathcal{X}, q_{Y_1,\beta_1|X_1,\beta_0}, \mathcal{Y})$, where $\mathcal{B}$ is a finite set of possible states of the channel, $\mathcal{X}$ is a finite set of possible channel inputs, $\mathcal{Y}$ is a finite set of possible channel outputs, and $q_{Y_1,\beta_1|X_1,\beta_0}$ is the conditional probability mass function (pmf) describing the statistical relationship between channel inputs and outputs as well as the state evolution. Given the state of the channel, the channel acts as a memoryless channel, and the next state of the channel depends on the previous state as well as the most recent input and output. To be more precise, we make the following definition:

Definition 3.1.1 Let $\{X_k\}_{k=1}^{\infty}$, $\{Y_k\}_{k=1}^{\infty}$ and $\{\beta_k\}_{k=0}^{\infty}$ random processes. The processes $\{Y_k\}_{k=1}^{\infty}$ and $\{\beta_k\}_{k=0}^{\infty}$ are the *output process* and *state process*, respectively, resulting from *passing* $\{X_k\}_{k=1}^{n}$ *through a DFSC* $(\mathcal{B}, \mathcal{X}, q_{Y_1,\beta_1|X_1,\beta_0}, \mathcal{Y})$ (or just $q_{Y_1,\beta_1|X_1,\beta_0}$) *without feedback* if for all $n > 0$,

$$p_{Y^n,\beta^n|X^n,\beta_0}(y^n, \tilde{\beta}^n|x^n, \tilde{\beta}_0) = \prod_{k=1}^{n} q_{Y_1,\beta_1|X_1,\beta_0}(y_k, \tilde{\beta}_k|x_k, \tilde{\beta}_{k-1}) \tag{3.1}$$

for all $x^n \in \mathcal{X}^n$, all $y^n \in \mathcal{Y}^n$, and all $\tilde{\beta}_0^n \in \mathcal{B}^{n+1}$.

(Note that only the distribution of the output process is necessary to describe the channel; the state process is defined to facilitate analysis.) A DFSC with feedback (DFSC$_\mathrm{f}$) consists of a DFSC as the forward channel (or just "channel") and a feedback channel, which is assumed noiseless and free of delay—i.e., at time k, the transmitter knows with certainty the value of Y^{k-1}. A DFSC$_\mathrm{f}$ is defined analogously to the DMC$_\mathrm{f}$:

Definition 3.1.2 Let M be a random variable taking values in $\mathcal{M}$. For all $m \in \mathcal{M}$, let $\{f_{m,i}\}_{i=1}^{\infty}$ be a sequence of functions, where $f_{m,i} : \mathcal{Y}^{i-1} \to \mathcal{X}$ maps a sequence of $i-1$ channel outputs to a single channel input, and $f_{m,1}$ takes a constant value in $\mathcal{X}$. The processes $\{Y_k\}_{k=1}^{\infty}$ and $\{\beta_k\}_{k=0}^{\infty}$ are the *output process* and *state process*, respectively, resulting from *passing* M *through a DFSC*$_f$ $(\mathcal{B}, \mathcal{X}, q_{Y_1,\beta_1|X_1,\beta_0}, \mathcal{Y})$ (or just $q_{Y_1,\beta_1|X_1,\beta_0}$) *via* $\{\{f_{m,i}\}_{i=1}^{\infty}\}_m$ if for all $n > 0$,

$$p_{Y^n,\beta^n|M,\beta_0}(y^n, \tilde{\beta}^n|m, \tilde{\beta}_0) = \prod_{k=1}^{n} q_{Y_1,\beta_1|X_1,\beta_0}(y_k, \tilde{\beta}_k|f_{m,k}(y^{k-1}), \tilde{\beta}_{k-1}) \tag{3.2}$$

for all $m \in \mathcal{M}$, all $y^n \in \mathcal{Y}^n$, and all $\tilde{\beta}_0^n \in \mathcal{B}^{n+1}$.

We further restrict our attention to indecomposable DFSC$_\mathrm{f}$'s, which have the property that the effect of the channel's initial state β_0 dies out over time (see, for example, [23] for further details). As a result, the coding scheme we develop and the analysis of its asymptotic properties requires no knowledge of the the initial state β_0.

An expression for the capacity C_{nf} of indecomposable DFSC's without feedback was given by Gallager in [23] as

$$C_{\mathrm{nf}} = \lim_{n\to\infty} \max_{p_{X^n}} \frac{1}{n} I(X^n; Y^n), \tag{3.3}$$

where $\{Y_k\}$ is the output process resulting from passing $\{X_k\}$ through the DFSC, starting from any channel state. In most cases, it is unknown how to even compute (3.3), although Mushkin and Bar-David [40] derived algorithms for computing the capacity of a simple subclass of DFSC's with two states known as Gilbert-Elliott channels. Goldsmith and Varaiya [26] studied a somewhat broader subclass of DFSC's and developed a capacity formula for this subclass that is easier to compute than (3.3); they also identified a subclass of DFSC's for which capacity is achieved with an i.i.d. input distribution and gave a relatively simple capacity formula for this subclass.

While feedback can be used to increase the capacity of a DFSC in at least some —but not all[1]—cases, little appears to be known about the availability and extent of such an increase.

We focus here on using feedback to reduce computational complexity rather than to increase capacity. We describe a variable-length coding scheme that achieves error probabilities that decay exponentially with the number of bits N in a block, at any fixed rate below the mutual information induced by an input distribution in a class of finite-memory ergodic Markov processes. When the feedback-free capacity C_{nf} is achieved by a stationary and ergodic input distribution, then this implies that the scheme can approach rates near C_{nf}. By exploiting a computationally efficient method for Shannon-Fano source coding, the scheme's encoding and decoding computational complexity is made linear in the number of channel inputs used.

We now embark on a detailed development of the coding scheme. This coding scheme is most easily described by building on the results of Chapter 2. For this reason, we begin in Section 3.2 by discussing how those results can be generalized, showing that to achieve certain performance characteristics, it is sufficient that the subsystems have certain properties. We then follow in Section 3.3 by describing subsystems having these properties. We argue in Section 3.4 that the computational complexity for the scheme is linear (and in some cases uniformly linear) and end in Section 3.5 with a discussion of certain remaining issues and future directions.

3.2 Generalization of the Coding Scheme

In this section, we generalize the results we obtained for DMC$_{\mathrm{f}}$'s in Chapter 2 to DFSC$_{\mathrm{f}}$'s. In particular, for coding schemes that use precoding, source coding, termination coding, and synchronization subsystems in the same way they are used in Sections 2.3 and 2.5, we develop an analog of Theorem 2.4.1 that gives the reliability of such schemes in terms of certain key properties of these subsystems. As a result, the problem of developing a coding scheme for DFSC$_{\mathrm{f}}$'s that attains this reliability function is reduced to the problem of developing subsystems with certain properties.

We begin by appropriately modifying the formulation of variable-length coding. For DFSC$_{\mathrm{f}}$'s, a variable-length code is just as described in Section 2.2, containing the same four elements. To describe the code's rate and probability of error, let $\tilde{\beta}_0$ be a known starting state of the channel, let W^N be defined as in Section 2.2, and let $\{Y_k\}$ and $\{\beta_k\}_{k=0}^{\infty}$ be the output and state processes, respectively, resulting from passing W^N through the DFSC$_{\mathrm{f}}$ $q_{Y_1,\beta_1|X_1,\beta_0}$ via $\{\{\tilde{\varepsilon}_i(m,\cdot)\}_{i=1}^{\infty}\}_{m\in\{0,1\}^N}$ with starting state $\beta_0 = \tilde{\beta}_0$. With $L^*(\tilde{\beta}_0)$ and $\hat{W}^N(\tilde{\beta}_0)$ defined analogously to (2.5) and (2.6), respectively, the code's rate is defined to be $N/(\sup_{\tilde{\beta}_0\in\mathcal{B}} E[L^*(\tilde{\beta}_0)])$, and the code's probability of error is defined to be $\sup_{\tilde{\beta}_0\in\mathcal{B}} \Pr\{W^N \neq \hat{W}^N(\tilde{\beta}_0)$. A coding scheme and its properties are then defined in the same way as in Section 2.2. This formulation is appropriate for the case in which the transmitter and receiver do not know the starting state of the channel.

Proceeding with the development of the analog to Theorem 2.4.1, we first observe that the proof of Theorem 2.4.1 follows entirely from Subsystem Properties 1-4. That is, if we use another set of subsystems $\{\pi_n\}$, $\{\sigma_n\}$, c^{term}, and $\{(\delta_{t_n}, \emptyset^{t_n}[n])\}_n$ within the framework of Section 2.3, then Theorem 2.4.1 still holds as long as Subsystem Properties 1-4 are satisfied.

Suppose that c_{FS} is a coding scheme mapping N and ν to a code that is the same as the code

[1] For example, Alajaji [4] showed that if the channel adds (modulo-q) an independent noise process with memory, then feedback does not increase channel capacity.

described in Section 2.3 with the modifications from Section 2.5 except for the fact that it substitutes different subsystems for the corresponding ones in Sections 2.3 and 2.5. Suppose that these different subsystems are such that Subsystem Properties 1-4 hold, from all starting states of the DFSC$_\mathrm{f}$ of interest, with some constant H_π taking the place of $H(X)$ in Subsystem Property 1, some constant H_σ taking the place of $H(X|Y)$ in Subsystem Property 2, and some constant E_0 taking the place of F_{cw} in Subsystem Property 3.

Then the following theorem regarding the error exponent associated with c_FS holds instead of Theorem 2.4.1:

Theorem 3.2.1 The coding scheme c_FS attains the error exponent function $E_\mathrm{FS} : (0, H_\pi - H_\sigma) \to \mathbb{R}$ defined by

$$E_\mathrm{FS}(r) = \left(1 - \frac{r}{H_\pi - H_\sigma}\right) E_0, \tag{3.4}$$

Proof: Follows in the same way as the proof of Theorem 2.4.1. □

Note that Subsystem Property 3 may be satisfied with multiple values of the constant E_0. The largest value leads to the strongest statement of the theorem. As discussed in Section 2.4, the constant E_0 may be chosen arbitrarily large in some cases, causing (3.4) to degenerate and and become somewhat meaningless.

From the theorem above, we see that the achievable rate and reliability of the overall scheme hinge on what values of H_σ, H_π, and E_0 emerge from a particular precoder, source coder and termination coding scheme design. In Chapter 2, capacity-achieving precoder and source coder designs are given for DMC$_\mathrm{f}$'s, in which $H_\pi = H(X)$ and $H_\sigma = H(X|Y)$, where X is distributed according to the capacity-achieving input distribution, and Y is the random variable resulting from passing X through the DMC. Subsystems appropriate for DFSC$_\mathrm{f}$'s are given in what follows.

3.3 A Coding Scheme for DFSC$_\mathrm{f}$'s

Before giving a precise description of the various subsystems of which the coding scheme is composed, we must consider what sort of channel input process we should use.

In the development of our coding scheme for the DMC in Chapter 2, the achievable rate of the scheme is highly dependent on the choice of the channel input process. For DMC$_\mathrm{f}$'s, the optimal input process is within the class of i.i.d. processes, so a natural choice for the input process is the optimal one. Fortunately, such a process can be generated (approximately) with low complexity by the precoder described in Section 2.3.1.1.

For DFSC$_\mathrm{f}$'s, the choice of input process is not as clear, because (3.3) does not suggest an easily described ideal input process. On one hand, we would like the class of allowable input processes to be as large as possible. On the other hand, the input process must be sufficiently structured that the required precoder, which must generate the input process, has low complexity.

We can satisfy both of these desires by constraining the output process feeding the channel to be a Kth-order Markov process, where K is finite but can be arbitrarily large. We adopt the usual definition of such processes [14]:

Definition 3.3.1 A process $\{X_k\}_{k=1}^n$ is a *Kth-order Markov process with transition pmf* $q_{X_1|X_{1-K}^0}$

and initial state distribution q_{α_0} if

$$p_{X^n}(x^n) = \sum_{x_{1-K}^0} q_{\alpha_0}(x_{1-K}^0) \prod_{k=1}^{n} q_{X_1|X_{1-K}^0}(x_k|x_{k-K}^{k-1}). \tag{3.5}$$

Clearly, $p_{X_j|X_{j-K}^{j-1}} = q_{X_1|X_{1-K}^0}$ for all $j \geq K+1$. The *state* of the process at time k is defined as X_{k-K+1}^k. With $\alpha_k = X_{k-K+1}^k$, we note that $\{\alpha_k\}$ can be considered as a first-order Markov process or Markov chain.

The finite-order Markov distributions constitute a rich and flexible class of channel input distributions into which data can be precoded with low computational complexity. Furthermore, we conjecture that these processes are sufficient to make the mutual information between channel input and output arbitrarily close to the feedback-free capacity of the channel, but are aware of no existing proofs of this conjecture. As seen in the previous chapter, analysis of a scheme using the compressed-error-cancellation framework often requires that we work with the joint distribution for the channel output and input after passing the input through the channel without feedback. From Definition 3.1.1, we see that specifying the joint distribution for (X^n, β_0) determines the joint distribution of the channel input and channel output (X^n, Y^n). In this chapter, the input processes of interest take the following form: X^n is Markov according to $q_{X_1|X_{1-K}^0}$ with an initial state distribution q_{α_0}, and the joint distribution for (X^n, β_0) has the form

$$p_{X^n,\beta_0}(x^n, \beta_0) = \sum_{x_{1-K}^0} p_{\beta_0|\alpha_0}(\beta_0|x_{1-K}^0) q_{\alpha_0}(x_{1-K}^0) \prod_{j} q_{X_1|X_{1-K}^0}(x_j|x_{j-K}^{j-1}). \tag{3.6}$$

This form of the joint distribution models the case in which the starting state of the channel is dependent on the input to the channel only by being dependent on the initial state of the Markov process. Actually, in all of the analyses in this chapter, β_0 and α_0 are independent, but this definition allows some extra flexibility. It is convenient to make the following special definition:

Definition 3.3.2 Let $\{X_k\}_{k=1}^n$ be a Kth-order Markov process with transition pmf $q_{X_1|X_{1-K}^0}$. Then the processes $\{Y_k\}_{k=1}^n$ and $\{\beta_k\}_{k=0}^n$ are the *output process* and *state process*, respectively, resulting from *passing the Markov process* $\{X_k\}_{k=1}^n$ *through a DFSC* $q_{Y_1,\beta_1|X_1,\beta_0}$ *without feedback with initial states jointly distributed according to* q_{α_0,β_0} if

$$\begin{aligned} &p_{X^n,Y^n,\beta^n,\alpha_0,\beta_0}(x^n, y^n, \tilde{\beta}^n, \tilde{\alpha}_0, \tilde{\beta}_0) \\ &\qquad = q_{\alpha_0,\beta_0}(\tilde{\alpha}_0, \tilde{\beta}_0) \prod_{j=1}^{n} q_{X_1|X_{1-K}^0}(x_j|\tilde{\alpha}_{j-1}) \prod_{k=1}^{n} q_{Y_1,\beta_1|X_1,\beta_0}(y_k, \tilde{\beta}_k|x_k, \tilde{\beta}_{k-1}), \\ &\forall x^n \in \mathcal{X}^n, \forall y^n \in \mathcal{Y}^n, \forall \tilde{\beta}_0^n \in \mathcal{B}^{n+1}, \forall \tilde{\alpha}_0 \in \mathcal{X}^K, \end{aligned} \tag{3.7}$$

where $\tilde{\alpha}_j = x_{j-K+1}^j$, for $j = 1, \cdots, n$.

One of the benefits of using a Kth-order Markov input distribution is that it allows the joint channel input and output processes to be viewed as resulting from a Markov chain. Suppose that $\{X_k\}_{k=1}^\infty$ is a Kth-order Markov process with some initial state distribution, that $\{Y_k\}_{k=1}^\infty$ and $\{\beta_k\}_{k=0}^\infty$ are the output and state processes resulting from passing the Markov process $\{X_k\}$ through a DFSC without feedback with some joint initial state distribution. We can first view a Kth-order Markov process in terms of appropriately defined deterministic functions f and $\hat{f}$ as

$$\alpha_k = f(\alpha_{k-1}, V_k) \tag{3.8}$$

$$X_k = \hat{f}(\alpha_{k-1}, V_k), \tag{3.9}$$

where $\{V_k\}$ is a sequence of random variables that are i.i.d. uniformly over $[0, 1)$ and independent of α_0 and β_0. Next, we can view the finite-state channel in terms of appropriately defined deterministic functions g and $\hat{g}$ as

$$\beta_k = g(\beta_{k-1}, X_k, Z_k) \tag{3.10}$$

$$Y_k = \hat{g}(\beta_{k-1}, X_k, Z_k), \tag{3.11}$$

where $\{Z_k\}$ are i.i.d. uniformly over $[0, 1)$ and independent of $\{V_k\}$ and α_0 and β_0. We now see that if $\{X_k\}$ is passed through a DFSC, then we can view the joint state (α_k, β_k) as a stationary Markov process that evolves according to

$$(\alpha_k, \beta_k) = h(\alpha_{k-1}, \beta_{k-1}, V_k, Z_k), \tag{3.12}$$

$$(Y_k, X_k) = \hat{h}(\alpha_{k-1}, \beta_{k-1}, V_k, Z_k) \tag{3.13}$$

where h and $\hat{h}$ are deterministic functions. It can be shown that h and $\hat{h}$ can be defined so that $X^n, Y^n, \beta^n, \alpha_0, \beta_0$ have the distribution given in (3.7). Therefore, (α_k, β_k) is the state of a Markov chain, which we refer to as the *joint input/channel state.* This new description allows us to see certain facts more easily. For example, it is now clear that (X_k^n, Y_k^n) is independent of (X^{k-1}, Y^{k-1}) conditioned on knowledge of $(\alpha_{k-1}, \beta_{k-1})$.

As a final restriction on the channel input processes, we require that the Markov process corresponding to $\{(\alpha_k, \beta_k)\}$ contain only transient states and a single ergodic class[2]. In Appendix B.1, it is shown that all input processes for which the state process $\{\alpha_k\}$ is an ergodic plus transient Markov chain are admissible under this restriction. We restrict our attention to such input processes, the set of which we denote by $\mathcal{E}$. One convenient feature of Markov chains with transient states and a single ergodic class is the existence of a stationary distribution over the states of the chain [25]. If the distribution of (α_0, β_0) is equal to this stationary distribution, then the process $\{(X_k, Y_k)\}_{k=1}^{\infty}$ is said to be stationary.

When we choose an input distribution $q_{X_1|X_{1-K}^0}$ from $\mathcal{E}$ for use with the DFSC$_\text{f}$ $(\mathcal{B}, \mathcal{X}, q_{Y_1,\beta_1|X_1,\beta_0}, \mathcal{Y})$, the coding scheme we develop is able to achieve the rate

$$\lim_{n\to\infty} I(\bar{X}^n; \bar{Y}^n)/n = H_\infty(\bar{\mathcal{X}}) - H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}), \tag{3.14}$$

where $\bar{\mathcal{X}} = \{\bar{X}_k\}_{k=1}^{\infty}$ is a Kth-order Markov process with the stationary initial state distribution and transition pmf $q_{X_1|X_{1-K}^0}$, and $\bar{\mathcal{Y}} = \{\bar{Y}_k\}_{k=1}^{\infty}$ is the output process resulting from passing the Markov process $\bar{\mathcal{X}}$ through the channel $q_{Y_1,\beta_1|X_1,\beta_0}$ without feedback with the stationary initial joint input/channel state distribution for (α_0, β_0). In addition, in (3.14), $(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) = \{(\bar{X}_k, \bar{Y}_k)\}_{k=1}^{\infty}$ is a stationary process; also, $H_\infty(\bar{\mathcal{X}})$, $H_\infty(\bar{\mathcal{Y}})$, and $H_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}})$ denote the entropy rates for $\bar{\mathcal{X}}$, $\bar{\mathcal{Y}}$, and

[2] We refer to such processes as ergodic plus transient Markov processes or Markov chains.

the joint process $(\mathcal{X}, \mathcal{Y})$, respectively.[3] As additional notation, $H_\infty(\mathcal{X}|\mathcal{Y}) \triangleq H_\infty(\mathcal{X}, \mathcal{Y}) - H_\infty(\mathcal{Y})$.

To construct our coding scheme, we design subsystems that satisfy Subsystem Properties 1–4 with $H_\pi = H_\infty(\mathcal{X})$ and $H_\sigma = H_\infty(\mathcal{X}|\mathcal{Y})$, for all starting states of the channel. When we do, Theorem 3.2.1 establishes that the scheme using these subsystems achieves a rate equal to the right-hand side of (3.14).

3.3.1 Precoder

We now define a precoder $\pi_{\text{FS},n} : \{0,1\}^n \to \mathcal{X}^\dagger$ to be used in place of the function π_n in Chapter 2. This precoder also relies on the principles described in Section 2.3.1.1, but has a different target output distribution. Because the output of the precoder $\pi_{\text{FS},n} : \{0,1\}^n \to \mathcal{X}^\dagger$ should be a process distributed (approximately) according to $q_{X_1|X_{1-K}^0}$, we must define a different cdf $F_{\text{FS},\tilde{X}}$ whose inverse is used to transform the uniformly distributed variable $0_2.D^n + 2^{-n}Z$, where Z is as in Section 2.3.1.1. To define $F_{\text{FS},\tilde{X}}$, assume without loss of generality that $\mathcal{X} = \{0, 1, \cdots, M-1\}$, and let the base of all expansions in this section taken to be M. Choose $\bar{\alpha}$ to be any ergodic state, and let $\mathcal{X} = \{X_k\}_{k=1}^\infty$ be a Kth-order Markov process with transition pmf $q_{X_1|X_{1-K}^0}$ that starts in the state $\bar{\alpha}$. Mapping these sequences bijectively to the unit interval $[0,1)$ (i.e., discarding the sequences that end in an infinite sequence of $(M-1)$'s), we let $\tilde{X}$ be a random variable defined by $\tilde{X} = 0_M.X_1X_2\cdots$, and we let $F_{\text{FS},\tilde{X}}$ be the cdf for $\tilde{X}$.

The precoder $\pi_{\text{FS},n}$ is then defined by

$$T_{\text{FS},n}(u) = u^{[l_{\text{FS},n}(u)]} \tag{3.16}$$

$$\pi_{\text{FS},n}(d^n) = T_n(F_{\text{FS},\tilde{X}}^{-1}(0_2.d^n + 2^{-n}Z)), \tag{3.17}$$

where the expansion in (3.16) is M-ary, and $l_{\text{FS},n} : [0,1) \to \mathbb{N}$ is defined as follows to ensure that the output of the precoder stops in state $\bar{\alpha}$ after enough digits of the M-ary expansion of $F_{\text{FS},\tilde{X}}^{-1}(s)$ have been put out to uniquely determine the first n bits of s:

$$l_{\text{FS},n}(u) = \min\{k \ : \ F_{\tilde{X}}([0_M.u^{[k]}, 0_M.u^{[k]} + M^{-k})) \subseteq [i2^{-n}, (i+1)2^{-n}) \text{ and } u_{[k-K+1]}^{[k]} = \bar{\alpha}\} \tag{3.18}$$

$$\text{for } u \in F_{\tilde{X}}^{-1}([i2^{-n}, (i+1)2^{-n})) \text{ for } i = 0, \cdots, 2^n - 1. \tag{3.19}$$

This definition of $\pi_{\text{FS},n}$ gives rise to the analogs of the key precoder properties held by π_n from Section 2.3.1.1:

Losslessness For any $d^n\{0,1\}^n$, d^n can be recovered from n and $\pi_{\text{FS},n}(d^n)$.

Approximation If D^n is uniformly distributed over $\{0,1\}^n$, then the elements of $\pi_{\text{FS},n}(D^n)$ form a random process that can be generated by taking a Kth-order Markov process with transition pmf $q_{X_1|X_{1-K}^0}$ and truncating it according to a stopping rule.

[3] The entropy rate $H_\infty(\{P_i\}_{i=1}^\infty)$ of a process $\{P_i\}_{i=1}^\infty$ is defined as [14]

$$H_\infty(\{P_i\}_{i=1}^\infty) = \lim_{n \to \infty} \frac{H(P^n)}{n}. \tag{3.15}$$

Efficiency If D^n is uniformly distributed over $\{0,1\}^n$, then

$$E[\ell(\pi_{\text{FS},n}(S))] < (n + C_\pi)/H_\infty(\bar{\mathcal{X}}),$$

where C_π is a constant that is independent of n.

The losslessness and approximation pıoperties are evident from the precoder definition, and a proof of the efficiency property is provided in Appendix B.2. The efficiency property implies that Subsystem Property 1 is satisfied, while the approximation property is needed to show that the precoder with the source coder in the following section together satisfy Subsystem Property 2.

3.3.2 Source Coder

We define a source coder $\sigma_{\text{FS},n} : \mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}^\dagger$ to be used in place of the function σ_n in Chapter 2.

Like the source coder σ_n from Chapter 2, $\sigma_{\text{FS},n}$ uses a Shannon-Fano strategy but must now use a different statistical model for the dependence between the channel input and output. Let $\hat{\mathcal{X}} = \{\hat{X}_i\}$ denote a Markov process with transition pmf $q_{X_1|X_{1-K}^0}$ and uniform initial state pmf, and let $\hat{\mathcal{Y}} = \{\hat{Y}_i\}_{i=1}^\infty$ and $\{\hat{\beta}_i\}_{i=0}^\infty$ denote the output and state processes, respectively, resulting from passing the Markov process $\hat{\mathcal{X}}$ through the channel $q_{Y_1,\beta_1|X_1,\beta_0}$ without feedback with uniform initial joint channel/input state distribution. Then, with $\tilde{X} = 0_M.\hat{X}^n$, and with $F_{\tilde{X}|\hat{Y}^n}$ denoting the cdf for $\tilde{X}$ conditioned on $\hat{Y}^n$, $\sigma_{\text{FS},n}$ is defined by

$$\sigma_{\text{FS},n}(x^n, y^n) = u^{[l]}, \tag{3.20}$$

where the expansion in (3.20) is binary, and

$$u = F_{\tilde{X}|\hat{Y}^n}(0_M.x^n|y^n) + w_{X^n|Y^n}(x^n|y^n)/2 \tag{3.21}$$

$$l = \lceil -\log w_{X^n|Y^n}(x^n|y^n) \rceil + 1, \tag{3.22}$$

where

$$w_{X^n|Y^n}(x^n|y^n) = p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n), \text{ for all } x^n \in \mathcal{X}^n, y^n \in \mathcal{Y}^n. \tag{3.23}$$

While the statistical model used in the source coder is again somewhat inaccurate as it was in Chapter 2, it is sufficiently accurate to produce the following key bound, which is shown in Appendix B.3:

$$E[L_{i+1}^\sigma] \leq E[L_i]\, H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) + 5\log E[L_i] + C_\sigma, \tag{3.24}$$

where C_σ is a constant. Comparing to (2.40), we see that this source coder design satisfies Subsystem Property 2, with $H_\sigma = H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}})$.

While it seems initially that evaluating $F_{\tilde{X}|\hat{Y}^n}(0_M.x^n|y^n)$ is computationally expensive without the convenient properties of DMC's and i.i.d. channel inputs, we can use actually use principles from the Bahl algorithm [5] to perform this evaluation using a number of computations that is linear in n. An algorithm for performing the evaluation is presented in Appendix B.4. This algorithm leads, in turn, to a linear complexity algorithm for source coding and decoding.

3.3.3 Synchronization Sequence and Detector

Similar to the development of the replacement precoders and source coders, a replacement synchronization subsystem for the code with parameter N, consists of the detector

$$\delta_{\mathrm{FS},t_N}(y^{t_N}, \emptyset^{t_N}[N]) = \begin{cases} 1 & \text{if } p_{\hat{Y}^{t_N}|\hat{X}^{t_N}}(y^{t_N}|\emptyset^{t_N}[N]) > 2^{\omega(t_N)} p_{\hat{Y}^{t_N}}(y^{t_N}) \\ 0 & \text{otherwise,} \end{cases} \tag{3.25}$$

where $\{\hat{X}_i\}$ and $\{\hat{Y}_i\}$ are defined as in Section 3.3.2, and $\omega(t_N)$ is any function that is $o_{t_N}(t_N)$ but bounded below by $3 \log t_N + o_{t_N}(1)$. For concreteness, let us set $\omega(t_N) = \sqrt{t_N}$. As the form of (3.25) suggests, $\delta_{\mathrm{FS},t_N}(y^{t_N}) = 1$ signals that $\emptyset^{t_N}[N]$ has been detected.

With t_N set, as in Section 2.5, to $\sqrt{N}$, we show in Appendix B.5 that there is a choice of $\emptyset^{t_N}[N]$ such that the probability of both false alarm and missed detection satisfy (2.56) and (2.57) in Subsystem Property 4. The purpose of the term $2^{\omega(t_N)}$ in (3.25) is to make this proof easier than it might otherwise be.

3.3.4 Termination Coder—Modified Schalkwijk-Barron Coder

The termination coding scheme remains an mSB coding scheme, with the two parameters κ and ν taking the same meaning as in Section 2.3.1.3. The difference is that the inner codebook and verification sequences c^ν and w^ν are chosen appropriately for the DFSC$_\mathrm{f}$. The coding theorem for DFSC's [23] guarantees the existence of an inner codebook with probability of error decaying exponentially in κ. The same theorem also implies that the verification sequences can be chosen so that the probability P_{cw} of confusing w^ν for c^ν and the probability P_{wc} of confusing c^ν for w^ν decay exponentially in ν, although, unlike the mSB coder for the DMC$_\mathrm{f}$, it is unclear what choices of c^ν and w^ν give the minimum value of P_{cw}. Random coding arguments at least guarantee the possibility of choosing the inner codebook and verification sequences so that the error decays exponentially with ν with exponent $E_{0,\mathrm{r}}$, the zero-rate random coding exponent.

To show that Subsystem Property 3 holds, consider the sequence of mSB codes corresponding to the sequence of parameters $\kappa_N = N^{1/4}$ with $\nu_N = \lceil \alpha N \rceil$. Then, using the same arguments as in Section A.3, the expected output length is easily seen to be $\alpha N + o_N(N)$. Subsystem Property 3 is therefore met with E_0 equal to $E_{0,\mathrm{r}}$.

3.4 Complexity

Because these subsystems satisfy the Subsystem Properties 1-4, Theorem 3.2.1 holds for the overall compressed-error cancellation scheme that employs these subsystems. To complete the performance characterization, we show that the scheme has linear time and space complexity under the uniform cost criterion.

Because linear-complexity arithmetic coding algorithms exist for Markov sources [14], the pre- and postcoder, which operate in a similar way, also have time complexity that is linear in the number of precoder outputs under the uniform-cost criterion. This in turn, implies that the total arithmetic operations required by these subsystems are linear in L^*. Space cost is linear in L^* when space for buffering inputs and outputs is counted. Using a variation of the arithmetic coding algorithm in [63], we can also develop an implementation on a digital computer with fixed-length registerr that runs in linear time and space.

Using the linear-complexity method of Appendix B.4 for computing the relevant source coding cdf, the source coders and decoders also require a total number of computations that is linear in L^* under the uniform-cost criterion. When analyzing the running time of the algorithms on a digital computer, however, we have to be careful. Certain quantities that must be computed in the source coding algorithm need not be done with arbitrary precision, but the final answers that are computed should be close to the true answers. But under floating-point arithmetic with p digits allowed after the decimal point, with no limit on the size of the exponent, empirical studies indicate that only $O_p(2^p)$ computations can be performed with negligible loss in accuracy. A source coder based on the equations in Appendix B.4 requires that $O_n(n)$ computations be performed. To obtain acceptable accuracy, then, it seems that we must allow $O_n(\log n)$ digits after the decimal point for a length-n input. Since additions or multiplications of n-bit numbers takes time proportional to n, it appears that the actual time used on a computer may grow as $O_n(n \log n)$.

The synchronization subsystem and mSB coding subsystem can both be shown to have total time and space costs that are $O_{L^*}(L^*)$ under the uniform-cost criterion, because a recursion can easily be developed so that the hypothesis tests required for decoding take a number of computations that is linear in the length of the sequences being tested. On a digital computer, $O_N(t_N)$ computations are required for computing the relevant likelihoods each time δ_{FS,t_N} is used. Because these likelihoods may be in error by a multiplicative factor growing exponentially in t_N, $O_N(t_N)$ time seems to be sufficient for each use of δ_{FS,t_N}.

By combining these facts with the reasoning given in Section 2.6, we may conclude that the coding scheme described herein has $O_{L^*}(L^*)$ time and space complexity.

While the time and space complexities are linear in the number of channel inputs for a fixed Kth-order Markov input distribution, the number of computations needed per channel input increases with K. Consequently, in the case in which arbitrarily large values of K are required to approach the feedback-free channel capacity arbitrarily closely, the scheme cannot be said to have uniform linear complexity in the sense described in Section 2.6.2. Note, however, that there do exist channels for which the feedback-free capacity is achieved with a finite-order Markov input distribution; examples include the Gilbert-Elliott channels and the class of channels Goldsmith and Varaiya [26] identified as having i.i.d. capacity-achieving input distributions, which includes additive white noise channels with symmetric PSK inputs and time-varying noise statistics or amplitude fading [26].

3.5 Discussion

3.5.1 Choosing the Synchronization and mSB Sequences

In contrast to the coding scheme we developed for DMC$_f$'s, we have not given an explicit construction for the synchronization sequences $\{\emptyset^{t_N}[N]\}_N$ or for the various sequences used by the mSB coder. The reason is that it is not evident how to construct such sequences. It is well known, however, that random selection of sequences is adequate with high probability. That is, Markov's inequality dictates that a sequence yields an error probability less than K times the average with probability at least $1 - 1/K$. Our view is that practically, random selection should suffice. While random selection of codewords is traditionally objectionable because the resulting codebook has so little structure that decoding and encoding complexity is exponential in the codeword length, this

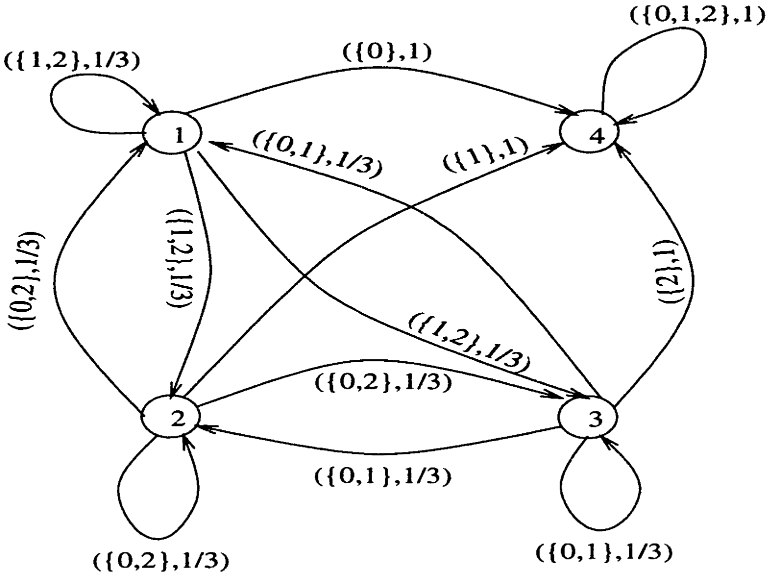


Figure 3-1: Diagram of a finite-state channel with feedback capacity of $1 - H(\epsilon)$ bits per channel input but with feedback-free capacity zero. The label $(\{0,1\}, 1/3)$ on the arc from state 3 to state 2 means that if 0 or 1 is put in to the channel in state 3, then with probability 1/3, a transition to state 2 occurs. The other labels have analogous meanings.

objection is not relevant in this case. An alternative to random selection of a single sequence is pseudorandom generation of a new sequence each time the synchronization sequence is sent or each time the mSB encoder is used. The performance of, say, the synchronization subsystem, would then be equal to the performance, averaged over all sequences, and the bounds on average error probability would apply directly.

3.5.2 Beyond Feedback-Free Capacity

Unlike a DMC, a DFSC may sometimes have a feedback capacity that is greater than its feedback-free capacity. An example of a DFSC with such a property, which we note is not indecomposable, is given in Figure 3-1. The channel in this figure has input alphabet $\mathcal{X} = \{0, 1, 2\}$, state space $\mathcal{B} = \{1, 2, 3, 4\}$, and output alphabet $\mathcal{Y} = \mathcal{X} \times \mathcal{B}$. The probability law associated with the channel is defined as follows:

$$\text{for all } (s, x_0, x_1, x^*) \in \{(1,1,2,0), (2,0,2,1), (3,0,1,2)\},$$

$$q_{Y,S_1|X,S_0}((x_1, j), j | x_0, s) = \epsilon/3, \quad j = 1,2,3, \tag{3.26a}$$

$$q_{Y,S_1|X,S_0}((x_0, j), j | x_1, s) = \epsilon/3, \quad j = 1,2,3, \tag{3.26b}$$

$$q_{Y,S_1|X,S_0}((x_0, j), j | x_0, 1) = (1-\epsilon)/3, \quad j = 1,2,3, \tag{3.26c}$$

$$q_{Y,S_1|X,S_0}((x_1, j), j | x_1, 1) = (1-\epsilon)/3, \quad j = 1,2,3, \tag{3.26d}$$

$$q_{Y,S_1|X,S_0}((x^*,4),4|x^*,1) = 1 \tag{3.26e}$$

The meaning of this probability law is as follows: State 4 is an absorbing state that has zero capacity when viewed as a DMC; the transmitter must avoid using the symbols 0, 1, and 2 from states 1, 2, and 3, respectively to avoid entering state 4. In state 1, an input of 1 yields output $(1,\cdot)$ with probability $1-\epsilon$ and $(2,\cdot)$ with probability ϵ; an input of 2 yields output $(2,\cdot)$ with probability $1-\epsilon$ and $(1,\cdot)$ with probability ϵ. In other words, in state 1, if only inputs 1 and 2 are used, then the channel is like a BSC with crossover probability ϵ and moves to states 1, 2, or 3 with equal probability. In state 2, the inputs 0 and 2 behave similarly, and in state 3, the inputs 0 and 1 also behave similarly.

The channel output symbol contains information about the next state. With feedback, the transmitter can know the current state of the channel and therefore which symbols it may use and which symbol it must avoid, allowing $1-H(\epsilon)$ bits per channel input to be transmitted. Without feedback, the transmitter has no way of knowing which symbols it may use and must avoid all the input symbols, resulting in zero capacity.

For this channel, we can still use the principles of the compressed-error-cancellation framework but must modify the precoder so that its input distribution adjusts according to the feedback. Assume that the initial state is known to be state 1. The source symbols input to the precoder are a sequence of 0's and 1's. Then the precoder performs the following mapping, which depends on the current state of the channel: in state 1, the source symbol 0 is mapped to channel input symbol 1 and the source symbol 1 to the channel input symbol 2; in state 2, 0 is mapped to channel input symbol 0 and 1 to channel input symbol 2; and in state 3, 0 is mapped to channel input symbol 0 and 1 to channel input symbol 1.

The subsequent source coding step is straightforward. The locations of all "crossovers" (e.g. an input of 0 becoming an output $(2,\cdot)$ in state 2) could be coded using about $NH(\epsilon)$ bits, and the same process could be used to precode this second message. Continuing in this fashion, it appears that rates approaching $1-H(\epsilon)$, far greater than the feedback-free capacity, might be achievable.

It is unclear to what extent this example can be generalized. One obstacle toward complete generalization seems to be that in the general case, the precoder does not necessarily seem to be able to generate even its first channel input without knowledge of the input distribution of second channel input. But the proper input distribution for the second channel input may depend on the first channel input. We were able to circumvent this problem in the above example only because it was so contrived. This difficulty may lend some insight into why so little progress has been made on finding feedback capacity of DFSC's and feedback-capacity-achieving coding strategies. Furthermore, we do not believe the difficulty is unique to our framework. Any feedback-capacity-achieving coding scheme would seem to involve adaptive adjustment of the input distribution. Yet even mapping the message bits to an appropriately distributed channel input sequence seems to present a significant challenge.

3.5.3 Computational Delays

We first observe that the issue of computational delays discussed in Section 2.8.1 is important for this coding scheme because the feedback sequence y^n is needed in its entirety before the source coder can perform its function. Because of this requirement, the interleaving technique outlined in Section 2.8.1 is especially relevant for this coding scheme.

But for the same reasons that proofs for fixed-length source coders and fixed-length precoders are difficult, it appears difficult to argue rigorously that interleaving would not affect the total rate asymptotically. Because the central limit theorem holds for sums of random variables from a wide variety of processes [10], we conjecture that interleaving does not reduce the rate of the scheme asymptotically.

3.5.4 Fixed-Length Precoding and Source Coding

We observe that the techniques used in Appendix A.9 to prove that the fixed-length precoders and source coders have exponentially decaying probability of decoding error would not carry over for the analogously defined fixed-length precoders and source coders for DFSC_{f}'s.

If the precoder were made fixed-length, we believe that it would be possible to use results like the central limit theorem for α-mixing processes [10] to prove similar results for the precoder. Whether we could prove the analogous results for an appropriate fixed-length source coder is less clear.

3.5.5 Summary and Future Directions

We have developed a linear-complexity coding scheme giving exponentially decaying error probabilities at any rate below what is essentially the mutual information induced by passing a given finite-order Markov input distribution through an indecomposable DFSC.

Aside from some of the research problems discussed above, we would like to determine higher moments or exact pmfs of L_i and L_i^q. We would also like to prove that finite-order Markov input distributions are sufficient to achieve feedback-free capacity. Since the amount of memory in the Markov input process does affect complexity, it is of interest to determine the relationship between memory and achievable rate.

Chapter 4

Unknown Channels

4.1 Introduction

The statistical characterization of a channel is often not given. Rather, it must measured. In many cases, the measurement process results in an accurate model of the channel for all time. The results of the previous two chapters are useful for these cases. But if the channel's statistical characterization varies with time in an unknown way, then a measurement at a given point in time may not describe the channel at other times. As an example of a channel that varies with time in an unknown way, suppose that one has a wireless link from portable computer to a network node fixed in the ceiling. One might send a message from the computer to the node, move the computer, send another message, move it again, etc. Because each position of the computer results in a new set of physical paths over which the signal travels, the channel varies with time. Because each message is only affected by one position of the computer, each message is effectively transmitted over a different channel. The transmitter has to send data reliably over a number of possible channels. We say in this case that the channel is *unknown*, because it is one of a set of possible channels. The channel that actually prevails is called the *realized channel.*

When feedback is not available on an unknown channel, a predetermined, fixed-rate codebook must be used independent of the realized channel, and information learned by the receiver about the channel (either implicitly or explicitly) can only be used to optimize the decoding portion of the system. In such scenarios, in addition to the task of designing the single codebook that must be usable on every realizable channel, one must construct *universal decoders* that can reliably decode the transmitted message without explicit knowledge of the channel no matter what channel is realized. Examples of powerful universal decoders that work with unknown DMC's and with a subclass of unknown DFSC's are described in [15] and [66], respectively; these decoders yield the same error exponent as maximum-likelihood decoders, which must know the channel. More recently, the results of [66] were extended to arbitrary unknown DFSC's [38].

For unknown channels with feedback, it is possible to develop communication schemes that implicitly or explicitly learn the channel parameters and use this information to adapt the transmission rate to the channel quality, jointly optimizing both the encoding and decoding processes; these can be viewed as *universal communication* schemes.

Simple forms of universal communication are increasingly used in practice. Examples include current voiceband modems, which use a two-way protocol for determining an acceptable rate based on the quality of the channel. More generally, a variety of simple universal communication schemes involve the use of training data. In particular, channel measurements obtained during a preliminary training phase are subsequently used to specify the transmitter and receiver parameters for the system. However, on time-varying channels, the system must be continually retrained, which can substantially reduce throughput. More efficient universal communication schemes avoid the use of training data and effectively learn what they need about the channel from the received data symbols, i.e., in a blind manner.

In this chapter, we develop universal communication counterparts to universal decoding schemes of this type for unknown channels with feedback. For unknown DFSC$_f$'s (UFSC$_f$'s) (i.e, the realized channel is a DFSC$_f$), we show that a compressed-error-cancellation approach can achieve some attractive asymptotic characteristics. In particular, exploiting low-complexity Lempel-Ziv type universal source coding [66] in this framework allows us to achieve rates approaching the mutual information between input and output determined by the realized channel and a fixed input distribution. Moreover, the complexity of the scheme is of the same order as that of the corresponding scheme for known DFSC$_f$'s, and the error probabilities also decay exponentially with average decoding delay. Only the mutual information rather than the feedback-free channel capacity of the realized channel is achieved (although the two may be the same in some cases), because we specifically avoid having the input distribution to the channel vary as a function of feedback to simplify both implementation and analysis.

We begin the detailed development of our universal communication scheme by defining what constitutes a variable-length universal communication code and how its performance is measured in Section 4.2. We then move to a description and analysis of a universal coding scheme in Sections 4.3–4.5, and end with a discussion of remaining issues and future directions in Section 4.6.

4.2 Formulation of Variable-Length Coding for Unknown Channels

We define an *unknown discrete finite-state channel with feedback* (UFSC$_f$) to be a set of DFSC$_f$'s $\mathcal{Q}$, with given starting states, with a common input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$.

A variable-length code for an unknown channel is specified just as for the known-channel case as the 4-tuple $(N, \varepsilon, \{\chi_i\}, \Delta)$, where the elements of the tuple have the same meaning as in the known-channel case (see Sections 2.2 and 3.2). If we apply this code to a channel $q \in \mathcal{Q}$, we obtain a certain rate and probability of error as defined in Section 3.2. Thus, the rate and probability of error for the code are functions of q. We call these functions the rate function and error function associated with the code.

A coding scheme is exactly as defined in Sections 2.2 and 3.2, namely, a function of two parameters p_1 and p_2 whose output is a variable-length code.

Diverging somewhat from the formulation in Section 2.2, we find that for analysis of universal communication schemes, it is convenient to consider properties of sequences of codes rather than coding schemes. We say a sequence of codes $\{a_n\}_n$ attains a rate function $R : \mathcal{Q} \to [0, \infty)$ if for each $q \in \mathcal{Q}$, the corresponding rate sequence $\{R_n(q)\}_n$ and probability of error sequence $\{P_n(q)\}_n$ satisfies

$$\liminf_{n\to\infty} R_n(q) \geq R(q) \tag{4.1}$$

$$\limsup_{n\to\infty} P_n(q) = 0. \tag{4.2}$$

We say the code sequence attains an error exponent function $F : \mathcal{Q} \rightarrow \mathbb{R}$ if for each $q \in \mathcal{Q}$, the corresponding rate sequence $\{R_n(q)\}_n$ and probability of error sequence $\{P_n(q)\}_n$ satisfy

$$\liminf_{n\to\infty} -\frac{R_n(q) \log P_n(q)}{N_n} \geq F(q), \tag{4.3}$$

where N_n is the number of message bits in the code a_n.

Because the notion of uniform convergence to rate functions or error exponent functions leads to some complex issues, we defer a discussion of this notion until Section 4.5.

4.3 A Universal Communication Scheme for a Class of UFSC's

The coding scheme developed in Chapters 2 and 3 for channels whose parameters are known *a priori* can be extended rather naturally to UFSC$_f$'s. In developing such extensions, it is convenient to restrict our attention to UFSC$_f$'s $\mathcal{Q}$ whose constituent DFSC$_f$'s are indecomposable.

To begin our development, we first choose a Kth-order Markov input distribution $q_{X_1|X_{1-K}^0}$ from the class $\mathcal{E}$ (defined in Section 3.3) of admissible distributions, just as we did at the outset of Section 3.3. Given a particular choice of input distribution, then the coding scheme we describe achieves the rate function $I : \mathcal{Q} \rightarrow (0, \infty)$ defined by

$$I(q) \triangleq H_\infty(\bar{\mathcal{X}}) - H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}(q)), \tag{4.4}$$

where $\bar{\mathcal{X}} = \{\bar{X}_i\}_{i=1}^\infty$ is a stationary process distributed according to $q_{X_1|X_{1-K}^0}$, and $\bar{\mathcal{Y}}(q) = \{\bar{Y}_i(q)\}_{i=1}^\infty$ is the stationary output process that results from passing the Markov process $\bar{\mathcal{X}}$ through the DFSC q without feedback.

Note that in contrast to the behavior described above, a feedback-free communication system cannot transmit at a rate above the capacity of the compound channel corresponding to $\mathcal{Q}$ [23], which is less than or equal to the capacity of every channel in $\mathcal{Q}$. At the same time, it is important to stress that because we do not consider adapting the input distribution to the realized channel, the rates our strategy achieves are not, in general, the best possible for a feedback scheme. Indeed, in principle, at least the feedback-free capacity of the realized channel can be achieved by varying the input distribution using, for example, a training-sequence based method. Nevertheless, there are many practical situations in which the capacity-achieving input distribution is the same for all the possible realizable channels, in which case the fixed input distribution causes no harm.

To achieve the rate function I, it is sufficient that the various component subsystems are such that the following four properties are satisfied:

Universal Subsystem Property 1 Subsystem Property 1 from Section 2.4 is satisfied with $H(X)$ replaced by $H_\infty(\bar{\mathcal{X}})$.

Universal Subsystem Property 2 The precoders $\{\pi_n\}_{n=1}^\infty$ and the source coders $\{\sigma_n : \mathcal{X}^n \times \mathcal{Y}^n \rightarrow \{0,1\}^\dagger\}_{n=1}^\infty$ are such that for each channel $q \in \mathcal{Q}$, there is a function λ_q

$$E[L_{i+1}^\sigma] \leq E[L_i]\, H(\bar{\mathcal{X}}|\bar{\mathcal{Y}}(q)) + \lambda_q(E[L_i]), \text{ for } i = 0, \cdots, B_N - 1, \tag{4.5}$$

where λ_q also has the properties that $\lambda_q(x) = o_x(x)$ and λ_q is non-negative, monotonically increasing, and concave ($\cap$) over $[1, \infty)$.

Universal Subsystem Property 3 The termination coding scheme c^{term} takes two parameters κ and ν and returns a code. For any $\alpha > 0$, there exists a sequence of parameters $\{(\kappa_n, \nu_n(\alpha))\}_{n=1}^{\infty}$ such that for each $q \in \mathcal{Q}$, $c^{\text{term}}(\kappa_n, \nu_n(\alpha))$, whose corresponding encoder is denoted $e^{\text{term}}_{\kappa_n,\nu_n}$, encodes $\kappa_n = \lceil n^{1/4} \rceil$ message bits into an average number of channel inputs $\eta_n(q) = \alpha n + \varphi_{1,q}(n)$, where $\varphi_{1,q}(n) = o_n(1)$. Furthermore, for each channel $q \in \mathcal{Q}$, there exists a constant $E_0(q) > 0$ and a function $\varphi_{2,q}(n) = o_n(1)$ such that the error probability $P_{\text{e,term},n}(q)$ associated with $c^{\text{term}}(\kappa_n, \nu_n(\alpha))$ operating on q is bounded according to

$$P_{\text{e,term},n}(q) < \exp_2\left\{ -\eta_n(q)\Big(E_0(q) - \varphi_{2,q}(n) \Big) \right\}. \tag{4.6}$$

Universal Subsystem Property 4 With $\{t_n\}_{n=1}^{\infty}$ a sequence satisfying $t_n = o_n(1)$, the sequence of detector-sequence pairs $\{(\delta_{t_n}, \emptyset^{t_n}[n])\}_{n=1}^{\infty}$ is such that on each channel $q \in \mathcal{Q}$, there exists a constant $M_{\text{MD}}(q) < 1$ and functions $\vartheta_{1,q}(n) {=} o_n(n^{-3})$ and $\vartheta_{2,q}(n) {=} o_n(1)$ such that the false-alarm and missed-detection probabilities $P_{\text{FA},t_n}(q)$ and $P_{\text{MD},t_n}(q)$, respectively, associated with each use of δ_{t_n} by the receiver satisfies

$$P_{\text{FA},t_n}(q) < \vartheta_{1,q}(n) \tag{4.7}$$

$$P_{\text{MD},t_n}(q) < M_{\text{MD}}(q) + \vartheta_{2,q}(n). \tag{4.8}$$

Then let c_{U} be a coding scheme mapping N and ν to a code that is identical to the code described in Section 2.3 with the modifications from Section 2.5 but for the fact that it substitutes subsystems having the properties above for the corresponding ones in Section 2.3.

