COMPLEXITY-FEEDBACK TRADEOFFS AND CAPACITY RESULTS FOR PACKET ERASURE NETWORKS

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Srinath Puducheri Sundaravaradhan

Thomas E. Fuja, Director

Graduate Program in Electrical Engineering
Notre Dame, Indiana
April 2010

COMPLEXITY-FEEDBACK TRADEOFFS AND CAPACITY RESULTS FOR PACKET ERASURE NETWORKS

Abstract

by

Srinath Puducheri Sundaravaradhan

Many communication networks are well-modeled as *packet erasure* networks, as packets transmitted over these networks are either received correctly at the destination or are "erased"; a packet erasure occurs when an error-corrupted packet is detected and discarded, or when a packet is dropped due to congestion in the network. This dissertation investigates two problems related to communicating reliably over packet erasure networks, adopting two different views of the network, viz., (i) a point-to-point erasure channel (that models either a single link or end-to-end communication), and (ii) a network of erasure links.

Reliable communication over a point-to-point erasure channel can be accomplished in one of two ways: 1) incorporating redundant packets in the transmitted packet sequence, i.e., via forward-error-correction (FEC) techniques, or 2) using feedback to request re-transmission of erased packets, i.e., automatic-repeat-request (ARQ) protocols.

This dissertation presents new constructions of hybrid ARQ protocols (i.e., protocols combining FEC and ARQ) for the point-to-point erasure channel. These protocols use Tornado codes (a class of LDPC codes) for erasure correction. The focus is on enabling and characterizing trade-offs between *costs* associated with

FEC (i.e., computational complexity of encoding/decoding) and ARQ (the amount of feedback utilized). The described protocols provide efficient trade-offs and can offer significant savings in computational/feedback requirements in several situations, compared to simple time-sharing between FEC and ARQ.

The second topic of this dissertation deals with reliable communication over two wireless relay networks – the multiple access relay channel (MARC) and the multiple relay channel (MRC) – wherein the links are memoryless erasure channels, and individual nodes *time-share* the use of the medium. The MARC is comprised of M independent sources that communicate with a common destination with the help of a single relay, while the MRC consists of a single source communicating with a single destination with the aid of M parallel relays.

The capacity region of the MARC and the capacity of the MRC are derived, assuming the destination has *perfect* knowledge of erasure patterns on all the links. Optimal bandwidth allocation strategies are obtained in closed-form as functions of the link parameters. These serve to highlight the utility of the relay(s) in various scenarios. Also, it is shown that easily-implemented capacity-approaching codes for the binary erasure channel, such as LDPC or Tornado codes, can be used at the link level to attain any achievable rate(s). Finally, these capacity results are *unchanged* in the presence of feedback of erasure location information to all nodes.

In loving memory of

my grandmother

Smt. Sakunthala Seshadri

CONTENTS

FIGURES

TABLES

ACKNOWLEDGMENTS

The successful completion of this dissertation is in no small measure due to the overwhelming support, help and encouragement I have received from numerous people, both before and during my tenure as a graduate student at Notre Dame.

Foremost, I would like to thank my advisor Prof. Thomas Fuja for what has been an exciting and intellectually fulfilling experience in pursuing my Ph.D. His guidance has been instrumental in shaping much of my research work; I have benefited enormously from his advice – both in identifying meaningful and interesting research problems, as well as in overcoming the challenges to solving them. Despite his extremely busy schedule as the department's chair, he always made time to meet with me on a regular basis; I have also had the freedom to be able to walk into his office any time to discuss research. As my advisor, he always let me conduct research in my own way, and was very patient and understanding during the times when I was stuck on a hard problem; his insight, advice and strong encouragement have often bailed me out of difficult situations. For this, I am deeply grateful to him.

I would like to thank Professors Daniel Costello, Nicholas Laneman and Vijay Gupta for agreeing to serve on my PhD committee, and for their careful reading of this dissertation and insightful comments.

Prof. Jörg Kliewer visited Notre Dame during 2005-2007 and co-advised me on my Master's thesis. I would like to thank him for several stimulating discussions

regarding our research and for his advice and encouragement during my formative years as a graduate student researcher.

The work in this dissertation was funded in part by the U.S. National Science Foundation and the Center for Applied Mathematics, University of Notre Dame. Their financial support is gratefully acknowledged.

I have been fortunate to have had excellent teachers in Professors Fuja, Costello, Laneman, Martin Haenggi and Yih-Fang Huang. Their courses laid the foundation of much of my knowledge and interest on advanced aspects of communication theory, and made for an enlightening and stimulating experience. My sincere thanks to all of them.

There are several current and former fellow grad students from whose friendship I have immensely benefited during my stay at Notre Dame. My close friend and former roommate Radhakrishna Ganti has been my go-to person for help and advice on innumerable topics, ranging from the abstract – the philosophy of doing research – to the very practical – installing Ubuntu Linux on my PC. My sincere thanks to him for all the help and support that he readily extended. I would also like to thank other roommates – Mahesh Mahadevan, Shivaprasad Kotagiri, Sundaram Vanka and Velmurugan Balaraman – whose company made my stay at 22 O'Hara Grace a very memorable and enjoyable one.

Many senior students in the department – Shyam Ranganathan, Krishnan Padmanabhan, Ajit Nimbalker, Ajay Gupta, Jagadish Venkataraman, Rajkumar Sankaralingam, Amaresh Malipatil, to name a few – extended invaluable support in helping me settle down, when I first arrived in Notre Dame. I extend my gratitude to all of them.

My sincere thanks to current and former group members Ajit Nimbalker, Lei

Xiao, Mahesh Mahadevan and Shashank Maiya for helpful discussions pertaining to my research – in particular, Ajit and Lei for being great “research mentors”. Thanks to Ali Pusane, Arvind Sridharan, Marcin Sikora and Christian Koller for illuminating various aspects of coding theory to me. Besides, many thanks are also due to Shashank, Shiva, Mahesh, Badri Tiwari, Sunil Srinivasa and Sundeep Venkatraman for their immense help and support – both within and outside of research.

I would also like to thank EE office staff members Tracy Cabello and Fanny Wheeler and international student advisor Samuel Lockhart at the ISO for their help and support with all administration and immigration related work, making it a very smooth and painless experience for me, and also for patiently answering the innumerable queries I always seemed to have.

None of what I have accomplished over the past six years at Notre Dame would have been possible but for the love and support of my parents and my little sister Preethi. They have been a constant source of joy, encouragement and motivation for me, particularly through some difficult phases of my PhD. Needless to say, I shall forever be indebted to them for their efforts, pains and sacrifices in helping me get to where I am. I would like to express my deepest gratitude and love to Appa, Amma and Preethi.

CHAPTER 1

INTRODUCTION

The model of an erasure channel, first introduced by Elias in 1956 ([1]), is perhaps the simplest among "noisy" communication system models. In its general form, an erasure channel between a transmitter and a receiver either allows error-free communication of a transmitted symbol, or replaces it with a special *erasure* symbol E at the output. More formally, if $\mathcal{X}$ denotes the (finite) input alphabet of the channel, then the output alphabet is given by $\mathcal{Y} = \mathcal{X} \cup \{E\}$; when any $x \in \mathcal{X}$ is transmitted over the channel, it is received as either x or E (an erasure), typically in a probabilistic fashion. Consequently, the receiver is able to identify exactly which transmitted symbols were erased, and the unerased symbols are detected without any ambiguity.

Erasure channels can be *memoryless*, wherein for any stream of transmitted symbols, erasures occur independently with some probability ϵ, or the channel could have memory, wherein the erasures follow a more complicated *correlated random process*. A special case of the erasure channel is when the input alphabet consists of *bits*, i.e, $\mathcal{X} = \{0, 1\}$ – this gives rise to the popular *binary erasure channel* (BEC) model, depicted in Fig. 1.1.