We can then prove the following theorem:

Theorem 4.3.1 Let $\{(\kappa_n, \nu_n(\alpha)\}_{n=1}^{\infty}$ be the sequence of termination-code parameter values described in Universal Subsystem Property 3. If $I(q) > 0$ for all $q \in \mathcal{Q}$, then the sequence of codes $\{(c_{\text{U}}(N, \nu_N(\alpha))\}_{N=1}^{\infty}$ attains the rate function $r : \mathcal{Q} \to (0, \infty)$ defined by

$$r(q) = I(q)/(\alpha I(q) + 1) \tag{4.9}$$

and the error exponent function $E_{\text{U},\alpha} : \mathcal{Q} \to (0, \infty)$ defined by

$$E_{\text{U},\alpha}(q) = \left(1 - \frac{r(q)}{I(q)}\right) E_0(q). \tag{4.10}$$

Remark: This theorem immediately implies that the rate function I is achievable, since as $\alpha \to 0$, $r(q) \to I(q)$.

Proof: Let us fix $q \in \mathcal{Q}$. Then Universal Subsystem Properties 1-4 imply that Subsystem Properties 1-4 from Section 3.2 hold for q, and the theorem follows by using the same arguments used to prove Theorem 2.4.1. □

In the following sections, we describe and analyze precoding, source coding, synchronization, and termination coding subsystems that satisfy the Universal Subsystem Properties. We discuss uniform convergence in Section 4.5.

4.3.1 Precoder and Source Coder

To begin, note that because the input distribution is fixed regardless of the realized channel, the precoder described in Section 3.3.1 for DFSC$_f$'s can be used as-is in our extension to UFSC$_f$'s. However, as we develop in the remainder of this section, the source coder for this case is substantially different. The source coder we develop, which we refer to as a *conditional Lempel-Ziv coder*, is based on the universal decoder introduced in [66].

The conditional Lempel-Ziv encoder $\sigma_{\text{CLZ}} : \mathcal{X}^\dagger \times \mathcal{Y}^\dagger \to \{0,1\}^\dagger$ codes a sequence $x^n \in \mathcal{X}^n$ based on a sequence $y^n \in \mathcal{Y}^n$ so that x^n can be recovered from $\sigma_{\text{CLZ}}(x^n|y^n)$ and y^n. This is accomplished via a Lempel-Ziv parsing (1978 version, [67]) of the sequence z^n, where $z_i = (x_i, y_i) \in \mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$. This parsing is completely described by the phrase indices $\{p_j\}_{j=1}^c$, where c is the resulting number of phrases. In particular, the jth phrase is $z_{p_j}^{p_{j+1}-1}$ and has length $p_{j+1} - p_j$.

In describing the encoding, it is convenient to define[1]

$$J[\tilde{y}] = \{j \ : \ y_{p_j}^{p_{j+1}-1} = \tilde{y}\}, \tag{4.11}$$

so that[2]

$$c[\tilde{y}] \triangleq |J[\tilde{y}]|$$

denotes the number of phrases in the parsing of z^n that match the tuple $\tilde{y}$ in their y-components.

With this notation, the x-component of the jth phrase of z^n, i.e., $x_{p_j}^{p_{j+1}-1}$, is encoded by the final letter $x_{p_{j+1}-1}$ of the phrase along with the index in $J[y_{p_j}^{p_{j+1}-2}]$ of the phrase whose x-component is $x_{p_j}^{p_{j+1}-2}$; this encoding requires approximately $\log|\mathcal{X}| + \log c[y_{p_j}^{p_{j+1}-2}]$ bits. The decoder can then recover x^n from this encoding provided the encoder also sends the phrase lengths $\{p_{j+1} - p_j\}_{j=1}^c$. The encoding process is depicted in Figure 4-1.

To analyze the expected length of the conditional Lempel-Ziv encoding, we exploit the following bound, which we prove in Appendix C.1:

$$\ell(\sigma_{\text{CLZ}}(x^n|y^n)) \leq -\log p_{\hat{X}^n|\hat{Y}^n(q)}(x^n|y^n) + \tilde{\lambda}_q(n), \tag{4.12}$$

where

$$\tilde{\lambda}_q(n) \triangleq C_q \frac{(n+2)\log\log(n+2)}{\log(n+2)},$$

with C_q a constant independent of n but dependent on q (in particular, on the size of the state space of q). The processes $\{\hat{X}_k\}$ and $\{\hat{Y}_k(q)\}$ referred to in (4.12) are defined in the same way as in Section 3.3.2, with the dependence of the channel output process on the realized channel q shown explicitly.

Using this property, it can be shown that a variant of (3.24) describing the interaction between

[1] For typographical reasons, we use square-bracket notation as a substitute for subscript notation.

[2] The cardinality of a set $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

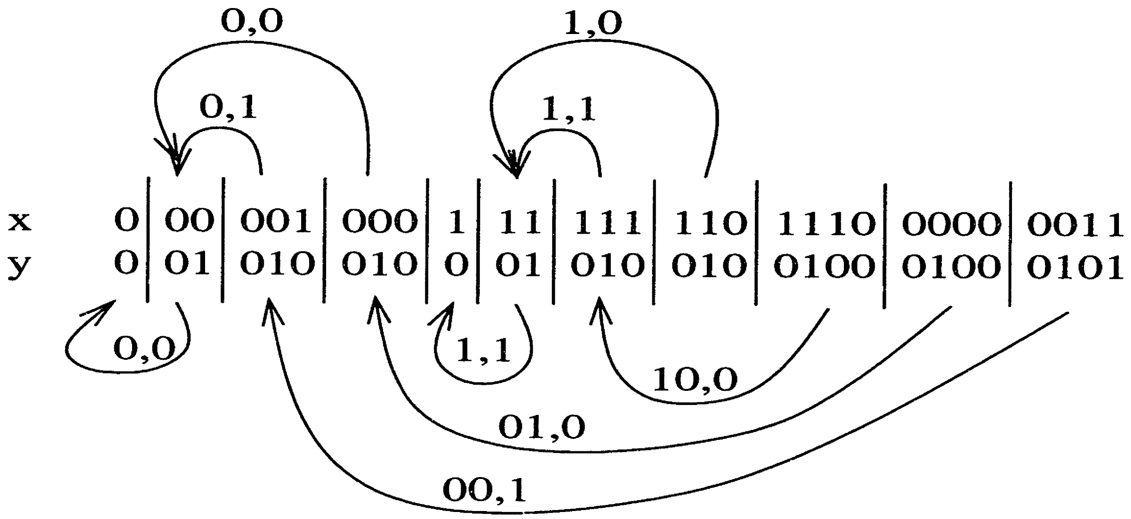


Figure 4-1: Graphical representation of conditional Lempel-Ziv encoding of $x^n \in \{0,1\}^n$ based on $y^n \in \{0,1\}^n$.

the source coder and precoder still holds. In particular, we have

$$E[L^{\sigma}_{i+1}] \leq E[L_i]\, H_\infty(\mathcal{X}|\mathcal{Y}(q)) + 5\log E[L_i] + C_{\sigma,q} + E[\tilde{\lambda}_q(L_i)] \tag{4.13}$$

$$\leq E[L_i]\, H_\infty(\mathcal{X}|\mathcal{Y}(q)) + 5\log E[L_i] + C_{\sigma,q} + \tilde{\lambda}_q(E[L_i]), \tag{4.14}$$

where $C_{\sigma,q}$ is a constant independent of $E[L_i]$ but dependent on q. To obtain (4.14), we exploited the concavity[3] of $\tilde{\lambda}_q$ with Jensen's inequality. Letting $\lambda_q(x) = \tilde{\lambda}_q(x) + 5\log x + C_{\sigma,q}$, we see that λ_q is a non-negative, monotonically increasing, concave function over $[1,\infty)$ and that $\lim_{x\to\infty} \lambda_q(x)/x \to 0$. Thus, Universal Subsystem Property 2 holds for this source coder.

4.3.2 Synchronization Subsystem

For the synchronization subsystem in a particular universal code with parameter N, we use a sequence $\emptyset^{t_N}[N]$ that is independent of the channel, and modify the detector for the sequence so as not to rely explicitly on knowledge of the channel statistics.

Following the general universal decoding strategy developed in [66], a universal method for

[3] The second derivative with respect to x of $(2+x)\log\log(2+x)/\log(2+x)$ is

$$-\frac{(\log(x+2)-2)(\log\log(x+2)-1)+1}{(x+2)\log^3(x+2)}, \tag{4.15}$$

which can be shown to be negative for all $x > -1$.

detecting the presence of the sequence $\emptyset^{t_N}[N]$ involves a detector of the form

$$\delta_{\mathrm{U},t_N}(y^{t_N}, \emptyset^{t_N}[N]) = \begin{cases} 1 & \text{if } u_2(y^{t_N} | \emptyset^{t_N}[N]) \leq u_1(y^{t_N}) - \zeta(t_N), \\ 0 & \text{otherwise,} \end{cases} \tag{4.16}$$

where

$$\zeta(t_N) = \frac{t_N (\log \log t_N)^2}{\log t_N}. \tag{4.17}$$

In (4.16), the function $u_1 : \mathcal{Y}^\dagger \to \mathbb{N}$ returns the length of a Lempel-Ziv encoding of $y^{t_N} \in \mathcal{Y}^\dagger$, while the function $u_2 : \mathcal{Y}^\dagger \times \mathcal{X}^\dagger \to \mathbb{N}$ returns the length of the conditional Lempel-Ziv encoding of y^{t_N} given $\emptyset^{t_N}[N]$.

As in Chapters 2 and 3, we let $t_N = \lceil \sqrt{N} \rceil$. We then show in Appendix C.2 that by choosing the sequence $\emptyset^{t_N}[N]$ randomly and using the detector δ_{U,t_N}, (4.7) and (4.8) are satisfied; in the process, why ζ is chosen as the strange looking function in (4.17) becomes evident. Hence, we design the synchronization subsystem to pseudorandomly generate a new sequence $\emptyset^{t_N}[N]$ each time a synchronization sequence is sent over the channel.

4.3.3 Termination Codes

We can again use mSB codes for encoding the block of data generated after B_N iterations of the compressed-error-cancellation procedure, though the decoding process required at the receiver is more elaborate. In particular, to decode the inner code and to distinguish c^ν from w^ν, we use the universal decoders developed by Ziv in [66]. It is shown in Ziv [66] and Lapidoth and Ziv [38] that under certain conditions, Ziv's universal decoder performs, in a certain sense, asymptotically as well as a maximum-likelihood decoder. While these results do not specifically address the universal decoding of a codebook containing exactly two codewords, which is of greatest interest for our purposes, they can be extended for this case [37]. In particular, the extended results state that if the codewords of a two-codeword codebook are drawn uniformly from a permutation invariant set, then the error exponent associated with average probability of error incurred by Ziv's universal decoder is asymptotically the same as the error exponent associated with the average probability of error incurred by a maximum-likelihood decoder.

The effective channel input distribution is restricted, because a uniform distribution over some permutation-invariant set corresponds to some i.i.d. input distribution. It is unclear which i.i.d. input distribution would be the "best" to use for $\mathcal{Q}$, so to be specific, let us consider the case in which the input distribution is uniform over all channel inputs, bearing in mind that this need not be the case. Using the extended results from [37], we can say that Ziv's universal decoder achieves the error exponent $E_{2,\mathrm{ML}}(q)$ associated with maximum-likelihood decoding on such a randomly chosen two-codeword codebook.

Using now the same mSB scheme as in Section 2.3.1.3, with all the necessary sequences assumed chosen according to the appropriate distributions, we find that Subsystem Property 3 is satisfied with $E_0(q) = E_{2,\mathrm{ML}}(q)$. Note that we must assume that the necessary sequences are chosen pseudorandomly. Since the sequences are assumed to be i.i.d. uniformly over the input distribution, it is particularly easy to generate such sequences pseudorandomly.

4.4 Complexity

The universal communication strategy has some attractive complexity characteristics. In particular, it is straightforward to verify that the complexity of the overall scheme for UFSC$_f$'s is of the same order as that for known DFSC$_f$'s under the uniform-cost criteria. To see this, it suffices to note that under both ccst criteria, the time and space complexity of conditional Lempel-Ziv encoding and decoding can be shown to be linear in the length of the input, and that the time and space complexity of Ziv's universal decoding procedure, which is based on conditional Lempel-Ziv coding, can also be shown to have the same time and space complexity as maximum-likelihood decoding. Since real numbers are not manipulated anywhere in the source coding, synchronization or termination coding subsystems, this characterization should accurately reflect the behavior of the algorithms on a digital computer. We note finally that the space complexity of the conditional Lempel-Ziv coder is linear and would be linear even if memory used for buffering inputs and outputs were not counted. This behavior differs from that of, say, an arithmetic source coder of an i.i.d. source, which requires only constant space when buffering requirements are ignored.

4.5 Uniform Convergence

For universal decoders, Lapidoth and Ziv [38] distinguish between non-uniform convergence of their decoder's performance to the maximum-likelihood decoder, which they call weak universality, and uniform convergence, which they call strong universality.

For a feedback scheme, one natural notion of uniform attainment of a rate function r would mean that some sequence of codes would have a sequence of rates $\{R_N(q)\}$ and probabilities of error $\{P_N(q)\}$ so that

$$\limsup_{N\to\infty} \inf_{q\in\Omega} R_N(q)/r(q) \geq 1 \tag{4.18a}$$

$$\liminf_{N\to\infty} \sup_{q\in\Omega} P_N(q) = 0. \tag{4.18b}$$

Similarly, uniform attainment of an error exponent function F would mean that

$$\limsup_{N\to\infty} \inf_{q\in\Omega} -R_N(q)\log P_N(q)/(NF(q)) \geq 1. \tag{4.19}$$

With c_{U} denoting the universal coding scheme that uses the subsystems described in Sections 4.3.1–4.3.3, the sequence of codes $\{c_{\mathrm{U}}(N, \lceil \alpha N \rceil)\}_N$ does not uniformly achieve the rate function r defined by $r(q) = I(q)/(\alpha I(q) + 1)$ without constraining Ω. For example, let Ω be the set of all BSC$_f$'s, and choose the input distribution to be i.i.d. and uniform over $\{0, 1\}$. Then for any value of N, a channel would exist such that the verification sequences used by the termination coder would be nearly indistinguishable. The probability of error could not, therefore, uniformly approach 0.

Could the error exponent function $E_{\mathrm{U},\alpha}$ be attained uniformly? The answer still seems to be "no." For any particular N, the number of iterations B_N would be fixed, and there would be a channel q with crossover probability $\epsilon(q)$ such that the expected number of final message bits would be about $N/2$, i.e., $H(\epsilon(q)) = 2^{-1/B_N}$, requiring more than $\alpha N^2/(2\kappa)$ channel inputs to transmit, giving an error exponent less than about $2E_0\kappa\alpha/N$. But $E_{\mathrm{U},\alpha}(q) \approx E_0\alpha(1 - 2^{-1/B_N})$ which approaches 0 much more slowly than $2E_0\kappa/N$, which implies that the error exponent function cannot be attained uniformly in the sense described above.

Even if we were allowed to know the channel and could use the scheme from Chapter 3, but had to fix N and ν before knowing the channel, we would not be able to uniformly attain these rate or error exponent functions. That they are not uniformly attainable is not so much a product of our universal scheme as a product of Ω. We would like to separate the analysis of properties of Ω from the properties of the universal coding scheme. In Lapidoth and Ziv's work on universal decoding, analogous difficulties are avoided by competing with the maximum-likelihood decoder, rather than by evaluating convergence to a particular error exponent function.

The maximum-likelihood decoder is a natural benchmark, because it is the optimal decoder in the sense of minimizing probability of error. But when feedback is available, it is not clear what scheme would represent a natural benchmark. One possibility is to compete with the scheme of Chapter 3. The drawback of such an approach is that we do not know if that scheme is optimal in any sense. If the universal scheme does not converge to that scheme in some sense, then nothing prevents us from changing that scheme for the worse until the universal scheme does achieve uniform convergence.

In the absence of a better alternative, let us explore this comparative approach. It happens that is it still difficult to prove that any interesting uniform convergence properties hold. Let $c_{q,\text{FS}}$ be the coding scheme described in Chapter 3, designed for channel q, with the additional constraint that the mSB verification sequences c^ν and w^ν are chosen randomly and uniformly from a permutation-invariant set and distinguished with a maximum-likelihood decoder. Also let c_U draw its mSB verification sequences from the same permutation-invariant set. If $P_{q,n}$ is the probability of error associated with $c_{q,\text{FS}}(n, \lceil \alpha n \rceil)$, and $P_{\text{U},q,n}$ is the probability of error associated with $c_\text{U}(n, \lceil \alpha n \rceil)$ operating on q, then an extension [37] of the results of Lapidoth and Ziv [38] imply that $\lim_{n\to\infty} \sup_{q\in\Omega} \log P_{\text{U},q,n}/P_{q,n} = 0$. The reason is that the probability of error for the universal scheme is essentially determined by the probability of confusing w^ν for c^ν. On the other hand, the rates of the two schemes cannot be shown to converge uniformly to one another. The fault lies with our inability to show that the probability of confusing c^ν for w^ν under maximum-likelihood decoding is uniformly small for all q. Without this fact, the degraded performance introduced by universal decoding can cause an unacceptably large number of mSB retransmissions. Similarly, we must also show that the probability of missed detection of the synchronization sequence is also uniformly small for all q. We believe restrictions on Ω may be required to show that these probabilities are uniformly small.

For uniform convergence to the actual rate function — rather than to the rate of an opponent scheme — to occur in the sense of (4.18), some constraints must be placed on Ω itself. For a universal coding scheme satisfying the Universal Subsystem Properties 1-4, the most straightforward constraints allowing uniform convergence to the rate $I(q)/(\alpha I(q) + 1)$ are the following:

$$\inf_{q\in\Omega} I(q) > 0 \tag{4.20a}$$

$$\inf_{q\in\Omega} E_0(q) > 0 \tag{4.20b}$$

$$\lim_{n\to\infty} \sup_{q\in\Omega} \lambda_q(n) = 0 \tag{4.20c}$$

$$\lim_{n\to\infty} \sup_{q\in\Omega} \varphi_{1,q}(n) = 0 \tag{4.20d}$$

$$\lim_{n\to\infty} \sup_{q\in\mathcal{Q}} \varphi_{2,q}(n) = 0 \tag{4.20e}$$

$$\lim_{n\to\infty} \sup_{q\in\mathcal{Q}} n^3\vartheta_{1,q}(n) = 0 \tag{4.20f}$$

$$\lim_{n\to\infty} \sup_{q\in\mathcal{Q}} \vartheta_{2,q}(n) = 0 \tag{4.20g}$$

$$\sup_{q\in\mathcal{Q}} M_{\mathrm{MD},t_n} < 1. \tag{4.20h}$$

These conditions are also sufficient to give uniform convergence to the error-exponent function $E_{\mathrm{U},\alpha}$ in the sense of (4.19) (actually, (4.20b) need not hold).

These conditions essentially say that for each subsystem, there is a most antagonistic channel in $\mathcal{Q}$. The parameter N must then be chosen large enough so that each subsystem copes reasonably well with its most antagonistic channel.

As an example showing that the constraints can actually be met, we note that if, for example, $\mathcal{Q}$ were composed of all BSC$_f$'s with crossover probabilities less than x, where $x < 1/2$, then $\mathcal{Q}$ would meet the above constraints. We have already shown that there are sets of channels $\mathcal{Q}$ such that the constraints are not met.

4.6 Discussion

4.6.1 Choice of input distribution

The range of rates achievable with this scheme depends on the choice of input distribution. Because the coding scheme achieves a different rate for each realized channel, trying to optimize the rate leads to a multicriteria optimization problem, whose solution depends in general on the goals of the system designer. Nevertheless, as a default, one reasonable choice of input process for an arbitrary set of channels $\mathcal{Q}$ is an i.i.d. sequence of random variables uniformly distributed over $\mathcal{X}$. This choice at least guarantees some positive mutual information for every channel having positive capacity. This is easily shown by verifying that the expression $n \log |\mathcal{X}| - H(X^n|Y^n) > 0$ for some n unless for all n, $p_{Y^n|X^n}(y^n|x^n) = p_{Y^n}(y^n)$ for all x^n and y^n, which would imply a channel with zero capacity.

4.6.2 Computational Delays and Maximum Quality

Recall from Section 2.8.1 that if a computer providing a fixed number of computational resources per unit time is used to carry out the encoding, then we require the computer to be at least fast enough to be able to source code and then precode n channel inputs in the time it takes to transmit $nF(q)$ samples, where $F(q) = H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}(q))/H_\infty(\bar{\mathcal{X}})$. In the case of universal communication, because $F(q)$ may be different for each q, it seems that $\inf_{q\in\mathcal{Q}} F(q)$ must be strictly positive to be able to choose a computer that is sufficiently powerful to carry out the interleaving solution described in Section 2.8.1.

4.6.3 Summary and Future Directions

We have developed a linear-complexity coding scheme giving exponentially decaying error probabilities at any rate below what is essentially the mutual information induced by passing a given finite-order Markov input distribution through an unknown indecomposable DFSC.

The universal communication scheme we have introduced is not practically viable at rates near the mutual information, because extremely long blocklengths are required before the rates of the conditional Lempel-Ziv coders become reasonably close to their asymptotic values. Since the computation per channel input used is approximately a constant, long blocklengths present no problems in terms of time spent computing, but at some point, delay and storage may become the limiting factors. It may be more practical to probe the channel with training data first, identify the channel, and then use a feedback scheme such as that described in Chapter 3.

Nevertheless, the universal communication scheme we have described is important, because it shows that universal communication at rates that vary with the realized channel can be accomplished without the use of training data, which in general reduces the attainable error exponent. While the universal communication scheme we have described may not always attain the best exponent, because the input distribution does not adapt to the channel, there are cases in which variants of the scheme could achieve the maximum error exponent achievable by any feedback scheme that knows the realized channel a priori. An example of such a case is when the unknown channel is the set of all $\mathrm{BSC}_{\mathrm{f}}$'s with crossover probability less than some upper bound $x < 1/2$. In this case, the mSB scheme would be able to use the all-zeros and all-ones sequences as c^{ν} and w^{ν}, which could be distinguished optimally without knowing the channel by using a simple majority vote. In this case, the error exponent of the universal coder would be the optimum achievable by any scheme. While this sort of performance is not necessarily typical, it demonstrates that the universal scheme we have developed can achieve performance that could seemingly never be achieved with a training-sequence-based approach.

Among a variety of interesting and promising directions for further research with this approach to communication with feedback are extensions to the scheme that involve adapting the input distribution according to the feedback to achieve the feedback capacity of the realized but unknown $\mathrm{DFSC}_{\mathrm{f}}$.

Chapter 5

Multiple-Access Channels

5.1 Introduction

In the preceding three chapters, we develop and apply the compressed-error-cancellation framework to the single-user case, i.e., the situation in which a single transmitter must send data to a single receiver. But often in practice, multiple transmitters must simultaneously send data to multiple receivers. If the transmitters share the same channel in some sense, then the complications introduced by the mutual interference invalidate the single-transmitter, single-receiver model.

Of particular importance are multiple-access channels, which, by definition, arise when multiple transmitters send data to a single, common receiver. Most wireless networks and many wired networks employ a multiple-access channel in the following way: a number of end users send data to a central hub (e.g, a base station) which forwards the data to another hub, which forwards the data to another end user; the channel from the end users to the common hub is a multiple-access channel.

An expansion in the number of data networks, especially wireless ones, such as cellular telephone networks, has given rise to a parallel expansion of interest in multiple-access channels. One of the central questions of interest is, just as in the single-user case, how to send data at high rates with a reasonable amount of computation and low probabilities of error.

Much of the information-theoretic work on multiple-access channels has focused on finding the maximum rates at which it is possible to send data reliably, given unlimited computation and delay. Liao [39] and Ahlswede [2] determined the set of achievable rates, known as the *capacity region*, for discrete memoryless multiple-access channels (DMMAC's), which are discussed further below. But since the proof of the achievability of these rates is non-constructive in the same sense as Shannon's coding theorem is non-constructive, much practical work has been focused on finding computationally feasible methods for achieving rates close to the frontier[1] of the capacity region. The dominant approach in practice has been to partition the channel according to time (time-division multiple access), frequency (frequency-division multiple access), or some other space (code-division multiple access) into several (nearly) independent channels. After partitioning, each user communicates to the receiver as though the channel were a single-user channel and can employ coding techniques developed for this case. In general, these methods do not allow rates arbitrarily close to the frontier

[1] The frontier of an achievable region in rate-space is the set of achievable points at which one user's rate is maximized given the other user's rate is fixed.

of the capacity region.[2] Moreover, excepting the time-division technique, these methods are inapplicable to all but a small (albeit very practical) subset of multiple-access channels, namely those in which the transmitters' input symbols and noise are additively combined to form the output symbol.

The question we address in this chapter is whether and if so, how feedback can be used to reduce the computation required for coding over a multiple-access channel. In particular, we focus on two-user DMMAC's with feedback (DMMAC$_\mathrm{f}$'s), primarily for the sake of simplicity.

A two-user DMMAC is described by a 4-tuple $(\mathcal{X}_1, \mathcal{X}_2, q_{Y|X_1,X_2}, \mathcal{Y})$, which specifies the forward channel. The finite set $\mathcal{X}_1$ is the range of allowable inputs to the channel by Transmitter 1 (Tx-1); the finite set $\mathcal{X}_2$ is the range of allowable inputs by Transmitter 2 (Tx-2); the finite set $\mathcal{Y}$ is the range of possible channel outputs; and the function $q_{Y|X_1,X_2}$ describes the statistical relationship between the two transmitters' inputs and the channel output. The following definition completes the meaning of this 4-tuple:

Definition 5.1.1 Let $\{Y_k\}_{k=1}^\infty$, $\{X_{1,k}\}_{k=1}^\infty$ and $\{X_{2,k}\}_{k=1}^\infty$ be random processes. Then $\{Y_k\}_{k=1}^\infty$ is the *output process* resulting from *passing* $\{X_{1,k}\}_{k=1}^\infty$ *and* $\{X_{2,k}\}_{k=1}^\infty$ *through a DMMAC* $(\mathcal{X}_1, \mathcal{X}_2, q_{Y|X_1,X_2}, \mathcal{Y})$ (or just $q_{Y|X_1,X_2}$) *without feedback* if for all[3] $n > 0$,

$$p_{Y^n|X_1^n,X_2^n}(y^n, x_1^n, x_2^n) = \prod_{k=1}^{n} q_{Y|X_1,X_2}(y_k|x_{1,k}, x_{2,k}) \tag{5.1}$$

for all $x_1^n \in \mathcal{X}_1^n$, $x_2^n \in \mathcal{X}_2^n$, and $y^n \in \mathcal{Y}^n$.

Like a DMC$_\mathrm{f}$, a DMMAC$_\mathrm{f}$ has feedback channels that are noiseless, delayless, and have sufficient capacity to feed back the receiver's complete observation. At time k, both transmitters know the value of the channel outputs from time 1 to time $k-1$. A DMMAC$_\mathrm{f}$ is then defined analogously to a DMC$_\mathrm{f}$ as follows:

Definition 5.1.2 Let M be a random variable taking values in $\mathcal{M}$. For all $m \in \mathcal{M}$, let $\{f_{m,i}\}_{i=1}^\infty$ be a sequence of functions, where $f_{m,i} : \mathcal{Y}^{i-1} \to \mathcal{X}_1 \times \mathcal{X}_2$ maps a sequence of $i-1$ channel outputs to a single channel input pair, and $f_{m,1}$ takes a constant value in $\mathcal{X}_1 \times \mathcal{X}_2$. Then $\{Y_k\}$ is the *output process resulting from passing* M *through a DMMAC*$_f$ $(\mathcal{X}_1, \mathcal{X}_2, q_{Y|X_1,X_2}, \mathcal{Y})$ (or just $q_{Y|X_1,X_2}$) *via* $\{\{f_{m,i}\}_{i=1}^\infty\}_{m\in\mathcal{M}}$ if for all $n > 0$,

$$p_{Y^n|M}(y^n|m) = \prod_{k=1}^{n} q_{Y|X_1,X_2}(y_k|f_{m,k}(y^{k-1})) \tag{5.2}$$

for all $y^n \in \mathcal{Y}^n$ and all $m \in \mathcal{M}$.

The capacity region of a DMMAC is defined as the closure of the set of achievable rate pairs. The capacity region of a two-user DMMAC $q_{Y|X_1,X_2}$ is known [39, 2] to be the closure of the set

[2] In theory, CDMA in conjunction with optimal decoding allows rates arbitrarily close to the frontier of the capacity region on additive white Gaussian noise channels. Optimal decoding, however, is computationally very expensive.

[3] We rely on context to distinguish between x_2^n's denoting $(x_{2,1}, \cdots, x_{2,n})$ versus $(x_2, \cdots, x_n)$.

$\mathcal{R}$, which is defined by

$$\mathcal{R} = \mathrm{co} \left\{ (r_1, r_2) \ : \ r_1 < I(Y; X_1 | X_2), r_2 < I(Y; X_2 | X_1), \text{and } r_1 + r_2 < I(Y; X_1, X_2) \right.$$
$$\text{for some random variables } X_1, X_2, \text{ and } Y \text{ satisfying}$$
$$p_{Y,X_1,X_2}(y, x_1, x_2) = p_{X_1}(x_1) p_{X_2}(x_2) q_{Y|X_1,X_2}(y|x_1, x_2)$$
$$\left. \text{for all } x_1 \in \mathcal{X}_1, \text{ all } x_2 \in \mathcal{X}_2, \text{ and all } y \in \mathcal{Y} \right\}, \qquad (5.3)$$

where co$\mathcal{A}$ denotes the convex hull of a subset $\mathcal{A}$ of Euclidean space. Any rate pair outside the closure of $\mathcal{R}$ is known not to be achievable. An example of a capacity region is depicted in Figure 5-1.

In this chapter, we design a linear-complexity coding scheme for two-user DMMAC$_f$'s that can be used to achieve rate pairs on the frontier of the feedback-free capacity region. We derive this low-complexity coding scheme by extending the compressed-error-cancellation framework introduced in Chapter 2. The presence of feedback is also known to extend the capacity region for DMMAC's. The schemes we develop do not achieve points in this extended region, although the compressed-error-cancellation framework does yield an interesting perspective on why feedback extends the capacity region.

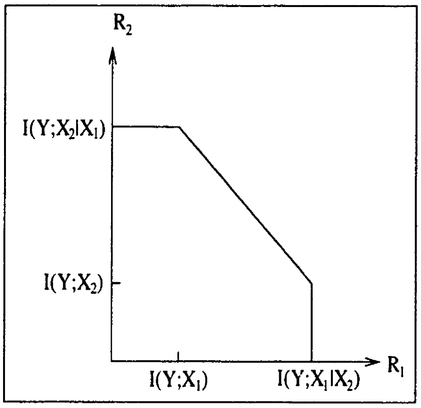


Figure 5-1: An example of a capacity region.

To facilitate the discussion in this chapter, let X_1 and X_2 be independent and distributed according to some given pmfs q_{X_1} and q_{X_2}, respectively. Let Y be such that $p_{Y|X_1,X_2}(y|x_1,x_2) = q_{Y|X_1,X_2}(y|x_1,x_2)$ for all $x_1 \in \mathcal{X}_1$, $x_2 \in \mathcal{X}_2$, and $y \in \mathcal{Y}$.

With this notation, we can discuss a natural extension of the framework for the two-user DMMAC$_f$. This extended framework is illustrated as follows, with the perspective of Transmitter 1 (Tx-1) on the left and that of Transmitter 2 (Tx-2) on the right, where corresponding events on each side are not necessarily synchronized in time:

- Source 1 produces N_1 message bits to be sent by Tx-1 to the receiver (Rx).
- Tx-1 precodes its N_1 bits into the $N_1' = N_1/H(X_1)$ channel inputs $X_1^{N_1'}$.
- The channel corrupts the transmitted data according to $q_{Y|X_1,X_2}$.

- Source 2 produces N_2 message bits to be sent by Tx-2 to Rx.
- Tx-2 precodes its N_2 bits into the $N_2' = N_2/H(X_2)$ channel inputs $X_2^{N_2'}$.
- The channel corrupts the transmitted data according to $q_{Y|X_1,X_2}$.

- Rx feeds the corrupted data $Y^{N_1'}$ back to Tx-1.
- Tx-1 compresses $X_1^{N_1'}$ into $N_1'H(X_1|Y_1)$ bits, and precodes these bits into $N_1'' = N_1'H(X_1|Y_1)/H(X_1)$ samples.

$\vdots$

- Rx feeds the corrupted data $Y^{N_2'}$ back to Tx-2.
- Tx-2 compresses $X_2^{N_2'}$ into $N_2'H(X_2|Y,X_1)$ bits, and precodes these bits into $N_2'' = N_2'H(X_2|Y,X_1)/H(X_2)$ samples.

$\vdots$

If both processes continued through an infinite number of iterations, the resulting rate for Tx-1 would be $I(Y;X_1)$, and that for Tx-2 would be $H(X_2) - H(X_2|Y,X_1) = H(X_2|X_1) - H(X_2|Y,X_1) = I(Y;X_2|X_1)$. This rate pair is on the frontier of the multiple access capacity region for certain choices of the input distributions of both transmitters.

But a substantial obstacle blocks the path from this high-level illustration to a low-complexity coding scheme. Namely, a sort of conditional Slepian-Wolf coding is required for Tx-2 to encode its previous block of inputs with $N_2'H(X_2|Y,X_1)$ bits without knowledge of what Tx-1 sends. Since no techniques for computationally efficient Slepian-Wolf coding and decoding at this rate are known, we cannot quite use this approach.

Notice, however, that if the receiver can eventually know what Tx-1 sends, then Tx-2 must also eventually be able to know what Tx-1 sends, since it observes the received data via feedback. The problem then is the time Tx-2 must wait to decode Tx-1's transmission. To make this waiting time small, suppose that $N_1 \ll N_2$. Let Tx-1 use the coding scheme for (single-user) DMC's from Chapter 2, and suppose that Tx-1 sends many N_1-bit messages in a row. Let Tx-2 send its message using the scheme described in the illustration above, which takes several iterations. After the ith iteration, suppose Tx-2 waits until the current message of Tx-1 is decodable, sending filler data during this waiting time, and decodes it. Since all previous messages of Tx-1 can also be decoded, Tx-2 can know what Tx-1 sent during iteration i, which allows Tx-2 to perform iteration $i+1$. Since $N_1 \ll N_2$, Tx-2's waiting times can be made small and so can the number of channel inputs that pass unused during these waiting times.

In the following sections, we describe and analyze a transmission scheme based on this idea. We begin with a formulation of variable-length coding for DMMAC$_f$'s in Section 5.2. We follow in Sections 5.3–5.4 with a description and analysis of a coding scheme, and end in Section 5.5 with a discussion of certain related topics and future directions.

5.2 Formulation of Variable-Length Coding for Two-User Multiple-Access Channels

A variable-length code for a DMMAC$_f$ is described by the 6-tuple $(N_1, \varepsilon_1, N_2, \varepsilon_2, \{\chi_i\}_{i=0}^{\infty}, \Delta)$. The encoders ε_1 and ε_2 have the following form that reflects the timing of the availability of the feedback:

$$\varepsilon_1(w_1^{N_1}, y^{\infty}) = (\tilde{\varepsilon}_{1,1}(w_1^{N_1}), \tilde{\varepsilon}_{1,2}(w_1^{N_1}, y_1), \tilde{\varepsilon}_{1,3}(w_1^{N_1}, y^2), \cdots) \tag{5.4}$$

$$\varepsilon_2(w_2^{N_2}, y^{\infty}) = (\tilde{\varepsilon}_{2,1}(w_2^{N_2}), \tilde{\varepsilon}_{2,2}(w_2^{N_2}, y_1), \tilde{\varepsilon}_{2,3}(w_2^{N_2}, y^2), \cdots), \tag{5.5}$$

where the range of $\varepsilon_{u,i}$ is $\mathcal{X}_u$ for $u = 1,2$ and $i = 1,2,3,\cdots$.

To define the rate and probability of error fcr this code, we first define a sequence of functions $\{f_i\}_{i=1}^{\infty}$, each of which defines the joint channel input at time i:

$$f_i(w_1^{N_1}, w_2^{N_2}) = (\tilde{e}_{1,i}(w_1^{N_1}, y^{i-1}), \tilde{e}_{2,i}(w_2^{N_2}, y^{i-1})) \tag{5.6}$$

Next, we let $W_1^{N_1}$ and $W_2^{N_2}$ be two independent random sequences, uniformly distributed over $\{0,1\}^{N_1}$ and $\{0,1\}^{N_2}$, respectively. We then let $\{Y_k\}$ be the output process resulting from passing $(W_1^{N_1}, W_2^{N_2})$ through the DMMACf $q_{Y|X_1,X_2}$ via $\{\{f_i(w_1^{N_1}, w_2^{N_2}, \cdot\,)\}_{i=1}^{\infty}\}_{w_1^{N_1}, w_2^{N_2}}$. With

$$L^* = \min\{k \;:\; \chi_k(Y^k) = 1\}, \tag{5.7}$$

the receiver makes an estimate $(\hat{W}_1^{N_1}, \hat{W}_2^{N_2})$ of both messages using the decoding function $\Delta : \mathcal{Y}^{\dagger} \to \{0,1\}^{N_1} \times \{0,1\}^{N_2}$. That is, the estimate is given by

$$(\hat{W}_1^{N_1}, \hat{W}_2^{N_2}) = \Delta(Y^{L^*}). \tag{5.8}$$

The rate of Tx-u is defined to be $N_u/E[L^*]$ for $u = 1, 2$, and the error probability for the code is defined as $\Pr\{(\hat{W}_1^{N_1}, \hat{W}_2^{N_2}) \neq (W_1^{N_1}, W_2^{N_2})\}$.

We define a coding scheme to be a function mapping three parameters (p_1, p_2, p_3) to a code, but as in Chapter 4, we work primarily with particular sequences of codes. A code sequence $\{a_k\}_k$ achieves a rate pair (r_1, r_2) if the corresponding rate-pair sequence $\{(R_{1,k}, R_{2,k})\}$ and probability-of-error sequence $\{P_k\}$ satisfy

$$\liminf_{k\to\infty} R_{u,k} \geq r_u, \quad u = 1, 2 \tag{5.9}$$

$$\limsup_{k\to\infty} P_k = 0. \tag{5.10}$$

The code sequence achieves the error exponent F, if

$$\liminf_{k\to\infty} -\log P_k / E[L_k^*] \geq F, \tag{5.11}$$

where $E[L_k^*]$ is expected number of channel inputs used by the code a_k.

Other formulations of variable-length codes for multiple-access channels are possible. This formulation is particularly simple, but somewhat inflexible in that it does not allow different probabilities of error and different transmission lengths for the transmitters. We discuss other possibilities in Section 5.5.2.

5.3 Low-Complexity Coding Scheme for Two-User Multiple-Access Channels

We now describe code sequences that achieve rate pairs arbitrarily close to $(I(Y;X_1), I(Y;X_2|X_1))$. Because the input distributions q_{X_1} and q_{X_2} are arbitrary, these sequences can effectively be used to achieve any of the points on the frontier of the feedback-free capacity region for a DMMAC (see Section 5.5.1). Furthermore, the computations per channel input required for encoding and decoding is upper bounded by a constant that is independent of rate and probability of error; i.e., the time

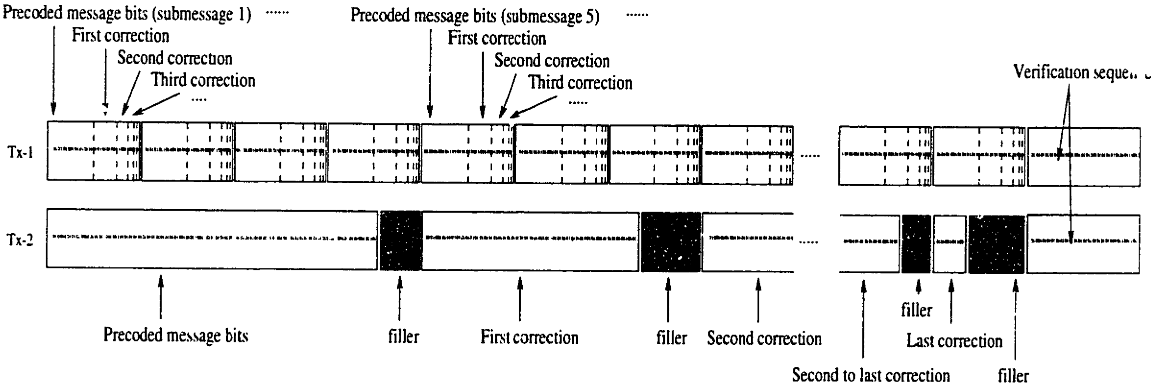


Figure 5-2: A graphical representation of the proposed multiple-access communication scheme.

complexity is uniformly linear.

As we state above, our scheme is based on the idea that with feedback, Tx-2 can know what Tx-1 sends with some delay. We must make this delay small, while maintaining a high probability of decoding Tx-1's data correctly. The scheme we propose operates at a high level as follows; a graphical representation of the scheme is also given in Figure 5-2:

- Source 1 produces $N_1 = k_1 n_1$ message bits to be sent by Tx-1 to Rx.
- Tx-1 breaks groups these bits into k_1 messages of length n_1 each.
- Tx-1 sends the first n_1-bit message using the fixed-length variant of the scheme from Section 2.7.1, taking η^{sub} channel inputs, where $n_1/\eta^{\text{sub}} = r_1 + o_{n_1}(1)$, where $r_1 < I(Y; X_1)$.
- Tx-1 sends the second n_1-bit message using the same fixed-length scheme.

⋮

- Tx-1 sends the k_1th n_1-bit message using the same fixed-length scheme.

- Source 2 produces N_2 message bits to be sent by Tx-2 to the Rx.
- Tx-2 precodes its N_2 bits into the $N_2' = N_2/H(X_2)$ channel inputs $X_2^{N_2'}$. It also puts $D_2 < \eta^{\text{sub}}$ random filler inputs into the channel.
- The channel corrupts the transmitted data according to $p_{Y|X_1,X_2}$.
- Rx feeds the corrupted data $Y^{N_2'+D_2}$ back to Tx-2.
- From $Y^{N_2'+D_2}$, Tx-2 determines $X_1^{N_2'}$ (with high probability of success).
- Tx-2 compresses $X_2^{N_2'}$ into $N_2' H(X_2|Y, X_1)$ bits, and precodes these bits into $N_2'' = N_2' H(X_2|Y, X_1)/H(X_2)$ channel inputs.

⋮

- After some fixed number of iterations, final residual data is FEC coded and sent.

- Simultaneously, Tx-1 and Tx-2 each send a length-ν sequence indicating whether retransmission is required for their respective data.

Note that because the length-ν verification sequences are sent by both transmitters simultaneously, N_1 and N_2 must have a particular relationship to avoid excessive waste of the channel caused by one transmitter's waiting for another to finish its main transmission. For now, we just say that $N_1 = h_\delta(N_2)$, and defer specification of h_δ until later.

We now give a more precise description of a coding scheme based on these ideas. With $\delta > 0$ given, we describe a scheme $c_{\mathrm{MA},\delta}$ capable of generating a sequence of codes that achieves the rate pair $(I(Y; X_1) - 2\delta, I(Y; X_2|X_1) - 2\delta)$. To do so, we let $(h_\delta(N_2), \varepsilon_1, N_2, \varepsilon_2, \{\chi_i\}_{i=0}^{\infty}, \Delta)$ be the code returned by $c_{\mathrm{MA},\delta}(N_2, \nu)$, and describe the encoders ε_1 and ε_2, which implicitly specify the other parts of the code as well. This coding scheme $c_{\mathrm{MA},\delta}$ takes only two parameters, which implies that there is a dependence among the rates of the two users and the probability of error. Nevertheless, this coding scheme can generate an appropriate sequence of codes achieving the rate pair $(I(Y; X_1) - 2\delta, I(Y; X_2|X_1) - 2\delta)$.

5.3.1 Encoders

Both ε_1 and ε_2 have the same structure. For $u = 1, 2$, Tx-u sends a length-η_u feedback inner codeword $e_u^{\mathrm{in}}(W_u^{N_u}, Y^\infty)$, where we enforce the constraint $\eta_1 = \eta_2$. If the receiver decodes this inner codeword incorrectly (via the corresponding inner decoder Δ_u^{in}) for either $u = 1$ or 2, then both transmitters retransmit their inner codewords, after first signaling whether their data need retransmission via a length-ν verification sequence. More precisely, for $u = 1, 2$, ε_u is the solution to the following set of equations:

$$\varepsilon_u(W_u^{N_u}, Y^\infty) = (e_u^{\mathrm{in}}(W_u^{N_u}, Y^\infty), v_u(W_u^{N_u}, Y^\infty), \rho_u(W_u^{N_u}; Y^\infty)), \tag{5.12}$$

$$\eta_u = \ell(e_u^{\mathrm{in}}(W_u^{N_u}, Y^\infty)) \tag{5.13}$$

$$v_u(W_u^{N_u}, Y^\infty) = \begin{cases} w_u^\nu & \text{if } \Delta_u^{\mathrm{in}}(Y^{\eta_u}) \neq W_u^{N_u} \\ c_u^\nu & \text{otherwise,} \end{cases} \tag{5.14}$$

$$\rho_u(W_u^{N_u}, Y^\infty) = \begin{cases} \varepsilon_u(W_u^{N_u}, Y_{\eta_u+\nu+1}^{\infty})) & \text{if } \delta_\nu(Y_{\eta_u+1}^{\eta_u+\nu}, c_1^\nu, w_1^\nu, c_2^\nu, w_2^\nu) = 1, \\ (c_u^\nu, c_u^\nu, \cdots) & \text{otherwise,} \end{cases} \tag{5.15}$$

where Δ_1^{in} and Δ_2^{in} are the decoders associated with the inner encoders e_1^{in} and e_2^{in}, which are defined farther below, and δ_ν and the verification sequences $c_1^\nu, w_1^\nu, c_2^\nu$, and w_2^ν are defined as follows: The verification sequences are chosen according to

$$c_1^\nu = (\underbrace{x_1, \cdots, x_1}_{\nu_1 \overset{\triangle}{=} \lceil \lambda\nu \rceil}, \underbrace{z_1, \cdots, z_1}_{\nu_2 \overset{\triangle}{=} \nu - \lceil \lambda\nu \rceil}) \tag{5.16a}$$

$$w_1^\nu = (\underbrace{x_1', \cdots, x_1'}_{\nu_1}, \underbrace{z_1, \cdots, z_1}_{\nu_2}) \tag{5.16b}$$

$$c_2^\nu = (\underbrace{z_2, \cdots, z_2}_{\nu_1}, \underbrace{x_2, \cdots, x_2}_{\nu_2}) \tag{5.16c}$$

$$w_2^\nu = (\underbrace{z_2, \cdots, z_2}_{\nu_1}, \underbrace{x_2', \cdots, x_2'}_{\nu_2}), \tag{5.16d}$$

where

$$(x_1, x_1', z_2) = \underset{(x_1, x_1', z_2) \in \mathcal{X}_1^2 \times \mathcal{X}_2}{\arg\max} D(q_{Y|X_1,X_2}(\cdot|x_1, z_2) \parallel q_{Y|X_1,X_2}(\cdot|x_1', z_2)) \tag{5.17}$$

$$(x_2, x_2', z_1) = \underset{(x_2, x_2', z_1) \in \mathcal{X}_2^2 \times \mathcal{X}_1}{\arg\max} D(q_{Y|X_1,X_2}(\cdot|z_1, x_2) \parallel q_{Y|X_1,X_2}(\cdot|z_1, x_2')), \tag{5.18}$$

and λ satisfies

$$\lambda D(q_{Y|X_1,X_2}(\cdot|x_1, z_2) \parallel q_{Y|X_1,X_2}(\cdot|x_1', z_2)) = (1-\lambda) D(q_{Y|X_1,X_2}(\cdot|z_1, x_2) \parallel q_{Y|X_1,X_2}(\cdot|z_1, x_2')). \tag{5.19}$$

The detector δ_ν used by the receiver to determine whether either transmitter's data need be retransmitted is defined by

$$\delta_\nu(z^\nu, c_1^\nu, w_1^\nu, c_2^\nu, w_2^\nu) = \begin{cases} 1 & \text{if } \delta_{1,\nu_1}(z^{\nu_1}, c_1^{\nu_1}, w_1^{\nu_1}) = 1 \text{ or } \delta_{2,\nu_2}(z_{\nu_1+1}^\nu, c_{2,\nu_1+1}^\nu, q_{2,\nu_1+1}^\nu) = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{5.20}$$

where δ_{1,ν_1} is the function that would be used by an mSB decoder to distinguish $c_1^{\nu_1}$ and $w_1^{\nu_1}$ on the DMC $p_{Y|X_1,X_2}(\cdot|\cdot, z_2)$, returning 1 after detecting $w_1^{\nu_1}$ and 0 otherwise; δ_{2,ν_2} is the analogous function for the DMC $p_{Y|X_1,X_2}(\cdot|z_1, \cdot)$. Note that δ_{1,ν_1} and δ_{2,ν_2} are not maximum-likelihood detectors (see Section 2.3.1.3 and [65] for details).

Note that the stopping functions $\{\chi_i\}$ return 0 unless i is an integer multiple of $\eta_1 + \nu = \eta_2 + \nu$, and $\delta_\nu(Y_{i-\nu+1}^i, c_1^\nu, w_1^\nu, c_2^\nu, w_2^\nu) = 0$, i.e., no retransmission is needed.

Let us now define the inner codes used with ε_1 and ε_2 in terms of the number of bits $N_1 = h_\delta(N_2)$ and N_2 they each encode. The resulting lengths of the inner codes, η_1 and η_2, are functions of N_1 and N_2, respectively. We then define h_δ so that the constraint $\eta_1 = \eta_2$ is satisfied. We defer a complete definition of h_δ until after the definition of the inner encoders; assume for now only that $h_\delta(N_2) = n_1 k_1$, where k_1 is some positive integer, and

$$n_1 = \lceil \sqrt{N_2} \rceil. \tag{5.21}$$

Inner Encoder for Tx-1

To send its N_1-bit message, the encoder e_1^{in} for Tx-1 breaks this message into k_1 n_1-bit sub-messages. The sub-message size n_1 given by (5.21) is one of a number of adequate choices that 1) makes the delay that Tx-2 must wait to decode Tx-1's bits small and 2) makes the probability of decoding error for any one of these n_1-bit messages small.

Each sub-message is then sent using a low-complexity encoder designed for the single-user channel $p_{Y|X_1}$. While the code developed in Chapter 2 seems like a good choice for such a code, its output has a random length, which is incompatible with the overall coding scheme. For this reason, it is convenient to define a low-complexity, capacity-achieving coding scheme with *fixed overall length.*

We define an auxiliary encoding function e^{sub} that encodes n_1 message bits using a fixed number of channel inputs as follows: Let $\pi_{1,n} : \{0,1\}^n \to \mathcal{X}_1^{f_{1,\delta}(n)}$, where $f_{1,\delta}(n) = \lceil n/(H(X_1) - \delta) \rceil$

be a fixed-length precoder as described in Section A.9, and let $\sigma_{1,n} : \mathcal{X}_1^n \times \mathcal{Y}^n \to \{0,1\}^{g_{1,\delta}(n)}$, where $g_{1,\delta}(n) = \lceil n(H(X_1|Y) + \delta) \rceil$ be a fixed-length source coder, also described in Section A.9. The encoder e^{sub}, which is essentially the same as the single-user encoding scheme described in Section 2.7.1, is defined in terms of these subsystems by

$$\begin{aligned}
&\text{Initialization:} \\
&\qquad l_0^\sigma = n_1, l_0 = \lceil l_0^\sigma / (H(X_1) - \delta) \rceil, A_0 = 0 && (5.22) \\
&\qquad \Sigma_0 = W_1^{n_1}, \epsilon_0 = \pi_{1,l_0^\sigma}(r(\Sigma_0, 0)) && (5.23) \\
&\text{for } i = 1, \cdots, B_1\text{:} \\
&\qquad l_i^\sigma = \lceil (H(X_1|Y) + \delta) l_{i-1} \rceil && (5.24) \\
&\qquad l_i = \lceil l_i^\sigma / (H(X_1) - \delta) \rceil && (5.25) \\
&\qquad A_i = A_{i-1} + l_{i-1} && (5.26) \\
&\qquad \Sigma_i = \sigma_{1,l_{i-1}}(\epsilon_{i-1}, Y_{A_{i-1}+1}^{A_i}) && (5.27) \\
&\qquad \epsilon_i = \pi_{1,l_i^\sigma}(r(\Sigma_i, i)) && (5.28) \\
&\text{Number of channel inputs:} \\
&\qquad \eta^{\text{sub}} = A_{B_1} + \ell(\Sigma_{B_1}) \lceil n_1^{1/4} \rceil && (5.29) \\
&\text{Auxiliary encoding function:} \\
&\qquad e^{\text{sub}}(W_1^{n_1}, Y^\infty) = (\epsilon_0, \cdots, \epsilon_{B-1}, e^{\text{rep}}_{\lceil n_1^{1/4} \rceil}(\Sigma_{B_1})) && (5.30) \\
& && (5.31)
\end{aligned}$$

where r is the same binary randomization function used in Section 2.3, and B_1 is chosen as

$$B_1 = \left\lceil \frac{3 \log n_1}{4(\log(H(X_1) - \delta) - \log(H(X_1|Y) + \delta))} \right\rceil \tag{5.32}$$

so that $l_{B_1}^\sigma \approx {n_1}^{1/4}$; the encoding function e_k^{rep} codes its data by using a two-codeword codebook of length k to encode each of its input bits.

The inner encoder e_1^{in} is then defined in terms of e^{sub} as follows: With $W_1^{N_1} = (W_1^{n_1}[1], \cdots, W_1^{n_1}[k_1])$,

$$\begin{aligned}
e_1^{\text{in}}(W_1^{N_1}, Y^\infty) = (&e^{\text{sub}}(W_1^{N_1}[1], Y^\infty), e^{\text{sub}}(W_1^{N_1}[2], Y_{\eta^{\text{sub}}+1}^\infty), \\
&\cdots, e^{\text{sub}}(W_1^{N_1}[k_1], Y_{(k_1-1)\eta^{\text{sub}}+1}^\infty)).
\end{aligned} \tag{5.33}$$

Note that the total length η_1 is $k_1 \eta^{\text{sub}}$.

Inner Encoder for Tx-2

As suggested by the high-level view of the scheme above, Tx-2's inner encoding scheme is, in most respects, the same as the single-user coding scheme discussed in Chapter 2. The two major differences are the following: 1) Tx-2 decodes the feedback to estimate the data sent by Tx-1 with a delay of at most η^{sub} samples, and 2) Tx-2 performs its compression step using a statistical model of the dependence of $X_{2,k}$ on both Y_k and $X_{1,k}$.

Tx-2's fixed-length inner encoding scheme e_2^{in}, which codes N_2 bits, is defined as follows: Let $\pi_{2,n} : \{0,1\}^n \to \mathcal{X}_2^{f_{2,\delta}(n)}$, where $f_{2,\delta}(n) = \lceil n/(H(X_2) - \delta)\rceil$ be the fixed-length precoder described in Section A.9, and let $\sigma_{2,n} : \mathcal{X}_2 \times \mathcal{Y}^n \times \mathcal{X}_1^n \to \{0,1\}^{g_{2,\delta}(n)}$, where $g_{2,\delta}(n) = \lceil n(H(X_2|Y, X_1) + \delta)\rceil$ be a fixed-length source coder described in Appendix D. Then define e_2^{in} by

Initialization:

$$l_0^\sigma = N_2, l_0 = \lceil l_0^\sigma/(H(X_2) - \delta)\rceil, A_0 = 0, g_0 = 0 \tag{5.34}$$

$$\Sigma_0 = W_2^{N_2}, \varepsilon_0 = \pi_{2,l_0^\sigma}(r(\Sigma_0, 0)) \tag{5.35}$$

for $i = 1, \cdots, B_2$:

$$l_i^\sigma = \lceil (H(X_2|Y, X_1) + \delta)l_{i-1}\rceil \tag{5.36}$$

$$l_i = \lceil l_i^\sigma/(H(X_2) - \delta)\rceil \tag{5.37}$$

$$A_i' = A_{i-1} + l_{i-1} \tag{5.38}$$

$$A_i = \lceil A_i'/\eta^{\text{sub}}\rceil \eta^{\text{sub}} \tag{5.39}$$

$$g_i = g_{i-1} + A_i - A_i' \tag{5.40}$$

$$\hat{X}_{A_{i-1}+1}^{A_i'} = \tilde{\Delta}^{\text{sub}}(Y_{A_{i-1}+1}^{A_i}) \tag{5.41}$$

$$\Sigma_i = \sigma_{2,l_{i-1}}(\epsilon_{i-1}, Y_{A_{i-1}+1}^{A_i'}, \hat{X}_{A_{i-1}+1}^{A_i'}) \tag{5.42}$$

$$\epsilon_i = \pi_{2,l_i^\sigma}(r(\Sigma_i, i)) \tag{5.43}$$

Number of channel inputs:

$$\eta_2' = A_{B_2} + \ell(\Sigma_{B_2})\lceil N_2^{1/4}\rceil \tag{5.44}$$

Length of final filler:

$$F = \lceil \eta_2'/\eta^{\text{sub}}\rceil \eta^{\text{sub}} - \eta_2' \tag{5.45}$$

Inner encoding functions:

$$e_2^{\text{in}}(W_2^{N_2}, Y^\infty) = (\varepsilon_0, F_1^{g_1}, \varepsilon_1, F_{g_1+1}^{g_2}, \cdots, \varepsilon_{B_2-1}, F_{g_{B_2-1}+1}^{g_{B_2}}, e_{\lceil N_2^{1/4}\rceil}^{\text{rep}}(\Sigma_{B_2}), F_{g_{B_2}+1}^{g_{B_2}+F}). \tag{5.46}$$

The sequence $\{F_i\}_{i=1}^\infty$ used in (5.46) is filler used to satisfy the causality conditions that must hold for a feedback coding scheme; the filler sequence is transmitted whenever Tx-2 is waiting to decode Tx-1's channel inputs. The length F of the final filler is chosen so that η_2 is an integer multiple of η^{sub}. The function $\tilde{\Delta}^{\text{sub}}$ uses the decoder for e^{sub} to estimate Tx-1's channel inputs by decoding the sub-messages encoded by e^{sub} and then finding the corresponding input sequence. The number of iterations B_2 is chosen according to

$$B_2 = \frac{3 \log N_2}{4(\log(H(X_2) - \delta) - \log(H(X_2|Y, X_1) + \delta))}, \tag{5.47}$$

which makes the $\ell(\Sigma_{B_2}) \approx N_2^{1/4}$.

Choosing N_1 as a function of N_2

To complete the specification of the encoders for Tx-1 and Tx-2, we must specify the relationship between N_1 and N_2, i.e., h_δ. We can see that to constrain η_1 and η_2 to be equal, we must simply define h_δ by

$$h_\delta(N_2) = n_1\eta_2/\eta^{\text{sub}}. \tag{5.48}$$

(Note that the right-hand-side of this equation is indeed a function of only N_2, δ, and the channel parameters.)

5.4 Performance Analysis

5.4.1 Reliability

In this section, we analyze the reliability function of the coding scheme $c_{\text{MA},\delta}$. Specifically, we prove the following theorem:

Theorem 5.4.1 Let δ, $0 < \delta < I(Y;X_1)/2$, be given. Then for any $\alpha > 0$, the code sequence $\{c_{\text{MA},\delta}(N_2, \lceil \alpha N_2 \rceil)\}_{N_2=1}^{\infty}$ has rates converging to the rate pair $(r_1(\alpha), r_2(\alpha))$, where

$$r_1(\alpha) = \frac{I(Y;X_1) - 2\delta}{\alpha(I(Y;X_1) - 2\delta) + 1} \tag{5.49}$$

$$r_2(\alpha) = \frac{I(Y;X_2|X_1) - 2\delta}{\alpha(I(Y;X_2|X_1) - 2\delta) + 1}, \tag{5.50}$$

and attains the error exponent F_α defined by

$$F_\alpha = \left(1 - \frac{r_1(\alpha)}{I(Y;X_1) - 2\delta}\right) F_0 \tag{5.51}$$

$$= \left(1 - \frac{r_2(\alpha)}{I(Y;X_2|X_1) - 2\delta}\right) F_0, \tag{5.52}$$

where

$$F_0 = \lambda D(q_{Y|X_1,X_2}(\cdot|x_1, z_2) \parallel q_{Y|X_1,X_2}(\cdot|x_1', z_2)) \tag{5.53}$$

$$= (1-\lambda) D(q_{Y|X_1,X_2}(\cdot|z_1, x_2) \parallel q_{Y|X_1,X_2}(\cdot|z_1, x_2')). \tag{5.54}$$

Remark 1: As $\alpha \to 0$, both r_1 and r_2 approach $I(Y;X_1) - 2\delta$ and $I(Y;X_2|X_1) - 2\delta$, respectively, uniformly. It follows that a sequence of codes achieving the rate pair $(I(Y;X_1) - 2\delta, I(Y;X_2|X_1) - 2\delta)$ can be generated by $c_{\text{MA},\delta}$.

Remark 2: By changing the verification sequences in (5.16), the value of F_0 can be changed. We do not know, however, what the optimal choice of sequences is or what the optimal error exponent is.

Proof: We begin the proof with some useful observations. First, we note that prior to the transmission of the length-ν sequence, where $\nu = \lceil \alpha N_2 \rceil$, the channel inputs of Tx-1 and Tx-2 each form a process that is i.i.d. according to q_{X_1} and q_{X_2}, respectively. This fact follows immediately from the way the precoder is defined and from the use of the randomizing function r in (5.28).

Second, we argue as follows that with the encoders defined as above, the channel input streams are statistically independent: Consider the beginning of the transmission when both transmitters are on their first iterations. Both of these transmissions are easily seen to be independent, because they are derived from independent sources, and they are not functions of feedback. Suppose Tx-1 finishes its first iteration first and then moves to its next iteration. It source codes the data it sent on its previous iteration conditioned on the feedback. This data is certainly dependent on what Tx-2 sent during Tx-1's first iteration. But then Tx-1 XORs a pseudorandom sequence of bits to this data via the function r, which makes the data independent of Tx-2's channel inputs. When Tx-2 finishes its first iteration, because of the use of r again, the data Tx-2 sends on its second iteration is independent of what Tx-1 sends. A continuation of these arguments shows that the two streams are independent. Of course, the streams are only "independent" to the extent that the data generated by the relevant pseudorandom number generators appear to be independent. Also note that to ensure that this independence holds, we must generate the codewords for the repetition coders independently each time they are used. These two observations allow vast simplification of the analysis of what would otherwise be an enormously complicated problem and allow us to prove the theorem easily, which we do as follows.

Consider the code $c_{\text{MA},\delta}(N_2, \lceil \alpha N_2 \rceil)$. We can see that this code uses the same principles as does an mSB code, sending an inner code followed by a verification sequence. Consequently, the probability of error is controlled entirely by the probability P_{cw} that the detector δ_ν returns a 0, which indicates that no retransmission is required, when it should return a 1, which indicates that retransmission is required. Retransmission is required whenever either Tx-1 or Tx-2's inner codeword is decoded incorrectly.

The total expected length of the sequence is then determined by the lengths η_1 and η_2 of Tx-1 and Tx-2's inner codewords, the probability P_{wc} that δ_ν returns a 1 when it should return a 0, and the probability that either Tx-1 or Tx-2's inner codeword is decoded incorrectly.

The lengths of the inner codewords for both Tx-1 and Tx-2, which are equal, can be easily shown to satisfy

$$\begin{aligned} \eta_1 = \eta_2 &= N_2/(I(Y;X_2|X_1) - 2\delta) + o_{N_2}(N_2) && (5.55) \\ &= N_1/(I(Y;X_1) - 2\delta) + o_{N_2}(N_1). && (5.56) \end{aligned}$$

What is left is to find the probability that either is decoded incorrectly, which we upper bound (union bound) as the sum of the probability $P_{1,\text{in}}$ that Tx-1's inner codeword is decoded incorrectly plus the probability $P_{2,\text{in}}$ that Tx-2's inner codeword is decoded incorrectly.

To prove the theorem, it is sufficient to show that $P_{1,\text{in}}$, $P_{2,\text{in}}$, and P_{wc} are $o_{N_2}(1)$, and that $P_{cw} < \exp_2\{-\nu(F_0 - o_\nu(1))\}$ for some $F_0 > 0$, which we do as follows, starting with analysis of $P_{1,\text{in}}$.

To find $P_{1,\text{in}}$, the probability of error for the inner code, note that the probability of error P_{sub} for e^{sub} satisfies

$$\begin{aligned} P_{\text{sub}} &< \exp_2\{-(E^{\text{sub}}(\delta) - o_{n_1}(1))n_1^{1/4}\} && (5.57) \\ &= \exp_2\{-(E^{\text{sub}}(\delta) - o_{N_2}(1))N_2^{1/8}\}, && (5.58) \end{aligned}$$

for some constant $E^{\text{sub}}(\delta) > 0$, where (5.58) follows from (5.21). Since the inner code uses the fixed-length code $k_1 = N_1/n_1 = O_{N_2}(\sqrt{N_2})$ times, $P_{1,\text{in}}$ satisfies

$$P_{1,\text{in}} < k_1 P_{\text{sub}} = o_{N_2}(1). \tag{5.59}$$

We move now to analysis of $P_{2,\text{in}}$, which can be upper bounded according to

$$\begin{aligned} P_{2,\text{in}} &\leq \Pr\{\tilde{e}_2^{\text{in}}(W_2^{N_2}, Y^{\eta_2}) \text{ is decoded incorrectly or } e_2^{\text{in}}(W_2^{N_2}, Y^{\eta_2}) \neq \tilde{e}_2^{\text{in}}(W_2^{N_2}, Y^{\eta_2})\} && (5.60) \\ &\leq \Pr\{\tilde{e}_2^{\text{in}}(W_2^{N_2}, Y^{\eta_2}) \text{ is decoded incorrectly}\} + \Pr\{e_2^{\text{in}}(W_2^{N_2}, Y^{\eta_2}) \neq \tilde{e}_2^{\text{in}}(W_2^{N_2}, Y^{\eta_2})\}, && (5.61) \end{aligned}$$

where $\tilde{e}_2^{\text{in}}$ encodes just like e_2^{in} but with perfect estimation of Tx-1's transmissions substituted for the estimates in (5.41). The second term in (5.61) is equal to the probability that Tx-1's transmissions are wrongly estimated, which, using results from the previous section, is less than $P_{1,\text{in}}$, which, as we just saw, is $o_{N_2}(1)$. The first term in (5.61) decays exponentially in $N_2^{1/4}$. Therefore, $P_{2,\text{in}} = o_{N_2}(1)$.

We now consider P_{cw}. With the choice of verification sequences given in (5.16), it is clear that we can regard the verification sequences as really two verification sequences, of lengths ν_1 and ν_2, respectively. The first ν_1 samples are used by Tx-1 to communicate whether retransmission of its codeword is necessary, while the last ν_2 samples are used by Tx-2 to communicate whether retransmission of its codeword is necessary. The sequence $(z_2, \cdots, z_2)$ sent by Tx-2 during the first ν_1 samples is designed to give Tx-1 the best possible channel to the receiver. Likewise for Tx-1's transmissions during the last ν_2 samples. It can be shown using the reasoning from [65] that P_{cw} is upper bounded according to

$$\begin{aligned} P_{cw} < &\exp_2\{-\nu\lambda(D(q_{Y|X_1,X_2}(\cdot|x_1, z_2) \parallel q_{Y|X_1,X_2}(\cdot|x_1', z_2)) - o_{N_2}(1))\} && (5.62) \\ &+ \exp_2\{-\nu(1-\lambda)(D(q_{Y|X_1,X_2}(\cdot|x_1, z_2) \parallel q_{Y|X_1,X_2}(\cdot|x_1', z_2)) - o_{N_2}(1)\}. && (5.63) \end{aligned}$$

It can also be easily shown that the probability $P_{wc} = o_{N_2}(1)$.

Since the total error probability for the scheme P_{e,N_2} is equal to P_{cw}, and the expected length $E[L^*]$ is upper bounded by $\eta_2/(1 - P_{1,\text{in}} - P_{2,\text{in}} - P_{wc}) = N_2/I(Y;X_2|X_1) + \alpha N_2 + o_{N_2}(N_2)$, (5.52) follows by taking the limit of $-\log P_{\text{e},N_2}/E[L^*]$ as $N_2 \to \infty$; (5.51) follows by also using (5.56). □

5.4.2 Complexity

The scheme we have described has the same order of growth of time and space as the single-user scheme we described for DMC's in Chapter 2. That is, for both transmitters and the receiver, the growth in computation time and space required per message is uniformly linear in the number of channel inputs used per message.

5.5 Discussion

5.5.1 Achieving Other Rate Pairs

By adjusting the rate of Tx-1's inner sub-message code, it is clear that the line in rate-space from $(0, I(Y;X_2|X_1))$ to $(I(Y;X_1), I(Y;X_2|X_1))$ can be achieved. By exchanging the roles of Tx-1 and Tx-2, the line from $(0, I(Y;X_1|X_2))$ to $(I(Y;X_2), I(Y;X_1|X_2))$ can also be achieved. By timesharing, the entire frontier of the pentagon in Figure 5-1 can therefore be achieved. By further timesharing, all the points on the frontier of the capacity region can be reached.

5.5.2 Variations

A number of variations on the scheme presented in Section 5.3 can be made. For example, the receiver could use separate stopping functions for each transmitter as well as separate decoders; the two transmitters' verification sequences would not need to be synchronized, and the retransmissions of their data could be done independently. This modification would allow the transmitters to satisfy independent delay and error requirements. But the stopping times for the transmitters' messages would not coincide, the technique would only make sense if each transmitter had an infinite stream of messages to send. Within a scheme such as this one, we could also substitute variable-length encoders for e^{sub} and e_2^{in}, which might give better performance for a given value of N_2. Unfortunately, the use of a variable-length encoder is difficult to analyze, because the output lengths are statistically dependent on the other transmitter's channel inputs.

Other variations include changing the verification sequences. We chose the verification sequences according to (5.16) for ease of analysis.

5.5.3 Beyond the Feedback-Free Capacity Region

While we have no concrete ideas for how our framework can be used to achieve rates beyond the feedback-free capacity region, we have made some observations that suggest a new interpretation of why feedback extends the capacity region. The research of Cover et al. [12] on sending data generated by correlated sources over multiple-access channels shows that if two sources are correlated, then their data can be sent more efficiently over a DMMAC than if they are not correlated. That is, one can achieve rates higher than would be achieved by Slepian-Wolf coding the correlated sources and then transmitting them as though they were not correlated. There seems to be a relationship between this fact and the fact that the feedback capacity region of a DMMAC$_\text{f}$ is larger than its feedback-free capacity region. This relationship seems to come out in our framework. In our framework, the two transmitters first send independent data uncoded over the channel. On the second iteration, the data they must transmit is correlated; hence, the need for Slepian-Wolf coding as discussed earlier. If used as discussed earlier, this Slepian-Wolf coding leads to rate pairs on the frontier of the feedback-free capacity region. But the result of Cover et al. suggests that the data on the second iteration can be sent even more efficiently than by Slepian-Wolf coding followed by transmission as though the sources were independent. A technique that sends the data in this more efficient manner may enable one to reach points beyond the feedback-free capacity region. This observation seems worthy of further study.

5.5.4 Summary and Future Directions

We have developed a linear-complexity coding scheme for DMMAC$_f$'s that, with timesharing, allows any point on the frontier of the capacity region of a two-user DMMAC to be achieved. The probability of error for the scheme decays exponentially with the number of channel inputs used.

An interesting direction for future research stems from a practical problem. While the scheme described in Section 5.3 achieves good performance asymptotically, we suspect that in practice very large blocklengths may be required for reasonable error probabilities at rates near the pair $(I(Y;X_1), I(Y;X_2|X_1))$. For example, (5.21) dictates that if $n_1 = 10^4$, then $N_2 \approx 10^8$. But experiences with running simulations for single-user channels indicate that use of a message length of 10^4 on the 0 dB quantized Gaussian channel described in Section 2.8.6 leads to rates of only about 0.3 bits per channel input (well below the channel capacity of 0.5 bits per channel input) for only modestly small probabilities of error. Although deviating from the relationship (5.21) — for example, by setting $n_1 = 3 \times 10^4$ and $N_2 = 10^8$ — allows significantly better rates for Tx-1 to be achieved with little sacrifice in Tx-2's rate, very large values of N_2 are still required to achieve rate pairs near the frontier of the capacity region. Even larger blocklengths are required when the scheme is extended under the framework developed in this chapter to accommodate more than two users. Although larger blocklengths do not require faster processors because of the coding scheme's linear time complexity, storage capacity and delay requirements may limit the allowable blocklength. Consequently, an interesting and important direction for future research is to modify the scheme so as to require smaller blocklengths.

It is also interesting to consider how one can apply the iterative coding framework to other multiuser channels with feedback. Thus far, it is unclear how the framework can be applied to such channels as the interference channel or the broadcast channel. Investigations of such channels represents an interesting direction for future research.

Other interesting problems include dealing with multiple-access channels with memory, and unknown multiple-access channels. We believe that the universal coding techniques of Chapter 4 can be directly adapted to create a universal coding scheme for unknown DMMAC$_f$'s. However, proofs of good performance may be hard to obtain, because the universal coding scheme must use a variable-length source coding subsystem.

Chapter 6

Channels with Partial and Noisy Feedback

6.1 Introduction

In Chapters 2–5, we consider channels with feedback links that provide complete noiseless feedback. But in practice, some feedback links may not be accurately modeled as being noiseless. Some may also have insufficient capacity to provide complete feedback. While some research, for example [6] and [33], has been done on coding for channels with noisy and partial feedback, it has been sparse, and many questions remain unanswered.

In this chapter, we consider channels with noisy and partial feedback within the compressed-error-cancellation framework. We focus in particular on coding for channels with partial noiseless feedback, and discuss only briefly coding for channels with noisy feedback. We justify this choice as follows: If the feedback channel is noisy, one can apply a low-complexity error-correcting code to the data sent over it; if the feedback rate after coding is large enough to provide complete feedback, then we have, effectively, complete noiseless feedback; if the feedback rate is not large enough to provide complete feedback, then we have, effectively, partial noiseless feedback. We should, however, remember that there may be other ways of coping with noisy feedback besides applying a block error-correcting code to the feedback data, so noisy feedback channels may still warrant more attention than is given here.

We also focus on single-user channels, exploring only briefly the multiple-access channel with partial feedback. For simplicity, within the class of single-user channels, we consider only discrete memoryless forward channels, at times restricting further to BSC's. For the remainder of the chapter, we let $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$ be the forward DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition pmf $q_{Y|X}$. We let C be the capacity of this channel, achieved with input pmf q_X. For convenience, we also let X and Y be random variables such that $p_{X,Y}(x, y) = q_X(x) q_{Y|X}(y|x)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Except in Section 6.5, where we consider noisy feedback, the feedback channel is considered noiseless and delayless but with limited capacity, so that at each time, the receiver chooses some element of the feedback channel's input alphabet $\mathcal{F}$ to send back to the transmitter. The feedback rate that can be supported by the feedback channel is then $\log |\mathcal{F}|$ bits per channel output.

In what follows, we show that the compressed-error-cancellation framework is, with appropriate modifications, useful for developing coding schemes requiring only partial noiseless feedback. We show that the framework leads to coding schemes with a wide range of required feedback rates and provides a systematic way of integrating FEC codes with feedback.

We begin in Sections 6.2–6.4 by discussing how Slepian-Wolf coding and concatenated coding principles can be used to modify the coding scheme of Section 2.5 so that the required feedback rate is less than $H(Y)$ bits per forward channel output (which is the rate required by that coding scheme if source coding of the feedback data is performed). We then briefly explore in Section 6.5 what modifications can be made to the coding scheme of Section 2.5 so that noisy feedback can be accommodated. We end in Section 6.6 with a discussion of some interesting related topics and future directions.

6.2 Feedback Rate Reduction

In this section, we introduce the idea of using Slepian-Wolf coding for feedback rate reduction, describe and analyze a coding scheme for BSC's based on this idea, and then comment on the extent to which the scheme can be generalized for DMC's.

6.2.1 An Approach Based on Slepian-Wolf Coding

Let Y^n be the result of passing a sequence X^n that is i.i.d. with marginal pmf q_X through the DMC $q_{Y|X}$ without feedback. Suppose that the receiver and transmitter know only Y^n and X^n, respectively. Then the Slepian-Wolf coding theorem [55] states that it is possible for the receiver to use only about $nH(Y|X)$ bits to describe the sequence Y^n to the transmitter (since the transmitter knows X^n). This remarkable theorem thus suggests that it ought to be possible to reduce the required feedback rate of the scheme in Section 2.5 to $H(Y|X)$ bits per output sample.

But at least one obstacle hinders construction and decoding of such an efficient description of Y^n to the transmitter: high computational complexity. Since we are intent on preserving the low computational complexity of the scheme in Chapter 2, we demand that Slepian-Wolf coding be performed with low complexity. Unfortunately, no techniques for computationally efficient Slepian-Wolf coding at the minimum possible rate $H(Y|X)$ have yet been discovered.

It is, however, possible in some cases to perform Slepian-Wolf coding at rates substantially above $H(Y|X)$ with low complexity. Indeed, it is well known [64] that parity-check matrices of linear error-correcting codes can be used for Slepian-Wolf coding of two sequences differing by an independent additive (mod q) noise. In other words, a low-complexity linear error-correcting code may serve as a low-complexity Slepian-Wolf code in many cases.

To understand how linear error-correcting codes may be used as Slepian-Wolf codes, let Y^n and X^n differ by an additive (mod 2) i.i.d. Bernoulli-$(\delta - \alpha)$ noise process, where $\alpha < \delta$. Let K_n be the $(1 - r)n \times n$ parity-check matrix of a binary linear code of rate r that corrects up to a fraction δ of errors, i.e., δn errors. Then Y^n can be correctly determined with high probability from X^n and the length-$(1 - r)n$ sequence $K_n Y^n$ as follows: Compute $K_n Y^n - K_n X^n$; find a sequence Z^n of (Hamming) weight less than δn that satisfies $K_n Z^n = K_n(Y^n - X^n)$. Since the code corrects a fraction of errors δ, and because $Y^n - X^n$ has weight less than δn with high probability, $Z^n = Y^n - X^n$ with high probability.

But Slepian-Wolf coding with an arbitrary linear code with the above properties does not allow the uniform linear complexity we desire. The computation required for Slepian-Wolf encoding is that required to compute $K_n Y^n$, and the computation required for Slepian-Wolf decoding is that required to compute $K_n X^n$ plus that required to find Z^n. Mere computation of $K_n Y^n$ is an $O_n(n^2)$ operation for a general linear code. We must instead use a special class of linear codes for which matrix multiplication by K_n and estimating $(Y^n - X^n)$ from $K_n(Y^n - X^n)$ can be done with $O_n(n)$ computations.

One method appropriate for additive (mod-2) channels is to let the K_n be the parity-check matrix of an expander code [54]. The important properties of these codes are as follows [54]:

1. They are a subset of low-density parity check codes, which were introduced by Gallager [24]. Low-density parity check codes are linear codes and have the property that $K_n Y^n$ can be computed in $O_n(n)$ time.

2. (Theorem 19 from [54]) For all ϵ such that $1 - 2H_2(\epsilon) > 0$, where $H_2(\cdot)$ is the binary entropy function, there exists a polynomial-time constructible family of expander codes of rate $1 - 2H_2(\epsilon)$ and minimum relative distance arbitrarily close to ϵ^2, in which any $\alpha < \epsilon^2/48$ fraction of errors can be corrected in linear time on a random-access machine under the uniform cost model.

3. A sequence Z^n can be recovered from $K_n Z^n$ in $O_n(n)$ time if the weight of Z^n is below αn.

While these properties of expander codes make them suitable for linear-complexity Slepian-Wolf encoding and decoding, they have only limited applicability for several reasons. First, expander codes are designed to cope only with binary symmetric channels (BSC's), not general DMC's. Second, it is not known how to construct expander codes that correct an arbitrarily large fraction ($< \frac{1}{2}$) of errors; rather, Theorem 19 of [54] only guarantees the existence of expander codes with positive rate correcting a fraction $0.11^2/48$ of errors (since $H_2(0.11) \approx 1/2$). Therefore, the use of expander codes to carry out Slepian-Wolf coding of the feedback is limited to situations in which the forward channel is a BSC with crossover probability less than $0.11^2/48$.

Still, this Slepian-Wolf coding scheme allows us, in the following section, to demonstrate the general concept of Slepian-Wolf coded feedback and to show that uniform linear complexity can be maintained with a feedback rate smaller than $H(Y)$ bits per channel output.

6.2.2 A Reduced-Feedback-Rate Coding Scheme for BSC's

In this section, we describe how we modify the transmission scheme of Chapter 2 to use low-complexity expander-code based Slepian-Wolf coding for the feedback. Because of the limitations of expander codes, we assume for the remainder of this section that $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$ is a BSC with crossover probability $\epsilon < (0.11)^2/48$. For general DMC's, we discuss in Section 6.2.4 how the same principles would apply were an appropriate low-complexity Slepian-Wolf coding scheme available.

To avoid repeating most of the details of the scheme in Chapter 2, we describe the modifications at a high level. As the base on which we make modifications, we use the *base code* $c_{\mathrm{DMC}}(N, \nu)$, which is a complete noiseless feedback code from Section 2.3, including the modifications from Section 2.5.

The main modification to $c_{\mathrm{DMC}}(N, \nu)$ is that the receiver uses Slepian-Wolf coding to send feedback to the transmitter in blocks of n_{f} samples using the parity check matrix $K_{n_{\mathrm{f}}} \in \{0,1\}^{(1-r)n_{\mathrm{f}} \times n_{\mathrm{f}}}$ of an expander code of rate r, where $r < 1 - 2H_2(\sqrt{48\epsilon})$, which corrects $\alpha n_{\mathrm{f}} > \epsilon n_{\mathrm{f}}$ errors. If $Y^{n_{\mathrm{f}}}$ is an n_{f}-sample block of received channel outputs to be fed back to the transmitter, we refer to the length-$((1 - r)n_{\mathrm{f}})$ sequence $K_{n_{\mathrm{f}}} Y^{n_{\mathrm{f}}}$ as Slepian-Wolf coded feedback.

The Slepian-Wolf coded feedback introduces two new problems, feedback errors and feedback delay, the seriousness of which are determined by the choice of Slepian-Wolf coding blocklength n_f. Feedback errors arise because Slepian-Wolf coding is not lossless — i.e., Y^{n_f} can be incorrectly estimated by the transmitter, an event we call a Slepian-Wolf decoding error, if X^{n_f} and Y^{n_f} do not behave typically, where X^{n_f} is the block of channel inputs that gave rise to Y^{n_f}. Feedback delay arises for three reasons: 1) the time required to transmit k bits over the noiseless partial feedback channel is $\lceil k/\log|\mathcal{F}|\rceil$ channel input (or output) samples; 2) the transmission of Slepian-Wolf coded feedback over the feedback channel cannot begin until the receiver obtains the entire block Y^{n_f} of channel outputs; and 3) the Slepian-Wolf coded feedback cannot be decoded by the transmitter until it receives the entire block $K_{n_\mathrm{f}}Y^{n_\mathrm{f}}$. If we assume instantaneous Slepian-Wolf encoding and decoding, the total delay is $\lceil(1-r)n_\mathrm{f}/\log|\mathcal{F}|\rceil$ channel input samples; for simplicity, we assume the delay to be exactly linear in n_f and let Δ denote the constant of proportionality relating delay to number of bits of feedback.

Note that there is a tradeoff between feedback delay and the rate of feedback errors. As n_f increases, feedback delay increases, but the rate of feedback errors decreases. As n_f decreases, the reverse happens. As a result, additional modifications to the base scheme are needed to cope adequately with both feedback errors and feedback delay. We next describe these modifications as well as how we choose n_f.

We set

$$n_\mathrm{f} = \lceil\sqrt{N}\rceil \tag{6.1}$$

for the following reason. In the compressed-error-cancellation framework, the transmitter communicates a message by sending the message followed by a series of correction messages. Feedback delay may require the transmitter to wait between iterations while it receives the feedback needed to construct the next correction message. We restrict attention to the case in which the feedback rate is sufficiently large so that the start of each iteration beyond the first one, of which there are $O_N(\log^2 N)$, is delayed by at most n_f samples, during which the transmitter sends information-free filler. (We determine in Section 6.2.3.2 more precisely what constitutes a "sufficiently large" feedback rate.) Thus, (6.1) makes the number of channel inputs consumed by information-free filler an asymptotically negligible $O_N(\sqrt{N}\log^2 N)$.

But we pay for choosing $n_\mathrm{f} = O_N(\sqrt{N})$ with an elevated rate of Slepian-Wolf decoding errors. Such errors occur when the forward BSC makes more than αn_f crossovers within an n_f-sample block. If the forward BSC's crossover probability ϵ is less than α, then the probability $P_{\mathrm{e,SW}}$ of such an event is upper bounded [21] according to

$$\begin{aligned} P_{\mathrm{e,SW}} &\le 2^{-n_\mathrm{f}D_2(\alpha\|\epsilon)} && (6.2)\\ &= 2^{-\sqrt{N}D_2(\alpha\|\epsilon)} && (6.3)\end{aligned}$$

where D_2 is the binary Kullback-Leibler distance defined by

$$D_2(\alpha \parallel \epsilon) = \alpha\log\frac{\alpha}{\epsilon} + (1-\alpha)\log\frac{1-\alpha}{1-\epsilon}, \tag{6.4}$$

which is always non-negative.

While $P_{\mathrm{e,SW}} \to 0$ as $N \to 0$, it does not do so exponentially with N. Were we to declare an

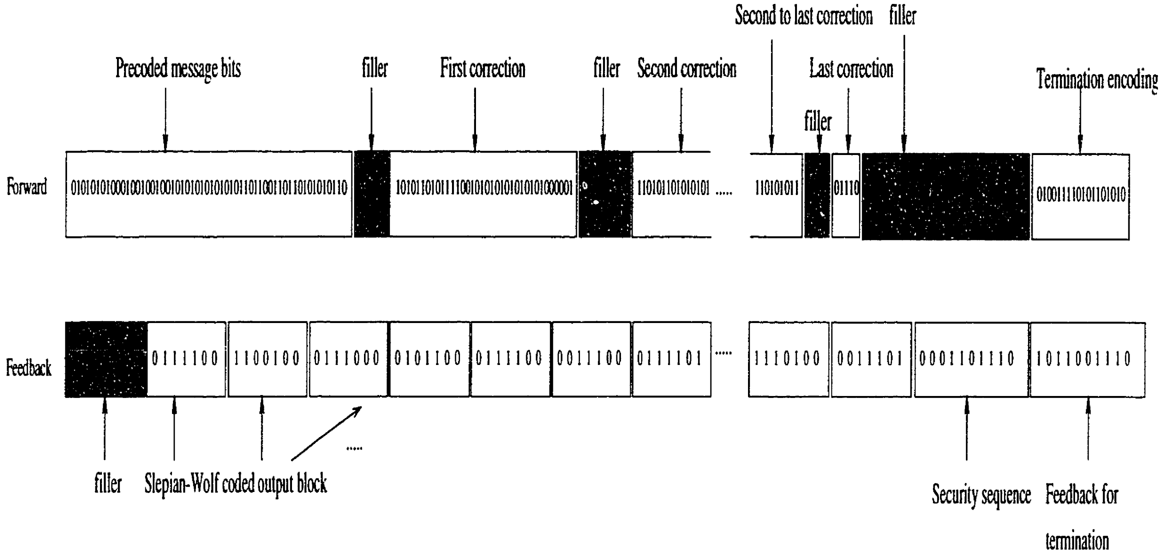


Figure 6-1: Graphical representation of the main elements of a coding scheme based on Slepian-Wolf coded feedback.

overall decoding error whenever a Slepian-Wolf decoding error occurred, the probability of error associated with the overall scheme would not decay exponentially with N as we desire.

To achieve an overall error probability that does decay exponentially with N, we install the following additional security mechanism: First, after each block of t_N samples is received, the receiver sends one bit back to the transmitter indicating whether or not it has detected the synchronization sequence. After the synchronization sequence is first detected by the receiver at, say, time $S + t_N$, the receiver sends to the transmitter a parity-check sequence of length proportional to S. This sequence enables the transmitter to determine whether it made any Slepian-Wolf decoding errors on the first S channel outputs. If it has — and the synchronization sequence detection is not a false alarm — then the transmitter restarts the transmission process from the beginning, first sending a special mSB coded message to the receiver indicating its intention. If the synchronization sequence detection is a false alarm, the original N message bits are simply mSB coded as described in Section 2.5. To be more precise, let X^S be the first S symbols that are put in to the forward channel. Let Y^S be the S corresponding receptions, and let $\hat{Y}^S$ be the transmitter's estimate of Y^S based on the Slepian-Wolf coded feedback. Let $\tilde{K}_S \in \{0,1\}^{2H_2(\sqrt{\delta})S \times S}$ be the parity-check matrix of a length-S binary expander code with minimum distance δS, where $\delta > P_{\text{e,SW}}$. Then the receiver sends the sequence $\tilde{K}_S Y^S$ over the feedback channel. The transmitter computes $\tilde{K}_S \hat{Y}^S$ and checks whether $\tilde{K}_S(Y^S - \hat{Y}^S) = 0$. If not, it initiates a retransmission.

Let us now summarize our modifications to the base code, the main elements of which are depicted graphically in Figure 6-1:

1. The receiver gives feedback as follows: After receiving its ith block of n_f outputs $Y^{n_\text{f}}[i]$, it forms $K_{n_\text{f}} Y^{n_\text{f}}[i]$ and sends it over the feedback channel; this action is performed for $i = 1, 2, \cdots$, until the receiver detects the synchronization sequence $\emptyset^{t_N}[N]$.

2. The transmitter decodes the ith block of feedback by forming $K_{n_\mathrm{f}} X^{n_\mathrm{f}}[i]$, where $X^{n_\mathrm{f}}[i]$ is the ith block of n_f channel inputs, computing $K_{n_\mathrm{f}}(Y^{n_\mathrm{f}}[i] - X^{n_\mathrm{f}}[i])$ and then finding an estimate $\tilde{Z}^{n_\mathrm{f}}[i]$ of $Y^{n_\mathrm{f}}[i] - X^{n_\mathrm{f}}[i]$.

3. Until it sends the synchronization sequence $\emptyset^{t_N}[N]$, the transmitter puts out the same channel inputs as the base code with one difference: since it now must wait a time proportional to n_f for the feedback, it sends a filler sequence during this waiting time.

4. Every t_N samples, the receiver also sends back a single bit that communicates whether or not the synchronization sequence has been detected.

5. Once the synchronization sequence is detected at time $S+t_N$, the receiver sends the sequence $\tilde{K}_S Y^S$ to the transmitter. The transmitter idly sends filler data during this time. The transmitter computes $\tilde{K}_S Y^S$ and compares this sequence with $\tilde{K}_S \hat{Y}^S$. If the two sequences differ, then the transmitter sends an mSB-coded message indicating that a retransmission must occur and then begins a retransmission, starting from the first iteration — unless the synchronization sequence detection is a false alarm, in which case it sends the original N bits mSB coded as described in Section 2.5.

6. If no retransmission is required, then the normal terminating procedure involving mSB coding of the final data and side information is begun.

7. During all mSB coding phases, decision feedback is given for both the mSB inner codeword, and the verification sequence. Decision feedback means that the decoded data is fed back rather than the raw received data; e.g., decision feedback for the verification sequence should only be 1 bit.

6.2.3 Performance Analysis of the Modified Scheme

We now determine bounds on the error exponent associated with the modified scheme and find that the error exponent is positive for all rates below capacity. We also calculate the feedback rate needed to carry out the scheme and comment on the overall complexity of the scheme.

6.2.3.1 Reliability

To bound the error exponent for the modified scheme, we first upper bound the probability of error as the sum of the probability of error associated with the base scheme and the probability of new error events introduced by the modifications. We next lower bound the rate of the modified scheme in terms of the rate of the base scheme. We then combine these bounds on rate and probability of error into a lower bound on error exponent that is positive at all rates below capacity.

Upper Bound on Probability of Error

We begin by assessing the effect of the modifications on the error probability. Compared to the base scheme, the modified scheme has one additional source of error: the possibility that a Slepian-Wolf decoding error goes undetected (an "undetected feedback error"). To upper bound the probability of error for the modified scheme, we assume that when an undetected feedback error occurs, the received data are unraveled incorrectly, and an overall decoding error occurs. An undetected feedback error can only occur in the unlikely event that so many of the Slepian-Wolf coded feedback blocks are decoded with errors that the final security mechanism is foiled.

To upper bound the probability P_{ufe} of an undetected feedback error, note that only if $\hat{Y}^S$ differs from Y^S in at least δS places is there any possibility that $\tilde{K}_S\hat{Y}^S = \tilde{K}_S Y^S$ when $Y^S \neq \hat{Y}^S$. Moreover, only if $\delta S/n_{\text{f}}$ of the S/n_{f} blocks sent are decoded erroneously can $\hat{Y}^S$ differ from Y^S in δS or more places. Therefore, P_{ufe} can be upper bounded as follows:

$$P_{\text{ufe}} < \exp_2\left\{-\frac{S}{n_{\text{f}}}(\delta \log(1/P_{\text{e,sw}}) + (1-\delta)\log(1/(1-P_{\text{e,sw}})) - H_2(\delta))\right\} \tag{6.5}$$

$$< \exp_2\left\{-S(\delta D_2(\alpha \parallel \epsilon) + o_N(1))\right\}, \tag{6.6}$$

where (6.6) follows from using (6.3) with (6.5). Since S is always greater than N/μ, where[1]

$$\mu = -\log(\min\{q_X(x) : q_X(x) \neq 0, x \in \mathcal{X}\}), \tag{6.7}$$

we can further upper bound the error in terms of N according to

$$P_{\text{ufe}} < \exp_2\{-N(\delta D_2(\alpha \parallel \epsilon)/\mu + o_N(1))\}. \tag{6.8}$$

With this upper bound on the probability of undetected feedback error, we can upper (union) bound the probability of error $P_{\text{e,m}}$ for the overall modified scheme as follows:

$$P_{\text{e,m}} < P_{\text{e,b}} + P_{\text{ufe}}, \tag{6.9}$$

where $P_{\text{e,b}}$ is the probability of error associated with the base code. With R_{b} denoting the rate of the base code, we set

$$\delta = \min\left(\left(\frac{1}{R_{\text{b}}} - \frac{1}{I(X;Y)}\right)\frac{E_{cw}\mu}{D_2(\alpha \parallel \epsilon)}, 0.11^2\right), \tag{6.10}$$

where E_{cw} is given by (2.38). This choice of δ makes the exponents for upper bounds on the two sources of error in (6.9) asymptotically equal for sufficiently small values of $(1/R_{\text{b}} - 1/I(X;Y))$ and otherwise minimizes the upper bound on undetected feedback error, since δ can be chosen to be at most 0.11^2. For future reference, let R_{b}^* be the largest value of R_{b} for which $\delta = 0.11^2$.

Lower Bound on Rate

Since (6.9) with (6.8) and (6.10) gives us an upper bound on $P_{\text{e,m}}$ in terms of the rate R_{b} of the base scheme, we need only find the rate of the modified scheme in terms of R_{b} to lower bound the error exponent associated with the modified scheme. We do this by examining the expected length of the modified transmission, which is increased by two effects: First, while the sequence $\tilde{K}_S Y^S$ is transmitted from receiver to transmitter, the transmitter sends filler data (it could begin its next message, but for convenience, we assume it sends filler), which consumes a number of channel inputs proportional to $2H_2(\sqrt{\delta})S$. Second, the possibility of needing to restart and retransmit after

[1] Actually, a false alarm could cause this inequality to not hold, but we can just change the receiver so that hypothesis testing for the synchronization sequence does not even begin until after N/μ samples have been received.

a feedback error is detected increases the expected length. Fortunately, this addition to the total expected length decreases to 0 as N approaches infinity. There may seem to be a third possibility for increasing the length, but it is an illusion. One might think that if a Slepian-Wolf decoding error occurred then the transmitter's next correction message might be unusually long because it sees an atypical channel output. Actually, the opposite is true. The Slepian-Wolf coding causes atypical noises to be decoded as typical noises, resulting in a typical-length correction message. This fact, combined with the fact that Slepian-Wolf decoding errors are rare allows us to ignore this effect.

Accounting for these increases in length, we write the expected total length $E[L^*]$ for the modified scheme in terms of the expected length N/R_b of the base scheme as

$$E[L^*] \leq \frac{N}{R_\mathrm{b}} + 2H_2(\sqrt{\delta})\Delta\frac{N}{I(X;Y)} + \frac{P_\mathrm{r}}{1-P_\mathrm{r}}\frac{N}{I(X;Y)} + o_N(N). \tag{6.11}$$

The second term accounts for the transmitter's waiting for the parity-check sequence $\tilde{K}_S Y^S$, since $E[S] < N/I(X;Y) + o_N(N)$; the third term accounts for the retransmissions that occur as a result of Slepian-Wolf decoding errors, where P_r is the probability of such a retransmission and satisfies

$$P_\mathrm{r} < P_{\mathrm{e,SW}} E[S]/n_\mathrm{f} \tag{6.12}$$

by the union bound; the $o_N(N)$ term includes the length of the filler transmitted during the between-iteration delays. Using (6.3) with (6.12), we can see that

$$\frac{N}{I(X;Y)}\frac{P_\mathrm{r}}{1-P_\mathrm{r}} = o_N(1), \tag{6.13}$$

which allows us to write

$$E[L^*] \leq \frac{N}{R_\mathrm{b}} + 2H_2(\sqrt{\delta})\Delta\frac{N}{I(X;Y)} + o_N(N). \tag{6.14}$$

To write the rate $R = N/E[L^*]$ of the modified scheme in terms of R_b, we let

$$\gamma = \frac{1}{2H_2(\sqrt{\delta})\Delta}, \tag{6.15}$$

and use (6.14) to write

$$R \geq \gamma R_\mathrm{b} I(X;Y)/(R_\mathrm{b} + \gamma I(X;Y)) + o_N(1), \tag{6.16}$$

which can also be expressed as

$$R_\mathrm{b} \leq \frac{RI(X;Y)\gamma}{I(X;Y)\gamma - R} + o_N(1). \tag{6.17}$$

Lower Bound on Error Exponent

We can now combine the bounds on rate and error probability to find a lower bound on the error exponent. Since, for $R_\mathrm{b} > R_\mathrm{b}^*$, (6.10) and (6.8) together say that

$$P_{\mathrm{e,m}} < \exp_2\left\{ -\frac{N}{R_\mathrm{b}} E_{cw}\left(1 - \frac{R_\mathrm{b}}{I(X;Y)}\right) + o_N(N) \right\}, \tag{6.18}$$

we can substitute (6.17) into (6.18) to obtain

$$E_\mathrm{m}(R) \triangleq \liminf_{N\to\infty} \frac{-\log P_{\mathrm{e,m}}}{E[L^*]} > E_{cw}\left(1 - \frac{R\gamma}{I(X;Y)\gamma - R}\right) \frac{I(X;Y)\gamma - R}{I(X;Y)\gamma}, \tag{6.19}$$

which holds for R corresponding to $R_\mathrm{b} > R_\mathrm{b}^*$. If we examine (6.17), we see that as $R_\mathrm{b} \uparrow I(X;Y)$, $\delta \to 0$, $\gamma \to \infty$, and $R \to R_\mathrm{b}$. Furthermore, from (6.18), we see that the error exponent is positive for any rate $R_\mathrm{b} < I(X;Y)$. There must therefore be rates arbitrarily close to $I(X;Y)$ for which the error decreases exponentially with N. For $R_\mathrm{b} \leq R_\mathrm{b}^*$, the probability of undetected feedback errors dominates the error behavior, and a lower bound on the error exponent can be determined from (6.8) and (6.17). This bound is positive for any fixed positive rate (although it is approaches zero as the rate approaches zero). Hence, the error exponent for the overall modified scheme is positive for any fixed rate below $I(X;Y)$.

6.2.3.2 Feedback Rate

Let us now examine the feedback rate required to carry out the scheme with the above reliability. That is, if the feedback channel carries $R_\mathrm{f} \triangleq \log|\mathcal{F}|$ bits of information per forward channel input (or, equivalently, output), what is the minimum value of R_f that will support the scheme described above?

The feedback rate R_f causes delay in the transmitter's acquisition of the feedback data. As mentioned above, the foregoing analysis was conditioned on R_f being large enough for us to assume that the delay between iterations was proportional to n_f. To satisfy this condition, it is sufficient that R_f be such that $K_{n_\mathrm{f}} Y^{n_\mathrm{f}}$ samples can be fed back in a time less than or equal to the time taken to input n_f symbols to the forward channel. Since $K_{n_\mathrm{f}} Y^{n_\mathrm{f}}$ has length $(1-r)n_\mathrm{f}$, we see that we must have $R_\mathrm{f} \geq 1 - r$.

Let us now verify that $R_\mathrm{f} = 1 - r$ is sufficient to support the other aspects of the scheme described above. First let us argue that this value of R_f supports the feedback required for the mSB coding stage. Modified Schalkwijk-Barron coding consists of two stages, the transmission of the inner code and the transmission of the one-bit verification message. During transmission of the inner code, we give decision feedback, which requires $\kappa_N = \sqrt{N}$ bits. No matter what the value of R_f (as long as $R_\mathrm{f} > 0$), a delay of $O_N(\sqrt{N})$ samples is introduced before the verification message can be sent; this delay is asymptotically negligible. During transmission of the verification message, no feedback is required, except at the end, at which point only one bit of feedback is needed to tell the transmitter whether a retransmission is expected. This feedback introduces a constant delay, which is also asymptotically negligible. Likewise, the one-bit messages required to tell the transmitter whether the synchronization sequence was detected adds negligible load to the feedback. Finally, for any $R_\mathrm{f} > 0$, the delay required for the transmitter to receive the sequence $\tilde{K}_S Y^S$ is proportional to S as assumed in the form of the constant Δ above.