Despite its simplicity, an erasure channel is often used to accurately model modern **packet-based** communication systems. In particular, data transfer over

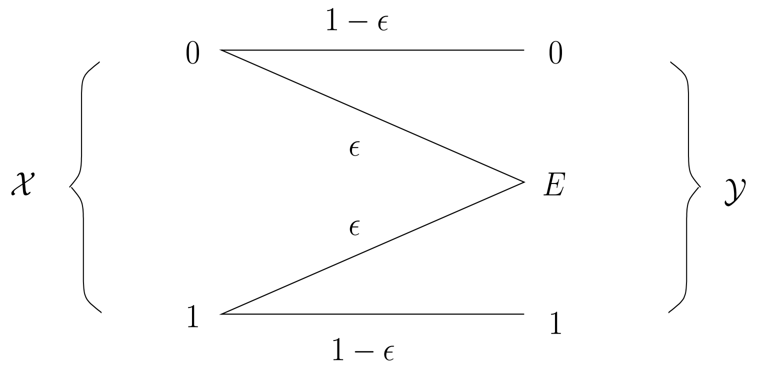


Figure 1.1. The binary erasure channel – transition probabilities

networks such as the Internet occurs in the form of *packets*, i.e., distinct formatted blocks of bits. When transmitted over a network, these packets are typically subject to impairments such as noise, interference from other transmissions, fading (such as in wireless networks), etc., which can result in errors in the received packets. Further, packets may be discarded or "dropped" at intermediate routers, as a result of congestion. Packets received with errors are detected via an internal check-sum mechanism; further, every corrupted or dropped packet can be *identified* by means of a unique, pre-assigned *sequence number*. Consequently, such packets can be thought of as "erasures" produced by the network (channel). In this situation, the end-to-end communication channel between a source and a destination can be modeled as either (i) a **single** point-to-point *packet erasure* channel, or (ii) a network of interconnecting links that individually behave as packet erasure channels.

In this dissertation, the model of a packet erasure channel is often simplified to its *binary* counterpart – the BEC. This is done primarily for convenience of analysis, and the results obtained with the BEC model are easily generalized to packet erasure channels. For example, since each packet is simply a binary string, binary linear codes naturally lead to packet coding schemes, in which binary XOR operations are replaced by bit-wise "packet-XOR" operations.

1.1 Reliable communication over the point-to-point erasure channel

Reliable communication over an erasure channel can be accomplished via two different methods:

1. **Feedback:** In the presence of a feedback channel from the destination to the source, a simple mechanism called Automatic Repeat reQuest (ARQ) can be used to reliably communicate data. Essentially, the status of every received symbol (bit/packet), i.e., whether erased or not, is fed back by the destination to the source, which then repeatedly re-transmits every erased symbol until it is received without erasure. This mechanism is employed in the popular transmission control protocol (TCP), used for communication over the Internet.

2. **Coding:** In the absence of feedback, reliable communication can still be accomplished by means of coding or forward error correction (FEC). Specifically, the source encodes a set of intended message symbols using an appropriate *erasure-correcting* code to generate a larger set of (redundant) code symbols, which are transmitted over the erasure channel. This pre-incorporated redundancy suffices to decode the message symbols from the subset of code symbols that are received without erasures at the destination.

   Examples of erasure correcting codes include Reed-Solomon codes, certain classes of low-density parity-check (LDPC) codes, Fountain codes, etc., as will be discussed in the next chapter.

Each of the above techniques has its own associated "cost". The drawback with ARQ is the need for an ideal feedback channel, which requires dedicated bandwidth and additional resources to ensure error-free communication. Likewise, one of the costs associated with FEC is the complexity in encoding and decoding redundant code symbols, each of which is a packet containing several hundred to few thousand bits.

Therefore, it is interesting to see if methods incorporating *both* coding and feedback for reliable communication, commonly referred to as **hybrid ARQ** protocols, can achieve a combination of costs that makes them more attractive than using only coding or only feedback. This is relevant in scenarios where limited resources may exist for feedback, such as (i) limited *bandwidth* (e.g., upstream channels in asymmetric digital subscriber loop (ADSL) systems, control channels in cellular systems, etc.), (ii) limited *power* (such as mobile wireless receivers with limited battery life) or, (iii) a combination of both. Likewise, there could exist situations where coding is feasible, albeit at reduced complexity.

The above topic forms the basis of discussion of the first part of this dissertation. In connection with this, different classes of hybrid ARQ protocols are derived, and the associated tradeoffs between coding complexity and feedback requirements are analyzed.

1.2 Networks of erasure channels

Conventionally, in packet-based networks, the intermediate nodes that handle data traffic act primarily as "routers", i.e., they merely forward received data along the appropriate path. On the other hand, allowing these nodes to *process* incoming data prior to routing it can result in significant improvements in the end-to-end throughput. To achieve these gains, it is necessary to take into account the individual link-level erasure statistics of the network.

In general, we would like to solve the problem of (i) characterizing the best end-to-end throughputs possible over a given network, as a *function* of all its link-level erasure statistics, and (ii) determining the processing strategy at each node that can help achieve these throughputs. The solution to this problem also has important implications concerning network architecture and design, viz., identifying potential traffic "bottlenecks", determining optimum *medium-sharing* strategies (in the case of wireless networks), etc.

This dissertation addresses the above problem for the case of two simple, yet fundamental, configurations of wireless networks, viz., the *multiple-access relay channel* (MARC) and the *multiple-relay channel* (MRC). The MARC consists of two or more sources (s_i) communicating with a common destination (d) with the aid of a single relay (r), as shown in Fig. 1.2(a) for the two-user case. This could represent a situation in which multiple mobile stations communicate with a common base station with the assistance of a common dedicated relay that could potentially aid in improving *coverage*. The MRC consists of a *single* source-destination pair (s, d) communicating with the aid of one or more relays (r_i), as shown in Fig. 1.2(b) for the two-relay case. This situation could arise in a wireless ad-hoc or sensor network, where certain wireless nodes may double up as relays for other pairs of nodes communicating with each other.

1.3 Contribution and organization of this dissertation

The rest of this dissertation is organized as follows.

Chapter 2 provides the relevant background for the topics addressed in this

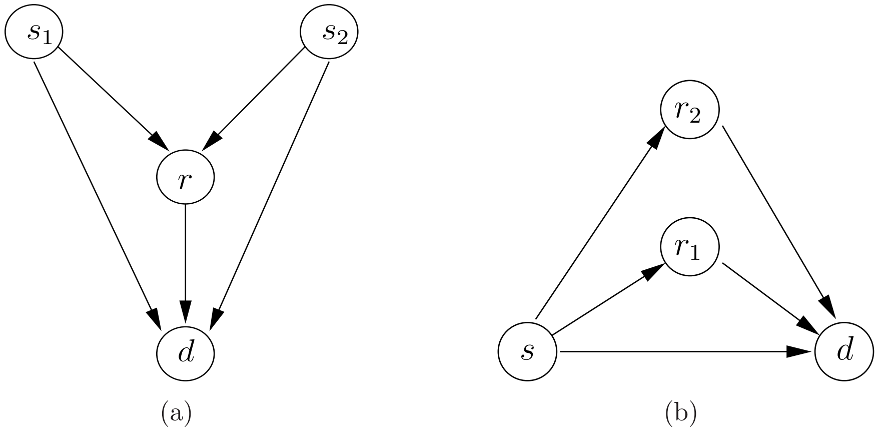


Figure 1.2. (a) The two-source multiple-access relay channel; (b) the two-relay multiple-relay channel.

dissertation and outlines previous work on erasure correcting codes, hybrid ARQ protocols and wireless erasure networks.

In Chapter 3, various hybrid ARQ protocols are presented for the point-to-point erasure channel, assuming the existence of a *noiseless* feedback channel. These protocols are based on a class of erasure-correcting codes called *Tornado* codes. The main goal in designing these protocols is to reduce the complexity incurred with coding by using a *limited* amount of feedback, which is measured as the number of bits communicated over the feedback channel. To this end, the amount of feedback required by different hybrid protocols is optimized using tools from rate-distortion theory.

For each hybrid protocol, the corresponding tradeoff between coding complexity and feedback requirements is characterized, and it is seen that these protocols can offer "better" tradeoffs than a benchmark *time-sharing* protocol in different regimes. Chapter 4 considers some practical aspects of these hybrid protocols, including: the choice of Tornado codes and how it affects the performance of these protocols, as well as the design of “practical” rate-distortion schemes.

Chapter 5 derives *capacity* results for two wireless relay networks – the MARC and the MRC – with links modeled by memoryless BECs, under the assumptions of: (i) orthogonal medium access (achieved via time-sharing) among the different transmitting nodes, and (ii) perfect knowledge at the destination of all erasure events that occur in the network. The approach adopted here consists of formulating cut-set outer bounds on the capacity (region), and demonstrating the achievability of these bounds using simple codes designed for the point-to-point erasure channel. As a consequence, optimum bandwidth-allocation strategies for both these networks are obtained as explicit functions of the link erasure statistics, which serve to highlight the utility of the relay node(s) under different scenarios.