These arguments imply that a feedback rate of $R_\mathrm{f} = 1 - r > 2H_2(\sqrt{48\epsilon})$ is sufficient to give exponentially decaying probability of error at any rate below $I(X;Y)$. This function is plotted on

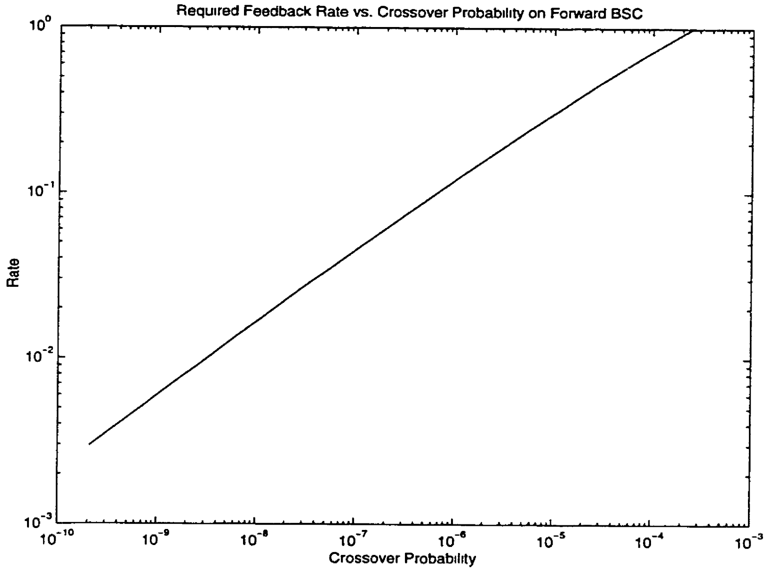


Figure 6-2: Plot of required feedback rate versus crossover probability on the forward BSC when using expander codes for Slepian-Wolf coded feedback.

a log-log scale in Figure 6-2. As is evident from this figure, the required feedback rate is less than one bit per channel output only when the crossover probability is less than about 2.5×10^{-4}. It follows that only BSC's with small crossover probabilities can benefit from expander-code based Slepian-Wolf coded feedback. Also note that larger values of R_f can improve the error exponent by reducing Δ.

6.2.3.3 Complexity

Because we have designed the Slepian-Wolf coding system to have linear encoding and decoding complexities that are essentially independent of the transmission rate, the overall complexity remains the same as for the scheme in Chapter 2, namely, uniformly linear in N. The same arguments in Section 2.8.1 apply when delays due to computation are taken into account.

6.2.4 Comments on Slepian-Wolf Coded Feedback for General DMC's

For general DMC's, the essential principles developed in Section 6.2.2 can still be applied to develop reduced-feedback-rate coding schemes. Similar rate and reliability performance — i.e., error probabilities decaying exponentially with average blocklength at all rates below capacity — can still be obtained. However, the complexity of the scheme depends on the complexity of the Slepian-Wolf coding scheme used to code the feedback. While we have found no low-complexity Slepian-Wolf coding schemes appropriate for general DMC's, there are special cases for which appropriate low-complexity schemes do exist; one such case is discussed in Section 6.4.

It is useful, therefore, to consider how we can use an appropriate given Slepian-Wolf coding scheme to construct a reduced-rate-feedback coding scheme for a general DMC. Suppose we have a Slepian-Wolf encoder $e^{\mathrm{SW}} : \mathcal{Y}^{n_{\mathrm{f}}} \rightarrow \{0,1\}^{R_{\mathrm{f}} n_{\mathrm{f}}}$ with corresponding decoder $d^{\mathrm{SW}} : \mathcal{X}^{n_{\mathrm{f}}} \times \{0,1\}^{R_{\mathrm{f}} n_{\mathrm{f}}}$ with the following property: with $(X_1, \cdots, X_{n_{\mathrm{f}}})$ and $(Y_1, \cdots, Y_{n_{\mathrm{f}}})$ distributed according to $p_{X^{n_{\mathrm{f}}}, Y^{n_{\mathrm{f}}}}(x^{n_{\mathrm{f}}}, y^{n_{\mathrm{f}}}) = \prod_{i=1}^{n_{\mathrm{f}}} q_X(x_i) q_{Y|X}(y_i|x_i)$,

$$P_{\mathrm{e,SW}} \triangleq \Pr\{d^{\mathrm{SW}}(X^{n_{\mathrm{f}}}, e^{\mathrm{SW}}(Y^{n_{\mathrm{f}}})) \neq Y^{n_{\mathrm{f}}}\} \leq \exp_2\{-n_{\mathrm{f}} E_{\mathrm{SW}} + o_{n_{\mathrm{f}}}(n_{\mathrm{f}})\}. \tag{6.20}$$

Then for each block of n_{f} output samples $Y^{n_{\mathrm{f}}}[i]$, the receiver feeds back $e^{\mathrm{SW}}(Y^{n_{\mathrm{f}}}[i])$. The transmitter decodes these data using d^{SW}.

The security mechanism that checks for Slepian-Wolf decoding errors can be implemented using expander codes as follows: Let b be any invertible mapping from $\mathcal{Y}$ to $\{0,1\}^v$, where $v = \lceil \log |\mathcal{Y}| \rceil$. Suppose that S channel inputs have been transmitted as above before the synchronization sequence. Let $\tilde{K}_{Sv}$ be a parity-check matrix of an expander code with minimum relative distance $\delta > P_{\mathrm{e,SW}}$. With

$$B^{Sv} = (b(Y_1), b(Y_2), \cdots, b(Y_S)), \tag{6.21}$$

the sequence $\tilde{K}_{Sv} B^{Sv}$ is sent as a check sequence. If the transmitter has made an estimate $\hat{Y}^S$ of the feedback, then it compares $\tilde{K}_{Sv} B^{Sv}$ with $\tilde{K}_{Sv} \hat{B}^{Sv}$, where

$$\hat{B}^{Sv} = (b(\hat{Y}_1), b(\hat{Y}_2), \cdots, b(\hat{Y}_S)), \tag{6.22}$$

initiating retransmission if the two terms are not equal. The probability of an undetected feedback error can easily be shown to be less than

$$\exp_2\{-(S/n_{\mathrm{f}}) D_2(\delta \parallel P_{\mathrm{e,SW}})\} < \exp_2\{-S(\delta E_{\mathrm{SW}} + o_N(1))\} \tag{6.23}$$

which is less than

$$\exp_2\{-N(\delta E_{\mathrm{SW}}/\mu + o_N(1))\}, \tag{6.24}$$

where μ is defined as in (6.7). This expression parallels (6.8).

Choosing δ in a way analogous to (6.10), the resulting coding scheme can be shown to have error probability decaying exponentially in N for all rates below $I(X;Y)$.

6.3 A Concatenated Coding Framework for Combining FEC Codes with Feedback Codes

While the scheme we describe in Section 6.2.2 requires feedback rate $2H_2(\sqrt{48\epsilon})$ bits per channel output, which can be substantially less than the one bit per channel output required by the scheme of Chapter 2, a scheme needing even lower feedback rate would, of course, be more desirable. That pure FEC codes achieve capacity with no feedback at all (albeit with higher complexity) suggests that integrating FEC codes with feedback codes could lead to feedback coding schemes requiring lower feedback rates. We develop a framework in this section for systematically integrating FEC codes with feedback codes, which results in a class of coding schemes capable of operating at any of a wide variety of feedback rates.

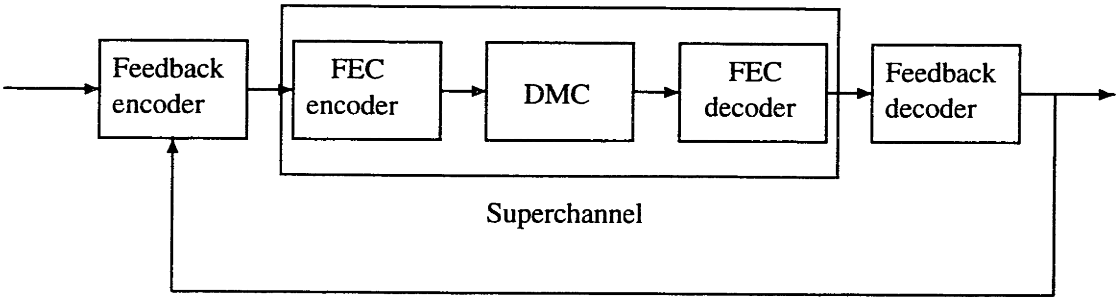


Figure 6-3: Illustration of concatenated coding with a feedback outer code.

6.3.1 Main Idea

The inspiration and main idea of the framework are as follows: For a forward BSC with crossover probability ϵ, the Slepian-Wolf coded feedback (SWCF) coding scheme described in Section 6.2.2 uses feedback rate $2H_2(\sqrt{48\epsilon})$, and the Slepian-Wolf coding scheme suggests that $H_2(\epsilon)$ bits per output sample is the lowest feedback rate possible under an SWCF strategy. Therefore, for small ϵ, the feedback rate decreases as ϵ decreases, suggesting that feedback rate can be lowered if ϵ can be lowered. But if ϵ is a fixed parameter of the forward channel, how can ϵ be lowered? One solution is to transform the channel into a channel with effectively smaller ϵ by applying an error-correcting code, which we call the *inner code*. After the channel is transformed, the SWCF scheme, which we call the *feedback outer code*, can then be applied. This solution is graphically represented in Figure 6-3. When FEC codes rather than feedback codes are used as outer codes, the family of codes known as concatenated codes, which were studied by Forney in [21], result. For this reason, we call the feedback schemes we develop based on this framework *concatenated SWCF* (CSWCF) coding schemes.

A precise description of CSWCF coding is most easily given using the concept of a superchannel [21]. A superchannel is induced when a block of symbols is considered as a single symbol. Of particular interest is the superchannel induced by the inner FEC block code and its decoder. To be precise, let $\mathcal{C}$ be a length-n, rate-r inner block code, and d its decoder. Let $x^n \in \mathcal{C}$ be sent over a DMC called the *original channel*, and let the original channel put out the n symbols Y^n in response to x^n. Then we consider x^n to be a single input to the superchannel and $d(Y^n) \in \mathcal{C}$ to be the corresponding (single) superchannel output. The probability distribution

$$\Pr\{d(Y^n) = \hat{x}^n \mid X^n = x^n\}, \tag{6.25}$$

for all $x^n \in \mathcal{C}$ and $\hat{x}^n \in \mathcal{C}$ then describes the superchannel, which is itself a DMC. The set of codewords $\mathcal{C}$ is both the input and output alphabet, and the above conditional probability distribution is the corresponding channel transition probability mass function (pmf). We call this channel the *super-DMC induced by* the code $(\mathcal{C}, d)$. Since the super-DMC can be regarded as just another DMC, we can apply the (SWCF) scheme described in Section 6.2.4 to this super-DMC.

Recall, however, that we can construct an SWCF scheme only if we can construct an appropriate Slepian-Wolf coding scheme for the feedback data. A low-complexity, expander-code-based implementation of Slepian-Wolf coding exists when the inner code has very low probability of error, as we show in the next section. In principle, though, the inner code need not have this property; we discuss such inner codes in Section 6.3.3.

6.3.2 Superchannels Induced by Strong Inner Codes

We can explicitly construct SWCF coding schemes requiring very low rate feedback for superchannels induced by strong inner codes, i.e., inner codes with low probability of decoding error. Note that such codes must have rates below the capacity of the original DMC. Since we already describe how SWCF codes work for arbitrary DMC's with a given Slepian-Wolf coding scheme in Section 6.2.4, all that remains for a construction of a coding scheme is to construct an appropriate Slepian-Wolf coding scheme.

To facilitate the discussion, let us introduce some notation. Let $(\mathcal{C}, d)$ be a rate-r_{in} length-n_{in} codebook and corresponding decoder inducing a superchannel, whose inputs we denote by $\tilde{X}_i$ and outputs we denote by $\tilde{Y}_i$. The superchannel transition pmf $p_{\tilde{Y}_i|\tilde{X}_i}$ satisfies $p_{\tilde{Y}_i|\tilde{X}_i}(x^n|x^n) = 1 - P_{\text{in}}$ for all $x^n \in \mathcal{C}$, where P_{in} is the probability of decoding error associated with $(\mathcal{C}, d)$.

Slepian-Wolf Coding Scheme

To create a Slepian-Wolf coder for this superchannel, we can adapt the expander-code based method of Section 6.2.2. Conceptually, the inner code transforms the DMC into a BSC with effective crossover probability less than P_{in}, and the expander-code based method of Section 6.2.2 can be applied.

To be more precise, let $b : \mathcal{C} \to \{0,1\}^{\lceil r_{\text{in}} n_{\text{in}} \rceil}$ be an invertible mapping from $\mathcal{C}$ to a sequence of $\lceil r_{\text{in}} n_{\text{in}} \rceil$ bits. To Slepian-Wolf code a length-n_{f} block of superchannel symbols $\tilde{Y}^{n_{\text{f}}}$, we let $\tilde{n}_{\text{f}} = n_{\text{f}} \lceil r_{\text{in}} n_{\text{in}} \rceil$ be the number of bits used to represent n_{f} superchannel symbols, and let $K_{\tilde{n}_{\text{f}}} \in \{0,1\}^{(1-r)\tilde{n}_{\text{f}} \times \tilde{n}_{\text{f}}}$ be the parity-check matrix of an expander code with rate $r > 1 - 2H_2(\sqrt{48P_{\text{in}}})$ that corrects $\alpha\tilde{n}_{\text{f}} > P_{\text{in}}\tilde{n}_{\text{f}}$ errors in linear time. Then let e^{SW} be defined by

$$e^{\text{SW}}(\tilde{Y}^{n_{\text{f}}}) = K_{\tilde{n}_{\text{f}}} Y^{\tilde{n}_{\text{f}}} \tag{6.26}$$

$$Y^{\tilde{n}_{\text{f}}} = (b(\tilde{Y}_1), b(\tilde{Y}_2), \cdots, b(\tilde{Y}_{n_{\text{f}}})). \tag{6.27}$$

The decoder d^{SW} is defined by

$$d^{\text{SW}}(\tilde{X}^{n_{\text{f}}}, B^{r\tilde{n}_{\text{f}}}) = d^{\text{exp}}(K_{\tilde{n}_{\text{f}}} X^{\tilde{n}_{\text{f}}} - B^{\tilde{n}_{\text{f}}}) \tag{6.28}$$

$$X^{\tilde{n}_{\text{f}}} = (b(\tilde{X}_1), b(\tilde{X}_2), \cdots, b(\tilde{X}_{n_{\text{f}}})), \tag{6.29}$$

where d^{exp} is a decoder for an expander code. The rate of the Slepian-Wolf code is therefore $2H_2(\sqrt{48P_{\text{in}}})\lceil r_{\text{in}} n_{\text{in}} \rceil$ bits per superchannel symbol, each of which consumes n_{in} symbols in the original channel. Hence, the total feedback rate is $2H_2(\sqrt{48P_{\text{in}}})\lceil r_{\text{in}} n_{\text{in}} \rceil / n_{\text{in}}$ bits per original channel symbol.

To analyze the probability of Slepian-Wolf decoding error for this coder, we use that an error cannot occur unless $Y^{\tilde{n}_{\text{f}}}$ differs from $X^{\tilde{n}_{\text{f}}}$ in more than $\alpha\tilde{n}_{\text{f}}$ places. But this event cannot happen unless $\tilde{Y}^{n_{\text{f}}}$ differs from $\tilde{X}^{n_{\text{f}}}$ in more than αn_{f} places. The probability that this event happens is well known [21] to be less than

$$\exp_2\{-n_{\text{f}} D_2(\alpha \parallel P_{\text{in}})\}. \tag{6.30}$$

Properties of the CSWCF Scheme

According to Section 6.2.4, this Slepian-Wolf coder can be used to construct an SWCF coding scheme that operates at any rate below the capacity of the superchannel with error probabilities that decay exponentially with the average number of superchannel inputs used, which, in turn, is proportional to the average number of original channel inputs used since each superchannel input uses a fixed number n_{in} of original channel inputs.

To determine the maximum rate of the CSWCF scheme for which arbitrarily small probabilities of error are possible, we need only determine the capacity of the superchannel and divide it by n_{in}, the number of original channel inputs used per superchannel input. The capacity of the superchannel can be lower bounded by the capacity of the corresponding "equierror" channel [21]. In brief, the equierror channel corresponding to our superchannel is a channel with additive (mod $|\mathcal{C}|$) noise Z satisfying $\Pr\{Z = 0\} = 1 - P_{\text{in}}$, and $\Pr\{Z = j\} = P_{\text{in}}/(|\mathcal{C}| - 1)$ for all $j = 1, \cdots, |\mathcal{C}| - 1$. The capacity of this equierror channel $\log |\mathcal{C}| - H(Z)$, where it can easily be shown that $H(Z) = P_{\text{in}} \log(|\mathcal{C}| - 1) + H_2(P_{\text{in}})$. The capacity of the superchannel is therefore greater than $n_{\text{in}} r_{\text{in}} (1 - P_{\text{in}}) - H_2(P_{\text{in}})$. Note that the channel coding theorem implies that the inner code can be chosen so that the superchannel capacity is arbitrarily close to n_{in} times the capacity of the original channel, implying in turn that rates arbitrarily close to the original channel's capacity can be achieved with the CSWCF scheme.

Moving to an analysis of the feedback rate required by the scheme, we can use the arguments of Section 6.2.3.2 to see that $2H_2(\sqrt{48P_{\text{in}}})$ bits per channel output are needed. Since P_{in} can be made arbitrarily small by appropriately choosing the inner code, the feedback rate can be made arbitrarily small.

The complexity of this concatenated coding scheme is non-uniform linear complexity. While the CSWCF scheme can approach the capacity of the superchannel with uniform linear complexity, more and more complex inner codes are required for the superchannel's capacity to approach the capacity of the original channel. Only by using inner codes that are capacity-achieving with uniform linear complexity can the CSWCF scheme have uniform linear complexity. Since no uniform-linear-complexity capacity-achieving codes are known, we can only say the CSWCF scheme has non-uniform linear complexity.

As a practical matter, it may be of concern that the precoder and source coder for the superchannel may be quite complex owing to large input and output alphabets associated with a given superchannel. However, the ideal precoding distribution is just the uniform distribution over $\mathcal{C}$ and therefore a simple mapping can be used. In addition, the source coder can easily be simplified so that it simply notes the locations in which $\tilde{Y}_i$ differs from $\tilde{X}_i$ and then repeats the data in that location. It is straightforward to show that these precoder and source coder designs give the same asymptotic performance as those described in Chapter 2.

6.3.3 Superchannels Induced by Weak Inner Codes

The superchannel we just described has the attractive property that a very low feedback rate is required. Many other combinations of inner and outer code parameters are available to give different combinations of inner code rate, outer code rate, CSWCF code rate, and feedback rate. The inner code may even operate at a rate significantly above capacity. The extreme case of this, when the inner code has no redundancy, corresponds to the original case of the iterative coding scheme developed in Chapter 2. This extreme case gives no reduction of the feedback rate.

As a middle ground, we might choose an inner code with rate r satisfying $C < r < H(X)$. For a forward BSC with crossover probability ϵ, this translates to choosing $r \in (1 - H_2(\epsilon), 1)$. Assuming a corresponding feedback rate of $H(\tilde{Y}_i|\tilde{X}_i)/n$, we then have a construction for a continuum of capacity-achieving schemes with feedback rates ranging from $H_2(\epsilon)$ at $r = 1$ to 0 at $r = 1 - H_2(\epsilon)$. But can a scheme that is not at these two extremes achieve capacity? This question is equivalent to the question of what the capacity of the induced superchannel is. The scheme approaches capacity only if

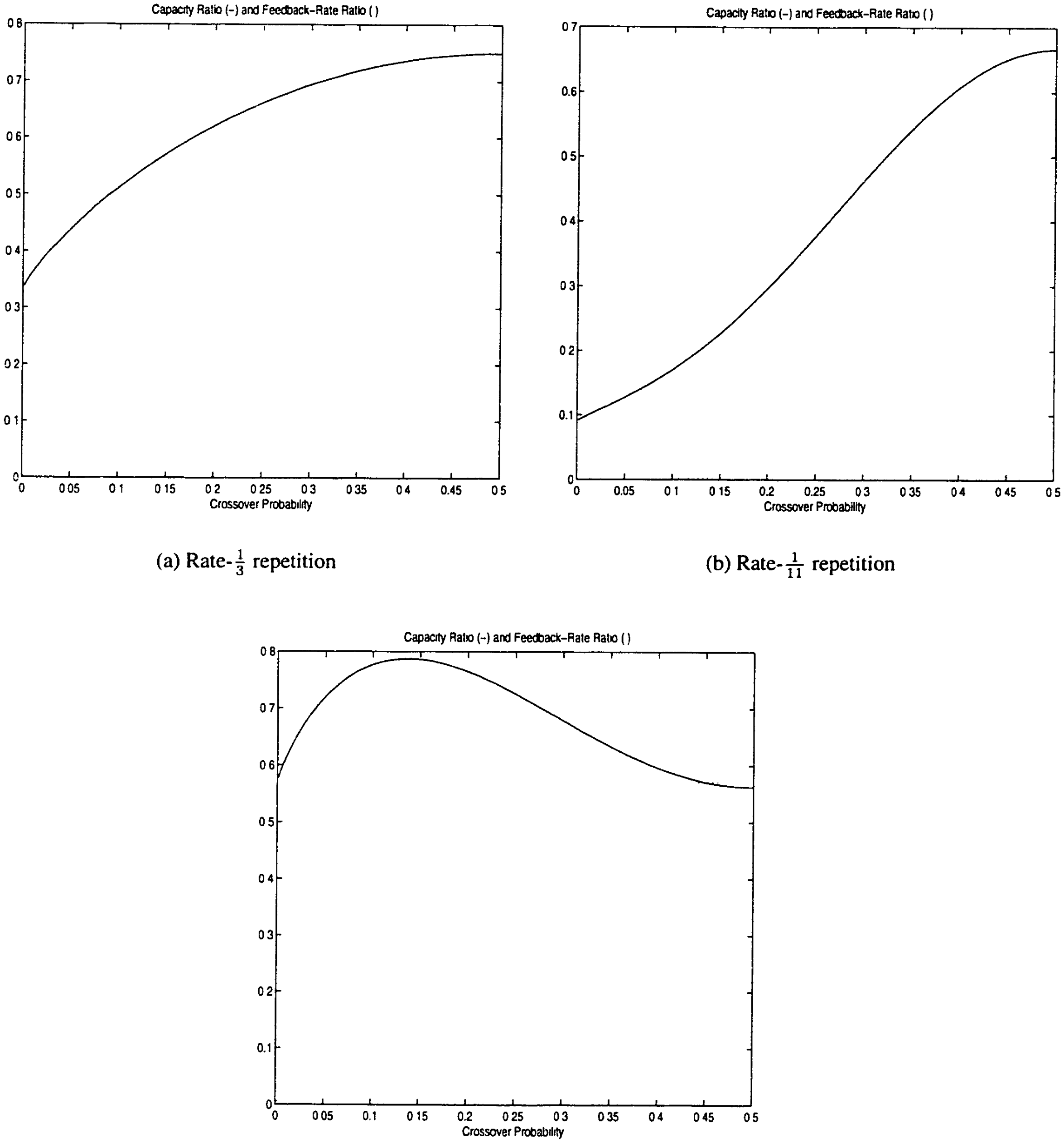
$$H(\tilde{X}_i|\tilde{Y}_i)/n \approx r - C. \tag{6.31}$$

For the case of the BSC, the results on list decoding (see the generalized Gilbert-Varshamov bound in [19]) imply that for sufficiently long blocklengths, there do exist codes so that (6.31) holds. This fact suggests that our framework gives rise to a continuum of capacity-achieving schemes requiring feedback rates from 0 to $H(Y|X)$ (assuming the possibility of Slepian-Wolf coding at the minimum possible rate.)

Although it is unclear why one would ever favor a long inner code with rate greater than capacity — since the length of the inner code would render the complexity high, and the high rate would cause the feedback rate to be high as well — short inner codes with rates above capacity may have use under certain sets of constraints.

Let us demonstrate this potential usefulness with the following examples, which show that simple, short inner codes used in a CSWCF scheme can substantially reduce feedback rate, while only moderately reducing the achievable forward rate. Consider a very simple example in which we have a BSC with partial noiseless feedback (BSC_{pf}) with a forward capacity of 0.25 bits per input (i.e., crossover probability $\approx$ 0.21). The Slepian-Wolf coding theorem suggests that to use an SWCF scheme, we require at least 0.75 bits of feedback per channel output used. Now suppose we use a 3-symbol repetition code as an inner code. This 2-codeword repetition code transforms the channel into a super-BSC with capacity 0.48, taking three channel inputs per superchannel input for an effective capacity of $0.48/3 = 0.16$, significantly below the original capacity of 0.25 (a "capacity ratio" of 0.16/0.25=0.63). However, according to the Slepian-Wolf coding theorem, the feedback can now be coded at rate of 0.52 bits per superchannel output, which is effectively a feedback rate of $0.52/3 = 0.17$ bits per original channel output, which is far below the original 0.75 bits per channel output that were needed originally (a "feedback-rate ratio" of 0.17/0.75=0.23). With this same repetition code used as the inner code, we plot in Figure 6-4(a) the resulting capacity ratio (solid line) and feedback-rate ratio (dotted line) for crossover probabilities in $(0, 1/2)$. That is, we let $\delta = \epsilon^3 + 3\epsilon^2(1 - \epsilon)$ be the "crossover" probability associated with the superchannel and plot $\frac{1}{3}(1 - H_2(\delta))/(1 - H_2(\epsilon))$ with the solid line and $\frac{1}{3}H_2(\delta)/H_2(\epsilon)$ with the dotted line. In Figures 6-4(b) and 6-4(c), we plot the analogous curves for a rate-1/11 repetition code and a (7,4) Hamming code, respectively.

As an aside, we note that weak inner codes are practically useless for the linear-complexity feedback-free concatenated codes based on Spielman outer codes discussed in Section 2.6.2. The reason is that the outer Spielman code is so weak that the outer code's rate would have to be very low to correct a large number of errors remaining from the weak inner code. In contrast, a feedback outer code as we have described is so strong that any capacity loss is effectively incurred only by the inner code. On the other hand, because our scheme is based on using expander codes for the Slepian-Wolf coding, the full, theoretically achievable, feedback rate reduction will not be completely realized.



(a) Rate-$\frac{1}{3}$ repetition

(b) Rate-$\frac{1}{11}$ repetition

(c) (7,4) Hamming

Figure 6-4: Capacity ratios (solid line) and feedback-rate ratios (dotted line) for BSC$_{\mathrm{pf}}$'s with various crossover probabilities using rate 1/3 and rate 1/11 repetition codes and a (7,4) Hamming code.

6.3.4 Computational Comparison with Feedback-Free Linear Complexity Concatenated Coding Scheme

Although feedback-free codes exist with similar performance to the CSWCF coding scheme described above, the superior reliability of the feedback outer code compared to (feedback-free) Spielman codes [56] gives it a computational advantage over the linear-complexity concatenated coding scheme mentioned in Section 2.6.2.

To compare the performance of the CSWCF scheme with the concatenated code using an inner Spielman code (CS codes), we make several approximations to bring out the essential behavior of both schemes. Let us assume that the outer code rates are the same for each coder and that the inner code rates are the same for each coder, resulting in the same combined rate. Also assume that the rate r of the inner code is close to capacity C and that the outer code rate R is close to one, so that the combined rate is close to capacity. Finally, assume that the inner code is decoded using maximum-likelihood exhaustive search.

We begin by assessing the relative powers of the Spielman outer code compared to the SWCF outer code. For CS codes, since the Spielman outer code operates at rate R, it corrects a fraction of errors $K_1^2\epsilon_1^2$, where K_1^2 is a small constant given in Theorem 19 of [56], and ϵ_1 is a function of R satisfying $H_2(\epsilon_1) \approx (1-R)/2$. In contrast, for the CSWCF codes, since the outer feedback code has rate R, it usually corrects a fraction of errors ϵ_2, where, because the code is capacity achieving, ϵ_2 is a function of R satisfying $H_2(\epsilon_2) \approx 1-R$. Note also that $\epsilon_2 \approx 2\epsilon_1$ for R very close to 1.

The power of the outer code affects the computation in the following way: The length of the inner code must be adjusted so that the probability of error is less than $K_1^2\epsilon_1^2$ and ϵ_2 for the CS code and CSWCF code, respectively. Because $r \approx C$, the computation needed for decoding the inner code is approximately $n2^{nC}$. To relate the required computation to the probability of inner decoding error P_{in}, we make the approximation $P_{\text{in}} \approx 2^{-nE_{\text{in}}(r)}$, where $E_{\text{in}}(r)$ is the error exponent for the inner code, which allows us to rewrite the computation in terms of P_{in} as $nP_{\text{in}}^{-nC/E_{\text{in}}(r)}$. We can substitute $K_1^2\epsilon_1^2$ for P_{in} to find the computation required per superchannel input for the CS code and substitute ϵ_2 for P_{in} to find the corresponding quantity for the CSWCF code. For the CS case, we require about $n(K_1\epsilon_1)^{-2C/E_{\text{in}}(r)}$ computations to decode the inner code. For the CSWCF case, we require about $n\epsilon_2^{-C/E_{\text{in}}(r)}$. Using that $\epsilon_2 \approx 2\epsilon_1$ for small ϵ_1 and small ϵ_2, we see that the computation per original channel input sample for the CS code is approximately $(K_1^2/2)^{-C/E_{\text{in}}(r)} > 1$ times the *square* of the computation required for the CSWCF code.

In summary, when operating at about the same rate, while the complexity of both the CS and CSWCF is linear, the computation per bit for the CS scheme is far larger than that required for the CSWCF scheme.

6.4 Feedback Coding on the Erasure Channel and ARQ

While computationally efficient expander codes can, as we showed in Section 6.2.2, be used for linear-complexity Slepian-Wolf coding in some scenarios, different low-complexity Slepian-Wolf codes can be developed for another scenario, namely, for communicating over an erasure channel, an example of which is depicted in Figure 6-5.

Erasure channels are convenient approximations to a number of practical channels, e.g., channels on which error-detection codes are used. In part because of the ubiquity of such channels, the automatic-repeat-request (ARQ) protocol, which operates at the capacity of the erasure channel

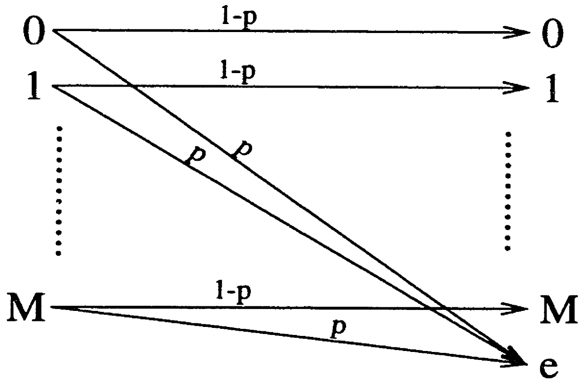


Figure 6-5: Depiction of an erasure channel.

with zero probability of error, is perhaps the feedback scheme most widely used in practice. In this section, we develop new perspectives on the ARQ protocol by showing that it can be interpreted as a special version of the SWCF and CSWCF coding schemes described in the previous sections.

Consider an erasure channel in which the channel output Y is equal to the channel input X with high probability $1 - P_{\text{er}}$, and the channel output Y is equal to the erasure symbol e with probability P_{er}. We can easily apply the SWCF coding scheme to this channel, because Slepian-Wolf coding of Y^n for a decoder that knows X^n is straightforward. The Slepian-Wolf encoder for Y^n need only convey the location of all the erasures to the associated decoder, which requires about $nH_2(P_{\text{er}})$ bits of feedback.

On this channel, a simple ARQ scheme achieves similar performance but with smaller delay. An ARQ scheme gives one bit of feedback for each received symbol indicating whether or not the reception is an erasure. Then if erased symbols are immediately retransmitted, the average number of retransmissions is $P_{\text{er}}/(1 - P_{\text{er}})$, the total number of transmissions per symbol is $1/(1 - P_{\text{er}})$, and the rate is therefore $\log|\mathcal{X}|(1 - P_{\text{er}})$ bits per channel input, which is the capacity of such a channel. The feedback rate of one bit per received sample is generally substantially less than the rate of $H(Y)$ bits per received sample required for complete feedback.

The ARQ protocol can be viewed within the framework of this chapter, with retransmission interpreted as a source coding step and one-bit feedback interpreted as efficient low-complexity Slepian-Wolf coding. In particular, we can interpret the ARQ transmission scheme as a special version of the SWCF coding scheme of Section 6.2 in which the message length is fixed at $N = 1$ rather than being arbitrarily large, and the number of iterations is variable rather than fixed.

It is straightforward to see that error/no-error feedback is a very efficient code that allows an agent with knowledge of X to know Y, which is the essential property of a Slepian-Wolf code. Furthermore, error/no-error feedback is the most efficient such code if one insists on coding Y in "blocks" of only one sample each. Because it is also an error-free encoding of Y, the unity blocklength does not cause a high rate of Slepian-Wolf decoding errors as it would in most cases in which X and Y were not related through an erasure channel.

We can also see that retransmission (and refraining from it appropriately) is an efficient source encoding of X based on Y. When an erasure occurs (i.e., when Y = e), then the conditional probability distribution for X is just the unconditional probability distribution for X, which means that $H(X)$ bits must be expended to describe X to the receiver — i.e., X must be retransmitted; when an erasure does not occur, then the conditional probability distribution for X says that X is deterministically equal to Y; in this case, no bits need be expended to describe X to the receiver, since it has knowledge of Y.

Other applications of ARQ also have interpretations within the frameworks developed in this chapter. For example, ARQ transmission schemes are also used on channels such as relatively clean BSC's. In this technique, the channel is transformed into what is effectively an erasure channel by using an "inner" error-detecting code. We can regard such schemes as a form of concatenated coding, where the outer code is the ARQ protocol.

To be more precise, let X^n be a codeword in a rate-r, length-n error-detection code transmitted to a receiver, and let Y^n be the receiver's reception of it. Let d be a decoder for the code that operates as follows

$$d(y^n) = \begin{cases} y^n & \text{if } y^n \in \mathcal{C}, \\ \mathrm{e} & \text{otherwise.} \end{cases} \tag{6.32}$$

Suppose that $d(Y^n) = X^n$ with high probability $1 - P_{\text{er}} - P_{\text{ud}}$, that $d(Y^n) \in \mathcal{C}$ and $d(Y^n) \neq X^n$ (an undetected error event), with probability P_{ud}, and that $d(Y^n) = \mathrm{e}$ with probability P_{er}. We now have a new channel from X^n to $d(Y^n)$, which we call a *pseudo-erasure superchannel*, since there is a possibility that the symbol put in to this superchannel will result in the superchannel's putting out a different non-erasure symbol.

If ARQ is still used for this new pseudo-erasure superchannel, then its behavior is almost the same as for a true erasure channel except that there is now the possibility of error caused by undetected error events. We can now interpret the error/no-error feedback as a Slepian-Wolf encoding of $d(Y^n)$ whose probability of decoding error is P_{ud}. These Slepian-Wolf decoding errors cause the overall transmission scheme to have a probability of error P_{ud} and rate $r(1 - P_{\text{er}})$.

That ARQ transmission schemes can be viewed as variants of an SWCF coding scheme if the original channel is an erasure channel, or a CSWCF coding scheme if the original channel is an arbitrary DMC, suggests opportunities for enhancing ARQ systems. In particular, to designers of low-rate feedback systems, the CSWCF framework may help to provide an extended range of coding options that build on or supplant ARQ. For example, error detection codes for the BSC used with ARQ have the following fundamental limitation: As the error-detection codes become longer, the probability of retransmission increases, but as they become shorter, the probability of an undetected error increases. One solution suggested by our framework is to use longer error-detection codes. More erasures then occur. When they do, we might feed back the full superchannel symbol corresponding to the erasure (possibly incorporating expander-code-based Slepian-Wolf coding) and then the next correction message need not be a full retransmission of the superchannel input symbol but only a conditional source coding of the superchannel input symbol. In this way, we can mitigate the loss in rate due to additional erasures.

6.5 Coping with Noisy Feedback

In this section, we undertake a preliminary investigation of coding for channels with noisy feedback. In particular, we propose and analyze, at a high level, modifications to the coding scheme from

Chapter 2 that are appropriate when forward channel is a DMC *and* the feedback channel is a (possibly different, noisy) DMC. We restrict attention to the case in which the capacity of this feedback channel is large enough to support complete feedback even after low-rate coding is applied to the data being fed back. Our main result is that probabilities of error decaying exponentially with the number of channel inputs used are possible at all rates below capacity with uniform linear complexity.

The general strategy we adopt is to apply a low-complexity error correcting code to the feedback data. This approach has the two problems of delay and errors. Of the two, errors are more serious. Consider, for example, an mSB coding scheme in which the transmitter sends an inner code, and depending on whether it is received correctly, sends a verification sequence. If one verification sequence is received, the receiver expects a retransmission; if the other is received, it does not. Suppose the receiver feeds back, after it receives the verification sequence, 1 bit indicating whether it expects a retransmission. If this 1 bit of feedback is corrupted, then the transmitter and receiver may have different opinions about whether a retransmission is to occur. If only one message is sent in total, then this difference of opinion can be made inconsequential. But if an infinite stream of messages is sent, then this difference of opinion can wreak havoc. For example, if after the first inner codeword and verification sequence, the receiver expects a retransmission but the transmitter does not, then the next message the receiver obtains is interpreted as the first message, when it is in fact the second. If all subsequent messages are received correctly, then they are all in the wrong position. In practice, one can install many mechanisms to make the possibility of complete havoc acceptably small. For example, each message can carry a number indicating its position in the message stream, or the receiver can tell the transmitter which number message it expects. The analysis of such schemes is cumbersome, however. To circumvent these difficulties, we discuss how a fixed-length scheme with error correction on the feedback link could work.

To develop a fixed-length, linear-complexity scheme with exponentially decaying error probability, we first note that there is a two-codeword codebook for the DMC with length n_2 and probability of error less than 10^{-12}, a constant we have chosen somewhat arbitrarily, although it should be smaller than $0.11^2/48$. Then there for every n, there is an expander code with rate greater than $1-2H_2(\sqrt{48 \times 10^{-6}}) \approx 0.8807$ that corrects at least $10^{-6}n$ errors in linear time. If we take a codeword in this binary expander codebook and substitute one codeword in our two-codeword codebook for each 1 and the other codeword for each 0, then we will have constructed a codebook for the DMC with rate $r_{\text{term}} = 0.8807/n_2$, probability of error less than $\exp_2\{-nD_2(10^{-6} \parallel 10^{-12})\}$, and linear encoding and decoding complexity.

Now, let $L = N(1/I(X;Y) + \epsilon)$, be the target length for our fixed-length code that encodes N bits, where $\epsilon \ll 1$. We construct a scheme that gives exponentially decaying error probability at this rate, albeit with a poor exponent, as follows: First, we let $\delta = \epsilon/(8\bar{I}(X;Y)^2)$ and use an iterative scheme similar to that in Section 2.7.1, using the fixed-length precoding and source coding subsystems $\{\pi_{\delta,n}\}_n$ and $\{\sigma_{\delta,n}\}_n$. We then let the number of iterations B_N be the smallest integer such that

$$\frac{N}{H(X)-\delta}\left(\frac{H(X|Y)+\delta}{H(X)-\delta}\right)^{B_N} < N\epsilon r_{\text{term}}/8. \tag{6.33}$$

Notice that B_N is then a constant independent of N, so we denote it by B. After B iterations, the residual message left to be coded is less than $N\epsilon r_{\text{term}}/8$ bits. Using the termination code discussed above, the final termination coded block is less than $N\epsilon/8$ samples long. Now, let the feedback be coded using some low-complexity, length-$N\epsilon/(8B)$ block error-correcting code, whose error probability decays exponentially in this length. We could, for example, use the design described in the preceding paragraph. Then the delay between two iterations incurred by waiting for the coded feedback is less than $N\epsilon/(8B)$, which in total is less than $N\epsilon/8$. The total length of this coding scheme is therefore upper bounded by

$$\frac{N}{I(X;Y)-2\delta}+N\frac{\epsilon}{8}+N\frac{\epsilon}{8}, \tag{6.34}$$

which, for small ϵ, is less than $N(1/I(X;Y)+\epsilon)$. All sources of error can be shown to decay exponentially in N, so the total error probability can be union bounded by a term decaying exponentially in N.

Because this scheme has fixed-length, the transmitter and receiver always agree on the actions they are taking. An error in the feedback data causes at most a forward decoding error and does not affect subsequent transmissions.

Note that if the rate of the feedback channel after low-complexity block coding is too low to support full feedback after coding, we could in principle exploit the notion of combining block coding on the feedback channel with the techniques for dealing with noiseless partial feedback described above.

6.6 Discussion

6.6.1 Multiple-Access Channels with Partial Feedback

Since multiple-access channels may also return only partial feedback, we are naturally curious to see how if at all the frameworks we develop in this chapter apply to this case. This section contains a preliminary exploration of this question.

We begin by considering the role of Slepian-Wolf coding when communicating over a 2-user DMMAC with partial feedback. Immediately, we encounter an obstacle: Assuming both transmitters receive the same feedback data, if we use only $H(Y|X_1,X_2)$ bits of Slepian-Wolf coded feedback, neither Transmitter 1 (Tx-1) nor Transmitter 2 (Tx-2) can obtain Y, because neither knows the other's transmission. A crude solution is to feed back $H(Y|X_1)$ bits to Tx-1 and $H(Y|X_2)$ bits to Tx-2 for a total rate of $H(Y|X_1)+H(Y|X_2)$. We do not know whether a more efficient method exists, although intuitively it stands to reason that one should: If D is the description of Y intended for Tx-1, then only $H(Y|X_2,D)$ bits should be required for the description of Y to Tx-2. But how small can $H(Y|X_2,D)$ be and what choice of D minimizes this quantity? This problem is beyond the scope of this discussion, but we believe that some of the work on multiple descriptions [18], successive refinement [20], and the broadcast channel [14] may be relevant.

That Slepian-Wolf coding in this case need not be more efficient in terms of rate than no coding suggests that the SWCF framework developed in the previous sections may not be well-suited to multiple-access channels.

Let us consider another approach to gaining insight into the problem. In Section 6.4, we were able to interpret the ARQ protocol, which requires very-low-rate feedback, in terms of the SWCF framework. Is there perhaps an analog to the ARQ protocol for multiple-access channels that will shed light on the problem? The most relevant set of protocols seems to be the various protocols for communicating over the packet-switched multiple-access broadcast channel (MABC), which is a useful model for a variety of packet switched communication systems.

The MABC is defined as follows. Time is discrete and is known by all users; at the start of each time slot, each of M users makes a decision regarding whether to transmit or not transmit; if two or more users in the system transmit a packet at the same time, then a collision results, and no packet is successfully received; if only a single user transmits a packet, a packet is received successfully, while if no user transmits, the channel is wasted; after every attempt at using the channel, all users are immediately informed as to whether a collision (two or more users transmit), success (one user transmits), or idle (no users transmit) occurred; packets arrive during each time slot with probability p to each user, and each user has no buffer (sometimes called a single buffer), i.e., he can hold only one packet waiting to be transmitted. All of the above are common and useful assumptions that capture the essence of a class of practical problems in which a common channel is shared. The goal is to choose protocols so that the probability of a successfully received packet in each slot, i.e., the throughput, is maximized.

A problem with using the MABC to garner insight into ways of handling partial feedback multiple-access channels is that the optimal method for communicating over such channels is unknown. As a finding that is interesting in its own right, we have established the near-optimality of the Hluchyj-Gallager protocol [28] for two and three users. This result along with a detailed discussion of the MABC and its relationship to decentralized control is given in Appendix E. Unfortunately, the Hluchyj-Gallager protocol is, in general, relatively complicated, although, as shown in Appendix E, it reduces in some cases to the TDMA protocol, which actually ignores the feedback altogether. In light of this fact, it is questionable and remains to be determined whether the MABC is an appropriate canonical example. It may be more fruitful to instead develop other multiple-access analogs of the erasure channel.

6.6.2 An Alternative View of Coding with Partial Feedback

We discuss here another perspective on the use of the compressed-error cancellation framework for partial-feedback channels. First, we observe that no feedback at all is required to use the compressed-error cancellation framework if a Slepian-Wolf source coder is used as the source coding subsystem. This observation provides an important link between Slepian-Wolf coding and forward error-correcting coding: if a reliable Slepian-Wolf coding scheme is available, then one can use it to construct a forward-error-correcting code. That Slepian-Wolf coding is, in general, computationally difficult, limits the practical utility of this observation. But the observation leads to a new perspective on partial feedback: When the conditioning variable is fully known (full feedback), the conditional source coding task is easy, as we saw in Chapter 2. When it is unknown (no feedback), the task is hard, as suggested above. So how hard is the conditional source coding task with partial knowledge of the conditioning variable (partial feedback)?

To provide some insight into this quesiton, we examine a particularly natural form of partial knowledge of the receiver's observation Y^n that could be fed back: a rate-distortion coded version of Y^n. The following example suggests how hard the source coder's task is when such partial knowledge of Y^n is given. Consider the following procedure for a BSC_{pf} with forward crossover probability ϵ and feedback rate $1 - H_2(\alpha)$ bits per channel output:

- The transmitter (Tx) sends a block B^N of N data bits uncoded.
- The channel adds (modulo-2) N samples of Bernoulli-ϵ noise Z^N.

- The receiver (Rx) forms a distorted version $\hat{Y}^N$ of its N noisy observations Y^N, where $D^N = \hat{Y}^N - Y^N$ has Hamming weight αN. It feeds $\hat{Y}^N$ back to Tx using $N(1 - H_2(\alpha))$ bits.
- Tx forms $\hat{Z}^N = \hat{Y}^N - B^N = Z^N + D^N$. It then uses Slepian-Wolf coding to compress $\hat{Z}^N$ into $H(\hat{Z}^N|D^N) = NH_2(\epsilon)$ new data bits and sends these data bits over the channel uncoded. (Since the receiver knows D^N, it can recover $\hat{Z}^N$ and therefore Z^N.)
- The channel adds (modulo-2) $NH_2(\epsilon)$ samples of Bernoulli-ϵ noise.

⋮

This procedure, which attains the channel capacity $1 - H_2(\epsilon)$, still seems to require a potentially costly Slepian-Wolf source coding step, even though partial knowledge of Y^N is fed back. It would be intuitively pleasing if the Slepian-Wolf coding step were somehow less and less computationally costly as the feedback rate increased, but we do not know if this is the case. Note, however, that if we avoid this Slepian-Wolf coding step altogether and encode $\hat{Z}^N$ with $H(\hat{Z}^N) = NH_2(\epsilon')$ bits, where $\epsilon' = \alpha(1 - \epsilon) + (1 - \alpha)\epsilon$, then the rate becomes $1 - H_2(\epsilon')$, which is less than capacity, but approaches capacity as the feedback rate goes to 1.

Although only a preliminary discussion of it is given here, this alternative perspective, in mapping the problem of compressed-error-cancellation coding with partial feedback to the problem of conditional source coding with partial knowledge of the conditioning random variable, could lead to new insights into the problem of coding with partial feedback.

6.6.3 Summary and Future Directions

We have shown by construction that a uniform-linear-complexity feedback coding scheme can be devised with less than rate-$H(Y)$ feedback. While the scheme gives a reduction in required feedback rate that can be dramatic, the reduction is ultimately due to Slepian-Wolf coding of the feedback. The transmitter still uses essentially complete knowledge of Y. We then developed a concatenated-coding type of framework in which truly incomplete knowledge of Y is fed back. However, we saw that the computational complexity increased. We have been unable to determine whether or not the complexity can be made to gracefully increase as feedback rate decreases to zero. We showed that if one does not demand that the rate of the coding scheme be near capacity, then the use of short, weak inner codes within the concatenated-coding framework provides a set of very computationally efficient feedback strategies with arbitrarily low probability of error with truly incomplete knowledge of Y. Still, much work remains to be done in assessing the tradeoffs between the power of the inner code, complexity, and rate.

We conjecture that there is no capacity-achieving uniform-linear-complexity coding scheme for DMC$_\text{f}$'s requiring fewer than $H(Y|X)$ bits of feedback per channel output. The origin of this conjecture is simplt that we have yet to find any uniform-linear-complexity capacity-achieving feedback coding scheme that does not require the transmitter to have nearly complete knowledge of the channel outputs. An interesting research problem would be to prove this conjecture to be wrong (or right, but this proof is likely to be much harder to obtain.)

Although the analysis in this section has been for DMC's with partial noiseless feedback, these techniques should be extendable to DFSC's and UFSC's with partial noiseless feedback (DFSC$_\text{pf}$'s and UFSC$_\text{pf}$'s, respectively). The main obstacle for DFSC$_\text{pf}$'s is finding an appropriate method for Slepian-Wolf coding the feedback data. For UFSC$_\text{pf}$'s, one would have to contend with some sort of universal Slepian-Wolf coding.

We have also shown briefly how error-correcting codes can be applied to a noisy feedback channel without reducing the achievable rate on the forward channel and maintaining low overall complexity. One somewhat unsatisfactory property of the resulting scheme, however, is that the feedback channel and forward channel are treated asymmetrically. The notion of using the compressed-error cancellation scheme on both the forward and feedback channel would be attractive if there were a way to carry it out. An exploration into this interesting area might begin with the case in which both the forward and backward channels are erasure channels.

Chapter 7

Conclusions

For decades, the availability of complete, noiseless feedback has been known to be useful in reducing the computational complexity of coding schemes. This knowledge has come mostly by way of examples of low-complexity feedback coding schemes. But no general framework for low-complexity coding with feedback has emerged as a particularly natural and useful one. In this thesis, we developed one such framework and showed it to be useful for a number of different feedback communication problems.

The major contributions of this thesis occur at two levels. At one level, new classes of low-complexity feedback coding schemes were introduced. The range of channels for which these schemes provide low-complexity coding solutions is remarkably broad, encompassing discrete memoryless channels (DMC$_f$'s), discrete finite-state channels (DFSC$_f$'s), unknown finite-state channels (UFSC$_f$'s), discrete memoryless multiple-access channels (DMMAC$_f$'s), and certain channels with only partial feedback. The coding schemes all have the lowest possible asymptotic time complexity — i.e., the asymptotic order of growth of the number of computations used for encoding or decoding is linear in the number of channel inputs used. Further, analysis of the performance of each coding scheme revealed that each has high rate and low probability of error.

At a broader level, the thesis introduced a systematic, tractable framework for low-complexity coding with feedback — i.e., the compressed-error-cancellation framework — which formed the basis of the new coding schemes. Within this framework, coding with feedback can be done by iterating a two-step process. In the first step, a message is precoded and sent without redundancy over the channel. In the second step, a new message is constructed that can resolve the uncertainty held by the receiver about the original message. The flexibility and generality of this framework suggest that it may form the basis of still more sophisticated schemes.

But the compressed-error-cancellation framework may have shortcomings. For example, whenever the feedback capacity exceeds the feedback-free capacity, it appears difficult to exploit this excess capacity via the framework. This difficulty has multiple roots, an influential one being that the precoding distribution is always held fixed rather than being allowed to adapt to the feedback. Developing new types of precoders for different communication scenarios is an interesting and promising direction for future research.

It is also not clear how to view many existing feedback coding schemes in the context of the compressed-error-cancellation framework. Consequently, another interesting direction for future research is the examination of existing low-complexity coding schemes such as Horstein's [30] to see how they relate to the compressed-error-cancellation framework. Such an examination has the potential to reveal some very deep principles of coding with and without feedback and may ultimately lead to further advances in coding theory and practice.