In particular, for the MARC, it is shown that the relay is useful *only* for those sources that have a weaker direct link to the destination than the relay itself - regardless of the quality of the source-to-relay links. On the other hand, for the MRC, the participation of a relay r in the optimum strategy is determined by a more complex, inductive criterion – it depends on the best throughput achievable using only those relays with a *better* link to the destination than r.

Finally, Chapter 6 presents a summary and discussion of the main results of this dissertation.

CHAPTER 2

BACKGROUND

2.1 Erasure correcting codes

The capacity of the binary erasure channel with erasure probability ϵ is given by $1-\epsilon$ bits/channel use [2]. More generally, the capacity of an M-ary erasure channel with the same erasure probability can be shown to be $1-\epsilon$ M-ary symbols/channel use, or $(1-\epsilon) \cdot \log_2 M$ bits/channel use. Consequently, in order to communicate k message symbols reliably, at least $n = k/(1-\epsilon)$ code symbols need to be transmitted over an erasure channel.

The class of Reed-Solomon (RS) codes [3] comes close to achieving this lower bound in the following sense: for an (n, k) RS code that maps k message symbols to n code symbols, any subset of size k of the code symbols suffices to decode all k message symbols. However, RS codes are defined over a finite field containing at least as many elements as the codeword length n. Further, the encoding and decoding complexities of typical implementations of these codes grow at least as $O(k^2)$. This makes these codes computationally unattractive for larger blocklengths.

The class of Tornado codes was proposed in [4] as capacity-approaching (c.a.) code constructions for the BEC. These codes are described using sparse bipartite graphs and are similar in construction to low-density parity-check (LDPC) codes

[5]. Consequently, they have encoding and decoding complexities that are *linear* in the blocklength, i.e., $O(n)$. These codes are constructed with a non-zero gap δ to capacity, where δ can be chosen arbitrarily small provided n is chosen sufficiently large, and their encoding/decoding complexity grows as $O\big(\log(1/\delta)\big)$ for fixed n, as δ is made small.

The code construction presented in [4] led to the discovery of other families of c.a. Tornado and irregular LDPC codes, a systematic study of which is presented in [6]. While Tornado codes have both linear encoding and decoding complexity, these irregular LDPC codes have only linear decoding complexity. Another class of codes that approaches capacity with linear encoding and decoding complexity is the family of systematic irregular-repeat accumulate (SIRA) codes [7], [8]. These codes have a very simple encoder structure, similar to Tornado codes, consisting of the cascade of an irregular single-parity-check (SPC) code and an accumulator.

All these families of codes have the common property that their complexity grows *unbounded* as $O\big(\log(1/\delta)\big)$, when the gap-to-capacity δ goes to zero. In contrast, the class of non-systematic irregular-repeat-accumulate (NSIRA) codes [9] and accumulate-repeat-accumulate (ARA) codes [10] approach capacity with bounded complexity (that is linear in the blocklength), as δ approaches zero.

Finally, the above codes are all designed for a fixed value of the erasure probability ϵ that is assumed to be known *a priori*. In contrast, it is possible to construct erasure-correcting codes that can achieve capacity for *any* value of ϵ, i.e., the value of ϵ is not needed in designing the code. Consequently, these codes are *universal* and can adapt to varying channel statistics. Examples of these codes are LT codes [11] and Raptor codes [12]. For the LT codes developed in [11], the encoding complexity is $O(\log(k))$ per code bit and the overall decoding complexity is $O(k\log(k))$; further, their gap-to-capacity is $O(\frac{\log^2 k}{\sqrt{k}})$. On the other hand, the behavior of Raptor codes is similar to Tornado codes – they have encoding complexity $O(\log(1/\delta))$ per code bit and overall decoding complexity $O(k\log(1/\delta))$ for gap to capacity δ.

2.2 Feedback, coding and hybrid ARQ protocols for packet erasure networks

For packet-based networks such as the Internet, feedback has traditionally been used to achieve reliable communication. Various types of ARQ protocols are used at both the link layer and the transport layer (such as in TCP) of the network protocol stack, as discussed in [13].

In [14], the use of coding or FEC as an alternative to feedback in networks was proposed for two different cases: *multicast* over networks, and end-to-end transmission over links with a high *bandwidth-delay* product. In the context of multicast, the same redundant packets generated using FEC can compensate for potentially different packets erased at different receivers; thus, FEC avoids the problem of "feedback implosion" that occurs when each of several receivers attempts to provide feedback about its own erasure status. In the case of links with a high bandwidth-delay product, the round trip time can exceed the total transmission time; using feedback in this context can be wasteful as it leads to long wait times at the transmitter, and FEC can reduce this inefficiency. The use of Reed-Solomon codes for reliable multicast was explored in [15].

The idea of using FEC for multicast in networks led to the notion of "digital fountain" codes [16], [17]. With Fountain codes, a transmitter continuously broadcasts distinct code packets encoded from a single message. Receivers listening to the broadcast experience different erasure processes with potentially different statistics, unknown to the transmitter; nonetheless, each receiver is able to decode the message on receiving a sufficient number of unerased code packets, that is roughly equal to the total number of message packets. The notion of Fountain codes led to the invention of LT and Raptor codes, outlined earlier.

Besides the above FEC-only approaches, there has also been work on developing hybrid ARQ protocols for packet-erasure networks. Some of these schemes are outlined below.

In [18], simple hybrid ARQ schemes using very short blocklength RS codes are presented for the case of multicast transmission; these help reduce the number/content of feedback requests from each receiver and improve the efficiency of re-transmissions.

In [19], a hybrid *rateless* scheme based on binary codes is developed that uses a very small amount of feedback to acknowledge only the *number* of decoded message packets. The resulting scheme has a "real-time decoding" property (i.e., decoding occurs as packets are received) and low memory requirements at the destination, at the cost of higher transmission overhead, roughly by a factor of two.

In [20], hybrid ARQ schemes using binary sparse-graph codes are proposed for multicast transmission with quality-of-service (QoS) constraints – the code-structure is adapted on-the-fly based on limited feedback from each receiver, regarding the number of lost packets.

In [21], the use of feedback to improve the *reliability* of erasure correcting codes is explored. In particular, this work introduces an LDPC coding scheme that uses a limited amount of noiseless feedback in order to improve the decoder's performance in the presence of *stopping sets* [22]. Tradeoffs are established between the resulting reliability (i.e., frame error rate) and the amount of feedback utilized.

In [23], feedback is used in the context of network coding for multicast, with the goal of reducing the queue-size (of packets waiting to be delivered) at the sender and decoding delay at the receivers. The encoding process consists of generating random linear combinations of message packets in a *queue*, treating them as symbols over a sufficiently large finite field. Conventional ARQ is used by all receivers to acknowledge received *code* packets. Based on this feedback, the sender identifies a particular common message packet as being “seen” by all receivers, and this packet is removed from the queue for encoding future code packets. This also has the effect of reducing coding complexity.

The hybrid protocols introduced in this dissertation differ from the approaches adopted earlier in many fundamental aspects. Our primary goal is to design “flexible” hybrid protocols for *unicast* transmission (i.e. for a single receiver) that allow for *tradeoffs* between the amount of coding and feedback used. Within this framework, our focus is explicitly on minimizing coding complexity for a given amount of feedback and vice versa. In contrast, much of the earlier work has dealt with *fixed* coding/feedback strategies, primarily for multicast transmission, where the focus has been different from establishing complexity-feedback tradeoffs. As mentioned earlier, our protocols use *binary* erasure correcting codes (i.e., Tornado codes) with XOR as the only operation, unlike some previous approaches that make use of codes defined over larger finite fields (e.g., RS codes) requiring far more complex operations. Further, all protocols considered in this dissertation are capacity-approaching, in the sense that there is no waste resulting from excessive transmission of redundant packets, i.e., beyond what is needed by the receiver to decode the message.

On a final note, it is worthwhile highlighting the difference between the hybrid

ARQ protocols considered here for erasure-recovery, and conventional hybrid ARQ protocols used at the link-level for *error-correction* [24]. While the former are used for losslessly communicating a *collection* of packets, the latter are used to communicate *individual* packets reliably. Consequently, the nature of feedback used in these two approaches is quite different; for example, while it is practically impossible to feedback the status of every “noisy” bit within a received packet, feeding back locations of erased packets is quite feasible.

2.3 Wireless relay networks with erasure links – MARC and MRC

The MARC and MRC can be viewed as extensions of the three-terminal relay channel (with a single source and single relay) studied by van der Meulen [25] and Cover and El Gamal [26]. The MARC was first introduced in [27], and bounds for the capacity region of the *discrete memoryless* MARC were derived in [27], [28], [29]. Similarly, achievable rates for the general discrete memoryless multiple relay channel have been derived, cf. [29], [30].

The focus of the cited work is primarily on establishing performance bounds for the most general descriptions of the MARC and the MRC, as is typical in information theory. While such an approach is quite powerful, the underlying problems are invariably extremely hard to tackle, and so as of yet there is no exact formulation for the capacity (region).

In contrast, this dissertation seeks a formulation for the capacity (region) by restricting attention to networks composed of erasure channels, which is a special case of the discrete memoryless channel (DMC) model. This allows us to considerably simplify the problem of solving for the capacity (region) and gain insights that are not easily accessible via more complex models. Also, as we have seen, the model of an erasure channel is quite relevant in packetized communication. The shortcoming of the erasure-channel assumption, of course, is that the results do not generalize to other instances of the DMC that arise in practice.

The study of wireless networks using the simple erasure-link model has been undertaken with great success by several researchers, cf. [31], [32], [33], [34], [35]. In [31], the capacity region of multi-source multicast over arbitrary wireless networks with erasures is derived, assuming: (*i*) orthogonal links that carry one packet per channel use, and (*ii*) perfect knowledge of erasure patterns on all links at the destination; moreover, it is shown in [31] that the capacity region coincides with that described by cut-set bounds. Further, [32] demonstrates that the capacity of single-source unicast and multicast transmission over such networks is achievable using *random network coding* at the intermediate nodes. In [33], [34] and [35], the authors consider particular relay erasure networks where the destination nodes do *not* have access to the above side information; converse bounds and achievability results using maximum distance separable (MDS) codes are presented for this scenario.

The models for the MARC and the MRC assumed in this dissertation are quite similar to that in [31] – in particular, we also assume the presence of perfect side information of all erasure patterns in the network, at the destination. In fact, the capacity results of [31] can be specialized to the MARC and the MRC. However, in [31] the available bandwidth is apportioned *equally* among all transmitting nodes; in contrast, we do *not* pre-determine the allocation of the wireless medium. As a consequence, the region of all achievable rates we obtain is strictly a superset of the region that follows from the results of [31]. Further, we demonstrate that points in the capacity region may be achieved using low-complexity capacity-achieving (c.a.) codes designed for the BEC (such as Tornado or LDPC codes), in place of the random coding arguments employed in [31].

CHAPTER 3

HYBRID ARQ PROTOCOLS FOR THE ERASURE CHANNEL

3.1 Introduction

As noted earlier, coding and feedback constitute two fundamentally different means of communicating reliably over erasure channels. One of the main costs of coding is the associated *computational complexity* in generating and decoding code packets. Likewise, the use of feedback necessitates dedicated *bandwidth* on a feedback link. Hybrid ARQ protocols combine coding and feedback to reliably communicate data. In this chapter, we explore a new class of hybrid ARQ protocols designed to permit *trade-offs* between computational complexity of encoding/decoding and the amount of feedback utilized.

We consider the model of a memoryless binary erasure channel (BEC) between a source s and destination d. The erasure probability of the BEC is denoted ϵ. The destination is able to convey information to the source by means of a *noiseless* feedback link. The source has k message bits that need to be conveyed reliably to the destination. We restrict ourselves to capacity-achieving protocols, i.e., protocols that achieve reliable communication with $n = k/(1-\epsilon)$ bits transmitted from s, on average.

We adopt the following metrics to evaluate different protocols:

- **Coding complexity C:** This is the total number of XORs needed to generate all the bits that are transmitted by s, i.e., the encoding complexity. For the codes we consider (Tornado codes), this is the same as the decoding complexity.

- **Feedback channel usage F:** This is given by the total number of bits communicated by d to s over the feedback link.

We begin with an overview of feedback-only and coding-only approaches.

3.2 Feedback-only (ARQ) protocol

The simplest form of a feedback-only protocol is as follows: s re-transmits every message bit until it is received unerased and d feeds back the status of every received bit. For this scheme, it follows that both s and d need to transmit $k/(1-\epsilon)$ bits on average. However, it is possible to reduce the amount of feedback, as shown below.

For n transmissions over the BEC, the erasure locations can be thought of as the output of a Bernoulli $\{0, 1\}$ source, where 1 denotes an erasure and 0 a non-erasure. Since the probability of a 1 is ϵ, it follows that the n-bit Bernoulli sequence describing the erasure locations can be efficiently represented with $n \cdot h(\epsilon)$ bits, using an appropriate source code. Here, $h(x) = -x \log_2(x) - (1-x)\log_2(1-x)$ is the binary entropy function.

The above observation leads to the **compressed-feedback** protocol. In this protocol, feedback from d and re-transmission from s occur in several *rounds*, as follows:

> **Round** 0**:** s transmits the k uncoded message bits and d feeds back the erasure locations, after appropriate compression using a source code.

Round i **($1 \leq i \leq N-1$):** s re-transmits the bits erased in the $(i-1)^{st}$ round; d feeds back the erasure locations among these bits using compression.

Round N**:** The erased bits in round $N-1$ are communicated to d using naive (uncompressed) feedback, i.e., s re-transmits every erased bit until it is received correctly and d immediately feeds back the status of every transmission.

In round i, $0 \leq i \leq N-1$, the average number of bits transmitted by s is $\epsilon^i k$; consequently, the average number of bits transmitted by d in feedback is $\epsilon^i k \cdot h(\epsilon)$. In the N^{th} round, both s and d transmit $\epsilon^N k/(1-\epsilon)$ bits each, on average. While the total number of transmissions from s is unchanged from the uncompressed case, the total amount of feedback from d is as follows:

$$\sum_{i=0}^{N-1} \epsilon^i k \cdot h(\epsilon) + \frac{\epsilon^N k}{1-\epsilon} \;=\; k \cdot \Big(\frac{h(\epsilon)}{1-\epsilon} + \frac{\epsilon^N\big(1-h(\epsilon)\big)}{1-\epsilon}\Big). \tag{3.1}$$

As $N \to \infty$, the contribution of the second term tends to zero. Since we are primarily interested in order behavior, this term will be neglected in the future.

Henceforth, we shall assume that the feedback-only protocol uses compressed-feedback. Summarizing, the coding complexity and feedback usage for this protocol are as follows:

$$\mathsf{C}_{ARQ} \;=\; 0 \tag{3.2}$$

$$\mathsf{F}_{ARQ} \;=\; \frac{k \cdot h(\epsilon)}{1-\epsilon} \tag{3.3}$$

Note that we have disregarded the complexity involved in compressing the feedback, relative to coding (FEC) for the erasure channel; this is primarily because compression involves bit-level operations, whereas in practice, erasure-correction coding is done at the level of packets. Consequently, the complexity of coding may often be much larger than the complexity of compression. The same reasoning is applied in evaluating hybrid ARQ protocols which may employ some form of compression/processing for the feedback part.

3.3 Coding-only approach – Tornado codes

As noted in Chapter 2, there are several classes of practical capacity-approaching erasure correcting codes, with encoding and decoding complexities that grow linearly with the message size k. We restrict our attention to the class of Tornado codes [4], as these codes are easily adapted for the hybrid protocols developed later.

Thus, the coding-only approach is as follows: s encodes the k message bits using an (n, k) Tornado code and transmits the resulting n code bits over the BEC; d can decode the message from the unerased portion of the received codeword with high probability, without having to use the feedback channel. We now briefly describe the structure and encoding/decoding process for Tornado codes.

The n code bits of an (n, k) Tornado code are partitioned into *layers*. These layers are encoded in a recursive manner as follows:

1. Layer 0 consists of the k message bits themselves; hence, these codes are *systematic*.

2. Layer i $(1 \leq i \leq N-1)$ consists of $\beta^i k$ parity bits, where $\beta = \epsilon/(1-\delta)$ and $\delta > 0$ is a small *overhead* parameter. Layer i is obtained by forming different linear combinations (XORs) of bits in layer $i-1$.

3. The final layer N consists of $\beta^N k/(1-\beta)$ parities derived from layer $N-1$ parities.

The blocklength of this code is $n = \sum_{i=0}^{N-1} \beta^i k + \beta^N k/(1-\beta) = k/(1-\beta)$, and hence its rate is $R = 1-\beta$. For the case when δ is small, the blocklength and rate are approximately $k/(1-\epsilon)$ and $1-\epsilon$, respectively.