Appendix A

Detailed Technical Information for Chapter 2

A.1 Proof of Subsystem Property 1

To upper bound $E[\ell(\pi_n(D^n))]$ as in (2.39), it is useful to upper bound l_n by a different function $\tilde{l}_n$, which is easier to analyze:

Lemma A.1.1 Let $b_i = F_{\tilde{X}}^{-1}(i2^{-n})$ for $i = 0, \cdots, 2^n$, and let $\gamma(x, y)$ denote the index of the first M-ary expansion digit at which x and y differ, i.e., let $\gamma : [0,1)^2 \to \mathbb{N}$ be defined by $\gamma(x, y) = \min\{k \in \mathbb{N} : x_{[k]} \neq y_{[k]}\}$. Then define $\tilde{l}_n$ by

$$\tilde{l}_n(u) = \begin{cases} \gamma(u, b_{i+1}) & \text{if } \bar{b}_{i,i+1} \leq u < b_{i+1} \\ \gamma(u, b_i) & \text{if } b_i \leq u < \bar{b}_{i,i+1} \end{cases}, \qquad \forall i = 0, 1, \ldots, 2^n - 1 \tag{A.1}$$

where

$$\bar{b}_{i,i+1} = 0_M.b_{i+1}^{[\gamma(b_i, b_{i+1})]}$$

(for example, if $\mathcal{X} = \{0, 1\}$, $b_3 = 0_2.010011\cdots$, and $b_4 = 0_2.01100101\cdots$ then $\bar{b}_{3,4} = 0_2.011$).

Then

$$l_n(u) \leq \tilde{l}_n(u) \text{ for all } u \in [0, 1). \tag{A.2}$$

Proof: To prove the lemma, we need only show that $[0_M.\tilde{T}_n(u), 0_M.\tilde{T}_n(u) + M^{-k}) \subseteq [b_i, b_{i+1})$, where $\tilde{T}_n(u) = u^{[\tilde{l}_n(u)]}$.

To see that this fact holds, first suppose that $u \in [\bar{b}_{i,i+1}, b_{i+1})$. Next, notice that $\bar{b}_{i,i+1} = 0_M.b_{i+1}^{[\gamma(b_i, b_{i+1})]}0000\cdots$. Since b_{i+1} starts with the same first $\gamma(b_i, b_{i+1})$ digits, any element in $[\bar{b}_{i,i+1}, b_{i+1})$ must also start with the same first $\gamma(b_i, b_{i+1})$ digits. Since $\tilde{T}_n$ truncates u only after its M-ary expansion *differs* from that of b_{i+1}, $0_M.\tilde{T}_n(u)$ must also begin with these digits. Hence, $0_M.\tilde{T}_n(u) \geq \bar{b}_{i,i+1}$. Next, since $u < b_{i+1}$, the M-ary expansion digit at which u first differs from

b_{i+1} must be smaller than the corresponding digit of b_{i+1}. Therefore, $0_M.\tilde{T}_n(u) + M^{-\ell(\tilde{T}_n(u))}$, which corresponds to $\tilde{T}_n(u)$ with its last element incremented by one, may at most equal b_{i+1}. Hence, $[0_M.\tilde{T}_n(u), 0_M.\tilde{T}_n(u) + M^{-\ell(\tilde{T}_n(u))}) \subseteq [\bar{b}_{i,i+1}, b_{i+1})$.

Next, suppose that $u \in [b_i, \bar{b}_{i,i+1})$. First, it is clear that $0_M.T_n(u) > b_i$, since $u > b_i$, and $u_{[\tilde{l}_n(u)]} > b_{i,[\tilde{l}_n(u)]}$. Then note that

$$\bar{b}_{i,i+1} = 0_M.b_i^{[\gamma(b_i,b_{i+1})-1]}((b_{i+1})_{[\gamma(b_i,b_{i+1})]} - 1)(M-1)(M-1)(M-1)\cdots . \tag{A.3}$$

Therefore, $0_M.T_n(u) + M^{-\tilde{l}_n(u)} = 0_M.T_n(u)(M-1)(M-1)(M-1)\cdots \le \bar{b}_{i,i+1}$, since every M-ary expansion digit of u is less than or equal to the corresponding expansion digit of $\bar{b}_{i,i+1}$. Hence, the lemma follows. ∇

We now prove that $E[\tilde{l}_n(F_{\bar{X}}^{-1}(S))] < (n + C_\pi)/H(X)$ for some $C_\pi < \infty$, where S is defined in (2.18). Subsystem Property 1 then follows from this bound with the lemma above.

Consider a device that takes the random process $F_{\bar{X}}^{-1}(S)_{[1]}, F_{\bar{X}}^{-1}(S)_{[2]}, \ldots$, and traverses an M-ary tree until a leaf is reached. (An M-ary tree is a tree in which every node has M children.) We let the stopping rule $\tilde{l}_n$, which is non-anticipatory (as a stopping rule must be) and deterministic, define the leaves and hence the tree. An example of how the tree is traversed is the following: If $M = 2$, starting from the root of the tree, we branch to the left if the first process value is zero, and branch to the right if it is one; now being at a new node, we branch to the left if the second process value is zero, and branch to the right if it is one; we continue until a leaf is reached. Each value of S leads to a different leaf of the tree. We can think of $V = \tilde{T}_n(F_{\bar{X}}^{-1}(S))$ as the random leaf at which we end, and of $\ell(V)$ as the depth of this leaf. It is shown in [16] that

$$E[\ell(V)] = H(V)/H(X). \tag{A.4}$$

To see that (A.4) holds, assume (A.4) holds for some tree (stopping rule). Take any leaf in this tree and create M children that emanate from it. If this leaf has probability p, then the expected length increases by p, and the entropy of the leaves increases by $pH(X)$. Since (A.4) trivially holds for the tree with one level and M leaves, (A.4) must hold for all trees.

We now show that $H(V) < n + C_\pi$, where C_π is independent of n, which gives us the desired upper bound $E[\ell(V)] \le (n + C_\pi)/H(X)$. Let $\mathcal{V} = \{\tilde{T}_n(u)\}_{u\in[0,1)} \subset \{0, 1, \ldots, M-1\}^\dagger$ be the set of all leaves of the tree. Since the stopping rule that defines the leaves is deterministic, the probability of a leaf $v \in \mathcal{V}$ can be written $p_V(v) = \Pr\{F_{\bar{X}}^{-1}(S) \in [0_M.v, 0_M.v + M^{-\ell(v)})\}$, which implies that $p_V(v) = \prod_{i=1}^{\ell(v)} q_X((0_M.v)_{[i]})$. Now, let us evaluate the entropy of V, which is, by definition,

$$H(V) = \sum_{v\in\mathcal{V}} p_V(v) \log \frac{1}{p_V(v)}.$$

To do so, we divide the sum into manageable portions as follows. Consider for now only the leaves $v \in \mathcal{V}$ for which $0_M.v \in [b_i, b_{i+1})$, for some i. Define the sets $\mathcal{V}_i = \{v \in \mathcal{V} \mid 0_M.v \in [b_i, b_{i+1})\}$, $\mathcal{V}_i^+ = \{v \in \mathcal{V} \mid 0_M.v \in [\bar{b}_{i,i+1}, b_{i+1})\}$, and $\mathcal{V}_i^- = \{v \in \mathcal{V} \mid 0_M.v \in [b_i, \bar{b}_{i,i+1})\}$, all of which have a countable number of elements. Further restrict consideration to the leaves $v \in \mathcal{V}_i^+$. Let v_i^+ be the element of $\mathcal{V}_i^+$ such that $0_M.v_i^+$ is the smallest member of $\{0_M.v\}_{v\in\mathcal{V}_i^+}$. Note that $\ell(v_i^+)$ is also the smallest element of $\ell(\mathcal{V}_i^+)$, although other elements of $\mathcal{V}_i^+$ may have equal length. Label the elements of $\mathcal{V}_i^+$ according to their length l and their lth M-ary-expansion digit. That is, let $v_i^{l,m}$ be the element of $\mathcal{V}_i^+$ that satisfies $\ell(v_i^{l,m}) = l$ and $(0_M.v_i^{l,m})_{[l]} = m$. Note that $(b_{i+1})_{[l]}$ is the number of elements in $\mathcal{V}_i^+$ with length l. Now, the probability of a leaf $v_i^{l,m} \in \mathcal{V}_i^+$ can be written as

$$p_V(v_i^{l,m}) = \frac{p_V(v_i^+)}{q_X((0_M.v_i^+)_{[\ell(v_i^+)]})} \prod_{k=\ell(v_i^+)}^{l} q_X((0_M.v_i^{l,m})_{[k]}), \tag{A.5}$$

because all leaves in $\mathcal{V}_i^+$ have their first $\ell(v_i^+) - 1$ elements in common. For convenience, let $p_i^+ = p_V(v_i^+)/q_X((0_M.v_i^+)_{[\ell(v_i^+)]})$, and note that

$$\Pr\{V \in \mathcal{V}_i^+\} \le p_i^+ \le \Pr\{V \in \mathcal{V}_i^+\}/p_{\min}. \tag{A.6}$$

This inequality holds because $0_M.v_i^+$ comes from expanding $\bar{b}_{i,i+1}$ until and including the first digit it differs from b_{i+1}. Therefore, if we set the last digit of $0_M.v_i^+$, $(0_M.v_i^+)_{[\ell(v_i^+)]}$, to zero to form a number q, then the interval $[q, q + M^{-\ell(v_i^+)+1})$ includes both $\bar{b}_{i,i+1}$ and b_{i+1}. Since $p_i^+ = \Pr\{0_M.V \in [q, q + M^{-\ell(v_i^+)+1})\}$, we arrive at the left half of (A.6). The right half of (A.6) holds because $p_V(v_i^+) \le \Pr\{V \in \mathcal{V}_i^+\}$. The part of the entropy contributed by the leaves in $\mathcal{V}_i^+$ can be upper bounded as

$$\sum_{v \in \mathcal{V}_i^+} p_V(v) \log \frac{1}{p_V(v)} = \sum_{l=1}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} p_V(v_i^{l,m}) \log \frac{1}{p_V(v_i^{l,m})} \tag{A.7}$$

$$= \sum_{l=\ell(v_i^+)}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} p_V(v_i^{l,m}) \left(\log \frac{1}{p_i^+} + \log \frac{p_i^+}{p_V(v_i^{l,m})} \right) \tag{A.8}$$

$$= \sum_{l=\ell(v_i^+)}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} p_V(v_i^{l,m}) \log \frac{1}{p_i^+} + p_i^+ \sum_{l=\ell(v_i^+)}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} \prod_{k=\ell(v_i^+)}^{l} q_X((0_M.v_i^{l,m})_{[k]}) \sum_{k'=\ell(v_i^+)}^{l} \log \frac{1}{q_X((0_M.v_i^{l,m})_{[k']})} \tag{A.9}$$

$$\le \Pr\{V \in \mathcal{V}_i^+\} \log \frac{1}{p_i^+} + p_i^+ M \sum_{l=1}^{\infty} p_{\max}^l \left(l \log \frac{1}{p_{\min}} \right) \tag{A.10}$$

$$\le \Pr\{V \in \mathcal{V}_i^+\} \left(\log \frac{1}{\Pr\{V \in \mathcal{V}_i^+\}} \right) + p_i^+ K_1 \tag{A.11}$$

$$\le \Pr\{V \in \mathcal{V}_i^+\} \left(\log \frac{1}{\Pr\{V \in \mathcal{V}_i^+\}} \right) + 2^{-n} K_{p_1}, \tag{A.12}$$

where K_1 and K_{p_1} are chosen appropriately. Equation (A.9) follows from (A.5), and Inequalities (A.11) and (A.12) follow from (A.6). The part of the entropy contributed by the leaves in $\mathcal{V}_i^-$

can be similarly upper bounded as

$$\sum_{v \in \mathcal{V}_i^-} p_V(v) \log \frac{1}{p_V(v)} \leq \Pr\{V \in \mathcal{V}_i^-\} \left(\log \frac{1}{\Pr\{V \in \mathcal{V}_i^-\}} \right) + 2^{-n} K_{p_2}.$$

Summing the bounds on the contributions to the entropy from $\mathcal{V}_i^-$ and $\mathcal{V}_i^+$, we can upper bound the entropy contributed by the leaves in $\mathcal{V}_i$ by the sum of the two:

$$\begin{aligned} \sum_{v \in \mathcal{V}_i} p_V(v) \log \frac{1}{p_V(v)} &\leq \Pr\{V \in \mathcal{V}_i^-\} \log \frac{1}{\Pr\{V \in \mathcal{V}_i^-\}} + \Pr\{V \in \mathcal{V}_i^+\} \log \frac{1}{\Pr\{V \in \mathcal{V}_i^+\}} \\ &\quad + 2^{-n}(K_{p_1} + K_{p_2}) \qquad \text{(A.13)} \\ &= 2^{-n}(n+1) + 2^{-n}(K_{p_1} + K_{p_2}), \qquad \text{(A.14)} \end{aligned}$$

where (A.14) follows from the fact that

$$\Pr\{V \in \mathcal{V}_i^-\} = \frac{\Pr\{V \in \mathcal{V}_i^-\}}{\Pr\{V \in \mathcal{V}_i^-\} + \Pr\{V \in \mathcal{V}_i^+\}} \cdot 2^{-n} \qquad \text{(A.15)}$$

$$\Pr\{V \in \mathcal{V}_i^+\} = \frac{\Pr\{V \in \mathcal{V}_i^+\}}{\Pr\{V \in \mathcal{V}_i^-\} + \Pr\{V \in \mathcal{V}_i^+\}} \cdot 2^{-n}, \qquad \text{(A.16)}$$

and that the entropy of a binary random variable is less than 1. Summing over all 2^n intervals, we have that $H(V) < n + C_\pi$ for an appropriate constant C_π that is independent of n. □

As an additional note, it is straightforward to show that the lower bound

$$E[\ell(\pi_n(D^n))] > n/H(X) \qquad \text{(A.17)}$$

holds as well. The proof follows from the fact that the tree associated with l_n has leaves whose probabilities are always less than 2^{-n}. The entropy of the leaves is therefore greater than n, which, combined with (A.4), proves the lower bound.

A.2 Proof of Subsystem Property 2

Subsystem Property 2 gives an upper bound on the expected length $E[L_{i+1}^\sigma]$ in terms of $E[L_i]$, which is obtained as follows.

Our approach is to first find an upper bound on $E[\ell(\Sigma_{i+1})|L_i^\sigma = n]$ in terms of $E[L_i|L_i^\sigma = n]$. To find such a bound, we must know the distribution of the precoder's output ε_i' (the source coder's input) conditioned $L_i^\sigma = n$. To find this distribution, we require the distribution of the precoder's input Ψ_i conditioned on $L_i^\sigma = n$. Fortunately, because of r in (2.14), we can assume that Ψ_i is, conditioned on $L_i^\sigma = n$, uniformly distributed over the set $\{0,1\}^n$. Given such an input, we asserted in Section 2.3.1.1 that the precoder π_n produces output that is a stopped sequence of random variables that are i.i.d. according to q_X. Using this characterization of the precoder π_n's output, we model the source coding of this precoder's output as follows: Let $\{\hat{X}_k\}$ and $\{\hat{Y}_k\}$ be as defined in Section 2.3.1.2. Let $\hat{L}_n = l_n(0_M.\hat{X}^\infty)$, where l_n is defined as in (2.22). The precoder π_n's output is then represented by the random variable-length tuple $\hat{X}^{\hat{L}_n}$. The transmitter sends $\hat{X}^{\hat{L}_n}$

over the channel. The receiver feeds back $\hat{Y}^{\hat{L}_n}$, and the transmitter source codes $\hat{X}^{\hat{L}_n}$ according to $\prod_{j=1}^{\hat{L}_n} p_{X|Y}(\cdot \mid \hat{Y}_j)$, which results in a stream of bits of length $\hat{L}_{\text{out},n}$. Because

$$\left\lceil -\log \prod_{j=1}^{l} p_{X|Y}(\hat{X}_j \mid \hat{Y}_j) \right\rceil + 1$$

bits are used to represent $\hat{X}^{\hat{L}_n}$, we can write the expected value of $\hat{L}_{\text{out},n}$ as

$$E[\hat{L}_{\text{out},n}] \leq 2 + \sum_{l=1}^{\infty} p_{\hat{L}_n}(l) \sum_{\hat{y}^l \in \mathcal{Y}^l} \sum_{\hat{x}^l \in \mathcal{X}^l} p_{\hat{Y}^l|\hat{X}^l}(\hat{y}^l|\hat{x}^l) p_{\hat{X}^l|\hat{L}_n}(\hat{x}^l|l) \log \frac{p_{\hat{Y}^l}(\hat{y}^l)}{p_{\hat{Y}^l|\hat{X}^l}(\hat{y}^l|\hat{x}^l) p_{\hat{X}^l}(\hat{x}^l)}. \tag{A.18}$$

We now prove the following lemma that bounds the length of $E[\hat{L}_{\text{out},n}]$:

Lemma A.2.1

$$E[\hat{L}_{\text{out},n}] \leq E[\hat{L}_n] H(X|Y) + H(\hat{L}_n) + 2 \tag{A.19}$$

Proof: We begin by expanding the logarithm in (A.18) as

$$\log \frac{p_{\hat{Y}^l}(\hat{y}^l)}{p_{\hat{Y}^l|\hat{X}^l}(\hat{y}^l|\hat{x}^l) p_{\hat{X}^l}(\hat{x}^l)} = \log \frac{p_{\hat{Y}^l|\hat{L}_n}(\hat{y}^l|l)}{p_{\hat{Y}^l|\hat{X}^l}(\hat{y}^l|\hat{x}^l) p_{\hat{X}^l|\hat{L}_n}(\hat{x}^l|l)} + \log \frac{p_{\hat{X}^l|\hat{L}_n}(\hat{x}^l|l)}{p_{\hat{X}^l}(\hat{x}^l)} - \log \frac{p_{\hat{Y}^l|\hat{L}_n}(\hat{y}^l|l)}{p_{\hat{Y}^l}(\hat{y}^l)},$$

we then obtain

$$E[\hat{L}_{\text{out},n}] = 2 + \sum_{l=1}^{\infty} p_{\hat{L}_n}(l) \left[H(\hat{X}^l|\hat{Y}^l, \hat{L}_n = l) + D(p_{\hat{X}^l|\hat{L}_n=l} \parallel p_{\hat{X}^l}) - D(p_{\hat{Y}^l|\hat{L}_n=l} \parallel p_{\hat{Y}^l}(\hat{y}^l)) \right] \tag{A.20}$$

$$\leq 2 + \sum_{l=1}^{\infty} p_{\hat{L}_n}(l) \left[H(\hat{X}^l|\hat{Y}^l, \hat{L}_n = l) + D(p_{\hat{X}^l|\hat{L}_n=l} \parallel p_{\hat{X}^l}) \right]. \tag{A.21}$$

Since $\hat{L}_n$ is a deterministic stopping rule for $\{\hat{X}_j\}$, and because $\{\hat{X}_j\}$ is an i.i.d. process and the channel is memoryless, we have that for all integers $m > 0$,

$$p_{\hat{X}_{l+1}^{l+m}|\hat{L}_n=l, \hat{X}^l} = p_{\hat{X}_{l+1}^{l+m}} \tag{A.22a}$$

$$p_{\hat{Y}_{l+1}^{l+m}|\hat{L}_n=l, \hat{Y}^l} = p_{\hat{Y}_{l+1}^{l+m}} \tag{A.22b}$$

$$p_{\hat{X}_{l+1}^{l+m}, \hat{Y}_{l+1}^{l+m}|\hat{L}_n=l, \hat{X}^l, \hat{Y}^l} = p_{\hat{X}_{l+1}^{l+m}, \hat{Y}_{l+1}^{l+m}}. \tag{A.22c}$$

From (A.22), it follows that for all integers $m > 0$,

$$\sum_{l=1}^{m} p_{\hat{L}_n}(l) D(p_{\hat{X}^l|\hat{L}_n=l} \parallel p_{\hat{X}^l}) = \sum_{l=1}^{m} p_{\hat{L}_n}(l) D(p_{\hat{X}^m|\hat{L}_n=l} \parallel p_{\hat{X}^m}) \tag{A.23}$$

$$\leq I(\hat{X}^m; \hat{L}_n) \tag{A.24}$$

$$\leq H(\hat{L}_n), \tag{A.25}$$

and

$$\sum_{l=1}^{m} p_{\hat{L}_n}(l) H(\hat{X}^l|\hat{Y}^l, \hat{L}_n = l)$$

$$= \sum_{l=1}^{m} \sum_{\hat{x}^l} \sum_{\hat{y}^l} p_{\hat{X}^l,\hat{Y}^l,\hat{L}_n}(\hat{x}^l, \hat{y}^l, l) \log \frac{p_{\hat{Y}^l|\hat{L}_n}(\hat{y}^l \mid l)}{p_{\hat{X}^l,\hat{Y}^l|\hat{L}_n}(\hat{x}^l, \hat{y}^l \mid l)} \tag{A.26}$$

$$= \sum_{l=1}^{m} \sum_{\hat{x}^m} \sum_{\hat{y}^m} p_{\hat{X}^m,\hat{Y}^m,\hat{L}_n}(\hat{x}^m, \hat{y}^m, l) \log \frac{p_{\hat{Y}^m|\hat{L}_n}(\hat{y}^m \mid l)}{p_{\hat{X}^m,\hat{Y}^m|\hat{L}_n}(\hat{x}^m, \hat{y}^m \mid l)}$$

$$- \sum_{l=1}^{m} p_{\hat{L}_n}(l) \sum_{\hat{x}^m_{l+1}} \sum_{\hat{y}^m_{l+1}} p_{\hat{X}^m_{l+1},\hat{Y}^m_{l+1}}(\hat{x}^m_{l+1}, \hat{y}^m_{l+1}) \log \frac{p_{\hat{Y}^m_{l+1}}(\hat{y}^m_{l+1})}{p_{\hat{X}^m_{l+1},\hat{Y}^m_{l+1}}(\hat{x}^m_{l+1}, \hat{y}^m_{l+1})} \tag{A.27}$$

$$\leq H(\hat{X}^m|\hat{Y}^m, \hat{L}_n) - \left(\sum_{l=1}^{m} p_{\hat{L}_n}(l)(m-l) \right) H(\hat{X}_1|\hat{Y}_1) \tag{A.28}$$

$$\leq mH(\hat{X}_1|\hat{Y}_1) - \left(\sum_{l=1}^{m} p_{\hat{L}_n}(l)(m-l) \right) H(\hat{X}_1|\hat{Y}_1) \tag{A.29}$$

$$\leq H(\hat{X}_1|\hat{Y}_1) \left(\sum_{l=1}^{m} p_{\hat{L}_n}(l) l + m \Pr\{\hat{L}_n > m\} \right) \tag{A.30}$$

$$\leq H(\hat{X}_1|\hat{Y}_1) E[\hat{L}_n]. \tag{A.31}$$

Inequality (A.29) follows from the fact that conditioning reduces entropy and that entropy is positive; (A.31) follows from writing out the sum corresponding to $E[\hat{L}_n]$ and using basic algebra. With (A.25) and (A.31) holding for all integers $m > 0$, substituting the appropriate quantities into (A.21) and taking the limit as $m \to \infty$ allows us to upper bound $E[\hat{L}_{\text{out},n}]$ according to

$$E[\hat{L}_{\text{out},n}] \leq 2 + H(\hat{L}_n) + E[\hat{L}_n]\, H(\hat{X}_1|\hat{Y}_1), \tag{A.32}$$

which completes the proof of the lemma. ▽

Lemma A.2.1 immediately implies that

$$E[\ell(\Sigma_{i+1})|L_i^{\sigma}] \leq 2 + H(L_i|L_i^{\sigma}) + E[L_i|L_i^{\sigma}]\, H(X|Y), \tag{A.33}$$

which implies, after averaging both sides over L_i^{σ}, that

$$E[\ell(\Sigma_{i+1})] \leq 2 + H(L_i) + E[L_i]\, H(X|Y) \tag{A.34}$$

$$\leq 4 + \log E[L_i] + E[L_i]H(X|Y), \tag{A.35}$$

where (A.35) follows because $H(L) < 2+\log E[L]$ for any positive, integer-valued random variable L (see, for example, Corollary 3.12 in [15] for a proof).

We may conclude from (A.35) that for $i = 0, 1, \ldots, B_N - 1$,

$$E[L_{i+1}^{\sigma}] \leq E[L_i]\, H(X|Y) + \log E[L_i] + 4 + 4 + 2E[\lceil \log L_i \rceil] + 2E[\lceil \log L_i^{\sigma} \rceil], \tag{A.36}$$

where the last three terms are due to the encodings of L_i and L_i^{σ} in Ψ_i. To express the right-hand side completely in terms of L_i, we first prove the following lemma:

Lemma A.2.2 For all $d^n \in \{0,1\}^n$,

$$\ell(\pi_n(d^n)) > n/M_{\pi}, \tag{A.37}$$

where $M_{\pi} = -\log p_{\text{min}}$ and $p_{\text{min}} = \min\{q_X(x) : q_X(x) \neq 0, x \in \mathcal{X}\}$.

Proof: The probability of the sequence $\pi_n(d^n)$ must be less than 2^{-n}. The shortest such sequence would consist entirely of the element of $\mathcal{X}$ with lowest nonzero probability, giving the sequence probability $p_{\text{min}}^{\ell(\pi_n(d^n))}$. The lemma then follows. $\bigtriangledown$

This lemma implies that $L_i^{\sigma} < L_i M_{\pi}$, which implies that we can write

$$E[L_{i+1}^{\sigma}] \leq E[L_i]\, H(X|Y) + 5\log E[L_i] + 12 + \log M_{\pi}. \tag{A.38}$$

That the logarithm is concave and monotonically increasing completes the proof that Subsystem Property 2 holds. □

A lower bound analogous to that given in Theorem B.3.2 can also be proven using the same arguments used in the proof of that theorem.

A.3 Proof of Subsystem Property 3

To bound the probability of decoding error for the mSB coding subsystem, we observe that an mSB decoding error occurs only if the transmitter sends w^{ν}, and the decoder mistakes it for c^{ν}. Let P_{cw} be the probability of this event, and let P_{wc} be the probability of the reverse error (mistaking c^{ν} for w^{ν}). If c^{ν} and w^{ν} are distinguished with the detector described in [65], which makes P_{cw} very small at the expense of making P_{wc} large (but sufficiently small), then it is shown in [65] that the choice of c^{ν} and w^{ν} given in Section 2.3.1.3 yields the bound [65] that

$$P_{cw} < \exp_2\{-\nu(E_{cw} - o_{\nu}(1))\}, \tag{A.39}$$

where E_{cw} is defined as in (2.38).

The average length of the mSB encoder's output η can be upper and lower bounded, respectively, according to

$$\kappa^2 + \nu < \eta < (\kappa^2 + \nu)/(1 - P_{\text{in}} - P_{wc}), \tag{A.40}$$

It can easily be shown that $P_{wc} = o_{\nu}(1)$ and $P_{\text{in}} = o_{\kappa}(1)$.

If we set $\nu = \lceil \alpha N \rceil$ and $\kappa = \lceil N^{1/4} \rceil$, then (A.40) implies that $\eta = \alpha N + o_N(N)$, which can be used with (A.39) to arrive at the inequality

$$P_{cw} < \exp_2\{-\eta(E_{cw} - o_N(1))\}. \tag{A.41}$$

The probability of error $P_{e,\text{term},N}$ for this mSB scheme can be written

$$P_{e,\text{term},N} = \frac{P_{\text{in}} P_{cw}}{P_{\text{in}} P_{cw} + (1 - P_{\text{in}})(1 - P_{wc})}, \tag{A.42}$$

which is less than P_{cw} for sufficiently large values of N, proving that the property holds.

As a final remark, note that we need not have chosen the "inner" code to be a bit-by-bit repetition code, but could have used a more complex code. The repetition code happens to be have very low complexity and is simple to construct. □

A.4 Proof of Lemma 2.4.1

To bound μ_{II}, we first use Subsystem Property 1 to conclude that

$$E[L_i | L_i^\sigma] \leq (L_i^\sigma + C_\pi)/H(X). \tag{A.43}$$

Taking expectations of both sides of this bound and combining with (2.40) in Subsystem Property 2, we find that

$$\begin{aligned} E[L_{i+1}] &\leq FE[L_i] + \lambda(E[L_i])/H(X) + C_\pi/H(X) && \text{(A.44)} \\ &= FE[L_i] + \tilde{\lambda}(E[L_i]), && \text{(A.45)} \end{aligned}$$

where $F = H(X|Y)/H(X)$, and $\tilde{\lambda}(x) = (\lambda(x) + C_\pi)/H(X)$. Note that $\tilde{\lambda}(x)/x \to 0$ as $x \to \infty$ and that $\tilde{\lambda}$ is a non-negative, monotonically increasing, concave function over $[1, \infty)$.

Using the recursion (A.45), we obtain

$$E[L_i] \leq F^i E[L_0] + \sum_{k=0}^{i-1} F^k \tilde{\lambda}(E[L_{i-1-k}]). \tag{A.46}$$

Using that $F > 0$, that $\tilde{\lambda}(x) > 0$ for all $x \geq 1$, that $E[L_k] \geq 1$ for $k = 0, \cdots, B_N$, and that

$$E[L_0] < (N + C_\pi)/H(X), \tag{A.47}$$

we can see that

$$\mu_{\text{II}} = \sum_{i=0}^{B_N - 1} E[L_i] \leq \frac{1}{1-F}\left(\frac{N + C_\pi}{H(X)} + \sum_{i=0}^{B_N - 1} \tilde{\lambda}(E[L_i])\right). \tag{A.48}$$

Using that $\tilde{\lambda}$ is concave over $[1, \infty)$ with the fact that $E[L_k] \geq 1$ for $k = 0, \cdots, B_N$, we can bound the second term in (A.48) according to

$$\sum_{i=0}^{B_N-1} \tilde{\lambda}(E[L_i]) \leq B_N \tilde{\lambda}\Big(\sum_{i=0}^{B_N-1} E[L_i]/B_N\Big) \tag{A.49}$$

$$= B_N \tilde{\lambda}(\mu_{\text{II}}/B_N). \tag{A.50}$$

Using (A.48) and (A.50), we can write

$$\frac{\mu_{\text{II}}}{N} \leq \frac{1}{N}\frac{1}{1-F}\Big(\frac{N+C_\pi}{H(X)} + B_N\tilde{\lambda}(\mu_{\text{II}}/B_N)\Big), \tag{A.51}$$

which, using elementary algebra, implies that

$$\frac{\mu_{\text{II}}}{N} \leq \Big(\frac{1}{H(X)-H(X|Y)}\Big)\Big(\frac{N+C_\pi}{N}\Big)\Big(\frac{1-F}{1-F-B_N\tilde{\lambda}(\mu_{\text{II}}/B_N)/\mu_{\text{II}}}\Big). \tag{A.52}$$

Since $\lim_{x\to\infty} \tilde{\lambda}(x)/x = 0$, and $\mu_{\text{II}}/B_N \to \infty$ as $N \to \infty$, the lemma follows. □

A.5 Proof of Lemma 2.4.2

To bound μ_{I}, we first use (A.37) with (2.28) to obtain

$$\tilde{L} \leq 2\ell(\Sigma_{B_N}) + 2\log t_N + 4\log L_{B_N-1} + 2\log M_\pi + 16. \tag{A.53}$$

We next bound L_{B_N-1} as follows: Since $\tilde{\lambda}$ from (A.45) is $o_N(N)$ and also represents a quantity that is finite, it follows that there is some constant C such that

$$\tilde{\lambda}(x) \leq \frac{1-F}{2}x + C, \tag{A.54}$$

so that the bound

$$E[L_{i+1}] \leq \frac{1+F}{2}E[L_i] + C \tag{A.55}$$

holds. Using (A.47) with this recursion, we see that

$$E[L_i] \leq \Big(\frac{1+F}{2}\Big)^i \frac{N}{H(X)} + C', \tag{A.56}$$

where C' is a constant. With (2.9), it follows immediately that

$$E[L_{B_N-1}] \leq C' + o_N(1). \tag{A.57}$$

This inequality in turn implies via (A.55) that $E[\ell(\Sigma_{B_N})]{=}O_N(1)$. Assuming $t_N \sim o_N(N)$, which is ensured by Subsystem Property 4, we may conclude that $E[\tilde{L}]{=}O_N(\log N)$.

Since $\kappa = \Theta_N(N^{1/4})$, it follows from (2.31) that

$$E[B_G] = 1 + o_N(1). \tag{A.58}$$

Coupled with the fact that $P_{\text{e,term},N} = o_N(1)$, the lemma follows. □

A.6 Proof of Subsystem Property 4

Here, we prove that both false alarm and missed detection probability decay at least exponentially with t_N.

To determine the probability of false alarm, suppose that $Y_1, Y_2, \ldots, Y_{t_N}$ are i.i.d. with marginal pmf p_Y. With $\emptyset^{t_N} = (a, \cdots, a)$, a being defined by (2.53), we can upper bound the probability of false alarm according to

$$\Pr\Big\{\delta_{t_N}(Y^{t_N}) = 1\Big\} \le \sum_{y^{t_N} \in \mathcal{Y}^{t_N}} p_{Y^{t_N}}(y^{t_N}) \left(\frac{p_{Y^{t_N}|X^{t_N}}(y^{t_N}|\emptyset^{t_N}[N])}{p_{Y^{t_N}}(y^{t_N})} \right)^{1/2} \tag{A.59}$$

$$\le \sum_{y^{t_N} \in \mathcal{Y}^{t_N}} \prod_{i=1}^{t_N} p_Y(y_i)^{1/2} p_{Y|X}(y_i|a)^{1/2} \tag{A.60}$$

$$= \prod_{i=1}^{t_N} \left(\sum_{k=1}^{|\mathcal{Y}|} p_Y(k)^{1/2} p_{Y|X}(k|x_i)^{1/2} \right) \tag{A.61}$$

$$\triangleq (E_\emptyset)^{t_N} \tag{A.62}$$

By the Schwartz inequality, $E_\emptyset \le 1$, with equality if and only if X and Y are independent. The probability of missed detection can also be bounded above by $(E_\emptyset)^{t_N}$. With t_N defined by (2.55), Subsystem Property 4 follows. □

A.7 Modified Scheme Reliability

To find the reliability of the modified scheme, we must find the new expected length and probability of error associated with the new coding scheme. Let $\tilde{c}(N, \nu_N)$ denote the modified counterpart to $c_{\text{DMC}}(N, \nu_N)$ defined in Section 2.4.

We have the following lemma that characterizes the length for the $\tilde{c}(N, \nu_N)$ in terms of that for the length L^* under the perfect-detection assumption:

Lemma A.7.1 With L^*_+ denoting the length associated with the $\tilde{c}(N, \nu_N)$,

$$E[L^*_+] = E[L^*] + o_N(N) \tag{A.63}$$

Proof: The increase in length $L^*_+ - L^*$ has two sources: false alarms and missed detections of the synchronization sequence.

The number of additional transmissions Δ_1^* due to false alarms satisfies

$$E[\Delta_1^*] \leq \left(\frac{E[A_{B_N}]}{t_N} + 1\right) P_{\mathrm{FA},t_N} \left(\frac{N}{\kappa}\right) \eta_N(r)(1 + o_N(1)) \tag{A.64}$$

$$= o_N(1) \tag{A.65}$$

where P_{FA,t_N} is the probability of false alarm associated with detecting $\emptyset^{t_N}[N]$. (Also recall that $A_{B_N} + 1$ is the time at which transmission of the sequence corresponding to the B_Nth iteration of the coding algorithm, and thus indicates the time at which the symbol $\emptyset^{t_N}[N]$ is transmitted.) The reasoning behind (A.64) is as follows: There are about A_{B_N}/t_N opportunities for false alarms in a particular transmission. If a false alarm occurs, the N source bits to be sent via the termination coder, where each block of κ_N bits uses about $\eta_N(r)$ channel inputs. Equation (A.65) follows by using Subsystem Property 4 with the fact that $E[A_{B_N}]$ grows linearly with N (Lemma 2.4.1).

Next, the number of additional transmissions Δ_2^* due to missed detections satisfies

$$E[\Delta_2^*] \leq \frac{t_N P_{\mathrm{MD},t_N}}{1 - P_{\mathrm{MD},t_N}} \tag{A.66}$$

$$< \frac{M_{\mathrm{MD}} t_N}{1 - M_{\mathrm{MD}}}, \tag{A.67}$$

which is derived as follows. Suppose we send the stream $\emptyset^{t_N}[N], \emptyset^{t_N}[N], \ldots$, and the detector does its hypothesis test every t_N samples. Then the number of times that $\emptyset^{t_N}[N]$ is transmitted before the first detection is a random variable with mean less than $1/(1 - P_{\mathrm{MD},t_N})$. Because we counted the first transmission of $\emptyset^{t_N}[N]$ in μ_{II} of (2.46), the number of *additional* transmissions is bounded above by (A.66). Equation A.67 follows from exploiting (2.57). Note that the additional transmissions due to transmission of $\phi(C_{\mathrm{MD}})$ is on average a constant and therefore negligible, as κ is large enough that $E[B_G]$ remains as in (A.58).

Combining the sources of additional channel uses, we obtain (A.63) as desired. □

The probability of error associated with $\tilde{c}(N, \nu_N)$ is characterized in terms of the probability of error associated with the scheme under the perfect-detection assumption by the following lemma:

Lemma A.7.2 With $P_{\mathrm{e},N}$ and $P_{\mathrm{e},+,N}$ denoting the probabilities of error associated with $c_{\mathrm{DMC}}(N, \nu_N)$ and $\tilde{c}(N, \nu_N)$, respectively,

$$P_{\mathrm{e},+,N} < P_{\mathrm{e},N} + P_{B,N}, \tag{A.68}$$

where

$$P_{B,N} \leq \left(\frac{E[A_{B_N}]}{t_N} + 1\right) P_{\mathrm{FA},t_N} \left(\frac{N}{\kappa}\right) P_{\mathrm{e,term},N} \tag{A.69}$$

$$= o_N(1)\, P_{\mathrm{e,term},N}. \tag{A.70}$$

Proof: The modified scheme introduces only one additional error event: the event that a false alarm occurs and then one of the subsequent $\frac{N}{\kappa}$ termination-coded blocks is received in error. Since the existence of a false alarm does not affect the probability of error associated with the termination-coded blocks, we arrive at (A.69) via the union bound. □

Since $P_{e,N} = (1 + o_N(1))P_{e,\text{term},N}$, Lemma A.7.2 says that the probability of error is effectively unchanged. Since Lemma A.7.1 says that the expected length is effectively unchanged as well, Lemmas A.7.1 and A.7.2 together imply that the modified scheme attains the error exponent function E_{CEC} defined in Theorem 2.4.1.

A.8 Precoding with Linear Complexity

Without loss of generality, assume that $\mathcal{X} = \{0, 1, \dots, M-1\}$. Given a sequence $d^n \in \{0,1\}^n$ to precode with π_n, let $s = 0_2.d^n + 2^{-n}Z$, where Z is a random variable uniformly distributed over $[0, 1)$. Then let $s_b = 0_2.s^{[n]}$, and let $s_t = s_b + 2^{-n}$. The precoder then finds the longest interval I of the form $I = [0_M.u_1 \cdots u_j, 0_M.u_1 \cdots u_j + M^{-j})$ such that $F_{\tilde{X}}(I) \subseteq (s_b, s_t)$ and then puts out u^j. The following algorithm gives a method for doing these steps efficiently:

1. $S_b := 0, S_t := 1, R := 1, l := 1.$

2. If $S_b > s_b$ and $S_t \leq s_t$ then go to 4. Otherwise, go to 3.

3. Compute $b_{l,m} := b_{l,m-1} + Rq_X(m-1)$ for $m = 1, 2, \dots, M$, starting with $b_{l,0} = S_b$. For $k = 0, 1, \dots, M-1$, if $s \in [b_{l,k}, b_{l,k+1})$, then set $U_l := k$, $R := Rq_X(k)$, $S_b := b_{l,k}$, $S_t := b_{l,k+1}$, $l := l + 1$, break out of the loop over k, and go to 2.

4. Halt, and return $(U_1, U_2, \dots, U_{l-1})$.

Assuming real arithmetic requires a single computation, this algorithm has complexity that is linear in the final value of $l - 1$, which equals $\ell(\pi_n(s))$.

To recover d^n, we need only recover the first n binary-expansion digits of s from $\pi_n(d^n)$. To do so, we compute $x = F_{\tilde{X}}(0_M.\pi_n(d^n))$. Then s and x share their first n binary-expansion digits. A convenient formula for $F_{\tilde{X}}$ is

$$F_{\tilde{X}}(y) = \sum_{k=1}^{\infty} \sum_{m=0}^{y_{[k]}-1} q_X(m) \prod_{i=1}^{k-1} q_X(y_{[i]}). \tag{A.71}$$

Because the M-ary-expansion digits of $0_M.\pi_n(d^n)$ are, by definition, all zero after the first $\ell(\pi_n(d^n))$ digits, and because the product term in (A.71) can be computed recursively, $F_{\tilde{X}}(0_M.\pi_n(d^n))$ can be computed with complexity that is linear in $\ell(\pi_n(d^n))$.

Like arithmetic source encoders and decoders, these methods for precoding and inversion of precoding suffer from numerical precision problems on finite-precision computers. To avoid these problems, we must use systems for arithmetic with arbitrary precision (see, for example, [43]), or we must use special rescaling methods similar to those that are used carry out arithmetic source encoding and decoding (see, for example, [63]).

A.9 Fixed-Length Precoding and Source Coding Subsystems

A.9.1 Fixed-Length Precoder

The fixed-length precoder $\pi_{\delta,n}$ maintains the general resemblance to a decoder for an arithmetic source coder held by its variable-length brother π_n. With $d^n \in \{0,1\}^n$, $i = 2^n(0_2.d^n)$, $m =$

$2^{-n}(i+1/2)$ the midpoint of the interval $[i2^{-n}, (i+1)2^{-n})$, $\tilde{X}$ defined as in Section 2.3.1.1, and Z uniformly distributed over $[0,1)$, we define $\pi_{\delta,n}$ by

$$a = F_{\tilde{X}}^{-1}(m + 2^{-n}Z) \tag{A.72}$$

$$\pi_{\delta,n}(d^n) = a^{[f_\delta(n)]}, \tag{A.73}$$

where $f_\delta(n) = \lceil n/(H(X)-\delta) \rceil$. It is clear that if $p_{X^{f_\delta(n)}}(a^{[f_\delta(n)]}) < 2^{-n-2}$, then i and therefore d^n can be recovered from $\pi_{\delta,n}(d^n)$ if Z is known. Furthermore, it is clear that the addition of $2^{-n}Z$ in (A.72) makes $\pi_{\delta,n}(W^n)$ statistically identical to $f_\delta(n)$ random variables that are i.i.d. according to q_X if W^n is uniformly distributed over $\{0,1\}^n$.

A possibility of decoding error now exists with this precoder. The probability of such an error is upper bounded by the probability that $p_{X^{f_\delta(n)}}(a_{[f_\delta(n)]}) \geq 2^{-n-2}$. Let us prove that this probability decays exponentially with n, beginning with the following lemma.

Lemma A.9.1 If $A_1, \cdots, A_n$ are i.i.d. random variables with mean $\bar{A} < \infty$, then for any $\delta > 0$ there exists a constant $F_+(\delta) > 0$ such that

$$\Pr\left\{\sum_{i=1}^{n} A_i > n(\bar{A}+\delta)\right\} < \exp\{-F_+(\delta)n\}. \tag{A.74}$$

Similarly, for any $\delta > 0$, there exists a constant $F_-(\delta)$

$$\Pr\left\{\sum_{i=1}^{n} A_i < n(\bar{A}-\delta)\right\} < \exp\{-F_-(\delta)n\}. \tag{A.75}$$

Proof: To demonstrate the first inequality, we use the Chernoff bound (see, for example, [41]), which says that for all $s > 0$,

$$\Pr\left\{\sum_{i=1}^{n} A_i > n(\bar{A}+\delta)\right\} < V(s)^n, \tag{A.76}$$

where

$$V(s) = \left(\exp\{-s\delta\}\frac{E[\exp\{sA\}]}{\exp\{s\bar{A}\}}\right). \tag{A.77}$$

Note that $V(0) = 1$ and that the derivative of V with respect to s evaluated at $s = 0$ is equal to $-\delta$. Therefore, there must exist some $s > 0$ such that $V(s) < 1$, which proves the lemma. The proof of the second inequality follows analogously. □

Corollary A.9.1 If $X_1, \cdots, X_n$ are i.i.d. according to q_X, then for all $\delta > 0$, there exists a constant $F_1(\delta) > 0$ such that

$$\Pr\{-\log p_{X^n}(X^n) < n(H(X_1)-\delta)\} < \exp\{-F_1(\delta)n\}. \tag{A.78}$$

This corollary implies that the probability of a precoding decoding error decays exponentially with $f_\delta(n)$, where the exponent is a function of δ.

A.9.2 Fixed-Length Source Coder

The fixed-length source coder $\sigma_{\delta,n}$ resembles its variable-length counterpart in Section 2.3.1.2 because it uses the same principles of arithmetic coding. With $\{X_i\}_i$ and $\{Y_i\}_i$ jointly marginally distributed according to $p_{X,Y}$ and mutually independent at different times and with $\mathcal{X} = \{0, \cdots, M_1 - 1\}$, let $\tilde{X} = 0_M.X_1X_2X_3\cdots$, and define the source coder $\sigma_{\delta,n}$ as follows:

$$a = F_{\tilde{X}|Y^n}(x^n|y^n) + p_{X^n|Y^n}(x^n|y^n)/2 \tag{A.79}$$

$$\sigma_{\delta,n}(x^n, q) = (a^{[g_\delta(n)]}), \tag{A.80}$$

where $F_{\tilde{X}_1|Y^n}$ is the conditional cdf of $\tilde{X}_1$ conditioned on Y^n. In this scenario, the sequence x^n is decodable as long as $p_{X^n|Y^n}(x^n|y^n) > 2^{-(g_\delta(n)-2)}$. Corollary A.9.2 below then implies that a decoding error occurs with probability that decays exponentially in n.

Corollary A.9.2 If $X_1, \cdots, X_n$ are i.i.d. according to q_X, and $Y_1, \cdots, Y_n$ are such that $p_{Y_i|X_i}(y|x) = q_{Y|X}(y|x)$ for all x and y, then for all $\delta > 0$, there exists a constant $F_2(\delta) > 0$ such that

$$\Pr\{-\log p_{X^n|Y^n}(X^n|Y^n) > n(H(X|Y) + \delta)\} < \exp\{-F_2(\delta)n\}. \tag{A.81}$$

A.10 Analysis of Variable-Length Interleaving

Suppose we use the same technique in Section 2.8.1 of interleaving the transmissions for two messages. Consider the first two transmissions. Their lengths L_1 and L_2 are identically distributed. Suppose computation to process L samples of data takes the same amount of time as it takes to transmit αL samples, where $\alpha = H(X|Y)/(2H(X)) < 1$. Because L_1 and L_2 are random and not guaranteed to be equal, the transmitter is guaranteed to be ready to send the second iteration of the first message only at time $L_1 + L_2 + \max(0, \alpha L_1 - L_2)$. This means the channel is idle for a number of samples

$$I_1 = \max(0, \alpha L_1 - L_2) \tag{A.82}$$

$$\leq \max(0, L_1 - L_2) \tag{A.83}$$

$$\leq |L_1 - L_2| \tag{A.84}$$

$$\leq \text{std}(L_1 - L_2) \tag{A.85}$$

$$\leq \sqrt{2}\,\text{std}(L_1), \tag{A.86}$$

where std($\cdot$) denotes the standard deviation of a random variable. Inequality (A.85) follows from the concavity of the square root function, and (A.86) follows from the independence of L_1 and L_2. Now, after sending at most I_1 samples of filler, the length-L_3 correction message for the first message is sent. We then encounter a similar situation in which the correction message for the second message cannot be sent until time $L_1 + L_2 + I_1 + L_3 + \max(0, \alpha L_2 - L_3)$. So the channel is idle for a number of samples

$$
\begin{aligned}
I_2 &= \max(0, \alpha L_2 - L_3) && \text{(A.87)}\\
&\leq \max(0, kL_2 - L_3)) && \text{(A.88)}\\
&\leq |kL_2 - L_3| && \text{(A.89)}\\
&\leq \sqrt{2}\ \max(\mathrm{std}(L_2), \mathrm{std}(L_3)), && \text{(A.90)}
\end{aligned}
$$

where $k > \alpha$ is chosen so that $E[kL_2 - L_3] = 0$. We believe that $\mathrm{std}(L_i){=}O_N(\sqrt{N}), i = 1, 2, 3$. If this belief is true, then the wasted channel inputs during the first iteration are $O_N(\sqrt{N})$. If similar behavior holds on subsequent iterations, then total channel inputs wasted is $B_N O_N(\sqrt{N})$, which is $o_N(N)$.

It remains to be shown that $\mathrm{std}(L_i){=}O_N(\sqrt{N})$, $i = 1, 2, 3$. It is easiest to show that $\mathrm{std}(L_1){=}O_N(\sqrt{N})$ if we use a modified precoder $\tilde{\pi}_n$ that behaves like $\pi_{\delta,n}$ except that it stops at a random length as soon as the probability of its output sequence falls below 2^{-n}. If L_1 is the output length of this precoder, then the following lemma implies that $\mathrm{std}(L_1){=}O_N(\sqrt{N})$:

Lemma A.10.1 If $L = \ell(\tilde{\pi}_n(D^n))$, where D^n is uniformly distributed over $\{0, 1\}^n$, then $\mathrm{var}(L){=}O_n(n)$.

Proof: It can be shown that $n/H(X) \leq E[L] \leq n/H(X) + C$. If we let $A \triangleq \sum_{i=1}^{n/H(X_1)} - \log q_X(X_i)$, where $\{X_i\}$ is a sequence of i.i.d. random variables with marginal pmf q_X, then the variance of A, $\mathrm{var}(A)$, is $O_n(n)$, and $E[A] = n$. When $n - A = d > 0$, $L - n/H(X_1) < -d/\log p_{\max}$. When $A - n = d > 0$, $n/H(X_1) - L < -d/\log p_{\max}$. Therefore, $E[(L - E[L])^2]{=}O_n(\mathrm{var}(A))$, which proves the lemma. □

A similar lemma would seem to hold for the source coder to give us the standard deviations of L_2 and L_3, but it is difficult to show because the precoder's output is not strictly i.i.d. For this reason, we are unable to show rigorously that the standard deviation of lengths of transmissions beyond the first two is $O_N(\sqrt{N})$. But, if the precoder output were exactly i.i.d. and stopped at a time independent of the data, then L_3, conditioned on L_1 would have a variance that is proportional to L_1. Averaging over L_1, L_3 has an unconditional variance proportional to expected value of L_1, which is proportional to N. Proceeding in a similar fashion for subsequent iterations, we would see that the time spent computing while using the channel idly is negligible.

Appendix B

Detailed Technical Information for Chapter 3

B.1 Ergodic Finite-Order Markov Input to Indecomposable DFSC$_f$'s

Theorem B.1.1 Let $q_{Y_1,\beta_1|X_0,\beta_0}$ be the the channel transition pmf for an indecomposable DFSC$_f$ with state-space $\mathcal{B}$. Let K be a finite positive integer, and let $q_{X_1|X_{1-K}^0}$ be a transition pmf of a Kth-order Markov process representing the channel input so that its corresponding Markov chain has only transient plus ergodic states; let the state-space associated with this process be denoted by $\mathcal{A}$. Then the set of joint input/channel states associated with $q_{Y_1,\beta_1|X_0,\beta_0}$ and $q_{X_1|X_{1-K}^0}$ contains a single ergodic class plus transient states.