The layers 1 through $N-1$ are constructed such that each layer, in its *entirety*, can be used to decode a random fraction ϵ of erased bits in the preceding layer, with high probability. The final layer N comprises the parity bits of a $\left(\frac{\beta^{N-1}k}{1-\beta}, \beta^{N-1}k\right)$ *systematic* erasure correcting code, whose message bits are the parities of layer $N-1$. Thus, the final layer can help recover a random fraction ϵ of erased bits in layer $N-1$, even when a fraction ϵ within itself is erased. For our discussion, we shall assume that the generator matrix of this final erasure-correcting code is of the form $[\mathbf{I}\ \mathbf{P}]$, where $\mathbf{I}$ is the identity matrix of order $\beta^{N-1}k$, and $\mathbf{P}$ is a $\beta^{N-1}k \times \frac{\beta^N k}{1-\beta}$ matrix whose every element is chosen to be either a 0 or a 1 with equal probability. It can be shown that this choice of the final code works in practice because the sub-matrix formed by randomly picking a fraction ϵ of the rows and a fraction $1-\epsilon$ of the columns of $\mathbf{P}$ has full-rank with high probability (Proposition 2 in [12]).

The connections between successive layers from 0 through $N-1$ can be represented by a cascade of bipartite graphs as shown in Fig. 3.1, where each node represents a code bit. Each right node of a bipartite graph is generated by XORing its *left neighbors* – i.e., nodes that it is connected to on the left. The average number of neighbors of the left and right nodes within each bipartite graph constitute the average left and right *degrees*, denoted by a_l and a_r, respectively. Since the total number of edges emanating from the left nodes must equal the number

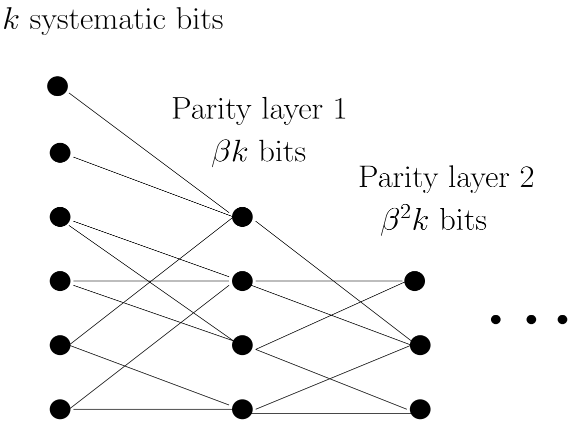


Figure 3.1. Tornado codes - encoding graph

converging on the right nodes, we have the following relation:

$$a_l = \beta a_r \tag{3.4}$$

The bipartite graphs connecting layers 0 through $N-1$ are characterized by their **node-degree distributions**, viz., the fractions of left and right nodes of different degrees (i.e., number of neighbors). Alternatively, they may equivalently be characterized using **edge-degree distributions** which are more commonly used in the literature [4], viz., the left edge-degree distribution $\{\lambda_i\}_{i=1}^{\infty}$ and the right edge-degree distribution $\{\rho_j\}_{j=1}^{\infty}$. Here λ_i represents the fraction of edges that originate from *left* nodes of degree i. Likewise, ρ_j is the fraction of edges terminating on *right* nodes of degree j. We shall use the short-hand notation $\lambda := \{\lambda_j\}_{j=1}^{\infty}$ and $\rho := \{\rho_j\}_{j=1}^{\infty}$ henceforth. It follows that the average degrees a_l and a_r are functions of the degree distribution (d.d.) pair (λ, ρ). For Tornado codes, all bipartite graphs in the cascade (excluding the final layer N) have the same d.d. pair, and hence the same a_l and a_r.

For layers 1 through $N-1$, the average number of XORs needed to generate a parity bit is $a_r - 1$. For layer N, the average number is $\beta^{N-1}k/2$. Summing over all layers, the total encoding complexity is given by

$$\begin{aligned} & k \cdot \frac{\beta(1-\beta^{N-1})}{1-\beta} \cdot (a_r - 1) \; + \; \frac{\beta^N k}{1-\beta} \cdot \frac{\beta^{N-1}k}{2} \\ = \; & \frac{\beta k}{1-\beta}\Big((1-\beta^{N-1}) \cdot (a_r - 1) \; + \; \frac{\beta^{2(N-1)}k}{2}\Big). \end{aligned} \tag{3.5}$$

Tornado codes are decoded in the reverse direction of encoding. Starting with the unerased parities of layer N, the erased parities of layer $N-1$ are first recovered by solving a system of linear equations. This is done using Gaussian elimination, which needs a total of $O\big((\beta^{N-1}k)^3\big)$ XOR operations. Next, each layer i $(< N-1)$ is recursively decoded from the reconstructed higher layer $i+1$, via the belief propagation (BP) algorithm. The number of XORs needed for this is again $a_r - 1$ per right node (in the decoding graph). Thus, the total decoding complexity is given by:

$$k \cdot \frac{\beta(1-\beta^{N-1})}{1-\beta} \cdot (a_r - 1) \; + \; O\big((\beta^{N-1}k)^3\big). \tag{3.6}$$

Usually, the number of layers N is chosen to be large so that $\beta^N = O(k^{-3/4})$. Consequently, for large k, we can neglect the contribution of the last layer (in comparison with the remaining layers) to both the encoding and decoding complexity, which are the same otherwise.

Thus, for the coding-only protocol using Tornado codes, the coding complexity and feedback usage are:

$$\begin{aligned} \mathsf{C}_{FEC} &= \frac{k\beta}{1-\beta} \cdot (a_r - 1) \\ &= \frac{k}{1-\beta} \cdot (a_l - \beta) \qquad (3.7) \\ \mathsf{F}_{FEC} &= 0 \qquad (3.8) \end{aligned}$$

The d.d. pair (λ, ρ) depends on the erasure probability ϵ and overhead δ but is independent of k. Consequently, the encoding and decoding complexity scale *linearly* with message size k. However, for any d.d. pair, the values of a_r and a_l are proportional to $\log(1/\delta)$ [6]. Consequently, the coding complexity of Tornado codes is unbounded as $\delta \to 0$.

In contrast, certain c.a. codes, such as ARA codes [10], possess *bounded* complexity per message bit even as $\delta \to 0$. Therefore, to ensure similar complexity, Tornado codes need to operate at a larger overhead δ in practice. Consequently, we only consider applications where this is not much of a penalty.

3.4 A hybrid protocol based on time-sharing

The simplest hybrid protocol consists of *time-sharing* between the coding-only and feedback-only protocols. Specifically, for some $\theta \in [0, 1]$, a subset of θk message bits are communicated using only coding, and the remaining bits using only feedback. We use this protocol as a benchmark for comparing other hybrid protocols.

The performance of any hybrid protocol may be characterized by the following parameters:

Fractional feedback f – defined as the ratio of its feedback usage to that of the feedback-only protocol, and

Fractional complexity $\mathtt{c}$ – defined as the ratio of its coding complexity to that of the coding-only protocol.

For the time-sharing protocol, these parameters are given by:

$$\begin{aligned} \mathtt{c}_{\mathtt{TS}} &= \theta && (3.9) \\ \mathtt{f}_{\mathtt{TS}} &= 1 - \theta && (3.10) \end{aligned}$$

The plot of $\mathtt{c}_{\mathtt{TS}}$ versus $\mathtt{f}_{\mathtt{TS}}$ is a straight line as shown in Fig. 3.2. Our goal is to design hybrid schemes with $\mathtt{c}$ versus $\mathtt{f}$ curves that lie *below* the time-sharing curve, i.e., for a fixed amount of feedback (coding complexity), we want the amount of coding complexity (feedback) of the hybrid schemes to be strictly smaller than the corresponding complexity (feedback) of the time-sharing scheme.

3.5 Hybrid-A: “Quantized” feedback

The first hybrid protocol we consider is essentially a simple adaptation of the coding-only protocol. The basic idea is to reduce the encoding and decoding complexity by feeding back a “quantized” version of the erasure locations.

In the hybrid-A protocol, the transmission of code bits from s and feedback from d occur in $N-1$ rounds as follows:

1. In round i ($0 \le i \le N-2$), s transmits the $\beta^i k$ code bits of layer i of the Tornado code used in the coding-only protocol (layer 0: systematic bits). The $\beta^i k$ channel outputs at the destination (comprising both erased and unerased bits) are partitioned into $\beta^i k/\ell$ contiguous blocks of size ℓ bits each. The locations of blocks that contain *at least* one erased bit are then

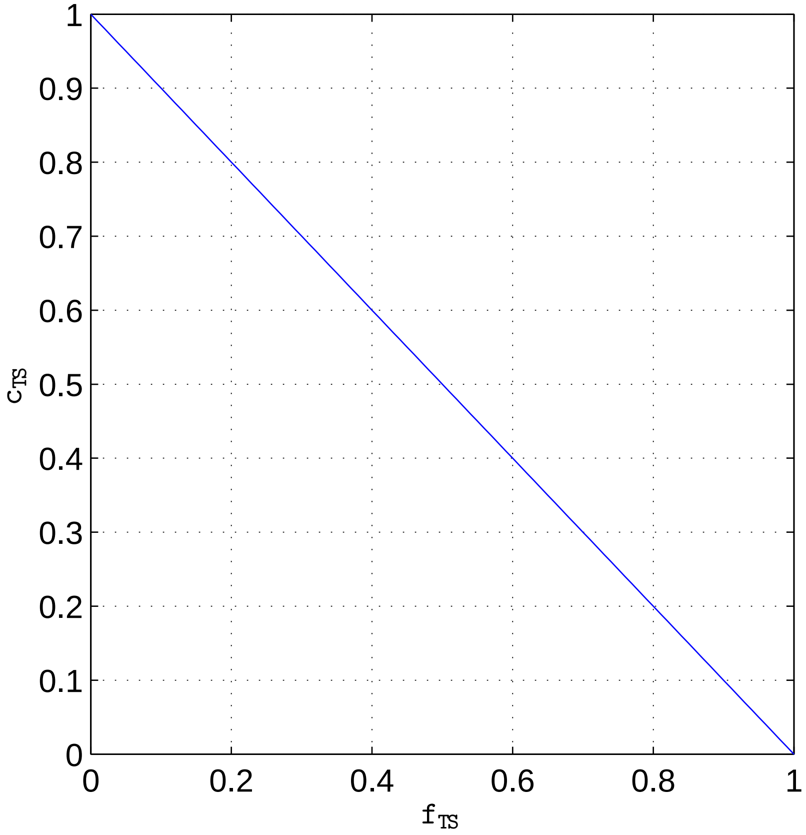


Figure 3.2. Performance of the time-sharing protocol

fed back to s via compressed feedback. We shall refer to such blocks as "erased" blocks.

2. The parity bits to be transmitted in round i ($1 \leq i \leq N-1$) are encoded *only after* the feedback from round $i-1$ has been received. In particular, the layer $i-1$ parity bits belonging to unerased blocks in round $i-1$ are simply *omitted* while encoding the layer i parities, as illustrated in Fig. 3.3.

3. In round $N-1$, the $\beta^{N-1}k$ layer $N-1$ parities are encoded as above and

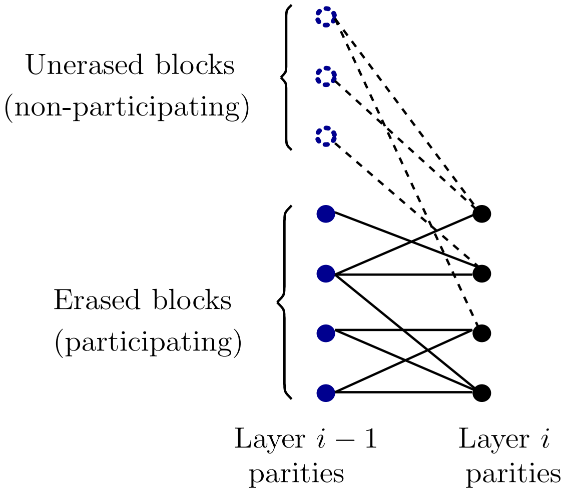


Figure 3.3. Hybrid-A scheme: Encoding process

communicated to d using the feedback-only protocol.

By excluding layer $i-1$ bits known to have been unerased at d, s can save on computations (XORs) needed to generate the layer i parities. Likewise, while decoding erasures in layer $i-1$ via the BP algorithm, d can avoid having to remove the contribution of these unerased bits, which is unavoidable with the coding-only approach. We shall refer to the subset of layer $i-1$ parities (belonging to erased blocks) that participate in the encoding process during round i as the **encoding set** for that round.

The probability p of a block erasure is given by:

$$p = 1 - (1-\epsilon)^{\ell}. \tag{3.11}$$

Thus, the total number of bits that must be fed back (using compression) in round i $(1 \leq i \leq N-2)$ is given by $h(p) \cdot \beta^i k/\ell$, where $h(\cdot)$ is the binary entropy function. Also, the number of feedback bits required for the last round $N-1$ is given by

$\beta^{N-1} k \cdot h(\epsilon)/(1-\epsilon)$. Hence, the total amount of feedback is given by:

$$\begin{aligned} & \sum_{i=0}^{N-2} \beta^i k \cdot \frac{h(p)}{\ell} + \beta^{N-1} k \cdot \frac{h(\epsilon)}{1-\epsilon} \\ = \; & k \cdot \Big(\frac{1-\beta^{N-1}}{1-\beta} \cdot \frac{h(p)}{\ell} \; + \; \beta^{N-1} \cdot \frac{h(\epsilon)}{1-\epsilon} \Big). \end{aligned} \tag{3.12}$$

For large N, we neglect β^{N-1} and the contribution of the second term, and we have for the hybrid-A protocol:

$$\mathsf{F}_A \;=\; \frac{k}{1-\beta} \cdot \frac{h(p)}{\ell}. \tag{3.13}$$

The bits belonging to the encoding set for round i (i.e., the erased blocks of round $i-1$) essentially constitute a random subset of layer $i-1$ parities, chosen regardless of their degree profile; therefore, these bits have the same left-node degree distribution and average left degree a_l as the original Tornado code. Then, it is easily verified that the new average right degree of the graph in Fig. 3.3, after removing the unerased bits, is given by $a'_r = pa_l/\beta = pa_r$. Consequently, the coding complexity is given by:

$$\sum_{i=1}^{N-1} \beta^i k \cdot (a'_r - 1) \;=\; k \cdot \frac{\beta(1-\beta^{N-1})}{1-\beta} \cdot (pa_r - 1). \tag{3.14}$$

Again, ignoring β^{N-1} for large N yields the coding complexity:

$$\mathsf{C}_A \;=\; \frac{\beta k}{1-\beta} \cdot (pa_r - 1). \tag{3.15}$$

Under the assumption that the overhead of the Tornado code satisfies $\delta \ll 1$, which yields $a_r \gg 1$ and $\beta \approx \epsilon$, the fractional complexity and feedback are given by:

$$
\begin{aligned}
\mathtt{c}_{\mathtt{A}} &= \frac{\mathsf{C}_A}{\mathsf{C}_{FEC}} = p, && (3.16)\\
\mathtt{f}_{\mathtt{A}} &= \frac{\mathsf{F}_A}{\mathsf{F}_{ARQ}} = \frac{h(p)}{\ell \cdot h(\epsilon)}. && (3.17)
\end{aligned}
$$

Note that $\mathtt{c}_{\mathtt{A}}$ and $\mathtt{f}_{\mathtt{A}}$ can be changed by varying ℓ between 1 and k. The resulting $\mathtt{c}_{\mathtt{A}}$ versus $\mathtt{f}_{\mathtt{A}}$ curves for different values of ϵ are shown in Fig. 3.4. (For these curves, the value of ℓ was varied between 1 and 10^4.)

It is seen that for small values of ϵ (e.g. 0.05), the hybrid-A scheme performs significantly better than the time-sharing scheme, in the sense that the fractional complexity for a given value of fractional feedback is substantially lower (and vice-versa) for the hybrid-A scheme. This behavior is observed over a large range of the fractional feedback, with the exception of $\mathtt{f}_{\mathtt{A}}$ close to 1. For larger ϵ (≥ 0.2), the performance of hybrid-A degrades and is worse than the time-sharing scheme over a wider range of values for $\mathtt{f}$.

For all ϵ, it is seen that $\mathtt{c}_{\mathtt{A}} \neq 0$ when $\mathtt{f}_{\mathtt{A}} = 1$, unlike the time-sharing scheme. In other words, even when all the erasure (and non-erasure) locations have been fed back to s, there is still some residual coding complexity in the hybrid-A scheme; this is because, while the unerased bits in any round are omitted from the encoding process, the contribution of erased bits to computing the parities in the next round remains *unchanged* in the presence of feedback.