Proof: Let the ergodic states of $\mathcal{A}$ be denoted $\mathcal{A}_\varepsilon = \{a_1, \cdots, a_{|\mathcal{A}_\varepsilon|}\}$. Call the chain corresponding to these states the input chain. Let the states of $\mathcal{B}$ be labeled $b_1, \cdots, b_{|\mathcal{B}|}$. Then consider the chain corresponding to the product state-space consisting of the pairs (a_i, b_j).

Let n be an integer such that for every $x^n \in \mathcal{X}^n$, there is a state b_f such that b_f is reachable under input x^n from all states in $\mathcal{B}$. Theorem 4.6.3 from [23] establishes that such an integer n exists. For some i, consider the set $U(a_i) = \{(a_i, b) : b \in \mathcal{B}\}$. There must be at least one recurrent state in $U(a_i)$, or else a_i would not be recurrent in the input chain as assumed. Let $R(a_i)$ be all the recurrent states (not necessarily in the same recurrent class) in $U(a_i)$. Select some input sequence x^n that has positive probability under the input distribution conditioned on starting in the ergodic state a_i. Then the ending state corresponding to x^n must be an ergodic state as well, because in the input chain, only ergodic states can be reached from a_i. Call this ending state a_e. Then the state (a_e, b_f) is reachable from all states in $R(a_i)$ as well as those in $U(a_i)$. Suppose s is in $R(a_i)$. Then (a_e, b_f) must be in the same class as s, since recurrent states can only reach states in their own class. But this holds for all states in $R(a_i)$, which implies that all states in $R(a_i)$ are in the same class. Now, since i was arbitrary, we can make the same argument for every i, concluding that for each $i = 1, \cdots, |\mathcal{A}_\varepsilon|$, the states in $R(a_i)$ are in the same (recurrent) class. Furthermore, there must be a way to go from a state in $R(a_i)$ to a state in $R(a_j)$ for $j \neq i$, or else a_j and a_i could not be in the same ergodic class in the input chain. Therefore, all states in $R = \cup_i R(a_i)$ are in the same (recurrent) class, i.e., there is a single recurrent class.

Now suppose R is periodic with period d. Then we can divide R into d subclasses $T_1, \cdots, T_d$ such that states in T_1 must go on the next time step to states in T_2, and so on. For any $i =$

$1, \cdots, |\mathcal{A}_\varepsilon|$, a_i is ergodic in the input chain, so each subclass T_k must contain at least one state of the form (a_i, b_j) (i.e., $U(a_i) \cap T_k \neq \emptyset$ for all $k = 1, \cdots, d$). If n from above is such that (n mod d) $= m$, then if (a_i, b_j) is in subclass T_k, then (a_e, b_f) must be in subclass T_{m+k-1}. But since (a_i, b_j) must reach (a_e, b_f) for all $j = 1, \cdots, |\mathcal{B}|$, and because $U(a_i) \cap T_k \neq \emptyset$ for all $k = 1, \cdots, d$, we find that $(a_e, b_f) \in T_{m+k-1}$ for $k = 1, \cdots, d$. Since the subclasses $\{T_k\}$ are disjoint, there can be only one subclass, and d must be 1. Hence, R is aperiodic and therefore an ergodic class. All states not in R must be transient since they are not recurrent. □

B.2 Efficiency Property of the Precoder

In the following, $\bar{\alpha}$, $\mathcal{X}$ and $\tilde{X}$ are as defined in Section 3.3.1, and $\bar{\mathcal{X}}$ is defined as in Section 3.3. The states at time k of $\mathcal{X}$ and $\bar{\mathcal{X}}$ are denoted by α_k and $\bar{\alpha}_k$, respectively.

We need the following lemma later in the proof, so we prove it now to avoid future disruptions.

Lemma B.2.1 If $\{X_k\}_k$ is a Kth-order ergodic plus transient[1] Markov process taking values in $\mathcal{X}$, then there exist constants $C < \infty$ and $p_{\max} < 1$ such that for any state $\alpha \in \mathcal{X}^K$,

$$p_{X^n|X_{1-K}^0}(x^n|\alpha) < p_{\max}^n. \tag{B.1}$$

Proof: Let P be the probability transition matrix associated with the $A = |\mathcal{X}|^K$ states of the chain. Suppose $P^A = \prod_{j=1}^A P$ contains an element with value 1. Then a deterministic trajectory through $A + 1$ states must exist. But since there are only A states in total, this trajectory must pass through one state more than once. There must then exist a deterministic trajectory that is a cycle, which implies that the chain associated with P is not ergodic plus transient, which is a contradiction. Therefore, P^A contains only values strictly less than 1. Let $\tilde{p}_{\max} < 1$ be the maximum value in the matrix P^A. Now, let $s_1, \cdots, s_{nA}$ be a state trajectory, and form the sampled state sequence $\tilde{s}^n = (s_1, s_{A+1}, s_{2A+1}, \cdots, s_{(n-1)A+1})$. Then the probability of this sampled state sequence is less than $\tilde{p}_{\max}^n$. Since the probability of s^{nA} must be less than the probability of $\tilde{s}^n$, the lemma follows. $\bigtriangledown$

We now move to a general lemma about stopping rules.

Lemma B.2.2 Let L be a deterministic stopping rule for $\mathcal{X}$ obeying $\alpha_L = \bar{\alpha}$. Then

$$E[L] = H(X^L)/H_\infty(\mathcal{X}). \tag{B.2}$$

Proof: Parse X^∞ into phrases J_i, $i = 1, 2, \cdots$, such that J_i is a trajectory from state $\bar{\alpha}$ to state $\bar{\alpha}$ that does not otherwise pass through $\bar{\alpha}$. Then $(X_1, X_2, \cdots) = (J_1, J_2, \cdots)$, and we can let Λ be a stopping rule for $\{J_i\}_{i=1}^\infty$ such that $(J_1, \cdots, J_\Lambda) = (X_1, \cdots, X_L)$. Since the J_i are i.i.d., we can use Wald's equality to conclude that $E[\Lambda] = H(J^\Lambda)/H(J_1)$. Ekroot [17] proves that $H(J_1) = H_\infty(\mathcal{X})/\mu_{\bar{\alpha}}$, where $1/\mu_{\bar{\alpha}}$ is both the probability of being in state $\bar{\alpha}$ in steady state and one over the expected length of J_i for all i. Thus,

$$E[\Lambda] = H(J^\Lambda)\mu_{\bar{\alpha}}/H_\infty(\mathcal{X}). \tag{B.3}$$

[1] For our purposes, an ergodic plus transient Markov process is always assumed to have more than one ergodic state.

It is left only to show that

$$E[L] = E[\Lambda]/\mu_{\tilde{\alpha}}. \tag{B.4}$$

Suppose (B.4) holds for some stopping rule L and corresponding stopping rule Λ. The stopping rule Λ corresponds to some ∞-ary tree in which each node has an infinite number of children, and each child corresponds to a realization of J_i. Consider extending this tree by creating an infinite number of children for some leaf whose probability is p. Then $E[\Lambda]$ increases by p, and $E[L]$ increases by $pE[\ell(J_i)] = p/\mu_{\tilde{\alpha}}$, so the equality still holds. Since the equality holds for the stopping rule for which $\Lambda = 1$ always, (B.4) holds for all deterministic stopping rules by induction. Substituting (B.4) into (B.3) and using the equivalence of J^{Λ} to X^L proves the lemma. ∇

Let S be uniformly distributed over the unit interval $[0,1)$. Letting $\hat{X}_1, \hat{X}_2, \ldots = F_{\tilde{X}}^{-1}(S)_{[1]}, F_{\text{FS},\tilde{X}}^{-1}(S)_{[2]}, \ldots$, and letting

$$L_n = l_n(F_{\text{FS},\tilde{X}}^{-1}(S)), \tag{B.5}$$

we can see that Lemma B.2.2 applies. Therefore, to complete the proof, we must show that $H(\hat{X}^{L_n}) \le n + C_\pi$, which is the purpose of the next lemma.

Lemma B.2.3 There exists a constant C_π independent of n such that

$$H(\hat{X}^{L_n}) \le n + C_\pi. \tag{B.6}$$

Proof: We find it easier to prove statements about a related stopping rule L_n' defined as follows: With

$$\begin{aligned} l_n'(u) = \min\{k\ : F_{\text{FS},\tilde{X}}([0_M.u^{[k]}, 0_M.u^{[k]} + M^{-k})) \subseteq [i2^{-n}, (i+1)2^{-n})\} \\ \text{for } u \in F_{\text{FS},\tilde{X}}^{-1}([i2^{-n}, (i+1)2^{-n})) \text{ for } i = 0, \cdots, 2^n - 1, \end{aligned} \tag{B.7}$$

let

$$L_n' = l_n'(F_{\text{FS},\tilde{X}}^{-1}(S)). \tag{B.8}$$

We upper bound $H(\hat{X}^{L_n'})$ shortly, but let us first examine the relationship between $H(\hat{X}^{L_n'})$ and $H(\hat{X}^{L_n})$. Let the final trajectory from $\tilde{\alpha}$ to $\tilde{\alpha}$ in $\hat{X}^{L_n}$ be called J_Λ. It is clear that there exists a mapping f such that $\hat{X}^{L_n} = f(\hat{X}^{L_n'}, J_\Lambda, L_n' - L_n + \ell(J_\Lambda))$, which implies that

$$\begin{aligned} H(\hat{X}^{L_n}) &\le H(\hat{X}^{L_n'}) + H(J_\Lambda) + H(L_n' - L_n + \ell(J_\Lambda)) && \text{(B.9)} \\ &\le H(\hat{X}^{L_n'}) + H(J_\Lambda) + \log E[L_n' - L_n + \ell(J_\Lambda)] + C && \text{(B.10)} \\ &\le H(\hat{X}^{L_n'}) + H(J_\Lambda) + \log E[\ell(J_\Lambda)] + C && \text{(B.11)} \\ &= H(\hat{X}^{L_n'}) + H_\infty(\tilde{X})/\mu_{\tilde{\alpha}} - \log \mu_{\tilde{\alpha}} + C, && \text{(B.12)} \end{aligned}$$

where C is a constant, and (B.10)–(B.11) follow from the fact that $0 \le L_n' - L_n + \ell(J_\lambda) \le \ell(J_\lambda)$. With this relationship, the final step is to upper bound $H(\hat{X}^{L_n'})$. To do so, let us first define a function $\tilde{l}_n$ that upper bounds l_n':

Lemma B.2.4 Let $b_i = F^{-1}_{\mathrm{FS},\tilde{X}}(i2^{-n})$ for $i = 0, \cdots, 2^n$, and let γ be the function defined in Lemma A.1.1. Next, define $\tilde{l}_n$ by

$$\tilde{l}_n(u) = \begin{cases} \gamma(u, b_{i+1}) & \text{if } \bar{b}_{i,i+1} \leq u < b_{i+1} \\ \gamma(u, b_i) & \text{if } b_i \leq u < \bar{b}_{i,i+1} \end{cases}, \qquad \forall i = 0, 1, \ldots, 2^n - 1 \tag{B.13}$$

where

$$\bar{b}_{i,i+1} = 0_M . b_{i+1}^{[\gamma(b_i, b_{i+1})]}.$$

Then

$$l'_n(u) \leq \tilde{l}_n(u) \text{ for all } u \in [0, 1). \tag{B.14}$$

Proof: Same as the proof of Lemma A.1.1. $\bigtriangledown$

With $\tilde{L}_n = \tilde{l}_n(F^{-1}_{\mathrm{FS},\tilde{X}}(S))$, and $\tilde{T}_n(u) = u^{[\tilde{l}_n(u)]}$, it is clear that $L'_n \leq \tilde{L}_n$. This inequality in turn implies that $H(\hat{X}^{L'_n}) \leq H(\hat{X}^{\tilde{L}_n})$. Letting $V = \hat{X}^{\tilde{L}_n}$, we complete proof of Lemma B.2.3 by showing that $H(V) < n + C_\pi$, where C_π is independent of n.

Let $\mathcal{V} = \{\tilde{T}_n(u)\}_{u \in [0,1)} \subset \{0, 1, \ldots, M-1\}^\dagger$. Let $\bar{x}^0_{1-K} = \bar{\alpha}$. Since the stopping rule that defines the leaves is deterministic, the probability of a leaf $v \in \mathcal{V}$ is $p_V(v) = \Pr\{F^{-1}_{\mathrm{FS},\tilde{X}}(S) \in [0_M.v, 0_M.v + M^{-\ell(v)})\}$, which implies that

$$p_V(v) = \begin{cases} \prod_{j=1}^{\ell(v)} q_{X_1|X^0_{1-K}}((0_M.v)_{[j]} | \bar{x}^0_{j-K}, (0_M.v)_{[1]}, \cdots, (0_M.v)_{[j-1]}) & \text{if } \ell(v) < K+1, \\ \prod_{j=1}^{\ell(v)} q_{X_1|X^0_{1-K}}((0_M.v)_{[j]} | (0_M.v)_{[j-K]}, \cdots, (0_M.v)_{[j-1]}) & \text{if } \ell(v) \geq K+1. \end{cases} \tag{B.15}$$

As in Section A.1, to evaluate the entropy of V, we divide the sum into manageable portions as follows. We consider for now only the leaves $v \in \mathcal{V}$ for which $0_M.v \in [b_i, b_{i+1})$ for some i. With $\mathcal{V}_i$, $\mathcal{V}_i^+$, and $\mathcal{V}_i^-$ defined as in Section A.1, we further restrict consideration to the leaves $v \in \mathcal{V}_i^+$. Defining v_i^+ and $v_i^{l,m}$ as in Section A.1, and assuming $\ell(v_i^+) > K$, we bound the probability of a leaf $v_i^{l,m} \in \mathcal{V}_i^+$ using Lemma B.2.1 according to

$$p_V(v_i^{l,m}) = \frac{p_V(v_i^+)}{q_{X_1|X^0_{1-K}}((0_M.v_i^+)_{[\ell(v_i^+)]} | (0_M.v_i^+)_{[\ell(v_i^+)-K]}, \cdots, (0_M.v_i^+)_{[\ell(v_i^+)-1]})}$$

$$\prod_{k=\ell(v_i^+)}^{l} q_{X_1|X^0_{1-K}}((0_M.v_i^{l,m})_{[k]} | (0_M.v_i^{l,m})_{[k-K]}, \cdots, (0_M.v_i^{l,m})_{[k-1]}) \tag{B.16}$$

$$\leq \frac{C p_V(v_i^+) p_{\max}^{l-\ell(v_i^+)+1}}{q_{X_1|X^0_{1-K}}((0_M.v_i^+)_{[\ell(v_i^+)]} | (0_M.v_i^+)_{[\ell(v_i^+)-K]}, \cdots, (0_M.v_i^+)_{[\ell(v_i^+)-1]})}, \tag{B.17}$$

where $p_{\max} < 1$ and $C < \infty$ are constants. An analogous bound holds if $\ell(v_i^+) \leq K$. For convenience, we also let $p_{\min}$ be the minimum of the non-zero transition probabilities. Next, we let

$$p_i^+ = p_V(v_i^+)/q_{X_1|X_{1-K}^0}((0_M.v_i^+)_{[\ell(v_i^+)]}|(0_M.v_i^+)_{[\ell(v_i^+)-1]}, \cdots, (0_M.v_i^+)_{[\ell(v_i^+)-K]}), \qquad \text{(B.18)}$$

and note that

$$\Pr\{V \in \mathcal{V}_i^+\} \leq p_i^+ \leq \Pr\{V \in \mathcal{V}_i^+\}/p_{\min} \qquad \text{(B.19)}$$

for the same reasons given for (A.6) in Section A.1. From here, Equations (A.7)–(A.16) can be used in essentially the same way as in Section A.1 to show that there is a constant $C_\pi < \infty$ such that $H(V) < n + C_\pi$, which, with (B.12), proves the lemma. ∇

The efficiency property of the precoder follows immediately from Lemmas B.2.2 and B.2.3. □

B.3 Interaction between Precoder and Source Coder

In this section, we analyze how many bits are required to encode the precoder's output conditioned on knowledge of the channel output.

The Approximation Property of the precoder allows us view the precoder's output, which is the source coder's input, as a stopped version of a Kth-order Markov process. This is one of the keys to being able to analyze the interaction of the precoding and source coding subsystems, which are each quite complicated. Another key is to note that the state of the channel at the beginning of an iteration is independent of the data the precoder puts into the channel during that iteration, conditioned on the length of the input to the precoder. To see this fact, note first that it certainly holds for the first iteration, because the input to the precoder is the sequence of original message bits, which is assumed independent of the initial channel state. For the second iteration, because of the randomization function r, we can again assume the input to the precoder $\pi_{\mathrm{FS},n}$ to be uniformly distributed over $\{0,1\}^n$, independent of the channel state. These statistical relationships continue to hold for all subsequent iterations.

These facts allow us to model the interaction between the precoder and source coder as in the following theorem, which is the main theorem of this appendix and relates the average length of the output of the source coder, which can be expressed, to within an additive constant, by the left-hand side of (B.20) below, to the average length of the input to the source coder.

Theorem B.3.1 Let $\mathcal{X} = \{X_i\}$ be a Markov process with transition pmf $q_{X_1|X_{1-K}^0}$ and initial state pmf q_{α_0}. Let $\mathcal{Y} = \{Y_i\}_{i=1}^\infty$ and $\beta = \{\beta_i\}_{i=1}^\infty$ denote the output and state processes, respectively, resulting from passing the Markov process $\mathcal{X}$ through the channel $q_{Y_1,\beta_1|X_1,\beta_0}$ without feedback with α_0 and β_0 jointly distributed according to q_{α_0,β_0}. Also let α_k be the state of $\mathcal{X}$, and let $S_k = (\alpha_k, \beta_k)$. Let L_n be the stopping rule for $\mathcal{X}$ corresponding to the action of $\pi_{\mathrm{FS},n}$ for some n.

Then

$$\sum_{l=1}^{\infty}\sum_{x^l}\sum_{y^l} p_{L_n,X^l,Y^l}(l,x^l,y^l)\log\frac{1}{w_{X^l|Y^l}(x^l|y^l)} \leq E[L_n]H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) + H(L_n) + C, \qquad \text{(B.20)}$$

where $w_{X^l|Y^l}(x^l|y^l)$ is as in (3.23), and C is a constant independent of $E[L_n]$.

Proof: For any integer $m > 0$, we can write

$$\sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} p_{L_n,X^l,Y^l}(l, x^l, y^l) \log \frac{1}{w_{X^l|Y^l}(x^l|y^l)} =$$

$$\sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} p_{L_n,X^l,Y^l}(l, x^l, y^l) \left(\log \frac{1}{p_{X^l|Y^l,L_n}(x^l|y^l, l)} + \log \frac{p_{X^l|Y^l,L_n}(x^l|y^l, l)}{w_{X^l|Y^l}(x^l|y^l)} \right). \qquad \text{(B.21)}$$

Next, we bound the first term of (B.21) according to

$$\sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} p_{L_n,X^l,Y^l}(l, x^l, y^l) \log \frac{1}{p_{X^l|Y^l,L_n}(x^l|y^l, l)}$$

$$= \sum_{l=1}^{m} \sum_{x^m} \sum_{y^m} p_{L_n,X^m,Y^m}(l, x^m, y^m) \log \frac{1}{p_{X^l|Y^l,L_n}(x^l|y^l, l)} \qquad \text{(B.22)}$$

$$= \sum_{l=1}^{m} \sum_{x^m} \sum_{y^m} p_{L_n,X^m,Y^m}(l, x^m, y^m) \left(\log \frac{p_{Y^l|L_n}(y^l|l) p_{Y_{l+1}^m|Y^l,L_n}(y_{l+1}^m|y^l, l)}{p_{X^l,Y^l|L_n}(x^l, y^l|l) p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l)} \right.$$

$$\left. + \log \frac{p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l)}{p_{Y_{l+1}^m|Y^l,L_n}(y_{l+1}^m|y^l, l)} \right) \qquad \text{(B.23)}$$

$$= \sum_{l=1}^{m} p_{L_n}(l) \sum_{x^m} \sum_{y^m} p_{X^m,Y^m|L_n}(x^m, y^m|l) \left(\log \frac{1}{p_{X^m,Y^m|L_n}(x^m, y^m|l)} - \log \frac{1}{p_{Y^m|L_n}(y^m|l)} \right) -$$

$$\sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} p_{L_n,X^l,Y^l}(l, x^l, y^l) \left(\sum_{x_{l+1}^m} \sum_{y_{l+1}^m} p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l) \right.$$

$$\left. \log \frac{1}{p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l)} \right)$$

$$+ \sum_{l=1}^{m} \sum_{y^l} p_{L_n,Y^l}(l, y^l) \sum_{y_{l+1}^m} p_{Y_{l+1}^m|Y^l,L_n}(y_{l+1}^m|y^l, l) \log \frac{1}{p_{Y_{l+1}^m|Y^l,L_n}(y_{l+1}^m|y^l, l)} \qquad \text{(B.24)}$$

$$\leq H(X^m|Y^m, L_n) - \sum_{l=1}^{m} p_{L_n}(l)(m - l)(H_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) - H_\infty(\bar{\mathcal{Y}})) + C \qquad \text{(B.25)}$$

$$\leq m H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) - \sum_{l=1}^{m} p_{L_n}(l)(m - l) H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) + C' \qquad \text{(B.26)}$$

$$= H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) \left(\sum_{l=1}^{m} p_{L_n}(l) l + m \Pr\{L_n > m\} \right) + C' \qquad \text{(B.27)}$$

$$\leq H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) E[L_n] + C' \qquad \text{(B.28)}$$

where C and C' are constants; the first term of (B.25) follows from the fact that the first term of (B.24) is upper bounded by summing from $l = 1$ to ∞; the second and third terms of (B.25) follow from Lemmas B.6.3 and Lemmas B.6.4 and the fact that Y^{L_n}, and L_n are independent of process samples at time indices greater than L_n conditioned on knowledge of the state at time L_n; (B.27) and (B.28) follow from basic algebra.

Next, consider the second term in (B.21). First, for notational clarity, for all $n > 0$, define

$$w_{X^n,Y^n}(x^n, y^n) \triangleq p_{\hat{X}^n,\hat{Y}^n}(x^n, y^n) \tag{B.29}$$

$$w_{Y^n}(y^n) \triangleq p_{\hat{Y}^n}(y^n) \tag{B.30}$$

Then the second term in (B.21) can be written

$$\begin{aligned}
\sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} p_{L_n,X^l,Y^l}(l, x^l, y^l) \log \frac{p_{X^l,Y^l|L_n}(x^l, y^l|l) w_{Y^l}(y^l)}{p_{Y^l|L_n}(y^l|l) w_{X^l,Y^l}(x^l, y^l)} \\
= \sum_{l=1}^{m} \sum_{y^l} p_{Y^l|L_n}(y^l|l) \log \frac{w_{Y^l}(y^l)}{p_{Y^l|L_n}(y^l|l)} + \\
\Bigg(\sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} \sum_{s_0} p_{L_n,X^l,Y^l,S_0}(l, x^l, y^l, s_0) \\
\log \frac{\sum_{s_0} p_{X^l,Y^l|L_n,S_0}(x^l, y^l|l, s_0) p_{S_0|L_n}(s_0|l)}{\sum_{s_0} p_{X^l,Y^l|S_0}(x^l, y^l|s_0)/(AB)} \Bigg),
\end{aligned} \tag{B.31}$$

where $A = |\mathcal{X}|^K$ and $B = |\mathcal{B}|$. Since the first term of (B.31) is negative, we can upper bound the entire expression by the second term, which can be rewritten and upper bounded as follows:

$$\sum_{l=1}^{m} p_{L_n}(l) \sum_{x^l} \sum_{y^l} \sum_{s_0} p_{X^l,Y^l,S_0|L_n}(x^l, y^l, s_0|l) \log \frac{\sum_{s_0} p_{X^l,Y^l|L_n,S_0}(x^l, y^l|l, s_0) p_{S_0|L_n}(s_0|l)}{\sum_{s_0} p_{X^l,Y^l|S_0}(x^l, y^l|s_0)/(AB)}$$

$$\leq \sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} \sum_{s_0} p_{L_n,X^l,Y^l,S_0}(l, x^l, y^l, s_0) \log \frac{p_{X^l,Y^l|L_n,S_0}(x^l, y^l|l, s_0) p_{S_0|L_n}(s_0|l)}{p_{X^l,Y^l|S_0}(x^l, y^l|s_0)/(AB)} \tag{B.32}$$

$$\leq \sum_{l=1}^{m} \sum_{x^l} \sum_{y^l} \sum_{s_0} p_{L_n,X^l,Y^l,S_0}(l, x^l, y^l, s_0) \log \frac{p_{X^l,Y^l|L_n,S_0}(x^l, y^l|l, s_0)}{p_{X^l,Y^l|S_0}(x^l, y^l|s_0)} + \log(AB) \tag{B.33}$$

$$\begin{aligned}
= \sum_{l=1}^{m} \sum_{x^m} \sum_{y^m} \sum_{s_0} p_{L_n,X^m,Y^m,S_0}(l, x^m, y^m, s_0) \\
\Bigg(\log \frac{p_{X^m,Y^m|L_n,S_0}(x^m, y^m|l, s_0)}{p_{X^m,Y^m|S_0}(x^m, y^m|s_0)} \\
- \log \frac{p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n,S_0}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l, s_0)}{p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,S_0}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, s_0)} \Bigg) + \log(AB)
\end{aligned} \tag{B.34}$$

$$\leq \sum_{l=1}^{m} \sum_{x^m} \sum_{y^m} \sum_{s_0} p_{L_n,X^m,Y^m,S_0}(l, x^m, y^m, s_0)$$

$$\left(\log \frac{p_{X^m,Y^m|L_n,S_0}(x^m, y^m|l, s_0)}{p_{X^m,Y^m|S_0}(x^m, y^m|s_0)} \right) + \log(AB) \tag{B.35}$$

$$\leq \sum_{l=1}^{m} \sum_{s_0} p_{L_n,S_0}(l, s_0) D(p_{X^m,Y^m|L_n=l,S_0=s_0} \parallel p_{X^m,Y^m|S_0=s_0}) + \log(AB) \tag{B.36}$$

$$\leq \sum_{l=1}^{\infty} \sum_{s_0} p_{L_n,S_0}(l, s_0) D(p_{X^m,Y^m|L_n=l,S_0=s_0} \parallel p_{X^m,Y^m|S_0=s_0}) + \log(AB) \tag{B.37}$$

$$\leq I(X^m, Y^m; L_n|S_0) + \log(AB) \tag{B.38}$$

$$\leq H(L_n) + \log(AB) \tag{B.39}$$

We used the log-sum inequality [14] for (B.32); to go from (B.34) to (B.38), we used that

$$\sum_{x_{l+1}^m, y_{l+1}^m} p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n,S_0}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l, s_0)$$

$$\log \frac{p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,L_n,S_0}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, l, s_0)}{p_{X_{l+1}^m,Y_{l+1}^m|X^l,Y^l,S_0}(x_{l+1}^m, y_{l+1}^m|x^l, y^l, s_0)} \geq 0. \tag{B.40}$$

The theorem is proved by summing (B.39) and (B.28) and taking the limit as $m \to \infty$. □

Using a straightforward analog of Lemma A.2.2, Inequality (3.24) follows from this theorem in a similar way that (A.38) follows from Lemma A.2.1.

Finally, it is interesting to note that the following lower bound also holds:

Theorem B.3.2 With $\mathcal{X}$, $\mathcal{Y}$, β, and L_n as in Theorem B.3.1,

$$\sum_{l=1}^{\infty} \sum_{x^l} \sum_{y^l} p_{L_n,X^l,Y^l}(l, x^l, y^l) \log \frac{1}{w_{X^l|Y^l}(x^l|y^l)} \geq E[L_n] H_\infty(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) - H(L_n) - C' \tag{B.41}$$

Proof: Letting z be the right-hand-side of (B.24), we can write for all $m > 0$,

$$z \geq H(X^m|Y^m, L_n) - \sum_{l=m+1}^{\infty} p_{L_n}(l) H(X^m|Y^m, L_n = l) -$$

$$\sum_{l=1}^{m} p_{L_n}(l)(m-l)(H_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) - H_\infty(\bar{\mathcal{Y}})) - \tilde{C} \tag{B.42}$$

$$\geq \sum_{l=1}^{m} p_{L_n}(l) l H(\bar{\mathcal{X}}|\bar{\mathcal{Y}}) - H(L_n) - \log|\mathfrak{X}| m \Pr\{L_n > m\} - \tilde{C}'. \tag{B.43}$$

Taking the limit of (B.43) as $m \to \infty$, we have $z \geq E[L_n]H(\mathcal{X}|\mathcal{Y}) - H(L_n) - \tilde{C}'$, since

$$\lim_{m\to\infty} m \Pr\{L_n > m\} = 0 \tag{B.44}$$

if $E[L_n] < \infty$. Equation (B.44) can be proven as follows: For all $m > 0$,

$$m \Pr\{L_n > m\} \leq E[L_n] - \sum_{l=1}^{m} p_{L_n}(l)l, \tag{B.45}$$

so taking the limit of both sides as $m \to \infty$ proves (B.44). Since the left-hand side of (B.41) is greater than z, the theorem is proven. □

B.4 A Fast Algorithm for Conditional Shannon-Fano Source Coding

In this section, we describe an algorithm for computing the essential quantities that lead to a computationally efficient algorithm for conditional Shannon-Fano (arithmetic) coding for a Markov process passing through a finite-state channel without feedback.

Suppose we are given sequences $\hat{x}^n$ and $\hat{y}^n$ and would like to perform the coding described in Section 3.3.2. The primary obstacle then is the computation of

$$F_{\hat{X}|\hat{Y}^n}(0_M.\hat{x}^n|\hat{y}^n) = \sum_{k=1}^{n} \sum_{x=0}^{\hat{x}_k - 1} p_{\hat{X}^k|\hat{Y}^n}(\hat{x}^{k-1}, x|\hat{y}^n), \tag{B.46}$$

where $\{\hat{X}_k\}$, and $\{\hat{Y}_k\}$ are as in Section 3.3.2.

We focus then on efficient computation of $p_{\hat{X}^j|\hat{Y}^n}(\hat{x}^{j-1}, x_j|\hat{y}^n)$ for all $x_j \in \mathcal{X}$ and for all $j = 1, \cdots, n$. To do this we use concepts similar to those used in the algorithm of Bahl et al. [5]

First, assume we have the value of $p_{\hat{Y}^n_{j+1}|\hat{S}_j}(\hat{y}^n_{j+1}|s_j)$ for all possible values of s_j, where $\{\hat{S}_k\}$ is the joint input/channel state corresponding to $\{\hat{X}_k\}$, $\{\hat{Y}_k\}$, and $\{\hat{\beta}_k\}$. Also suppose we have computed $p_{\hat{X}^j,\hat{S}_j|\hat{Y}^j}(\hat{x}^{j-1}, x_j, s_j|\hat{y}^j)$ for all values of x_j and s_j. Then

$$p_{\hat{X}^j,\hat{S}_j|\hat{Y}^n}(\hat{x}^{j-1}, x_j, s_j|\hat{y}^n) = \frac{p_{\hat{X}^j,\hat{S}_j|\hat{Y}^j}(\hat{x}^{j-1}, x_j, s_j|\hat{y}^j) p_{\hat{Y}^n_{j+1}|\hat{S}_j,\hat{X}^j,\hat{Y}^j}(\hat{y}^n_{j+1}|s_j, \hat{x}^{j-1}, x_j, \hat{y}^j)}{p_{\hat{Y}^n_{j+1}|\hat{Y}^j}(\hat{y}^n_{j+1}|\hat{y}^j)} \tag{B.47}$$

$$= \frac{p_{\hat{X}^j,\hat{S}_j|\hat{Y}^j}(\hat{x}^{j-1}, x_j, s_j|\hat{y}^j) p_{\hat{Y}^n_{j+1}|\hat{S}_j}(\hat{y}^n_{j+1}|s_j)}{p_{\hat{Y}^n_{j+1}|\hat{Y}^j}(\hat{y}^n_{j+1}|\hat{y}^j)}, \tag{B.48}$$

which should be computed for all s_j and x_j.

To find $p_{\hat{Y}^n_{j+1}|\hat{Y}^j}(\hat{y}^n_{j+1}|\hat{y}^j)$ in the denominator of (B.48) we use

$$p_{\hat{Y}^n_{j+1}|\hat{Y}^j}(\hat{y}^n_{j+1}|\hat{y}^j) = \sum_{s_j} p_{\hat{Y}^n_{j+1}|\hat{S}_j}(\hat{y}^n_{j+1}|s_j) p_{\hat{S}_j|\hat{Y}^j}(s_j|\hat{y}^j), \tag{B.49}$$

where we compute $p_{\hat{S}_j|\hat{Y}^j}(s_j|\hat{y}^j)$ through the following recursive computation of $p_{\hat{S}_j,\hat{Y}^j}(s_j,\hat{y}^j)$:

$$p_{\hat{S}_{j+1},\hat{Y}^{j+1}}(s_{j+1},\hat{y}^{j+1}) = \sum_{s_j} p_{\hat{S}_j,\hat{Y}^j}(s_j,\hat{y}^j)p_{\hat{S}_{j+1},\hat{Y}_{j+1}|\hat{S}_j,\hat{Y}^j}(s_{j+1},\hat{y}_{j+1}|s_j,\hat{y}^j) \tag{B.50}$$

$$= \sum_{s_j} p_{\hat{S}_j,\hat{Y}^j}(s_j,\hat{y}^j)p_{\hat{S}_{j+1},\hat{Y}_{j+1}|\hat{S}_j}(s_{j+1},\hat{y}_{j+1}|s_j) \tag{B.51}$$

Then to find $p_{\hat{S}_j|\hat{Y}^j}(s_j|\hat{y}^j)$ we just divide by $p_{\hat{Y}^j}(\hat{y}^j)$, which can be computed by summing the joint pmf over s_j.

Now, to find all $p_{\hat{Y}^n_{j+1}|\hat{S}_j}(\hat{y}^n_{j+1}|s_j)$ for all j and all possible values of s_j, we start with $p_{\hat{Y}_n|\hat{S}_{n-1}}(\hat{y}_n|s_{n-1})$ as given for all s_{n-1}, and use the following recursion:

$$p_{\hat{Y}^n_j|\hat{S}_{j-1}}(\hat{y}^n_j|s_{j-1}) = \sum_{s_j} p_{\hat{S}_j,\hat{Y}_j|\hat{S}_{j-1}}(s_j,\hat{y}_j|s_{j-1})p_{\hat{Y}^n_{j+1}|\hat{S}_{j-1},\hat{S}_j,\hat{Y}_j}(\hat{y}^n_{j+1}|s_{j-1},s_j,\hat{y}_j) \tag{B.52}$$

$$= \sum_{s_j} p_{\hat{S}_j,\hat{Y}_j|\hat{S}_{j-1}}(s_j,\hat{y}_j|s_{j-1})p_{\hat{Y}^n_{j+1}|\hat{S}_j}(\hat{y}^n_{j+1}|s_j). \tag{B.53}$$

Finally, we find $p_{\hat{X}^{j+1},\hat{S}_{j+1}|\hat{Y}^{j+1}}(\hat{x}^j,x_{j+1},s_j|\hat{y}^j)$ recursively as follows:

$$p_{\hat{X}^{j+1},\hat{S}_{j+1}|\hat{Y}^{j+1}}(\hat{x}^j,x_{j+1},s_{j+1}|\hat{y}^{j+1})$$

$$= p_{\hat{X}^{j+1},\hat{S}_{j+1}|\hat{Y}^{j+1}}(\hat{x}^j,x_{j+1},s_{j+1}|\hat{y}^j,\hat{y}_{j+1}) \tag{B.54}$$

$$= \frac{\sum_{s_j} p_{\hat{X}^j,\hat{S}_j|\hat{Y}^j}(\hat{x}^j,s_j|\hat{y}^j)p_{\hat{Y}_{j+1},\hat{S}_{j+1}|\hat{X}_{j+1},\hat{S}_j}(\hat{y}_{j+1},s_{j+1}|x_{j+1},s_j)p_{\hat{X}_{j+1}|\hat{S}_j}(x_{j+1}|s_j)}{\sum_{s_j} p_{\hat{Y}_{j+1}|\hat{S}_j,\hat{Y}^j}(\hat{y}_{j+1}|s_j,\hat{y}^j)p_{\hat{S}_j|\hat{Y}^j}(s_j|\hat{y}^j)} \tag{B.55}$$

In this way, we determine $p_{\hat{X}^j,\hat{S}_j|\hat{Y}^n}(\hat{x}^{j-1},x_j,s_j|\hat{y}^n)$ for all s_j, $x_j \leq \hat{x}_j - 1$, and $j = 1,\cdots,n$. It is then straightforward to determine $p_{\hat{X}^j|Y^n}(\hat{x}^{j-1},x_j|\hat{y}^n)$ by summing over the appropriate states.

Thus, to encode, the evaluation of $F_{\tilde{X}|\hat{Y}^n}(0_M.\hat{x}^n|\hat{y}^n)$, which makes up the primary computational load, can be done with computational complexity that is linear in n. Decoding proceeds as follows: The decoder has a point $s \in [0,1)$; it chooses $\hat{x}_1$ such that $s \in [F_{\tilde{X}|\hat{Y}^n}(0_M.\hat{x}_1|\hat{y}^n),$ $F_{\tilde{X}|\hat{Y}^n}(0_M.(\hat{x}_1+1)|\hat{y}^n))$, requiring at most computation of $p_{\hat{X}_1|\hat{Y}^n}(x|\hat{y}^n)$ for all $x \in \mathcal{X}$; once it has determined $\hat{x}_1$, it finds $\hat{x}_2$ such that $s \in [F_{\tilde{X}|\hat{Y}^n}(0_M.\hat{x}_1\hat{x}_2|\hat{y}^n), F_{\tilde{X}|\hat{Y}^n}(0_M.\hat{x}_1(\hat{x}_2+1)|\hat{y}^n))$, which requires at most computation of $p_{\hat{X}^2|\hat{Y}^n}(\hat{x}_1x|\hat{y}^n)$ for all $x \in \mathcal{X}$. Thus, using the recursions that are developed in this section, decoding also requires a number of computations that is linear in n.

B.5 Synchronization-Sequence Detection

Here, we prove that a sequence $\emptyset^{t_N}[N]$ can be chosen from $\mathcal{X}^{t_N}$ so that the probability of false alarm and missed detection both decay exponentially with the sequence length t_N.

Let $\hat{\mathscr{X}}$, $\hat{\mathscr{Y}}$, and $\{\hat{\beta}_i\}$ be as defined in Section 3.3.2, and let $\mathscr{X}$, $\mathscr{Y}$, and $\{\beta_i\}$ be as in Theorem B.3.1. Let α_k, and $\hat{\alpha}_k$ be the states of $\mathscr{X}$ and $\hat{\mathscr{X}}$, respectively, let $S_k = (\alpha_k,\beta_k)$ and $\hat{S}_k = (\hat{\alpha}_k,\hat{\beta}_k)$, and let $A = |\mathcal{X}|^K$ and $B = |\mathcal{B}|$. All other notation is taken from Section 3.3.3.

We begin by proving that the average probability of false alarm decays exponentially with t when detecting any sequence x^t with $\delta_{\text{FS},t}$.

Theorem B.5.1 For all t, and for all $x^t \in \mathcal{X}^t$,

$$\Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1\} < B2^{-\omega(t)}. \tag{B.56}$$

Proof: Assume for the moment that β_0 is uniformly distributed over $\mathcal{B}$ so that $p_{Y^t}(y^t) = p_{\hat{Y}^t}(y^t)$ for all $y^t \in \mathcal{Y}^t$. Using standard techniques, it can be shown that

$$\Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1\} \leq \sum_{y^t} p_{Y^t}(y^t)\left(\frac{p_{\hat{Y}^t|\hat{X}^t}(y^t|x^t)}{p_{\hat{Y}^t}(y^t)2^{\omega(t)}}\right) \tag{B.57}$$

$$= 2^{-\omega(t)} \sum_{y^t} p_{\hat{Y}^t|\hat{X}^t}(y^t|x^t) \tag{B.58}$$

$$= 2^{-\omega(t)}. \tag{B.59}$$

It must be also true that

$$\Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1\} = \sum_{\tilde{\beta}_0} \frac{1}{B} \Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1 | \beta_0 = \tilde{\beta}_0\}, \tag{B.60}$$

which implies that for all $\tilde{\beta}_0 \in \mathcal{B}$

$$B \Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1\} \geq \Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1 | \beta_0 = \tilde{\beta}_0\}. \tag{B.61}$$

Now, relaxing the constraint that β_0 be uniformly distributed over $\mathcal{B}$, it must be true that for any distribution for β_0,

$$\Pr\{\delta_{\text{FS},t}(Y^t, x^t) = 1\} \leq B2^{-\omega(t)}, \tag{B.62}$$

which completes the proof of the theorem. □

We now adapt the coding theorem for DFSC's in [23] to prove that missed-detection probability decays exponentially with t as well:

Theorem B.5.2 There exists $E_{\text{MD}} > 0$ such that for all integers $t > 0$,

$$\Pr\{\delta_{\text{FS},t}(Y^t, X^t) = 0\} < \exp_2\{-t(E_{\text{MD}} + o_t(1))\}. \tag{B.63}$$

Proof: We first need the following lemmas.

Lemma B.5.1 For all $s \in [0, 1]$,

$$\Pr\{\delta_{\text{FS},t}(Y^t, X^t) = 0\} < \exp_2\{-t(F_t(s) - \omega(t) - \log(AB)/t)\}, \tag{B.64}$$

where

$$F_t(s) = -s\frac{\log(AB^2)}{t} + \min_{\tilde{\alpha}_0,\tilde{\beta}_0} E_t(\tilde{\alpha}_0,\tilde{\beta}_0,s) \tag{B.65}$$

$$E_t(\tilde{\alpha}_0,\tilde{\beta}_0,s) = -\frac{1}{t}\log\sum_{x^t,y^t} p_{X^t|\alpha_0}(x^t|\tilde{\alpha}_0)p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\tilde{\beta}_0)^{1-s}p_{\hat{Y}^t}(y^t)^s. \tag{B.66}$$

Proof: Assume for the moment that $p_{\alpha_0,\beta_0}(\alpha_0,\beta_0) = 1/(AB)$ for all α_0,β_0. Under this assumption, $p_{Y^t|X^t}(y^t|x^t) = p_{\hat{Y}^t|\hat{X}^t}(y^t|x^t)$ for all $y^t \in \mathcal{Y}^t, x^t \in \mathcal{X}^t$. Then for all $0 \le s \le 1$,

$$\Pr\{\delta_{\text{FS},t}(Y^t,X^t)=0\} = \sum_{x^t,y^t} p_{X^t}(x^t)p_{Y^t|X^t}(y^t|x^t)\Pr\{\delta_{\text{FS},t}(Y^t,X^t)=0|Y^t=y^t,X^t=x^t\} \tag{B.67}$$

$$\le \sum_{x^t,y^t} p_{X^t}(x^t)p_{Y^t|X^t}(y^t|x^t)\left(\frac{2^{\omega(t)}p_{\hat{Y}}(y^t)}{p_{\hat{Y}^t|\hat{X}^t}(y^t|x^t)}\right)^s \tag{B.68}$$

$$\le 2^{\omega(t)}\sum_{x^t,y^t} p_{X^t}(x^t)p_{\hat{Y}^t|\hat{X}^t}(y^t|x^t)^{1-s}p_{\hat{Y}^t}(y^t)^s \tag{B.69}$$

$$= 2^{\omega(t)}\sum_{x^t,y^t}\sum_{\alpha_0} p_{X^t|\alpha_0}(x^t|\alpha_0)/A\left(\sum_{\beta_0} p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\beta_0)/B\right)^{1-s} p_{\hat{Y}^t}(y^t)^s \tag{B.70}$$

$$\le 2^{\omega(t)}\sum_{x^t,y^t}\sum_{\alpha_0} p_{X^t|\alpha_0}(x^t|\alpha_0)/A\sum_{\beta_0}\left(p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\beta_0)/B\right)^{1-s} p_{\hat{Y}^t}(y^t)^s \tag{B.71}$$

$$= 2^{\omega(t)}\frac{1}{AB^{1-s}}\sum_{\beta_0}\sum_{\alpha_0}\sum_{x^t,y^t} p_{X^t|\alpha_0}(x^t|\alpha_0)p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\beta_0)^{1-s}p_{\hat{Y}^t}(y^t)^s \tag{B.72}$$

$$\le 2^{\omega(t)}\frac{AB}{AB^{1-s}}\max_{\alpha_0,\beta_0}\sum_{x^t,y^t} p_{X^t|\alpha_0}(x^t|\alpha_0)p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\beta_0)^{1-s}p_{\hat{Y}^t}(y^t)^s \tag{B.73}$$

$$\le 2^{\omega(t)}(AB^2)^s\max_{\alpha_0,\beta_0}\sum_{x^t,y^t} p_{X^t|\alpha_0}(x^t|\alpha_0)p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\beta_0)^{1-s}p_{\hat{Y}^t}(y^t)^s. \tag{B.74}$$

Inequality (B.68) is a standard bound adapted from [23], (B.69) follows from the fact that $2^{\omega(t)s} \le 2^{\omega(t)}$ for $s \in [0,1]$, (B.70) follows from the assumption that the initial joint state distribution is uniform, (B.71) follows from the fact that

$$\left(\sum_i a_i\right)^s \le \sum_i a_i^s \tag{B.75}$$

for $s \in [0,1]$ [23], and (B.72)–(B.74) are straightforward.

Since

$$\Pr\{\delta_{\mathrm{FS},t}(Y^t,X^t)=0\} = \sum_{\tilde{\alpha}_0,\tilde{\beta}_0} \Pr\{\delta_{\mathrm{FS},t}(Y^t,X^t)=0|\alpha_0=\tilde{\alpha}_0,\beta_0=\tilde{\beta}_0\}/(AB), \tag{B.76}$$

we must have that

$$\Pr\{\delta_{\mathrm{FS},t}(Y^t,X^t)=0|\alpha_0=\tilde{\alpha}_0,\beta_0=\tilde{\beta}_0\} \leq AB\Pr\{\delta_{\mathrm{FS},t}(Y^t,X^t)=0\}, \tag{B.77}$$

which implies that (B.64) holds for *any* initial state distribution p_{α_0,β_0}. ∇

Paralleling the development of the coding theorem for DFSC's in [23], we next prove the following lemma.

Lemma B.5.2 For any $s \in [0,1]$, and any positive integers l, t such that $l < t$,

$$tF_t(s) \geq lF_l(s) + (t-l)F_{t-l}(s) \tag{B.78}$$

Proof: Let α_0^*, and β_0^* be the values that minimize $E_t(\alpha_0,\beta_0,s)$ for a given value of s.

$$\begin{aligned}
\exp_2\{-tF_t(s)\} &= (AB^2)^s \sum_{x^t,y^t} p_{X^t|\alpha_0}(x^t|\alpha_0^*)p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(y^t|x^t,\beta_0^*)^{1-s}p_{\hat{Y}^t}(y^t)^s \qquad \text{(B.79)}\\
&= (AB^2)^s \sum_{x^l,y^l}\sum_{x_{l+1}^t,y_{l+1}^t} p_{X^l|\alpha_0}(x^l|\alpha_0^*)p_{X_{l+1}^t|X^l,\alpha_0}(x_{l+1}^t|x^l,\alpha_0^*)\\
&\quad \left(\sum_{\beta_l} p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)p_{\hat{Y}_{l+1}^t|\hat{\beta}_l,\hat{X}_{l+1}^t}(y_{l+1}^t|\beta_l,x_{l+1}^t)\right)^{1-s}\\
&\quad \left(\sum_{s_l} p_{\hat{Y}^l,\hat{S}_l}(y^l,s_l)p_{\hat{Y}_{l+1}^t|\hat{S}_l}(y_{l+1}^t|s_l)\right)^s \qquad \text{(B.80)}\\
&\leq (AB^2)^s \sum_{x^l,y^l}\sum_{x_{l+1}^t,y_{l+1}^t} p_{X^l|\alpha_0}(x^l|\alpha_0^*)p_{X_{l+1}^t|X^l,\alpha_0}(x_{l+1}^t|x^l,\alpha_0^*)\\
&\quad \left(\sum_{\beta_l} p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s}p_{\hat{Y}_{l+1}^t|\hat{\beta}_l,\hat{X}_{l+1}^t}(y_{l+1}^t|\beta_l,x_{l+1}^t)^{1-s}\right)\\
&\quad \left(\sum_{s_l} p_{\hat{Y}^l,\hat{S}_l}(y^l,s_l)\right)^s \left(\sum_{s_l} p_{\hat{Y}_{l+1}^t|\hat{S}_l}(y_{l+1}^t|s_l)\right)^s \qquad \text{(B.81)}\\
&\leq (AB^2)^s \sum_{x^l,y^l}\sum_{x_{l+1}^t,y_{l+1}^t} p_{X^l|\alpha_0}(x^l|\alpha_0^*)p_{X_{l+1}^t|X^l,\alpha_0}(x_{l+1}^t|x^l,\alpha_0^*)\\
&\quad \left(\sum_{\beta_l} p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s}p_{\hat{Y}_{l+1}^t|\hat{\beta}_l,\hat{X}_{l+1}^t}(y_{l+1}^t|\beta_l,x_{l+1}^t)^{1-s}\right)
\end{aligned}$$

$$p_{\hat{Y}^l}(y^l)^s (AB)^s \left(\sum_{s_l} \frac{1}{AB} p_{\hat{Y}^t_{l+1}|\hat{S}_l}(y^t_{l+1}|s_l) \right)^s \tag{B.82}$$

$$\leq (AB^2)^s \sum_{x^l,y^l} \sum_{x^t_{l+1},y^t_{l+1}} p_{X^l|\alpha_0}(x^l|\alpha_0^*) p_{X^t_{l+1}|X^l,\alpha_0}(x^t_{l+1}|x^l,\alpha_0^*)$$
$$\left(\sum_{\beta_l} p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s} p_{\hat{Y}^t_{l+1}|\hat{\beta}_l,\hat{X}^t_{l+1}}(y^t_{l+1}|\beta_l,x^t_{l+1})^{1-s} \right)$$
$$p_{\hat{Y}^l}(y^l)^s (AB)^s p_{\hat{Y}^t_{l+1}}(y^t_{l+1})^s \tag{B.83}$$

$$\leq (AB^2)^s \sum_{\beta_l} \sum_{x^l,y^l} \sum_{x^t_{l+1},y^t_{l+1}} p_{X^l|\alpha_0}(x^l|\alpha_0^*) p_{X^t_{l+1}|X^l,\alpha_0}(x^t_{l+1}|x^l,\alpha_0^*) p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s}$$
$$p_{\hat{Y}^t_{l+1}|\hat{\beta}_l,\hat{X}^t_{l+1}}(y^t_{l+1}|\beta_l,x^t_{l+1})^{1-s} p_{\hat{Y}\,l}(y^l)^s (AB)^s p_{\hat{Y}^t_{l+1}}(y^t_{l+1})^s \tag{B.84}$$
$$= (AB)^s (AB^2)^s \sum_{\beta_l} \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\alpha_0^*) p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s} p_{\hat{Y}^l}(y^l)^s$$
$$\sum_{x^t_{l+1},y^t_{l+1}} p_{X^t_{l+1}|X^l,\alpha_0}(x^t_{l+1}|x^l,\alpha_0^*) p_{\hat{Y}^t_{l+1}|\hat{\beta}_l,\hat{X}^t_{l+1}}(y^t_{l+1}|\beta_l,x^t_{l+1})^{1-s} p_{\hat{Y}^t_{l+1}}(y^t_{l+1})^s \tag{B.85}$$
$$\leq (AB)^s (AB^2)^s \sum_{\beta_l} \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\alpha_0^*) p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s} p_{\hat{Y}^l}(y^l)^s$$
$$\left(\max_{\tilde{\beta}_l,\tilde{\alpha}_l} \sum_{x^t_{l+1},y^t_{l+1}} p_{X^t_{l+1}|\alpha_l}(x^t_{l+1}|\tilde{\alpha}_l) p_{\hat{Y}^t_{l+1}|\hat{\beta}_l,\hat{X}^t_{l+1}}(y^t_{l+1}|\tilde{\beta}_l,x^t_{l+1})^{1-s} p_{\hat{Y}^t_{l+1}}(y^t_{l+1})^s \right) \tag{B.86}$$
$$\leq (AB)^s \sum_{\beta_l} \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\alpha_0^*) p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s} p_{\hat{Y}^l}(y^l)^s$$
$$\times \exp_2\{-(t-l)F_{t-l}(s)\} \tag{B.87}$$
$$= (AB)^s \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\alpha_0^*) B \left(\sum_{\beta_l} \frac{1}{B} p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*)^{1-s} \right) p_{\hat{Y}^l}(y^l)^s$$
$$\times \exp_2\{-(t-l)F_{t-l}(s)\} \tag{B.88}$$
$$\leq (AB)^s \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\alpha_0^*) B \left(\sum_{\beta_l} \frac{1}{B} p_{\hat{Y}^l,\hat{\beta}_l|\hat{X}^l,\hat{\beta}_0}(y^l,\beta_l|x^l,\beta_0^*) \right)^{1-s} p_{\hat{Y}^l}(y^l)^s$$
$$\times \exp_2\{-(t-l)F_{t-l}(s)\} \tag{B.89}$$
$$= (AB)^s \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\alpha_0^*) B^s p_{\hat{Y}^l|\hat{X}^l,\hat{\beta}_0}(y^l|x^l,\beta_0^*)^{1-s} p_{\hat{Y}^l}(y^l)^s$$
$$\times \exp_2\{-(t-l)F_{t-l}(s)\} \tag{B.90}$$
$$\leq (AB^2)^s \max_{\tilde{\alpha}_0,\tilde{\beta}_0} \sum_{x^l,y^l} p_{X^l|\alpha_0}(x^l|\tilde{\alpha}_0) p_{\hat{Y}^l|\hat{X}^l,\hat{\beta}_0}(y^l|x^l,\tilde{\beta}_0)^{1-s} p_{\hat{Y}^l}(y^l)^s$$

$$\times \exp_2\{-(t-l)F_{t-l}(s)\} \tag{B.91}$$

$$= \exp_2\{-lF_l(s) - (t-l)F_{t-l}(s)\}. \tag{B.92}$$

Inequality (B.80) follows from (3.8)–(3.13), and (B.81) follows (B.75) plus the fact that

$$\sum_i a_i b_i \leq \left(\sum_i a_i\right)\left(\sum_j b_j\right) \tag{B.93}$$

when $a_i \geq 0$ and $b_i \geq 0$ for all i. Eq. (B.82) is straightforward, (B.83) follows from the uniform initial state distribution used to generate $\{\hat{Y}_k\}$, and (B.84) and (B.85) are just rearrangements of (B.83). Expressions (B.86)–(B.88) are straightforward. Inequality (B.89) follows from the concavity of the function $f(x) = x^s$ for $s \in [0,1]$ and $x > 0$, which implies that for $s \in [0,1]$,

$$\sum_i p_i a_i^s \leq \left(\sum_i p_i a_i\right)^s \tag{B.94}$$

when $\sum_i p_i = 1$, $p_i \geq 0$, and $a_i \geq 0$ for all i [23]. Expressions (B.90)–(B.92) are straightforward and complete the proof. $\bigtriangledown$

Lemma B.5.3 For all $s \in [0,1]$ and for all $\tilde{\alpha}_0$ and $\tilde{\beta}_0$, $E_t(\tilde{\alpha}_0, \tilde{\beta}_0, s) \geq -\log|\mathcal{X}||\mathcal{Y}|$.

Proof: Follows from the fact that the summands in (B.66) are less than 1, and that the sum is over $|\mathcal{X}|^t|\mathcal{Y}|^t$ terms. $\bigtriangledown$

Lemma B.5.4 For each $s \in [0,1]$, if $\{F_t(s)\}_t$ is bounded above, then

$$F_\infty(s) \stackrel{\triangle}{=} \liminf_{t\to\infty} F_t(s) = \sup_t F_t(s) \tag{B.95}$$

Proof: For all $s \in [0,1]$, Lemma B.5.3 implies that the sequence $\{F_t(s)\}_t$ is bounded below. The lemma then follows immediately from using Lemma B.5.2 above with Lemma 4A.2 from [23]. $\bigtriangledown$

Lemma B.5.5 There exists $s \in [0,1]$ such that

$$F_\infty(s) > 0. \tag{B.96}$$

Proof: Suppose that $\{F_t(s)\}_t$ is bounded above for all s. By elementary calculus, we find that for all $\tilde{\alpha}_0$ and $\tilde{\beta}_0$,

$$Q_t(\tilde{\alpha}_0, \tilde{\beta}_0) \stackrel{\triangle}{=} \left.\frac{\partial E_t(\tilde{\alpha}_0, \tilde{\beta}_0, s)}{\partial s}\right|_{s=0} = \sum_{x^t} p_{X^t|\alpha_0}(x^t|\tilde{\alpha}_0) D(p_{\hat{Y}^t|\hat{X}^t,\hat{\beta}_0}(\cdot|x^t, \tilde{\beta}_0) \parallel p_{\hat{Y}^t}(\cdot)). \tag{B.97}$$

Since the channel is assumed to be indecomposable, for sufficiently large t, we must have $Q_t(\tilde{\alpha}_0, \tilde{\beta}_0) > 0$ unless X^t is independent of Y^t for all t, which would imply a mutual information of 0 under this input distribution. If this is the case, we must change the input distribution. If the mutual information is 0 for all input distributions, then the channel capacity is 0, and we cannot communicate reliably over it at any rate. But if the channel capacity is greater than 0, there exists some input distribution so that (B.97) is positive. Furthermore, (B.97) can easily shown to be non-decreasing in t.

Letting α_0^* and β_0^* be the minimizing values for $F_t(s)$, we have that for sufficiently large t,

$$\frac{1}{t}\log AB^2 < Q_t(\alpha_0^*,\beta_0^*)/2. \tag{B.98}$$

For such t, then

$$\left.\frac{\partial F_t(s)}{\partial s}\right|_{s=0} > 0. \tag{B.99}$$

Since $F_t(0) = 0$, and F_t is continuous in s, it must be true that there is some s for which $F_t(s) > 0$. Since we assumed $\{F_t(s)\}_t$ to be bounded above, $F_\infty(s) \geq F_t(s)$ according to Lemma B.5.4, and the lemma follows for this case.

Now suppose that there exists $s_0 \in [0,1]$ such that $\{F_t(s_0)\}_t$ is not bounded above. Since $\{F_t(s_0)\}_t$ is still bounded below, it can be shown via Lemma B.5.2 and the arguments in the proof of Lemma 4A.2 of [23] that $\liminf_{t\to\infty} F_t(s_0) > C$, for any constant C. This fact completes the proof of the lemma. ∇

Let $s \in [0,1]$ be such that $F_\infty(s) > 0$. If $\{F_t(s)\}_t$ is unbounded, then the theorem clearly follows for any value of E_{MD}. If not, we use that since $\liminf_{t\to\infty} F_t(s) = F_\infty(s)$, there exists t_0 such that for all $t > t_0$,

$$(F_t(s) - \omega(t) - \log(AB)/t) > F_\infty(s)/2 \tag{B.100}$$

Combined with (B.64), the theorem follows. □

This theorem states the average missed-detection probability decays exponentially with t_N when $\emptyset^{t_N}[N]$ is random with the same distribution as X^{t_N}, implying that there exists a sequence whose missed-detection probability decays exponentially with t_N. Theorem B.5.1 says that for any sequence $\emptyset^{t_N}[N]$ the false-alarm probability decays exponentially with $\omega(t_N)$. Since $\omega(t_N) = o_{t_N}(1)$ and is also greater than $3\log t_N$, the synchronization subsystem we have described has Subsystem Property 4.

B.6 Lemmas Relating to Entropy Rates

Let $\bar{\mathcal{X}}$, $\bar{\mathcal{Y}}$, and $\{\bar{\beta}_i\}$ be as defined in Section 3.3, and let $\mathcal{X}$, $\mathcal{Y}$, and $\{\beta_i\}$ be as in Theorem B.3.1. Let α_k, and $\bar{\alpha}_k$ be the states of $\mathcal{X}$ and $\bar{\mathcal{X}}$, respectively, let $S_k = (\alpha_k, \beta_k)$, and let $\bar{S}_k = (\bar{\alpha}_k, \bar{\beta}_k)$.

In this section, we show that the entropy of n samples of $\mathcal{X}$, $\mathcal{Y}$, or both is, asymptotically, n times the entropy rate of the corresponding stationary process.

Recall that $q_{X_1|X_{1-K}^0}$ and $q_{Y_1,\beta_1|X_1,\beta_0}$ are such that the Markov chain corresponding to the joint input/channel state process $\{S_k\}$ contains only transient states plus a single ergodic class. Clearly, the Markov chain corresponding to $\{\bar{S}_k\}$ then also contains only transient states and a single ergodic class. With the state space of the input process denoted $\mathcal{A}$ and that of the DFSC$_f$ denoted $\mathcal{B}$, let $\mathcal{S} = \mathcal{A} \times \mathcal{B}$, and let the ergodic states of $\mathcal{S}$ be denoted $\mathcal{S}_{\text{erg}}$.

Lemma B.6.1

1. $H(\bar{Y}_k|\bar{Y}^{k-1}, \bar{\alpha}_0, \bar{\beta}_0) \leq H_\infty(\bar{\mathcal{Y}})$
2. $H(\bar{X}_k|\bar{X}^{k-1}, \bar{\alpha}_0, \bar{\beta}_0) \leq H_\infty(\bar{\mathcal{X}})$
3. $H(\bar{X}_k, \bar{Y}_k|\bar{X}^{k-1}, \bar{Y}^{k-1}, \bar{\alpha}_0, \bar{\beta}_0) \leq H_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}})$

Proof: 1) Because of stationarity,

$$H(\bar{Y}_k|\bar{Y}^{k-1}, \bar{\alpha}_0, \bar{\beta}_0) = H(\bar{Y}_{k+n}|\bar{Y}_{n+1}^{k+n-1}, \bar{\alpha}_n, \bar{\beta}_n) \tag{B.101}$$

for all $n \geq 0$. Using (3.10) and (3.11), we see that $\{(\bar{X}_i, \bar{Y}_i)\}_{i>k}$ is, conditioned on $(\bar{\alpha}_k, \bar{\beta}_k)$, independent of $\{(\bar{X}_i, \bar{Y}_i)\}_{i\leq k}$, which implies that

$$H(\bar{Y}_{k+n}|\bar{Y}_{n+1}^{k+n-1}, \bar{\alpha}_n, \bar{\beta}_n) = H(\bar{Y}_{k+n}|\bar{Y}^{k+n-1}, \bar{\alpha}_n, \bar{\beta}_n) \tag{B.102}$$

$$\leq H(\bar{Y}_{k+n}|\bar{Y}^{k+n-1}) \tag{B.103}$$

for all n, where (B.103) holds because conditioning reduces entropy. Since the inequality holds for all n, it must hold in the limit as $n \to \infty$, which we know to be $H_\infty(\bar{\mathcal{Y}})$.

Proofs of 2) and 3) follow similarly. □

Lemma B.6.2

1. $H(\bar{Y}^n) \leq nH_\infty(\bar{\mathcal{Y}}) + C_1$
2. $H(\bar{X}^n) \leq nH_\infty(\bar{\mathcal{X}}) + C_2$
3. $H(\bar{X}^n, \bar{Y}^n) \leq nH_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) + C_3$

where C_1, C_2, and C_3 are constants that are independent of n.

Proof: 1) Note that

$$H(\bar{Y}^n) \leq H(\bar{Y}^n|\bar{\alpha}_0, \bar{\beta}_0) + H(\bar{\alpha}_0, \bar{\beta}_0) \tag{B.104}$$

$$\leq H(\bar{\alpha}_0, \bar{\beta}_0) + \sum_{k=1}^{n} H(\bar{Y}_k|\bar{Y}^{k-1}, \bar{\alpha}_0, \bar{\beta}_0) \tag{B.105}$$

$$\leq H(\bar{\alpha}_0, \bar{\beta}_0) + nH_\infty(\bar{\mathcal{Y}}) \tag{B.106}$$

Because $H(\bar{\alpha}_0, \bar{\beta}_0)$ is independent of n, the lemma follows. Statements 2) and 3) follow similarly. □

Lemma B.6.3

1. $H(Y^n) \leq nH_\infty(\bar{\mathcal{Y}}) + D_1$

2. $H(X^n) \leq nH_\infty(\bar{\mathcal{X}}) + D_2$

3. $H(X^n, Y^n) \leq nH_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) + D_3$

where D_1, D_2, and D_3 are constants that are independent of n.

Proof: 1) Since $H(\bar{Y}^n|\bar{S}_0)$ is an average over initial states with positive probability, and only states in $\mathcal{S}_{\text{erg}}$ have positive stationary probabilities, there exists $s \in \mathcal{S}_{\text{erg}}$ such that $H(\bar{Y}^n|\bar{S}_0 = s) \leq H(\bar{Y}^n|\bar{S}_0) \leq H(\bar{Y}^n)$. Let $L = \min_k\{\bar{S}_k \mid \bar{S}_k = s\}$ be the first-passage time from state $\bar{S}_0$ to state s. Then for any s_0,

$$H(\bar{Y}^n|L = l, \bar{S}_0 = s_0) = H(\bar{Y}_{l+1}^n|\bar{Y}^l, L = l, \bar{S}_0 = s_0) + H(\bar{Y}^l|L = l, \bar{S}_0 = s_0) \tag{B.107}$$

$$= H(\bar{Y}_{l+1}^n|\bar{S}_l = s) + H(\bar{Y}^l|L = l, \bar{S}_0 = s_0) \tag{B.108}$$

$$\leq nH_\infty(\bar{\mathcal{Y}}) + l(\log|\mathcal{Y}| - H_\infty(\bar{\mathcal{Y}})). \tag{B.109}$$

This inequality implies that

$$H(\bar{Y}^n|L, \bar{S}_0 = s_0) \leq nH_\infty(\bar{\mathcal{Y}}) + E[L|\bar{S}_0 = s_0](\log|\mathcal{Y}| - H_\infty(\bar{\mathcal{Y}})) \tag{B.110}$$

$$\leq nH_\infty(\bar{\mathcal{Y}}) + (\log|\mathcal{Y}| - H_\infty(\bar{\mathcal{Y}}))E_{\text{max}}, \tag{B.111}$$

where $E_{\text{max}} = \max_{s'} E[L|\bar{S}_0 = s']$. Then

$$H(\bar{Y}^n|\bar{S}_0 = s_0) \leq H(\bar{Y}^n|L, \bar{S}_0 = s_0) + H(L|\bar{S}_0 = s_0) \tag{B.112}$$

$$\leq H(\bar{Y}^n|L, \bar{S}_0 = s_0) + \log E[L|\bar{S}_0 = s_0] + 2 \tag{B.113}$$

$$\leq nH_\infty(\bar{\mathcal{Y}}) + (\log|\mathcal{Y}| - H_\infty(\bar{\mathcal{Y}}))E_{\text{max}} + \log E_{\text{max}} + 2 \tag{B.114}$$

Since the Markov chain corresponding to $\{\bar{S}_k\}$ contains only transient states and a single ergodic class, $E_{\text{max}} < \infty$ [8]. Since (B.114) holds for all $s_0 \in \mathcal{A} \times \mathcal{B}$, we must have $H(Y^n|S_0) \leq nH_\infty(\bar{\mathcal{Y}}) + C$, which finally implies that $H(Y^n) \leq H(Y^n|S_0) + H(S_0) \leq nH_\infty(\bar{\mathcal{Y}}) + C'$, which proves the lemma. The proofs of 2) and 3) are similar. □

Lemma B.6.4

1. $H(Y^n) \geq nH_\infty(\bar{\mathcal{Y}}) - F_1$

2. $H(X^n) \geq nH_\infty(\bar{\mathcal{X}}) - F_2$

3. $H(X^n, Y^n) \geq nH_\infty(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) - F_3$,

where F_1, F_2, and F_3 are constants that are independent of n.

Proof: Note first that $H(\bar{Y}^n) \geq nH_\infty(\bar{\mathcal{Y}})$ [23].

There exists a state $s \in \mathcal{S}_{\text{erg}}$ such that $H(\bar{Y}^n|\bar{S}_0 = s) \geq H(\bar{Y}^n|\bar{S}_0) \geq H(\bar{Y}^n) - H(\bar{S}_0)$. Let $L = \min_k\{\bar{S}_k \ : \ \bar{S}_k = s\}$. Then for any s_0,

$$H(\bar{Y}^n|L = l, \bar{S}_0 = s_0) = H(\bar{Y}_{l+1}^n|\bar{Y}^l, L = l, \bar{S}_0 = s_0) + H(\bar{Y}^l|L = l, \bar{S}_0 = s_0) \tag{B.115}$$

$$= H(\bar{Y}_{l+1}^n|\bar{S}_l = s) + H(\bar{Y}^l|L = l, \bar{S}_0 = s_0) \tag{B.116}$$

$$\geq nH_\infty(\bar{\mathcal{Y}}) - H(\bar{S}_0) - lH_\infty(\bar{\mathcal{Y}}). \tag{B.117}$$

It then follows that

$$\begin{aligned} H(\bar{Y}^n|L, \bar{S}_0 = s_0) &\geq nH_\infty(\bar{\mathcal{Y}}) - H(\bar{S}_0) - E[L|\bar{S}_0 = s_0]H_\infty(\bar{\mathcal{Y}}) && \text{(B.118)} \\ &\geq nH_\infty(\bar{\mathcal{Y}}) - H(\bar{S}_0) - E_{\text{max}}H_\infty(\bar{\mathcal{Y}}) && \text{(B.119)} \\ &\triangleq nH_\infty(\bar{\mathcal{Y}}) - C, && \text{(B.120)} \end{aligned}$$

where $E_{\text{max}} = \max_{s'} E[L|\bar{S}_0 = s']$. Since the Markov chain corresponding to $\{\bar{S}_k\}$ contains only transient states and a single ergodic class, C is finite. Since $H(\bar{Y}^n|\bar{S}_0 = s_0) \geq H(\bar{Y}^n|L, \bar{S}_0 = s_0) \geq nH_\infty(\bar{\mathcal{Y}}) - C$, it follows that $H(Y^n) \geq H(Y^n|S_0) \geq nH_\infty(\bar{\mathcal{Y}}) - C$, and the lemma follows. Proofs of 2) and 3) are similar. □

Appendix C

Detailed Technical Information for Chapter 4

C.1 Properties of the Conditional Lempel-Ziv Coder

Let $x^n \in \mathcal{X}^n$, $y^n \in \mathcal{Y}^n$ be the sequences given in Section 4.3.1, let $\mathcal{A} = \mathcal{X}^K$, let $s_0^{n-1} \in (\mathcal{A} \times \mathcal{B})^n$ be a given joint input/channel state sequence, and also let $\tilde{s}_j = s_{p_j - 1}$. Using the notation from Section 4.3.1, let

$$J[\tilde{y}, s] = \{j \ : \ j \in J[\tilde{y}], \tilde{s}_j = s\}, \tag{C.1}$$

and let $c[\tilde{y}, s] = |J[\tilde{y}, s]|$. Note that

$$c[\tilde{y}] = \sum_{s \in \mathcal{A} \times \mathcal{B}} c[\tilde{y}, s] \tag{C.2}$$

$$c = \sum_{l} \sum_{\tilde{y} \in \mathcal{Y}^l} c[\tilde{y}]. \tag{C.3}$$

Let Ω be as in Section 4.3, and let M_Ω be the (possibly infinite) supremum of the cardinalities of the state-spaces associated with the DFSC$_f$'s in Ω. Let the process $\{X_i\}$ be a Kth-order Markov process with transition pmf $q_{X_1 | X_{1-K}^0}$ and initial state density q_{α_0}. Let the processes $\{Y_i(q)\}$ and $\{\beta_i\}$ be the output and state processes resulting from passing the Markov process $\{X_i\}$ through the DFSC$_f$ $q \in \Omega$ without feedback with α_0 and β_0 distributed according to q_{α_0, β_0}. With $\{S_k\}$ denoting the associated joint input/channel state process, the following lemma, which is analogous to Ziv's Inequality [14], holds.

Lemma C.1.1

$$\sum_{l} \sum_{\tilde{y}^l} \sum_{s} c[\tilde{y}^l, s] \log c[\tilde{y}^l, s] \leq - \log p_{X^n | Y^n(q), S_{p_1 - 1}, \cdots, S_{p_c - 1}}(x^n | y^n, \tilde{s}_1, \cdots, \tilde{s}_c) \tag{C.4}$$

Proof:

$$\begin{aligned}
&\log p_{X^n|Y^n(q),S_{p_1-1},\cdots,S_{p_c-1}}(x^n|y^n,\tilde{s}_1,\cdots,\tilde{s}_c) && \text{(C.5)}\\
&\quad=\sum_{j=1}^{c}\log p_{X_{p_j}^{p_{j+1}-1}|Y_{p_j}^{p_{j+1}-1}(q),S_{p_j-1}}(x_{p_j}^{p_{j+1}-1}|y_{p_j}^{p_{j+1}-1},\tilde{s}_j) && \text{(C.6)}\\
&\quad=\sum_{l}\sum_{\tilde{y}^l}\sum_{s}\sum_{j\in J[\tilde{y}^l,s]}\log p_{X_{p_j}^{p_{j+1}-1}|Y_{p_j}^{p_{j+1}-1}(q),S_{p_j-1}}(x_{p_j}^{p_{j+1}-1}|\tilde{y}^l,s) && \text{(C.7)}\\
&\quad=\sum_{l}\sum_{\tilde{y}^l}\sum_{s}c[\tilde{y}^l,s]\sum_{j\in J[\tilde{y}^l,s]}\frac{1}{c[\tilde{y}^l,s]}\log p_{X_{p_j}^{p_{j+1}-1}|Y_{p_j}^{p_{j+1}-1}(q),S_{p_j-1}}(x_{p_j}^{p_{j+1}-1}|\tilde{y}^l,s) && \text{(C.8)}\\
&\quad\le\sum_{l}\sum_{\tilde{y}^l}\sum_{s}c[\tilde{y}^l,s]\log\sum_{j\in J[\tilde{y}^l,s]}\frac{1}{c[\tilde{y}^l,s]}p_{X_{p_j}^{p_{j+1}-1}|Y_{p_j}^{p_{j+1}-1}(q),S_{p_j-1}}(x_{p_j}^{p_{j+1}-1}|\tilde{y}^l,s), && \text{(C.9)}
\end{aligned}$$

where the inequality follows from Jensen's inequality and the concavity of the logarithm. Then, using that

$$\sum_{j\in J[\tilde{y}^l,s]}\frac{1}{c[\tilde{y}^l,s]}p_{X_{p_j}^{p_{j+1}-1}|Y_{p_j}^{p_{j+1}-1}(q),S_{p_j-1}}(x_{p_j}^{p_{j+1}-1}|\tilde{y}^l,s)\le\frac{1}{c[\tilde{y}^l,s]}, \tag{C.10}$$

the lemma follows. □

Theorem C.1.1 For all $q\in\mathfrak{Q}$, there exists a constant $C_q<\infty$, independent of n, such that

$$\sum_{l}\sum_{\tilde{y}^l}c[\tilde{y}^l]\log c[\tilde{y}^l]\le-\log p_{X^n|Y^n(q)}(x^n|y^n)+C_q\frac{n+2}{\log(n+2)}. \tag{C.11}$$

Furthermore, if $M_{\mathfrak{Q}}<\infty$, then (C.11) can be satisfied by setting C_q to a common value $C<\infty$ for all $q\in\mathfrak{Q}$.

Proof: We begin with Lemma C.1.1, which we may rewrite as

$$\begin{aligned}
&-\log p_{X^n|Y^n(q),S_{p_1-1},\cdots,S_{p_c-1}}(x^n|y^n,\tilde{s}_1,\cdots,\tilde{s}_c)\\
&\qquad\ge\sum_{l}\sum_{\tilde{y}^l}\sum_{s}c[\tilde{y}^l,s]\log c[\tilde{y}^l,s] && \text{(C.12)}\\
&\qquad=\sum_{l}\sum_{\tilde{y}^l}c[\tilde{y}^l]\sum_{s}\frac{c[\tilde{y}^l,s]}{c[\tilde{y}^l]}\log\frac{c[\tilde{y}^l]c[\tilde{y}^l,s]}{c[\tilde{y}^l]} && \text{(C.13)}\\
&\qquad=\sum_{l}\sum_{\tilde{y}^l}c[\tilde{y}^l]\sum_{s}\frac{c[\tilde{y}^l,s]}{c[\tilde{y}^l]}\left(\log c[\tilde{y}^l]+\log\frac{c[\tilde{y}^l,s]}{c[\tilde{y}^l]}\right) && \text{(C.14)}\\
&\qquad=\sum_{l}\sum_{\tilde{y}^l}c[\tilde{y}^l]\log c[\tilde{y}^l]+\sum_{l}\sum_{\tilde{y}^l}c[\tilde{y}^l]\sum_{s}\frac{c[\tilde{y}^l,s]}{c[\tilde{y}^l]}\log\frac{c[\tilde{y}^l,s]}{c[\tilde{y}^l]}. && \text{(C.15)}
\end{aligned}$$

Writing $\pi_{\tilde{y}^l}(s) = \frac{c[\tilde{y}^l,s]}{c[\tilde{y}^l]}$, we have that $\sum_s \pi_{\tilde{y}^l}(s) = 1$. Treating $\pi_{\tilde{y}^l}$ as a pmf over $\mathcal{A} \times \mathcal{B}$, let $V_{\tilde{y}^l}$ be a random variable distributed according to $\pi_{\tilde{y}^l}$. Then

$$\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \sum_s \frac{c[\tilde{y}^l, s]}{c[\tilde{y}^l]} \log \frac{c[\tilde{y}^l, s]}{c[\tilde{y}^l]} = -\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] H(V_{y^l}) \tag{C.16}$$

$$\geq -\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log |\mathcal{A} \times \mathcal{B}| \tag{C.17}$$

$$= -c \log |\mathcal{A} \times \mathcal{B}|. \tag{C.18}$$

Hence,

$$\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log c[\tilde{y}^l] \leq -\log p_{X^n|Y^n(q),S_{p_1-1},\cdots,S_{p_c-1}}(x^n|y^n, \tilde{s}_1, \cdots, \tilde{s}_c) + c \log |\mathcal{A} \times \mathcal{B}|. \tag{C.19}$$

Note that the left-hand side of (C.19) is functionally independent of $\tilde{s}_1, \cdots, \tilde{s}_c$, and that (C.19) holds for any state sequence $\tilde{s}_1, \cdots, \tilde{s}_c$, since it was arbitrary. Since

$$p_{X^n|Y^n(q)}(x^n|y^n) = \sum_{\hat{s}_1,\cdots,\hat{s}_c} p_{S_{p_1-1},\cdots,S_{p_c-1}|Y^n(q)}(\hat{s}_1, \cdots, \hat{s}_c|y^n)$$

$$p_{X^n|Y^n(q),S_{p_1-1},\cdots,S_{p_c-1}}(x^n|y^n, \hat{s}_1, \cdots, \hat{s}_c) \tag{C.20}$$

$$\leq \max_{\hat{s}_1,\cdots,\hat{s}_c} p_{X^n|Y^n(q),S_{p_1-1},\cdots,S_{p_c-1}}(x^n|y^n, \hat{s}_1, \cdots, \hat{s}_c), \tag{C.21}$$

there must be some state sequence $\hat{s}_1, \cdots, \hat{s}_c$ for which

$$-\log p_{X^n|Y^n(q),S_{p_1-1},\cdots,S_{p_c-1}}(x^n|y^n, \hat{s}_1, \cdots, \hat{s}_c) \leq -\log p_{X^n|Y^n(q)}(x^n|y^n). \tag{C.22}$$

Using (C.22), that c, which is only a function of the individual sequences x^n and y^n, can be bounded by a term that is $O_n(n/\log n)$ [14], and that the left-hand side of (C.11) is finite for all n, the theorem follows. □

Theorem C.1.2 For each $q \in \mathcal{Q}$, there exists a constant $C_q < \infty$, such that for all $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$,

$$\ell(\sigma_{\mathrm{CLZ}}(x^n|y^n)) \leq -\log p_{X^n|Y^n(q)}(x^n|y^n) + C_q \frac{(n+2)\log\log(n+2)}{\log(n+2)}, \tag{C.23}$$

Furthermore, if $M_{\mathcal{Q}} < \infty$, then (C.23) can be satisfied by setting C_q to a common value $C < \infty$ for all $q \in \mathcal{Q}$.

Proof: To code the jth phrase takes

$$\lceil \log c[y_{p_j}^{p_{j+1}-2}] \rceil + \lceil \log |\mathcal{X}| \rceil \tag{C.24}$$

bits. Additionally, the length of each phrase must be sent so that the decoder can initially parse the sequence y^n. The entire length of the encoding can therefore be written as

$$\ell(\sigma_{\mathrm{CLZ}}(x^n|y^n)) = \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l]\Big(\lceil \log c[\tilde{y}^{l-1}]\rceil + \lceil \log |\mathcal{X}| \rceil + 2\lceil \log l \rceil + 2\Big). \tag{C.25}$$

Consider for the moment just the term

$$a = \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log c[\tilde{y}^{l-1}], \tag{C.26}$$

which can be rewritten

$$\begin{aligned}
a &= \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log \frac{c[\tilde{y}^{l-1}]c[\tilde{y}^l]}{c[\tilde{y}^l]} && \text{(C.27)}\\
&= \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log c[\tilde{y}^l] + \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log \frac{c[\tilde{y}^{l-1}]}{c[\tilde{y}^l]} && \text{(C.28)}\\
&\leq \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log c[\tilde{y}^l] + \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log \frac{\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^{l-1}]}{\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l]}, && \text{(C.29)}
\end{aligned}$$

where the inequality follows from the log-sum inequality [14]. Letting M_l be the maximum phrase length, and using that

$$\begin{aligned}
\sum_{l=1}^{M_l} \sum_{\tilde{y}^l} c[\tilde{y}^{l-1}] &= \sum_{\tilde{y}_l} \sum_{l=1}^{M_l} \sum_{\tilde{y}^{l-1}} c[\tilde{y}^{l-1}] && \text{(C.30)}\\
&= |\mathcal{Y}| \sum_{l=1}^{M_l} \sum_{\tilde{y}^{l-1}} c[\tilde{y}^{l-1}] && \text{(C.31)}\\
&= |\mathcal{Y}| \sum_{l=1}^{M_l-1} \sum_{\tilde{y}^l} c[\tilde{y}^l] && \text{(C.32)}\\
&\leq |\mathcal{Y}| \sum_{l=1}^{M_l} \sum_{\tilde{y}^l} c[\tilde{y}^l], && \text{(C.33)}
\end{aligned}$$

we see that

$$\begin{aligned}
a &\leq \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log c[\tilde{y}^l] + \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log |\mathcal{Y}| && \text{(C.34)}\\
&\leq \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l] \log c[\tilde{y}^l] + c \log |\mathcal{Y}|. && \text{(C.35)}
\end{aligned}$$

Hence, the length can be upper bounded as

$$\ell(\sigma_{\rm CLZ}(x^n|y^n)) \le \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l](\log c[\tilde{y}^l] + \log|\mathcal{X}| + \log|\mathcal{Y}| + 2\log l + 6). \tag{C.36}$$

Note that

$$\begin{aligned}\sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l]\log l &\le c\log\frac{\sum_{l,\tilde{y}^l} lc[\tilde{y}^l]}{c} &\text{(C.37)}\\ &= c\log\frac{n}{c} &\text{(C.38)}\\ &= O_n(n\log\log n/\log n), &\text{(C.39)}\end{aligned}$$

where (C.37) follows from the concavity of the logarithm function, and (C.39) follows from the facts that (C.38) is, for fixed n, maximized at the maximum value of c, and that c is bounded by a term that is $O_n(n/\log n)$ [14]. We can write

$$\ell(\sigma_{\rm CLZ}(x^n|y^n)) \le \sum_l \sum_{\tilde{y}^l} c[\tilde{y}^l]\log c[\tilde{y}^l] + O_n\left(\frac{n\log\log n}{\log n}\right). \tag{C.40}$$

With Theorem C.1.1, the fact that $(n+2)/\log(n+2)=O_n(n\log\log n/\log n)$, and the fact that the left-hand side of (C.40) is finite for all n, the theorem follows. □

C.2 Universal Detection of the Synchronization Sequence

As stated in Section 4.3.2, to do universal detection of $\emptyset^{t_N}[N]$, we use a modified version of Ziv's universal function [66], which is based on Lempel-Ziv parsing. Let $\sigma_{\rm LZ}(y^n)$ be a standard 1978-version Lempel-Ziv encoding of y^n (see, for example, [14] for a description). Then the universal functions u_1 and u_2 in (4.16) are defined by

$$\begin{aligned}u_1(y^n) &= \ell(\sigma_{\rm LZ}(y^n)) &\text{(C.41)}\\ u_2(y^n|x^n) &= \ell(\sigma_{\rm CLZ}(y^n|x^n)) &\text{(C.42)}\end{aligned}$$

The following two lemmas show that $u_1(y^n)$ and $u_2(y^n|x^n)$ are not too much larger than certain associated negative log probabilities, which helps us to prove in Theorem C.2.2 that the missed-detection probability associated with $\delta_{{\rm U},t}$ is adequate.

Lemma C.2.1 For each $q \in \mathcal{Q}$, there exists a constant $C_{2,q} < \infty$, such that for all $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$,

$$u_2(y^n|x^n) \le -\log p_{Y^n(q)|X^n}(y^n|x^n) + C_{2,q}\frac{(n+2)\log\log(n+2)}{\log(n+2)}, \tag{C.43}$$

Furthermore, if $M_{\mathcal{Q}} < \infty$, then (C.43) can be satisfied by setting $C_{2,q}$ to a common value $C_2 < \infty$ for all $q \in \mathcal{Q}$.

Remark: The set $\mathcal{Q}$, the constant $M_{\mathcal{Q}}$, and processes $\{X_k\}$ and $\{Y_k(q)\}$ referred to above and throughout this section are as defined in Appendix C.1.

Proof: Follows in the same way as the proof of Theorem C.1.2. □

Lemma C.2.2 For each $q \in \mathcal{Q}$, there exists a constant $C_{1,q} < \infty$, such that for all $y^n \in \mathcal{Y}^n$,

$$u_1(y^n) \leq -\log p_{Y^n(q)}(y^n) + C_{1,q}\frac{(n+2)\log\log(n+2)}{\log(n+2)}. \tag{C.44}$$

Furthermore, if $M_{\mathcal{Q}} < \infty$, then (C.44) can be satisfied by setting $C_{1,q}$ to a common value $C_1 < \infty$ for all $q \in \mathcal{Q}$.

Proof: Proof is similar to proof of asymptotic optimality of Lempel-Ziv coding for stationary Markov sources (Theorem 12.10.1 in [14]). The major change is that a version of Ziv's Inequality (Lemma 12.10.3 in [14]) must be proven, but instead of the previous k samples forming the state of the process, a given sequence of channel states must be used as in Lemma C.1.1. □

The following lemma is stated and proven in, among other places, [14] (Theorem 5.11.1). It states that the source encoding of a sequence cannot be much shorter than the negative log probability of the sequence very often, a fact that is used to prove Theorem C.2.1.

Lemma C.2.3 Let $\tilde{X}^n$ be drawn according to the given pmf $q_{\tilde{X}^n}$, and let $\sigma : \mathcal{X}^n \to \{0,1\}^\dagger$ be some invertible mapping (e.g., the encoder for a uniquely decodable code). Then for any $\mu > 0$,

$$\Pr\{\ell(\sigma(\tilde{X}^n)) \leq -\log q_{\tilde{X}^n}(\tilde{X}^n) - \mu\} \leq 2^{-\mu}. \tag{C.45}$$

Proof:

$$\mathcal{K} = \{x^n \in \mathcal{X}^n \ : \ \ell(\sigma(x^n)) \leq -\log q_{\tilde{X}^n}(x^n) - \mu\}. \tag{C.46}$$

Then we can write

$$\begin{aligned}
\Pr\{\ell(\sigma(\tilde{X}^n)) \leq -\log q_{\tilde{X}^n}(\tilde{X}^n) - \mu\} &= \sum_{x^n \in \mathcal{K}} q_{\tilde{X}^n}(x^n) && \text{(C.47)}\\
&\leq \sum_{x^n \in \mathcal{K}} \exp_2\{-\ell(\sigma(\tilde{X}^n)) - \mu\} && \text{(C.48)}\\
&\leq \sum_{x^n \in \mathcal{X}^n} \exp_2\{-\ell(\sigma(\tilde{X}^n)) - \mu\} && \text{(C.49)}\\
&= 2^{-\mu} \sum_{x^n \in \mathcal{X}^n} \exp_2\{-\ell(\sigma(\tilde{X}^n))\} && \text{(C.50)}\\
&\leq 2^{-\mu}, && \text{(C.51)}
\end{aligned}$$

where the last inequality follows from Kraft's inequality (see Theorem 5.5.1 in [14]). □

With the functions $\delta_{\mathrm{U},t}$ and ζ defined as in (4.16), the following theorem states that, the false-alarm probability decays exponentially with $\zeta(t)$.

Theorem C.2.1 For all $q \in \mathcal{Q}$, there is a function $f_q(t) = o_t(1)$ such that for all $x^t \in \mathcal{X}^t$,

$$\Pr\{\delta_{\mathrm{U},t}(Y^t(q), x^t) = 1\} < \exp_2\{-\zeta(t)(1 - f_q(t))\} \tag{C.52}$$

Furthermore, if $M_{\mathcal{Q}} < \infty$, then (C.52) can be satisfied by setting f_q to a common $o_t(1)$ function for all $q \in \mathcal{Q}$.

Proof: Note that

$$\Pr\{u_2(Y^t(q)|x^t) \leq u_1(Y^t(q)) - \zeta(t)\} \leq \Pr\{u_2(Y^t(q)|x^t) \leq -\log p_{Y^t(q)}(Y^t(q)) - \zeta(t) + \epsilon(t)\}, \tag{C.53}$$

where $\epsilon_q(t) = C_{1,q}(t+2)\log\log(t+2)/\log(t+2)$, since, from Lemma C.2.2, $u_1(Y^t(q)) \leq -\log p_{Y^t(q)}(Y^t(q)) + \epsilon_q(t)$. Because u_2 is the length of an invertible encoding of y^t, and because the conditional Lempel-Ziv encoding is uniquely decodable, we can immediately see from Theorem C.2.3 that the right-hand side of (C.53) is less than $\exp_2\{-(\zeta(t) - \epsilon_q(t))\}$. Since $\epsilon_q(t) = o_t(\zeta(t))$ as $t \to \infty$, the first statement of the theorem follows. If $M_{\mathcal{Q}} < \infty$, then one can replace $C_{1,q}$ with C_1, and the last statement of the theorem follows. □

The following theorem states that the missed-detection probability associated with $\delta_{\mathrm{U},t}$, averaged over all length-t synchronization sequences, decays exponentially in t.

Theorem C.2.2 For each $q \in \mathcal{Q}$, there exists $F_{\mathrm{MD}}(q) > 0$ and a function $g_q(t) = o_t(1)$ such that for all t,

$$\Pr\{\delta_{\mathrm{U},t}(Y^t(q), X^t) = 0\} < \exp_2\{-t(F_{\mathrm{MD}}(q) - g_q(t))\}. \tag{C.54}$$

Furthermore, if $M_{\mathcal{Q}} < \infty$, then (C.54) can be satisfied by setting g_q to a common $o_t(1)$ function for all $q \in \mathcal{Q}$.

Proof: Again, assume for the moment that $q_{\alpha_0,\beta_0} = 1/(AB)$, where A and B are as in Appendix B.5. Then we can upper bound the probability of missed detection according to

$$\begin{aligned}
&\Pr\{\delta_{\mathrm{U},t}(Y^t, X^t) = 0\} \\
&\leq \sum_{x^t} \sum_{y^t \notin S(\mu_q(t))} p_{X^t}(x^t) p_{Y^t(q)|X^t}(y^t|x^t) \left(\exp_2\{\zeta(t) + u_2(y^t|x^t) - u_1(y^t)\}\right)^s \\
&\quad + \sum_{x^t} \sum_{y^t \in S(\mu_q(t))} p_{X^t}(x^t) p_{Y^t(q)|X^t}(y^t|x^t), \qquad \text{(C.55)} \\
&\leq \sum_{x^t} \sum_{y^t \notin S(\mu_q(t))} p_{X^t}(x^t) p_{Y^t(q)|X^t}(y^t|x^t) \left(\exp_2\{\zeta(t) + \epsilon_q(t) + \mu_q(t)\} \frac{p_{Y^t(q)}(y^t)}{p_{Y^t(q)|X^t}(y^t|x^t)}\right)^s \\
&\quad + \sum_{x^t} \sum_{y^t \in S(\mu_q(t))} p_{X^t}(x^t) p_{Y^t(q)|X^t}(y^t|x^t), \qquad \text{(C.56)}
\end{aligned}$$

where

$$S(\mu_q(t)) = \{y^t \in \mathcal{Y}^t \ : \ u_1(y^t) \leq -\log p_{Y^t(q)}(y^t) - \mu_q(t)\}, \tag{C.57}$$

and $\mu_q(t)$ is to be specified. Inequality (C.56) follows from the definition of $S(\mu_q(t))$ and Lemma C.2.1, which can be rewritten

$$\exp_2\{u_2(y^t|x^t)\} \leq \exp_2\{\epsilon_q(t)\}\frac{1}{p_{Y^t(q)|X^t}(y^t|x^t)}, \tag{C.58}$$

where $\epsilon_q(t) = C_{2,q}(t+2)\log\log(t+2)/\log(t+2)$, and $\lim_{t\to\infty}\epsilon_q(t)/t = 0$. The second term in (C.56) can be written $\Pr\{Y^t(q) \in S(\mu_q(t))\}$, which we know from Theorem C.2.3 to be less than $\exp_2\{-\mu_q(t)\}$.

From Theorem B.5.1, we know that

$$\begin{aligned}\sum_{x^t}\sum_{y^t} p_{X^t}(x^t)p_{Y^t(q)|X^t}(y^t|x^t)&\left(\exp_2\{\zeta(t)+\epsilon_q(t)+\mu_q(t)\}\frac{p_{Y^t(q)}(y^t)}{p_{Y^t(q)|X^t}(y^t|x^t)}\right)^s\\ &\leq \exp_2\{-tF_t(s;q)+s(\zeta(t)+\epsilon_q(t)+\mu_q(t))\} &\text{(C.59)}\\ &\leq \exp_2\{-tF_t(s;q)+\zeta(t)+\epsilon_q(t)+\mu_q(t)\}, &\text{(C.60)}\end{aligned}$$

where $F_t(s;q)$ is as defined in (B.65), but with its dependence on q made explicit.

Therefore,

$$\Pr\{\delta_{\mathrm{U},t}(Y^t(q),X^t)=0\} \leq \exp_2\{-tF_t(s;q)+\mu_q(t)+\zeta(t)+\epsilon_q(t)\}+\exp_2\{-\mu_q(t)\}. \tag{C.61}$$

Relaxing the constraint that $q_{\alpha_0,\beta_0} = 1/(AB)$, we find that for arbitrary q_{α_0,β_0},

$$\begin{aligned}\Pr\{\delta_{\mathrm{U},t}(Y^t(q),X^t)=0\} \leq\ &\exp_2\{-tF_t(s;q)+\mu_q(t)+\zeta(t)+\epsilon_q(t)+\log(AB)))\}\\ &+\exp_2\{-\mu_q(t)+\log(AB)\}. &\text{(C.62)}\end{aligned}$$

Now, let $s \in [0,1]$ be such that $F_\infty(s;q) \triangleq \liminf_{t\to\infty} F_t(s;q) > 0$ (see Lemma B.5.5). Then there must be a number t_0 such that for all $t > t_0$,

$$F_t(s;q) - (\log(AB)/t + \epsilon_q(t)/t + \zeta(t)/t) > F_\infty(s;q)/2, \tag{C.63}$$

which implies that, if we set $\mu_q(t) = tF_\infty(s;q)/4$, then

$$\begin{aligned}F_t(s;q) - (\log(AB)/t + \epsilon_q(t)/t + \zeta(t)/t + \mu_q(t)/t) &> F_\infty(s;q)/2 - \mu_q(t)/t &\text{(C.64)}\\ &= F_\infty(s;q)/4. &\text{(C.65)}\end{aligned}$$

Thus, the exponent for the first term in (C.62) is greater than $F_\infty(s;q)/4$, asymptotically. Since the second term on the right-hand side of (C.62) has, asymptotically, the exponent $F_\infty(s;q)/4$, the first statement of the theorem follows. The second statement follows upon realizing that if $M_{\mathrm{Q}} < \infty$, then $C_{2,q}$ above can be replaced by C_2, a constant independent of q. □

Appendix D

Detailed Technical Information for Chapter 5

D.1 Fixed-Length Source Coder for Tx-2

The fixed-length precoders used in both Tx-1 and Tx-2 as well as the fixed-length source coder used by Tx-1 are as described in Section A.9. In this section, we develop the fixed-length source coder used by Tx-2.

With the notation introduced in Section 5.3 holding, throughout this section, also let $\{X_{1,i}\}_{i=1}^n$ be i.i.d. with marginal pmf q_{X_1} and $\{X_{2,i}\}_{i=1}^n$ be i.i.d. with marginal pmf q_{X_2}. Then let $\{Y_i\}_{i=1}^n$ be such that $p_{Y^n|X_1^n,X_2^n}(y^n|x_1^n,x_2^n) = \prod_{i=1}^n q_{Y|X_1,X_2}(y_i|x_{1,i},x_{2,i})$ for all x_1^n, x_2^n, and y^n.

With this notation, the following Corollary to Lemma A.9.1 leads directly to Tx-2's fixed-length source coder.

Corollary D.1.1 For all $\delta > 0$, there exists a constant $F_3(\delta) > 0$ such that

$$\Pr\{-\log p_{X_2^n|Y^n,X_1^n}(X_2^n|Y^n,X_1^n) > n(H(X_2|Y,X_1)+\delta)\} < \exp\{-F_3(\delta)n\}. \tag{D.1}$$

The fixed-length source coder $\sigma_{2,n}$ then resembles $\sigma_{1,n}$. With $\mathcal{X}_2 = \{0,\cdots,M_2-1\}$, let $\tilde{X}_2 = 0_{M_2}.X_{2,1}X_{2,2}X_{2,3}\cdots$, and define the source coder $\sigma_{2,n}$ as follows: With the sequence $x_1^n \in \mathcal{X}_1^n$ representing Tx-1's inputs to the channel, with $x_2^n \in \mathcal{X}_2^n$ representing Tx-2's inputs to the channel, and with $y^n \in \mathcal{Y}$ representing the channel's output, the sequence x_2^n is coded according to

$$\sigma_{2,n}(x_2^n,q) = a^{[g_2(n)]} \tag{D.2}$$

$$a = F_{\tilde{X}_2|Y^n,X_1^n}(0_M.x_2^n|y^n,x_1^n) + p_{X_2^n|Y^n,X_1^n}(x_2^n|y^n,x_1^n)/2, \tag{D.3}$$

where $F_{\tilde{X}_2|Y^n,X_1^n}$ is the conditional cdf of $\tilde{X}_2$ conditioned on Y^n and X_1^n. With this definition of the source coder, the sequence x_2^n is decodable as long as $p_{X_2^n|Y^n,X_1^n}(x_2^n|y^n,x_1^n) > 2^{-(g_2(n)-2)}$. Corollary D.1.1 then implies that a decoding error occurs with probability that decays exponentially in n.

Appendix E

Performance Bounds for the Multiple-Access Broadcast Channel

In this Appendix, we present analysis of the MABC that shows that for two and three users, the Hluchyj-Gallager protocol is at least nearly optimal. The techniques we develop can be used to find bounds for a wide variety of decentralized control problems.

E.1 Background on the MABC

The multiple access broadcast channel (MABC) is a useful model for a variety of packet switched communication systems. Protocols for efficiently coordinating data transmission by multiple users over the MABC have long been sought by researchers. Although several variations of the MABC have been considered in the literature, e.g., [9], [27], [42], [28], [45], [51], and [58], we focus on a finite-user slotted system with immediate ternary feedback, retransmission of collisions, no buffering, and no communication among users; we refer to this system as the *canonical* system. For such a system, we show via bounds that the optimized window protocol developed by Hluchyj and Gallager [28] achieves the highest possible throughput in the two-user case and achieves a throughput that is at least very close to the highest possible in the three-user case.

Previous researchers have recognized that the MABC protocol design problem can be analyzed as a decentralized control problem [27], [42], [28], [45], [51], [58]. Invariably, these researchers have resorted to simplifications that make the problem tractable but also removed from the canonical problem. For example, Schoute [51] and Varaiya [58] both consider decentralized control of the MABC under a delayed sharing pattern and under the assumption that colliding packets incur a fixed cost rather than requiring retransmission; Rosberg [45] also assumes a fixed collision cost and no retransmissions, but differs by assuming no information sharing among controllers as well as control inputs that depend only on broadcast feedback. Although these two simplified problems are easier to analyze, their relationships to the canonical problem are unclear. On the other hand, the simplified problems considered by Hluchyj and Gallager [28], Grizzle et al. [27], and Paradis [42] yield solutions that can be used to find lower and upper bounds on the throughput of the canonical system. Hluchyj and Gallager consider the canonical system and find protocols that are optimal in the class of protocols known as the window protocols. Since window protocols are a proper subset of the set of all protocols, the throughput of Hluchyj and Gallager's optimized window protocol provides a *lower* bound to the throughput achievable in the canonical system. Conversely, Grizzle et al. and Paradis attack the problem of optimally controlling a MABC that is canonical except for a one-step delay sharing (OSDS) information pattern. Because the canonical system does not allow any communication among users, its throughput is *upper* bounded by the throughput achievable under OSDS. For two users, the OSDS bound is very close to the throughput of the Hluchyj-Gallager optimized window protocol [28], [27], [42]. Unfortunately, for more than two users, the OSDS bound is not close to the throughput of the Hluchyj-Gallager protocol or any other known protocol.

We introduce new upper bounds on the throughput of the canonical system via the K-step delay state information pattern, which is similar to the previously considered K-step delay sharing pattern [35], [36], [57], [62]. That is, we calculate the throughput of a MABC that is canonical except for a K-step delay state information pattern, and this provides an upper bound on the throughput of the canonical system. We use the dynamic programming method developed by Aicardi et al. [3] as a starting point for performing this calculation. However, we find this method to be unnecessarily computationally complex for a class of systems that includes the MABC with K-step delay state information. For this class, we present a more efficient version of the algorithm that can spell the difference between computational feasibility and infeasibility. As a result of the more efficient algorithm, we are able to find upper bounds on the throughput of the canonical system for two and three users that are tighter than the OSDS bound. The new bounds show that the performance of Hluchyj and Gallager's optimized window protocol is *exactly* optimal for two users and at least nearly optimal for three users, where optimality is with respect to maximizing throughput. In fact, the Hluchyj-Gallager protocol meets the upper bound exactly for three users when the packet arrival probability is moderately large.