We analyze the performance of this protocol for the case when $\epsilon \ll 1$. Let $\ell = c/\epsilon$, for some constant $c > 0$. Then, we have $p = 1 - (1-\epsilon)^{c/\epsilon} \approx 1 - e^{-c}$. Thus, the fractional complexity $\mathtt{c}_{\mathtt{A}}$ is roughly constant (independent of ϵ). On the

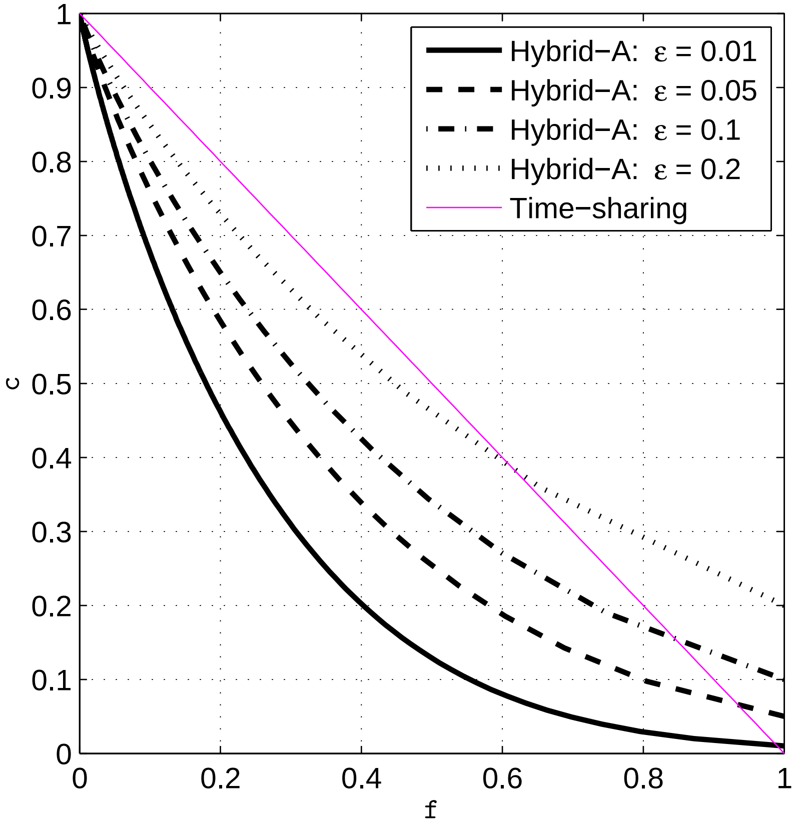


Figure 3.4. Performance of the hybrid-A protocol

other hand,

$$\begin{aligned} \frac{h(p)}{\ell \cdot h(\epsilon)} &= \frac{h(p)}{c} \cdot \frac{\epsilon}{h(\epsilon)} \\ &= \frac{h(p)}{c} \cdot \frac{1}{\log_2 \frac{1}{\epsilon} - (1-\epsilon)\frac{\log_2(1-\epsilon)}{\epsilon}} \\ &\approx \frac{h(1-e^{-c})}{c} \cdot \frac{1}{\log_2 \frac{1}{\epsilon}}, \qquad (3.18) \end{aligned}$$

which follows from the fact that $p \to 1 - e^{-c}$ and $\frac{\log_2(1-\epsilon)}{\epsilon} \to 0$ as $\epsilon \to 0$. Therefore, the fractional feedback $\mathtt{f_A}$ decays as $\frac{1}{\log_2(1/\epsilon)}$. This explains the improved performance of the hybrid-A scheme in the small-ϵ regime.

3.6 Hybrid-B: Optimizing hybrid-A's feedback

In this section, we describe a variation of the hybrid-A scheme that requires less feedback.

The basic idea of the hybrid-A protocol is to feed back partial information about the channel-outcomes (erasures and non-erasures) in each round so that a subset of the unerased bits in that round can be omitted from encoding the next round. We now pose the question as to whether there exist feedback strategies that achieve the *same* reduction in the size of the encoding set in each round as the hybrid-A scheme, but require *fewer* feedback bits than the quantized-feedback strategy. This is formally stated as follows.

Let the channel outcomes of m transmissions (for some m) over a BEC be represented by the m-tuple $\mathbf{x} \in \{0,1\}^m$, where 0 denotes a non-erasure and 1 an erasure. Suppose these channel outcomes are encoded with some *distortion* as the m-tuple $\mathbf{y} \in \{0, \mathcal{U}\}^m$, where again 0 represents a non-erasure while $\mathcal{U}$ represents an *unacknowledged* channel outcome (that could have been either an erasure or a non-erasure). Further, we encode $\mathbf{y}$ according to the following rules:

1. An element of $\mathbf{y}$ can be 0 only if the corresponding element in $\mathbf{x}$ is also 0. In other words, an erasure in $\mathbf{x}$ **cannot** be reported as a non-erasure in $\mathbf{y}$.

2. The total number of non-erasures (i.e., 0's) in $\mathbf{x}$ that are unacknowledged (i.e., reported as $\mathcal{U}$) in $\mathbf{y}$ cannot exceed mD. Here, $0 \leq D \leq 1-\epsilon$ is a *distortion parameter* that is fixed beforehand.

Suppose the vector $\mathbf{y}$ is fed back to s instead of $\mathbf{x}$; then s can identify a *subset* of the unerased transmissions, as in the hybrid-A protocol. Thus, if $\mathbf{y}$ can be compressed more efficiently than the quantization scheme used in hybrid-A, fewer feedback bits are needed to convey this information. We now formalize this problem of optimum compression as a problem in rate-distortion theory.

Let $\mathbf{y} = f(\mathbf{x})$ denote an encoding function from the space of true erasure outcomes to the space of distorted reconstructions, i.e., $f:\ \{0,1\}^m \rightarrow \{0,\mathcal{U}\}^m$.

Let the following function define the *per-letter* distortion measure:

$$\mathcal{D}(x,y) = \begin{cases} 0, & \text{if } x=0, y=0, \\ 1, & \text{if } x=0, y=\mathcal{U}, \\ \infty, & \text{if } x=1, y=0, \\ 0, & \text{if } x=1, y=\mathcal{U}. \end{cases} \tag{3.19}$$

Then, the sum-distortion measure between the vectors $\mathbf{x}$ and $\mathbf{y}$ is given by

$$\hat{d}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} \mathcal{D}(x_i, y_i), \tag{3.20}$$

and the average per-letter distortion $\bar{d}_f$ associated with an encoding function $f(\cdot)$ is given by

$$\bar{d}_f = \frac{1}{m} \sum_{\mathbf{x}\ \in\{0,1\}^m} p(\mathbf{x}) \cdot \hat{d}\big(\mathbf{x}, f(\mathbf{x})\big). \tag{3.21}$$

Note here that, for finite $\bar{d}_f$, the pair $(x_i, (f(\mathbf{x}))_i)$ can never be $(1,0)$; consequently, for each $\mathbf{x}$, $\hat{d}(\mathbf{x}, f(\mathbf{x}))$ is exactly the *number* of non-erasures in $\mathbf{x}$ that are unacknowledged in $f(\mathbf{x})$. Therefore, imposing the condition $\bar{d}_f \leq D$ (for $0 \leq D \leq 1-\epsilon$) on the encoding function $f(\cdot)$ simultaneously enforces both encoding rules outlined earlier.

Finally, for Bernoulli random variables X and Y, with $Pr(X = 1) = \epsilon$, define the following positive-valued function $\tilde{R}(\cdot)$:

$$\tilde{R}(D) = \min_{p_{Y|X}(y|x):\ E(\mathcal{D}(X,Y)) \leq D} I(X;Y), \tag{3.22}$$

where the expectation $E(\mathcal{D}(X,Y))$ is over the joint distribution of X and Y.