The bounding technique we present is quite flexible and can be adapted to handle a greater number of users as well as other variations on the problem. Indeed, the bounding technique applies to the general class of decentralized control problems with no information sharing. For this reason, we first describe the bounding technique in this general setting and later focus attention on the MABC. We begin with Section E.2, in which we describe the general decentralized control problem with no information sharing. We devote Section E.3 to a description of notational conventions. In Section E.4, we describe the bounding technique and a method of complexity reduction. In Section E.5, we formulate the canonical MABC protocol design problem as a decentralized control problem with no sharing, and give numerical results of calculating the bounds for this problem. Finally, we give a discussion of our results as well as concluding remarks in Section E.6.

E.2 Decentralized Systems with No Sharing

In this section, we describe the class of systems for which the bounding technique to be described in Section E.4 is applicable. This class of systems consists of decentralized systems with no sharing, i.e., systems in which no information is communicated among controllers. We show in Section E.5 that the MABC can be regarded as such a system.

Consider a discrete-time stochastic system that is regulated by M decentralized controllers, each with an associated measurement station. The system state variable x_t and the mth measurement station's observations y_t^m evolve over T time steps according to the equations

$$x_{t+1} = f_t(x_t, u_t^1, \cdots, u_t^M, v_t), \qquad t = 0, 1, \cdots, T-1 \tag{E.1}$$

$$y_t^m = g_t^m(x_t, w_t^m), \qquad t = 0, 1, \cdots, T-1, \qquad m = 1, 2, \ldots, M, \tag{E.2}$$

where f_t and g_t^m are given functions, u_t^m represents the mth controller's input at time t, and the quantities $x_0, (v_0, w_0^1, \cdots, w_0^M), \cdots, (v_{T-1}, w_{T-1}^1, \cdots, w_{T-1}^M)$ are mutually independent primitive random vectors whose distributions are known. At a particular time step t, the random vectors $v_t, w_t^1, \cdots, w_t^M$ are allowed to be statistically dependent. The vectors u_t^m, v_t, w_t^m, x_t, and y_t^m take values in the *finite* sets U_t^m, V_t, W_t^m, X_t, and Y_t^m, respectively. We assume that these sets are mutually disjoint, e.g., $X_1 \cap X_2 = \emptyset$, and $W_1^1 \cap Y_1^1 = \emptyset$.[1]

Each controller produces, according to a pre-designed control law, an input based only on local observations from its own measurement station. If the mth controller is governed at time t by a control law γ_t^m then

$$u_t^m = \gamma_t^m(y_0^m, y_1^m, \cdots, y_t^m). \tag{E.3}$$

It is this functional dependence of u_t^m on only the history of the mth measurement station's measurements that makes the problem one of no sharing.

The set of admissible control laws, Γ_t^m, is the set of all functions mapping $Y_0^m \times \cdots \times Y_t^m$ to U_t^m. The design objective is to choose the control laws

$$\gamma_t^m \in \Gamma_t^m, \qquad m = 1, \cdots, M, \qquad t = 0, \cdots, T-1 \tag{E.4}$$

to minimize the total expected cost per stage

$$\frac{1}{T} \sum_{t=0}^{T-1} E[h_t(x_{t+1}, u_t^1, \cdots, u_t^M)], \tag{E.5}$$

where h_t is a given cost function.

E.3 Notational Conventions

To analyze the MABC, we use a completely different framework from the rest of the thesis. Therefore, we introduce the following new notation, which supersedes any notation introduced earlier.

To ease the notational burden, we introduce some conventions before proceeding. We adopt the convention of using context to distinguish between values assumed by random variables and the random variables themselves. The dependence of densities and expectations on a choice of control laws $\gamma_{s:t}$ is indicated by $p(\cdot; \gamma_{s:t})$ and $E[\cdot; \gamma_{s:t}]$, respectively. Domains and ranges of functions are to be inferred from context, but sometimes may be explicitly given for emphasis or clarity.

To consolidate lists of related symbols, we define $y_t = (y_t^1, \cdots, y_t^M)$, $y_{s:t}^m = (y_s^m, \cdots, y_t^m)$, and $y_{s:t} = (y_s, \cdots, y_t)$ for $t \geq s$. If $t < s$, then $y_{s:t}$ and $y_{s:t}^m$ are empty tuples. Moreover, we denote the range of y_t by $Y_t = \prod_{m=1}^M Y_t^m$, the range of $y_{s:t}^m$ by $Y_{s:t}^m = \prod_{j=s}^t Y_j^m$, and the range of $y_{s:t}$ by $Y_{s:t} = \prod_{j=s}^t Y_j$. Analogous notation will be used for other variables and their ranges.

It is convenient to define Boolean set operations on tuples as follows. Suppose that $\mathcal{A} =$

[1] This assumption is not essential and is present only to facilitate notation in subsequent sections. Indeed, we will focus on the stationary case, which is at odds with this assumption, but this difficulty can be ignored.

$\{A_1, \cdots, A_n\}$ is an ordered collection of disjoint sets such that $A_1 < \cdots < A_n$, i.e., A_1 is the "smallest" element of $\mathcal{A}$, while A_n is the "largest." This ordering of the collection $\mathcal{A}$ is only important for the systematic construction of tuples from sets. Let $a = (a_1, \cdots, a_n) \in \prod_{i=1}^{n} A_i$ be a tuple. Let $b = (a_{j_1}, \cdots, a_{j_p})$, and $c = (a_{k_1}, \cdots, a_{k_q})$ be tuples which may be called "sub-tuples" of a. Then define $b \setminus c$ to be a tuple consisting of the elements of the set difference $\{a_{j_i}\}_{i=1}^{p} \setminus \{a_{k_i}\}_{i=1}^{q}$, ordered so that an element in the tuple precedes another if it belongs to a set that is "smaller" than the set to which the other belongs. Analogous notation holds for $b \cup c$, and $b \cap c$. Finally, if $d = (a_{\ell_1}, \cdots, a_{\ell_r})$, we denote the set product $\prod_{i=1}^{r} A_{\ell_i}$ by $\mathcal{S}(d)$.

In particular, the collection $\mathcal{X} = \{U_t^m, V_t, W_t^m, X_t, X_T, Y_t^m \mid m = 1, \cdots, M,\ t = 0, \cdots, T-1\}$ is, by assumption, a collection of disjoint sets. Let $\vartheta = (u_{0:T-1}, v_{0:T-1}, w_{0:T-1}, x_{0:T}, y_{0:T-1})$, and let the members of $\mathcal{X}$ have an ordering corresponding to the ordering of the elements in the tuple ϑ. We may now consider unions, intersections, and differences of the sub-tuples of ϑ and will do so in the following sections.

E.4 Bounding Techniques

The only known solution to the decentralized control problem with no sharing described in Section 2 involves exhaustive search, which is prohibitive in practice and impossible when the time horizon is infinite. While we do not develop an explicit solution to this problem, we present a technique for lower bounding the optimal cost achievable by the system in Section 2. The bounds are useful for the evaluation of suboptimal control laws, and when the bound is tight, an optimal solution is implicitly identified.

We lower bound the optimal cost achievable in the no-sharing system with the optimal cost achievable by a system with K-step delay state information, which will be described shortly. Control laws for a system with K-step delay state information can be efficiently found [3], and because more information is available to controllers than under the no sharing pattern, the optimal cost incurred in such a system is always lower than that incurred by the system with no sharing. As we will see, however, the algorithm in [3] is more computationally complex than necessary in the special case of common information. For this special case, we present a more computationally efficient version of the algorithm.

E.4.1 Systems with K-Step Delay State Information

In this section, we describe systems with K-step delay state information, which are similar to the K-step delay sharing systems considered in [35], [36], [57], and [62]. Indeed, K-step delay state and K-step delay sharing are equivalent in the case of noiseless observations i.e., the case in which x_t can be determined from y_t. For this noiseless case, Aicardi et al. [3] show how to efficiently find optimal control laws when the input and state spaces are finite. We adapt the algorithm from [3] to the general K-step delay state problem.

A system with K-step delay state information is the same as the system described in Section 2 except for the following modifications. With

$$\delta_t = (y_{0:t-1}, x_{0:t}) \tag{E.6}$$

and the range of δ_t denoted by Δ_t, an admissible control law γ_t^m for the K-step delay state problem must have the form

$$u_t^m = \gamma_t^m(y_{t-K:t}^m, \delta_{t-K}). \tag{E.7}$$

For convenience, we denote $(y_{t-K:t}^m, \delta_{t-K})$ by z_t^m and $Y_{t-K:t}^m \times \Delta_{t-K}$ by Z_t^m. The set of possible control laws for the mth controller at time t is denoted $\check{\Gamma}_t^m$, and consists of all possible maps from Z_t^m to U_t^m. To complete the problem modifications, we change the cost criterion as follows. Since we are interested primarily in the infinite horizon scenario, we assume for convenience that the first K optimal control laws $\gamma_{0:K-1}$ are given and that the goal is to minimize the expected cost per stage starting at stage K:

$$\min_{\gamma_{K:T-1} \in \check{\Gamma}_{K:T-1}} \frac{1}{T-K} E[\sum_{\tau=K}^{T-1} h_\tau(x_{\tau+1}, u_\tau); \gamma_{0:T-1}] \tag{E.8}$$

Since the first K steps will not affect the limiting behavior, we can just as well choose an arbitrary set of starting control laws $\gamma_{0:K-1}$.

It will be useful to introduce the notion of a "sub-law" as follows. Let $\gamma_t^m \in \check{\Gamma}_t^m$ be a control law, and let η be a sub-tuple of ϑ, and let $N = \mathcal{S}(\eta)$. A sub-law with respect to N is a map from $\mathcal{S}(z_t^m \setminus \eta)$ to U_t^m, and the set of all such maps is denoted $\check{\Gamma}_{t|N}^m$. Define $\gamma_{t|\eta}^m$ to be the sub-law in $\check{\Gamma}_{t|N}$ satisfying

$$\gamma_{t|\eta}^m(\beta) = \gamma_t^m(\beta \cup \eta), \qquad \forall \beta \in \mathcal{S}(z_t^m \setminus \eta), \tag{E.9}$$

where the elements in the tuple $\beta \cup \eta$ are assumed to be in the order required by γ_t^m. Similarly, if $\psi_t^m \in \check{\Gamma}_{t|N}^m$ is sub-law with respect to N, $\hat{\eta}$ is a sub-tuple of ϑ, and $\hat{N} = \mathcal{S}(\hat{\eta})$, then $\psi_{t|\hat{\eta}}^m$ is defined to be a sub-law in $\check{\Gamma}_{t|N \times \hat{N}}^m$ satisfying

$$\psi_{t|\hat{\eta}}^m(\beta) = \psi_t^m(\beta \cup \hat{\eta}), \qquad \forall \beta \in \mathcal{S}(z_t^m \setminus (\eta \cup \hat{\eta})). \tag{E.10}$$

Also define the expansion of a sub-law $\psi_t^m \in \check{\Gamma}_{t|N}^m$ to be $G_{t|N}^m(\psi_t^m) = \mu \in \check{\Gamma}_t^m$ such that

$$\mu(\beta \cup \bar{\eta}) = \psi_t^m(\beta), \qquad \forall \beta \in \mathcal{S}(z_t^m \setminus \breve{\eta}), \qquad \forall \bar{\eta} \in S(\breve{\eta}), \tag{E.11}$$

where $\breve{\eta}$ is any element of N. For convenience, we denote $G_{t|N}^m$ by G, assuming that a sub-law always expands into a control law for the corresponding time and controller. Also, we denote $(G(\gamma_{t|\eta}^1), \cdots, G(\gamma_{t|\eta}^M))$ by $G(\gamma_{t|\eta})$ and $(G(\gamma_{s|\eta}), \cdots, G(\gamma_{t|\eta}))$ by $G(\gamma_{s:t|\eta})$.

We can now state a theorem that characterizes the optimal control law for a system with K-step delay state. Note that this theorem is essentially the same as a theorem presented in [3].

Theorem E.4.1 Consider the following recursive equations that characterize optimal control laws for the K-step delay state system:

$$J_T^*(x_{T-K}, \psi_{T-K:T-1}) = 0, \qquad \forall \psi_{T-K:T-1} \in \check{\Gamma}_{T-K:T-1|\Delta_{T-K}}, \qquad \forall x_{T-K} \in X_{T-K} \tag{E.12}$$

$$\begin{aligned} J_t(x_{t-K}, \psi_{t-K:t-1}, \psi_t) = E[h_t(x_{t+1}, u_t) + J_{t+1}^*(x_{t-K+1}, \psi_{t-K+1:t|y_{t-K}})|x_{t-K}; G(\psi_{t-K:t})], \\ \forall \psi_{t-K:t} \in \check{\Gamma}_{t-K:t|\Delta_{t-K}} \end{aligned} \tag{E.13}$$

$$J_t^*(x_{t-K},\psi_{t-K:t-1}) = \min_{\psi_t\in\check{\Gamma}_{t|\Delta_{t-K}}} J_t(x_{t-K},\psi_{t-K:t-1},\psi_t), \qquad \forall\psi_{t-K:t-1}\in\check{\Gamma}_{t-K:t-1|\Delta_{t-K}} \tag{E.14}$$

Let $\gamma_{0:K-1}\in\check{\Gamma}_{0:K-1}$ be given starting optimal control laws. If control laws $\gamma_{K:T-1}\in\check{\Gamma}_{K:T-1}$ satisfy

$$J_t(x_{t-K},\gamma_{t-K:t-1|\delta_{t-K}},\gamma_{t|\delta_{t-K}}) = J_t^*(x_{t-K},\gamma_{t-K:t-1|\delta_{t-K}}) \tag{E.15}$$

for every $\delta_{t-K}\in\Delta_{t-K}$, and every $t=K,\cdots,T-1$, then $\gamma_{K:T-1}$ are optimal.

Proof: The proof of this theorem follows along the same lines as the proof of Theorem E.4.2. A proof of a similar theorem is provided in [3]. □

We can interpret the equations in Theorem 1 as the equations resulting from applying the dynamic programming algorithm to the following centralized stochastic control problem. Using the notation of Bertsekas [8], the state for the centralized problem is $x'_t=(x_{t-K},\psi_{t-K:t-1})\in X_{t-K}\times\check{\Gamma}_{t-K:t-1|\Delta_{t-K}}$, the input is $u'_t=\psi_t\in\check{\Gamma}_{t|\Delta_{t-K}}$, and the disturbance is $w'_t=(x_{t-K+1},y_{t-K})\in X_{t-K+1}\times Y_{t-K}$. The state transition function f'_t is defined by

$$\begin{aligned} f'_t(x'_t,u'_t,w'_t) &= f'_t((x_{t-K},\psi_{t-K:t-1}),\psi_t,(x_{t-K+1},y_{t-K})) && \text{(E.16)}\\ &= (x_{t-K+1},\psi_{t-K+1:t|y_{t-K}})\\ &\forall x_{t-K:t-K+1}\in X_{t-K+1},\quad \forall y_{t-K}\in Y_{t-K},\quad \forall\psi_{t-K+1:t}\in\check{\Gamma}_{t-K+1:t|\Delta_{t-K}}, && \text{(E.17)}\end{aligned}$$

and the cost function g'_t is defined by

$$\begin{aligned} g'_t(x'_{t+1},u'_t) &= g'_t((x_{t-K+1},\psi_{t-K+1:t|y_{t-K}}),\psi_t) && \text{(E.18)}\\ &= E[h_t(x_{t+1},u_t)|x_{t-K+1};G(\psi_{t-K+1:t|y_{t-K}})],\\ &\forall x_{t-K+1}\in X_{t-K+1},\quad \forall y_{t-K}\in Y_{t-K},\quad \forall\psi_{t-K+1:t}\in\check{\Gamma}_{t-K+1:t|\Delta_{t-K}}. && \text{(E.19)}\end{aligned}$$

It may appear strange to define the disturbance as the next state, but the disturbance defined as such satisfies the properties of a disturbance variable, namely that it is independent of previous disturbances given the current state and the input.

A stationary optimal undiscounted infinite horizon control law for the K-step delay state problem will exist if this equivalent centralized control problem has such a stationary optimal infinite horizon solution. The conditions under which such a solution exists are described by Bertsekas in [8]. If the conditions are satisfied, then an optimal control law can be found efficiently by known methods, including, for example, Howard's policy iteration algorithm [31].

We see immediately from this equivalent centralized stochastic control problem that the state space will likely be enormous, a situation that makes even the dynamic programming algorithm computationally expensive. More precisely, the size of the state space is

$$|X_{t-K}\times\check{\Gamma}_{t-K:t-1|\Delta_{t-K}}| = |X_{t-K}|\cdot\prod_{m=1}^{M}|U_t^m|^{|Y^m_{t-K:t-1}|}, \tag{E.20}$$

where $|Y^m_{t-K:t-1}|$ grows exponentially with K, implying that the state space grows *doubly* exponentially with K. For a certain class of systems that includes the MABC, the complexity can be substantially reduced, although the growth rate remains doubly exponential. We describe this class and the method by which complexity can be reduced in the following subsection.

E.4.2 Complexity Reduction

We describe a method of reducing complexity when the controllers in the problem of Section 4.1 have common information, e.g., the ternary feedback in the MABC described in Section 5. Although the case of common information can be handled using the methods described in Section 4.1, substantial computational savings result from exploiting the common information.

A system with common information and K-step delay state retains the elements of the system in Section 4.1 with one modification. Namely, the observation y^m_t can be partitioned into a local observation and a common observation. Specifically, we can write $y^m_t = (\lambda^m_t, \xi_t)$, for all $m = 1, \cdots, M$, where ξ_t is the common observation since every controller observes it, and λ^m_t is the mth controller's local observation. We denote their ranges by Ξ_t and Λ^m_t, respectively. The shared information δ_t can then be written

$$\delta_t = (\xi_{0:t-1}, \lambda_{0:t-1}, x_{0:t}), \tag{E.21}$$

and the control laws take the form

$$\begin{aligned} u^m_t &= \gamma^m_t(y^m_{t-K:t}, \delta_{t-K}) && \text{(E.22)} \\ &= \gamma^m_t(\lambda^m_{t-K:t}, \xi_{t-K:t}, \delta_{t-K}). && \text{(E.23)} \end{aligned}$$

For future convenience, we denote $(\delta_{t-K}, \xi_{t-K:t})$ by θ_t and its range by Θ_t.

Exploiting the common information, we arrive at the following modified theorem characterizing optimal control laws.

Theorem E.4.2 Consider the following recursive equations that characterize optimal control laws for the K-step delay state system with common information:

$$\begin{aligned} J^*_T(x_{T-K}, \xi_{T-K:T}, \psi_{T-K:T-1}) &= 0, \quad \forall \psi_{T-K:T-1} \in \check{\Gamma}_{T-K:T-1|\Theta_T} && \text{(E.24)} \\ J_t(x_{t-K}, \xi_{t-K:t}, \psi_{t-K:t-1}, \psi_t) &= E[h_t(x_{t+1}, u_t) + J^*_{t+1}(x_{t-K+1}, \xi_{t-K+1:t+1}, \psi_{t-K+1:t|\lambda_{t-K}}) \\ &\qquad\qquad |x_{t-K}, \xi_{t-K:t}; G(\psi_{t-K:t})], \\ &\quad \forall \psi_{t-K:t} \in \check{\Gamma}_{t-K:t|\Theta_t} && \text{(E.25)} \\ J^*_t(x_{t-K}, \xi_{t-K:t}, \psi_{t-K:t-1}) &= \min_{\psi_t \in \check{\Gamma}_{t|\Theta_t}} J_t(x_{t-K}, \xi_{t-K:t}, \psi_{t-K:t-1}, \psi_t), \\ &\quad \forall \psi_{t-K:t-1} \in \check{\Gamma}_{t-K:t-1|\Theta_t} && \text{(E.26)} \end{aligned}$$

Let $\gamma_{0:K-1} \in \check{\Gamma}_{0:K-1}$ be given optimal control laws. If control laws $\gamma_{K:T-1} \in \check{\Gamma}_{K:T-1}$ satisfy

$$J_t(x_{t-K}, \xi_{t-K:t}, \gamma_{t-K:t-1|\theta_t}, \gamma_{t|\theta_t}) = J^*_t(x_{t-K}, \xi_{t-K:t}, \gamma_{t-K:t-1|\theta_t}) \tag{E.27}$$

for all $\theta_t = (\lambda_{0:t-K-1}, \xi_{0:t}, x_{0:t-K}) \in \Theta_t$ and $t = K, \cdots, T-1$, then $\gamma_{K:T-1}$ are optimal.

Proof: See Appendix A. □

Again, we can interpret the above equations as the result of applying dynamic programming to the following centralized stochastic control problem. Let the state for the centralized problem be

$$x_t'' = (x_{t-K}, \xi_{t-K:t}, \psi_{t-K:t-1}) \in X_{t-K} \times \Xi_{t-K:t} \times \check{\Gamma}_{t-K:t-1|\Theta_t}, \tag{E.28}$$

let the input be $u_t'' = \psi_t \in \check{\Gamma}_{t|\Theta_t}$, and let the disturbance be

$$w_t'' = (x_{t-K+1}, \xi_{t-K+1:t+1}, \lambda_{t-K}) \in X_{t-K+1} \times \Xi_{t-K+1:t+1} \times \Lambda_{t-K}. \tag{E.29}$$

Define the transition function f_t'' by

$$\begin{aligned} f_t''(x_t'', u_t'', w_t'') &= f_t''((x_{t-K}, \xi_{t-K:t}, \psi_{t-K:t-1}), \psi_t, (x_{t-K+1}, \xi_{t-K+1:t+1}, \lambda_{t-K})) && \text{(E.30)} \\ &= (x_{t-K+1}, \xi_{t-K+1:t+1}, \psi_{t-K+1:t|\lambda_{t-K}}), && \text{(E.31)} \end{aligned}$$

and define the cost function g_t'' by

$$\begin{aligned} g_t''(x_{t+1}'', u_t'') &= g_t''((x_{t-K+1}, \xi_{t-K+1:t+1}, \psi_{t-K+1:t|\lambda_{t-K}}), \psi_t) && \text{(E.32)} \\ &= E[h_t(x_{t+1}, u_t)|x_{t-K+1}, \xi_{t-K+1:t+1}; G(\psi_{t-K+1:t|\lambda_{t-K}})]. && \text{(E.33)} \end{aligned}$$

Again, in [8], Bertsekas gives the conditions under which a stationary optimal undiscounted infinite horizon control law exists for this centralized problem.

Note that the size of the state space for this centralized stochastic control problem is now

$$|X_{t-K} \times \Xi_{t-K:t} \times \check{\Gamma}_{t-K:t-1|\Theta_t}| = |X_{t-K}| \cdot |\Xi_{t-K:t}| \cdot \prod_{m=1}^{M} |U_t^m|^{|\Lambda_{t-K:t-1}^m|}, \tag{E.34}$$

and that $|\Xi_{t-K:t}|$ is not in the exponent of $|U_t^m|$ as it would be had we used the method in Section 4.1. However, the growth rate of the state space remains doubly exponential with K since $|\Lambda_{t-K:t-1}^m|$ grows exponentially with K. Nevertheless, even for small problems. the efficiency of the above algorithm allows an enormous reduction in required computation when compared to the algorithm in Section 4.1.

E.5 Multiple Access Broadcast Channel

In this section, we focus our attention on the canonical MABC. We describe the canonical MABC and formulate the problem of designing protocols for the canonical MABC as a decentralized control problem with no information sharing. We then apply the bounding techniques described in Section 4.2 and develop specific results for the cases of two and three users, which are most computationally feasible.

We introduce notation to facilitate discussion of the MABC. Let q_t^{m-} be the number of packets in the mth user's buffer (either 1 or 0) prior to the arrivals for that time slot; let q_t^{m+} be the number of packets in the mth users' buffer (either 1 or 0) after the arrivals for that slot. Let b_t be the feedback after the tth slot. Let a_t^m be the number of arrivals (either 1 or 0) to user m during the tth slot. The control applied by the mth user is denoted u_t^m and specifies whether the user will ($u_t^m = 1$)

or will not ($u_t^m = 0$) transmit a packet in his buffer. The number of packets transmitted by user m (although not necessarily received by the receiver) is s_t^m (either 1 or 0).

Denoting the Boolean logic operators "and" and "or" by $\wedge$ and $\vee$, respectively, we describe the operation of the channel as follows:

$$\vdots$$

$t.1$: The pre-arrival buffer state is q_t^{m-}.

$t.2$: Arrivals a_t^m to each user occur independently with probability p.

$t.3$: Post-arrival queue state is $q_t^{m+} = a_t^m \vee q_t^{m-}$.

$t.4$: Number of packets transmitted is $s_t^m = q_t^{m+} \wedge u_t^m$.

$t.5$: Feedback is $b_t = \min\{\sum_{m=1}^{M} s_t^m, 2\}$.

$(t+1).1$: The next pre-arrival buffer state is

$$q_{t+1}^{m-} = \begin{cases} q_t^{m+} - s_t^m & \text{if } b_t = 1 \\ q_t^{m+} & \text{otherwise} \end{cases} \tag{E.35}$$

$$\vdots$$

By assumption, each user may use the history of feedback broadcasts $b_{0:t-1}$ and its history of local post-arrival buffer states $q_{0:t}^{m+}$ to decide its control input u_t^m.

Let us now put the canonical MABC design problem in the decentralized control framework of Section 2. Let the state be

$$x_t = (q_t^-, b_{t-1}). \tag{E.36}$$

Let the primitive random variables be $v_t = a_t$ and $w_t^m = a_t^m$. Then the state transition function f_t is given by

$$x_{t+1} = f_t(x_t, u_t, v_t) = (\phi(q_t^-, u_t, a_t), \beta(q_t^-, u_t; a_t)), \tag{E.37}$$

where the functions ϕ and β are defined as follows:

$$\beta(q_t^-, u_t, a_t) = \min\{\sum_{m=1}^{M}((q_t^{m-} \vee a_t^m) \wedge u_t^m), 2\} \tag{E.38}$$

$$\phi(q_t^-, u_t, a_t) = \begin{cases} (q_t^- \vee a_t) - ((q_t^- \vee a_t) \wedge u_t) & \text{if } \beta(q_t^-, u_t, a_t) = 1 \\ q_t^- \vee a_t & \text{otherwise} \end{cases}, \tag{E.39}$$

where binary operations on tuples are performed element by element. The function g_t^m relating observations to the state is defined as

$$y_t^m = (\lambda_t^m, \xi_t) = g_t^m(x_t, a_t^m) = (q_t^{m-} \vee a_t^m, b_{t-1}), \tag{E.40}$$

and the cost function h_t is defined by

$$h_t(x_{t+1}, u_t) = \begin{cases} -1 & \text{if } b_t = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{E.41}$$

so that minimizing the expected cost will maximize the probability of successful packet reception in a slot. As mentioned earlier, the control input u_t^m must be a function of $b_{0:t-1}$ and $q_{0:t}^{m+}$, i.e., it must be a function of $y_{0:t}^m$, indicating that no sharing is allowed.

With these definitions, we have formulated the canonical MABC protocol design problem as a decentralized control problem with common information but with no sharing. The corresponding problem with K-step delay state from Section 4.2 can now be solved, and the resulting optimal throughput will provide an upper bound to the throughput of the canonical MABC.

E.5.1 Results for Two and Three Users

We use Theorem 2 with Howard's policy iteration algorithm [31] to find the bound described in Section 4.2. The tightness of the bound increases as the delay K increases, so it is desirable to compute the bound for the largest K possible. For reference, with respect to the decentralized control formulation of the MABC given in Section 5, the OSDS problem considered by Grizzle et al. [27] and Paradis [42] is equivalent to a K-step delayed state problem with $K = 0$. For two users, we choose the delay to be $K = 1$, while for three users, we choose a delay of $K = 2$. These choices were made because for two users, with $K = 1$, the bound meets the performance of the Hluchyj-Gallager protocol, while for three users, computation prohibits choosing $K > 2$.

Before using the policy iteration algorithm, we eliminate all self-contradictory states. For example, the state with sub-laws that never transmit cannot have a success or collision feedback associated with it. The remaining states meet the sufficient conditions required for an optimal stationary infinite-horizon solution to exist.

In the two-user case, numerical calculation of the bound with $K = 1$ shows that Hluchyj and Gallager's optimized window protocol achieves the bound for $K = 1$ to within machine precision. In the three-user case, numerical calculations of the bound for $K = 2$ show that Hluchyj and Gallager's optimized window protocol approaches the bound for $p < .3$ and meets the bound for $p > .3$. Note that for $p > .2891$, the Hluchyj-Gallager protocol is the same as time-division multiple access (TDMA) [28]. Because much computation is required for the $K = 2$ bound, the bound is only computed for the values of p in Table 1. For these values of p, the performance of the optimized window protocol is at least 99.59% of the bound, suggesting near optimality. Numerical calculation of the bound for three users with $K = 1$ show that for $p > .5$, the Hluchyj-Gallager (TDMA) protocol is optimal. Thus, we conclude that for $p > .3$, the Hluchyj-Gallager (TDMA) protocol is optimal.

E.6 Discussion and Conclusions

We have presented a bounding technique for decentralized control problems with no sharing of information and showed how complexity of the bound calculation can be reduced in the special case of common information. The complexity reduction has allowed us to apply the bound to the canonical MABC design problem for two and three users. We have developed results that show that

p	*Probability of Success*		Ratio (≤ 1)
	Hluchyj-Gallager Protocol	$K = 2$ Bound	
.05	.1486	.1486	1.000
.10	.2881	.2882	.9996
.15	.4099	.4103	.9989
.20	.5081	.5101	.9961
.25	.5905	.5926	.9965
.28	.6301	.6327	.9959
.29	.6421	.6446	.9961
.30	.6570	.6570	1.000
.40	.7840	.7840	1.000
.50	.8750	.8750	1.000

Table E.1: Three-user case: bounds on probability of success with $K = 2$ compared to probability of success of Hluchyj and Gallager's optimized window protocol.

the performance of Hluchyj and Gallager's optimized window protocol meets and almost meets the bound in the two and three user cases, respectively.

The results open several avenues for further inquiry. That the Hluchyj-Gallager optimized window protocol is at least nearly optimal for two and three users suggests that this protocol may be nearly optimal for more than three users; it may be fruitful to try to show this analytically. Alternatively, since the optimized window protocol uses only the common information [28], i.e., the broadcast feedback, it may be possible instead to show that no performance loss results from restricting protocols to the class of protocols that use only common information. Another interesting task is to determine p such that TDMA is the optimal protocol for $M > 3$ users. It is known that TDMA is optimal as $p \to 1$, but our results suggest that TDMA is optimal for values of p that are substantially smaller than 1. Finally, since the main impediment to computing tighter bounds for larger numbers of users is the large state space and large input space, the application of neurodynamic programming methods and other efficient numerical optimization techniques should be considered.

E.7 Proof of Theorem E.4.2

We require 4 lemmas before we can prove the theorem.

Lemma E.7.1 Given control laws $\gamma_{t-K:t} \in \check{\Gamma}_{t-K:t}$, there exists a function $\tilde{\pi}$ such that

$$p(x_{t-K+1:t+1}, \lambda_{t-K:t+1}, \xi_{t-K:t+1} | \delta_{t-K}; \gamma_{t-K:t}) = \tilde{\pi}(x_{t-K:t+1}, \lambda_{t-K:t+1}, \xi_{t-K:t+1}, \gamma_{t-K:t|\theta_t}) \tag{E.42}$$

Remark: Note that many of the densities in subsequent proofs can be derived from the above via integration over the appropriate variables.

Proof: To prove this lemma, we use an inductive argument in which the main tool is the chain rule for probability density functions. If we put $j = t - K$, then the last link of the chain rule can be written

$$p(\lambda_j, \xi_j \mid \delta_j; \gamma_{j:j+K}) = p(\lambda_j, \xi_j \mid x_j). \tag{E.43}$$

Using the system dynamical equations (E.1) and (E.2), along with the fact that the primitive random variables are independent at distinct times, and the fact that the control laws $G(\gamma_{j:j+K|\theta_{j+K}})$ return a constant control input over the argument θ_{j+K}, we combine the last two links in the chain by

$$\begin{aligned}
&p(x_{j+1}, \lambda_{j:j+1}, \xi_{j:j+1} \mid \delta_j; \gamma_{j:j+K}) \\
&= p(\lambda_j, \xi_j \mid x_j) \cdot p(x_{j+1} \mid \delta_j, x_j, \lambda_j, \xi_j; G(\gamma_{j|\delta_j,\xi_j})) \cdot p(\lambda_{j+1}, \xi_{j+1} \mid \delta_j, x_{j:j+1}, \lambda_j, \xi_j) \qquad \text{(E.44)} \\
&= p(\lambda_j, \xi_j \mid x_j) \cdot p(x_{j+1} \mid x_j, \lambda_j; G(\gamma_{j|\delta_j,\xi_j})) \cdot p(\lambda_{j+1}, \xi_{j+1} \mid x_{j+1}). \qquad \text{(E.45)}
\end{aligned}$$

Examining the dependence of (E.45) on δ_j, we conclude that a function $\tilde{\pi}_1$ exists such that

$$p(x_{j+1}, \lambda_{j:j+1}, \xi_{j:j+1} \mid \delta_j; \gamma_{j:j+K}) = \tilde{\pi}_1(x_{j:j+1}, \lambda_{j:j+1}, \xi_{j:j+1}, \gamma_{j|\delta_j,\xi_j}). \tag{E.46}$$

Let us now set up a proof by induction by supposing that there exists a function $\tilde{\pi}_s$ such that

$$p(x_{j+1:j+s}, \lambda_{j:j+s}, \xi_{j:j+s} \mid \delta_j; \gamma_{j:j+K}) = \tilde{\pi}_s(x_{j:j+s}, \lambda_{j:j+s}, \xi_{j:j+s}, \gamma_{j:j+s-1|\delta_j,\xi_{j:j+s-1}}) \tag{E.47}$$

for some s such that $1 \leq s \leq K - 1$. Using the same argument used to derive (E.45), we write

$$\begin{aligned}
&p(x_{j+s+1}, \lambda_{j+s+1}, \xi_{j+s+1} \mid \delta_j, x_{j+1:j+s}, \lambda_{j:j+s}, \xi_{j:j+s}; \gamma_{j:j+K-1}) \\
&= p(x_{j+s+1}|\delta_j, x_{j:j+s}, \lambda_{j:j+s}, \xi_{j:j+s}; G(\gamma_{j+s|\delta_j,\xi_{j:j+s}})) \cdot \\
&\quad p(\lambda_{j+s+1}, \xi_{j+s+1}|\delta_j, x_{j:j+s+1}, \lambda_{j:j+s}, \xi_{j:j+s}) \qquad \text{(E.48)} \\
&= p(x_{j+s+1}|x_{j+s}, \lambda_{j:j+s}; G(\gamma_{j+s|\delta_j,\xi_{j:j+s}})) \cdot p(\lambda_{j+s+1}, \xi_{j+s+1}|x_{j+s+1}). \qquad \text{(E.49)}
\end{aligned}$$

Since (E.49) depends on δ_j only through $\gamma_{j+s|\delta_j,\xi_{j:j+s}}$, we can use the chain rule to write

$$p(x_{j+1:j+s+1}, \lambda_{j:j+s+1}, \xi_{j:j+s+1} \mid \delta_j; \gamma_{j:j+K}) = \tilde{\pi}_s(x_{j:j+s}, \lambda_{j:j+s}, \xi_{j:j+s}, \gamma_{j:j+s|\delta_j,\xi_{j:j+s}}) \cdot (\,*\,), \tag{E.50}$$

where ($*$) represents the right hand side of (E.49). Then (E.50) implies that there exists a function $\tilde{\pi}_{s+1}$ such that

$$p(x_{j+1:j+s+1}, \lambda_{j:j+s+1}, \xi_{j:j+s+1} \mid \delta_j; \gamma_{j:j+K}) = \tilde{\pi}_{s+1}(x_{j:j+s+1}, \lambda_{j:j+s+1}, \lambda_{j:j+s+1}, \gamma_{j:j+s|\delta_j,\xi_{j:j+s}}). \tag{E.51}$$

By induction, it follows that there exists a function $\tilde{\pi} = \tilde{\pi}_K$ such that

$$p(x_{j+1:j+K+1}, \lambda_{j:j+K+1}, \xi_{j:j+K+1}|\delta_j; \gamma_{j:j+K}) = \tilde{\pi}_K(x_{j:j+K+1}, \lambda_{j:j+K+1}, \xi_{j:j+K+1}, \gamma_{j:j+K|\delta_j,\xi_{j:j+K}}). \tag{E.52}$$

This is what we wished to show. □

Lemma E.7.2 Define the immediate cost function C_t by

$$C_t(\gamma_{0:T-1}, \theta_t) = E\left[h(x_{t+1}, u_t) \mid \theta_t; \gamma_{t-K:t}\right]. \tag{E.53}$$

Then there exists a well-defined function $\tilde{C}_t$ such that

$$C_t(\gamma_{0:T-1}, \theta_t) = \tilde{C}_t(\gamma_{t-K:t|\theta_t}, \xi_{t-K:t}, x_{t-K}). \tag{E.54}$$

Proof: We can write explicitly

$$\begin{aligned} &E[h(x_{t+1}, u_t) \mid \theta_t; \gamma_{t-K:t}] \\ &= \sum_{\lambda_{t-K:t}} h(x_{t+1}, \gamma_{t|\theta_t}(\lambda_{t-K:t})) \cdot p(\lambda_{t-K:t}|\theta_t; \gamma_{t-K:t}) && \text{(E.55)} \\ &= \sum_{\lambda_{t-K:t}} h(x_{t+1}, \gamma_{t|\theta_t}(\lambda_{t-K:t})) \cdot p(\lambda_{t-K:t}|x_{t-K}, \xi_{t-K:t}; \gamma_{t-K:t|\theta_t}), && \text{(E.56)} \end{aligned}$$

and the lemma follows immediately. □.

Lemma E.7.3 Let

$$D_t(\gamma_{0:T-1}, \theta_t) = E\left[\eta(\delta_{t-K+1}, \xi_{t-K+1:t+1}) \mid \theta_t; \gamma_{t-K:t}\right], \tag{E.57}$$

where η is some real-valued function. Then there exists a function D'_t such that

$$D'_t(\gamma_{t-K:t|\theta_t}\xi_{t-K:t}, x_{t-K}) = D_t(\gamma_{0:T-1}, \theta_t). \tag{E.58}$$

Proof: Follows immediately from Lemma 1. □

Lemma E.7.4 Consider the following recursive exhaustive search equations

$$\begin{aligned} J^*_T(\theta_T, \gamma_{0:T-1}) &= 0, \quad \forall \gamma_{0:T-1} \in \check{\Gamma}_{0:T-1}, \quad \forall \theta_T \in \Theta_T && \text{(E.59)} \\ J_t(\theta_t, \gamma_{0:t-1}, \gamma_t) &= E[h_t(x_{t+1}, u_t) + J^*_{t+1}(\theta_{t+1}, \gamma_{0:t})|\theta_t; \gamma_{0:t}] && \text{(E.60)} \\ J^*_t(\theta_t, \gamma_{0:t}) &= \min_{\gamma_t \in \check{\Gamma}_t} J_t(\theta_t, \gamma_{0:t-1}, \gamma_t). && \text{(E.61)} \end{aligned}$$

Let control laws $\gamma_{0:K-1} \in \check{\Gamma}_{0:K-1}$ be given optimal control laws. If control laws $\gamma_{K:T-1} \in \check{\Gamma}_{K:T-1}$ satisfy

$$J_t(\theta_t, \gamma_{0:t-1}, \gamma_t) = J^*_t(\theta_t, \gamma_{0:t-1}) \tag{E.62}$$

for all $\theta_t \in \Theta_t$ and $t = K, \cdots, T-1$, then $\gamma_{K:T-1}$ is optimal.

Proof: The proof follows from recognizing that $J^*_t(\theta_t, \gamma_{0:t-1})$ is the same as

$$\min_{\gamma_{t:T-1}} E[\sum_{\tau=t}^{T-1} h_\tau(x_{\tau+1}, u_\tau)|\theta_t; \gamma_{0:T-1}]. \tag{E.63}$$

Then it is clear that γ_K is the optimal control law at time K, given $\gamma_{0:K-1}$. Knowing that γ_K is optimal allows us to similarly conclude that γ_{K+1} given that $\gamma_{0:K}$ is optimal and so on. □

We are now prepared to prove Theorem 2.

Proof of Theorem 2:

We now use the lemmas to show that the equations can equivalently be written as stated in the theorem. First, let $\tilde{J}_t$ and $\tilde{J}_t^*$ be functions satisfying

$$\tilde{J}_t^*(x_{t-K}, \xi_{t-K:t}, \gamma_{0:t-1|\delta_{t-K},\xi_{t-K:t-1}}) = J_t^*(\theta_t, \gamma_{0:t-1}) \tag{E.64}$$

$$\tilde{J}_t(x_{t-K}, \xi_{t-K:t}, \gamma_{0:t-1|\delta_{t-K},\xi_{t-K:t-1}}, \gamma_{t|\delta_{t-K},\xi_{t-K:t-1}}) = J_t(\theta_t, \gamma_{0:t-1}, \gamma_t). \tag{E.65}$$

Then by Equation (E.24), $\tilde{J}_T^*$ exists, from which it follows that $\tilde{J}_{T-1}$ exists. It follows from Lemmas 3 and 4 that if $\tilde{J}_{t+1}^*$ exists, then so do $\tilde{J}_t$ and $\tilde{J}_t^*$. Therefore, by induction, the foregoing functions exist for all t. Replacing the equations in the statement of Lemma 4 with the corresponding tilde functions, we conclude that if a control law $\gamma_{K:T-1}$ satisfies

$$\tilde{J}_t(x_{t-K}, \xi_{t-K:t}, \gamma_{0:t-1|\delta_{t-K},\xi_{t-K:t-1}}, \gamma_{t|\delta_{t-K},\xi_{t-K:t}}) = \tilde{J}_t^*(x_{t-K}, \xi_{t-K:t}, \gamma_{0:t-1|\delta_{t-K},\xi_{t-K:t-1}}) \tag{E.66}$$

then $\gamma_{K:T-1}$ is optimal. This is equivalent to the theorem statement. □

Bibliography

[1] R. Ahlswede. A constructive proof of the coding theorem for discrete memoryless channels with feedback. In *Proceedings of the Sixth Prague Conference on Information Theory, Statistical Decision Functions, and Random Processes*, pages 39–50, 1971.

[2] R. Ahlswede. Multi-way communication channels. In *Proceedings of the 2nd International Symposium on Information Theory*, pages 23–53, 1971.

[3] M. Aicardi, F. Davoli, and R. Minciardi. Decentralized optimal control of Markov chains with a common past information set. *IEEE Transactions on Automatic Control*, 32(11):1028–1031, November 1987.

[4] F. Alajaji. Feedback does not increase the capacity of discrete channels with additive noise. *IEEE Transactions on Information Theory*, 41(2):546–549, March 1995.

[5] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(3):284–287, March 1974.

[6] M. S. Bargh and J. P. M. Schalkwijk. Coding for channels with low rate noiseless feedback. In *Proceedings of the 1997 IEEE International Symposium on Information Theory*, page 129, 1997.

[7] E. R. Berlekamp. *Block Coding with Noiseless Feedback.* PhD thesis, Massachusetts Institute of Technology, 1964.

[8] D. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* Prentice Hall, 1987.

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

[10] P. Billingsley. *Probability and Measure.* John Wiley and Sons, 1986.

[11] M. V. Burnashev. Data transmission over a discrete channel with feedback. Random transmission time. *Problems of Information Transmission*, 12(4):250–265, 1976.

[12] T. M. Cover, A. A. El Gamal, and M. Salehi. Multiple access channels with arbitrarily correlated sources. *IEEE Transactions on Information Theory*, 26(6):648–657, November 1980.

[13] T. M. Cover and C. S. K. Leung. An achievable rate region for the multiple-access channel with feedback. *IEEE Transactions on Information Theory*, 27(3):292–298, 1981.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[15] I. Csiszár and J. Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, 1981.

[16] L. Ekroot and T. M. Cover. The entropy of a randomly stopped sequence. *IEEE Transactions on Information Theory*, 37(6):1641–1644, November 1991.

[17] L. Ekroot and T. M. Cover. The entropy of Markov trajectories. *IEEE Transactions on Information Theory*, 39(4):1418–1421, July 1993.

[18] A. A. El Gamal and T. M. Cover. Achievable rates for multiple descriptions. *IEEE Transactions on Information Theory*, 28(6):851–857, November 1982.

[19] P. Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37(1):5–12, January 1991.

[20] W. H. R. Equitz and T. M. Cover. Successive refinement of information. *IEEE Transactions on Information Theory*, 37(2):269–275, March 1991.

[21] G. D. Forney. *Concatenated Codes*. MIT Press, 1966.

[22] N. T. Gaarder and J. K. Wolf. The capacity region of a multiple-access discrete memoryless channel can increase with feedback. *IEEE Transactions on Information Theory*, 21:100–102, January 1975.

[23] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley and Sons, Inc., 1968.

[24] R. G. Gallager. A perspective on multiaccess channels. *IEEE Transactions on Information Theory*, 31(2):124–142, March 1985.

[25] R. G. Gallager. *Discrete Stochastic Processes*. Kluwer Academic Publishers, 1996.

[26] A. J. Goldsmith and P. P. Varaiya. Capacity, mutual information, and coding for finite-state Markov channels. *IEEE Transactions on Information Theory*, 42(3):868–886, May 1996.

[27] J. W. Grizzle, S. I. Marcus, and K. Hsu. Decentralized control of a multiaccess broadcast network. In *Proceedings of the 20th IEEE Conference on Decision and Control*, pages 390–391, 1981.

[28] M. G. Hluchyj and R. G. Gallager. Multiaccess of a slotted channel by finitely many users. In *Proceedings of National Telecommunications Conference*, pages D4.2.1–D4.2.7, 1981.

[29] M. Horstein. Sequential transmission of digital information with feedback. *M.I.T. R.L.E. Technical Report*, September 1960.

[30] M. Horstein. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9:136–143, July 1963.

[31] R. Howard. *Dynamic Programming and Markov Processes.* The M.I.T. Press, 1960.

[32] J. Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18:652–656, September 1972.

[33] R. L. Kashyap. Feedback coding schemes for an additive noise channel with a noisy feedback link. *IEEE Transactions on Information Theory*, 14(3):471–480, May 1968.

[34] B. D. Kudryashov. Message transmission over a discrete channel with noiseless feedback. *Problems of Information Transmission*, 15(1):1–9, 1979.

[35] B. Kurtaran. Decentralized stochastic control with delayed sharing information pattern. *IEEE Transactions on Automatic Control*, 21:576–581, August 1976.

[36] B. Kurtaran. Corrections and extensions to 'Decentralized control with delayed sharing information pattern'. *IEEE Transactions on Automatic Control*, 24(4):656–657, August 1979.

[37] A. Lapidoth. Personal communication. August 1997.

[38] A. Lapidoth and J. Ziv. Universal decoding for noisy channels: An algorithmic approach. In *Proceedings of the 1997 IEEE International Symposium on Information Theory*, page 399, 1997.

[39] H. Liao. *Multiple-Access Channels.* PhD thesis, University of Hawaii, Honolulu, 1972.

[40] M. Mushkin and I. Bar-David. Capacity and coding for the Gilbert-Elliott channels. *IEEE Transactions on Information Theory*, 35(6):1277–1290, November 1989.

[41] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1991.

[42] A. Paradis. Application optimal control to the multiple access channel. Master's thesis, Massachusetts Institute of Technology, 1981.

[43] W. H. Press, B. P. Flannery, and S. A. Teukolsky. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1992.

[44] J. G. Proakis. *Digital Communication.* McGraw-Hill, 1995.

[45] Z. Rosberg. Optimal decentralized control in a multiaccess channel with partial information. *IEEE Transactions on Automatic Control*, 28(2):187–193, February 1983.

[46] J. P. M. Schalkwijk. A coding scheme for additive noise channels with feedback – Part II: Band-limited signals. *IEEE Transactions on Information Theory*, 12(2):183–189, April 1966.

[47] J. P. M. Schalkwijk. A class of simple and optimal strategies for block coding on the binary symmetric channel with noiseless feedback. *IEEE Transactions on Information Theory*, 17(3):283–287, May 1971.

[48] J. P. M. Schalkwijk and M. E. Barron. Sequential signaling under a peak power constraint. *IEEE Transactions on Information Theory*, 17(3):278–282, May 1971.

[49] J. P. M. Schalkwijk and T. Kailath. A coding scheme for additive noise channels with feedback – Part I: No bandwidth constraint. *IEEE Transactions cn Information Theory*, 12(2):172–182, April 1966.

[50] J. P. M. Schalkwijk and K. A. Post. On the error probability for a class of binary recursive feedback strategies. *IEEE Transactions on Information Theory*, 19(4):498–511, July 1973.

[51] F. C. Schoute. Decentralized control in packet switched satellite communication. *IEEE Transactions on Automatic Control*, 23:362–371, April 1979.

[52] C. E. Shannon. The zero-error capacity of a noisy channel. In *Transactions of the 1956 Symposium on Information Theory*, pages 8–19, September 1956.

[53] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois, 1948.

[54] M. Sipser and D. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, November 1996.

[55] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 4:471–480, July 1973.

[56] D. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, November 1996.

[57] P. Varaiya and J. Walrand. On delayed sharing patterns. *IEEE Transactions on Automatic Control*, 23(3):443–445, June 1978.

[58] P. Varaiya and J. Walrand. Decentralized control in packet switched satellite communication. *IEEE Transactions on Automatic Control*, 24(5):794–796, October 1979.

[59] T. Veugen. Capacity-achieving strategies for discrete memoryless channels with feedback. In *Proceedings of the 1994 IEEE International Symposium on Information Theory*, page 466, 1994.

[60] T. Veugen. Multiple repetition feedback coding for discrete memoryless channels. In *Proceedings of the 1995 IEEE International Symposium on Information Theory*, page 457, 1995.

[61] T. Veugen. *Multiple-repetition coding for channels with feedback.* PhD thesis, Eindhoven University of Technology, 1997.

[62] H. S. Witsenhausen. Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, 59(11):1557–1566, November 1971.

[63] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.

[64] A. D. Wyner. The common information of two dependent random variables. *IEEE Transactions on Information Theory*, 21:163–179, 1975.

[65] H. Yamamoto and K. Itoh. Asymptotic performance of a modified Schalkwijk-Barron scheme for channels with noiseless feedback. *IEEE Transactions on Information Theory*, 25(6):729–733, November 1979.

[66] J. Ziv. Universal decoding for finite-state channels. *IEEE Transactions on Information Theory*, 31:453–460, 1985.

[67] J. Ziv and A. Lempel. Compression of individual sequences by variable rate coding. *IEEE Transactions on Information Theory*, 24:530–536, 1978.

THESIS PROCESSING SLIP

FIXED FIELD: ill ______________ name ______________

index ______________ biblio ______________

▶ COPIES [✓ Archives] Aero Dewey [✓ Eng] Hum

Lindgren Music Rotch Science

TITLE VARIES ▶□ ______________

NAME VARIES ▶□ ______________

IMPRINT (COPYRIGHT) ______________

▶COLLATION 185p

▶ADD DEGREE: ______________ ▶ DEPT.: ______________

SUPERVISORS: ______________

NOTES:

cat'r: date:

page: ▶ F50

▶DEPT: E.E.

▶YEAR: 1998 ▶ DEGREE: Ph.D.

▶NAME: OOI, James Meng-Hsien