We now state the following result due to Gallager [36, Theorems 9.2.1 and 9.3.2] for arbitrary distortion measures:

Theorem 1 ([36])**.** *For the rate-distortion code $f(\cdot)$, if $\bar{d}_f \leq D$ then the entropy of the set of reconstruction sequences must satisfy:*

$$H(f(\mathbf{x})) \ \geq \ m \cdot \tilde{R}(D). \tag{3.23}$$

Further, given any $D \geq 0$ and $\kappa > 0$, there exists, for sufficiently large m, a rate-distortion code $\tilde{f}(\cdot)$ for which $\bar{d}_{\tilde{f}} \leq D + \kappa$ and

$$H(\tilde{f}(\mathbf{x})) \ \leq \ m \cdot (\tilde{R}(D) + \kappa). \tag{3.24}$$

Consequently, for a specified average per-letter distortion D between $\mathbf{x}$ and $\mathbf{y} = f(\mathbf{x})$, a minimum of $m \cdot \tilde{R}(D)$ bits on average are required to communicate $\mathbf{y}$ to s. Here, mD is the average number of 0's (non-erasures) in $\mathbf{x}$ that are reported as $\mathcal{U}$'s (unacknowledged outcomes) in $\mathbf{y}$. Further, it can be shown that

$$\tilde{R}(D) \ = \ (D+\epsilon)\log_2 \frac{1}{D+\epsilon} \ - \ D\log_2 \frac{1}{D} \ + \ (1-\epsilon)\log_2 \frac{1}{1-\epsilon}. \tag{3.25}$$

(See Appendix A for a derivation.)

In sum, by using a rate-distortion code that achieves the rate-distortion bound

$\tilde{R}(D)$, the feedback can potentially be more efficient than in the hybrid-A scheme.

For a fixed distortion parameter $0 \leq D \leq 1 - \epsilon$, the hybrid-B scheme is described as follows:

1. In round i, $0 \leq i \leq N-2$, s transmits the $n_i = \beta^i k$ code bits of layer i of the Tornado code used in the coding-only protocol. The n_i channel outcomes in round i consist of $n_i(1-\epsilon)$ non-erasures: d feeds back the channel outcomes to s with distortion D in the non-erasure locations (i.e., only a subset of $n_i(1-\epsilon-D)$ non-erasures are acknowledged without ambiguity), using a rate-distortion code that achieves the bound $\tilde{R}(D)$.

2. The feedback received in round i is used to reduce the encoding complexity in round $i+1$ exactly as in the hybrid-A scheme, viz., the subset of layer i parities that are *known* by s to be unerased are omitted while encoding layer $i+1$ parities. Likewise, d neglects the contribution of these bits while decoding.

3. In round $N-1$, the $\beta^{N-1}k$ parities of layer $N-1$ are computed as in the hybrid-A scheme and communicated to d using the feedback-only approach.

Thus, only $n_i(D+\epsilon)$ parities of layer i participate in encoding during round $i+1$. As in the hybrid-A scheme, the average left degree among these participating bits remains unchanged from the original Tornado code. Similar reasoning shows that the average right degree in this case is given by $(D+\epsilon)a_r$, where a_r is the average right degree of the original code. Consequently, the coding complexity is given by:

$$\begin{aligned}\mathsf{C}_B &= \frac{\beta k}{1-\beta}\cdot\Big((D+\epsilon)a_r - 1\Big),\\ &= \frac{k}{1-\beta}\cdot\Big((D+\epsilon)a_l - \beta\Big) \qquad (3.26)\end{aligned}$$

where we have ignored a β^{N-1} term, following the standard large N approximation.

The amount of feedback in round i $(1 \leq i \leq N-2)$ is given by $n_i\tilde{R}(D) = \beta^i k \cdot \tilde{R}(D)$. Thus, the total amount of feedback is:

$$\begin{aligned}&\sum_{i=1}^{N-2} n_i\tilde{R}(D) + \beta^{N-1}k\cdot\frac{h(\epsilon)}{1-\epsilon}\\ =\ & k\cdot\Big(\frac{1-\beta^{N-1}}{1-\beta}\cdot\tilde{R}(D)\ +\ \beta^{N-1}\cdot\frac{h(\epsilon)}{1-\epsilon}\Big). \qquad (3.27)\end{aligned}$$

Again, for large N, ignoring β^{N-1} yields:

$$\mathsf{F}_B = \frac{k}{1-\beta}\cdot\tilde{R}(D). \qquad (3.28)$$

Finally, assuming $\delta \ll 1$ as in the hybrid-A scheme, i.e., $a_r \gg 1$ and $\beta \approx \epsilon$, the fractional complexity and feedback for the hybrid-B protocol are given by:

$$\mathtt{c}_\mathtt{B} = D+\epsilon, \qquad (3.29)$$

$$\mathtt{f}_\mathtt{B} = \frac{\tilde{R}(D)}{h(\epsilon)}. \qquad (3.30)$$

The variation of $\mathtt{c}_\mathtt{B}$ with $\mathtt{f}_\mathtt{B}$ for different values of ϵ is plotted in Fig. 3.5. Also shown are the corresponding hybrid-A curves. It is seen that significant gains are obtained with an optimum rate-distortion code compared to the quantized feedback strategy of hybrid-A, especially for smaller values of ϵ: e.g., for $\epsilon = 0.01$

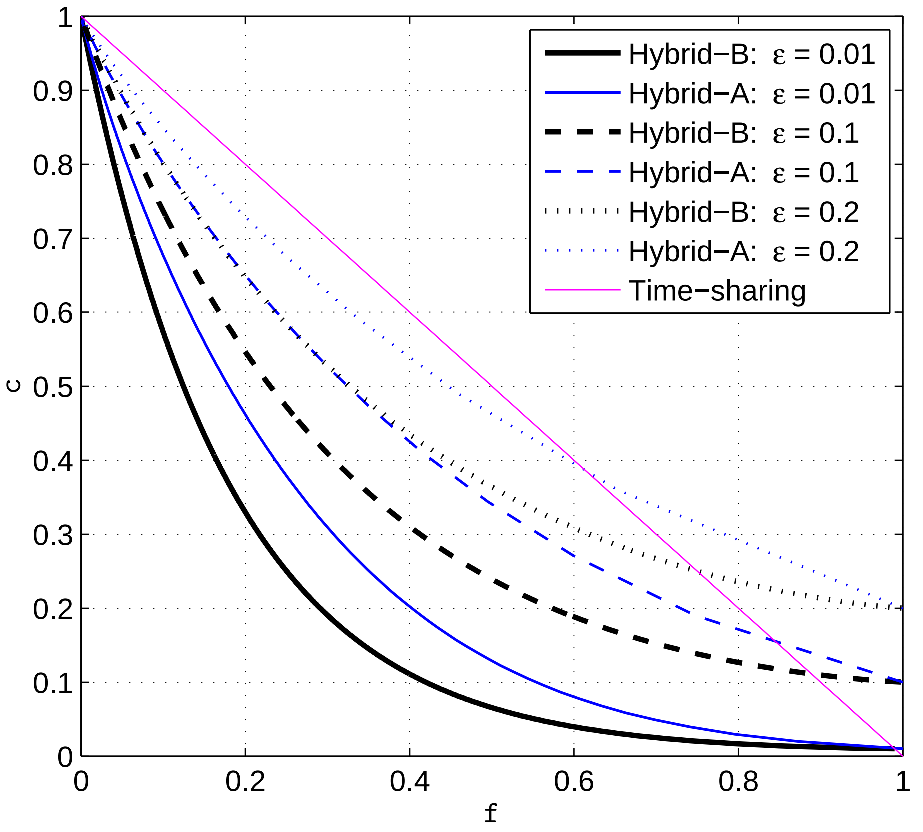


Figure 3.5. Performance of the hybrid-B protocol

and $\mathtt{f_A} = \mathtt{f_B} = 0.4$, we have $\mathtt{c_B} \approx \frac{1}{2} \cdot \mathtt{c_A}$. Since, the coding strategy in hybrid-B is unchanged from hybrid-A, there is the same residual coding complexity for hybrid-B when $\mathtt{f_B} = 1$, i.e., when the distortion $D = 0$.

It is instructive to compare the *rate* R_{QF} of the quantized feedback strategy, i.e., number of bits fed back per forward transmission, with the rate-distortion function $\tilde{R}(D)$. It is easily seen that $R_{QF} = h(p)/\ell$. We compare the two rates for a fixed distortion D; note that the distortion introduced by quantized feedback is simply $D = p - \epsilon$, i.e., the difference between the fraction of channel outcomes marked as erasures and the true fraction of erasures. The ratio $R_{QF}/\tilde{R}(D)$ is plotted as a function of D in Fig.3.6, for different values of ϵ (again, ℓ is varied between 1 and 10^4). It is seen that there is a significant gap to optimality with quantized feedback; this is not surprising, considering that it is a fairly naive strategy. It is also seen that this gap *increases* with decreasing ϵ, which is consistent with our observations regarding Fig. 3.5.

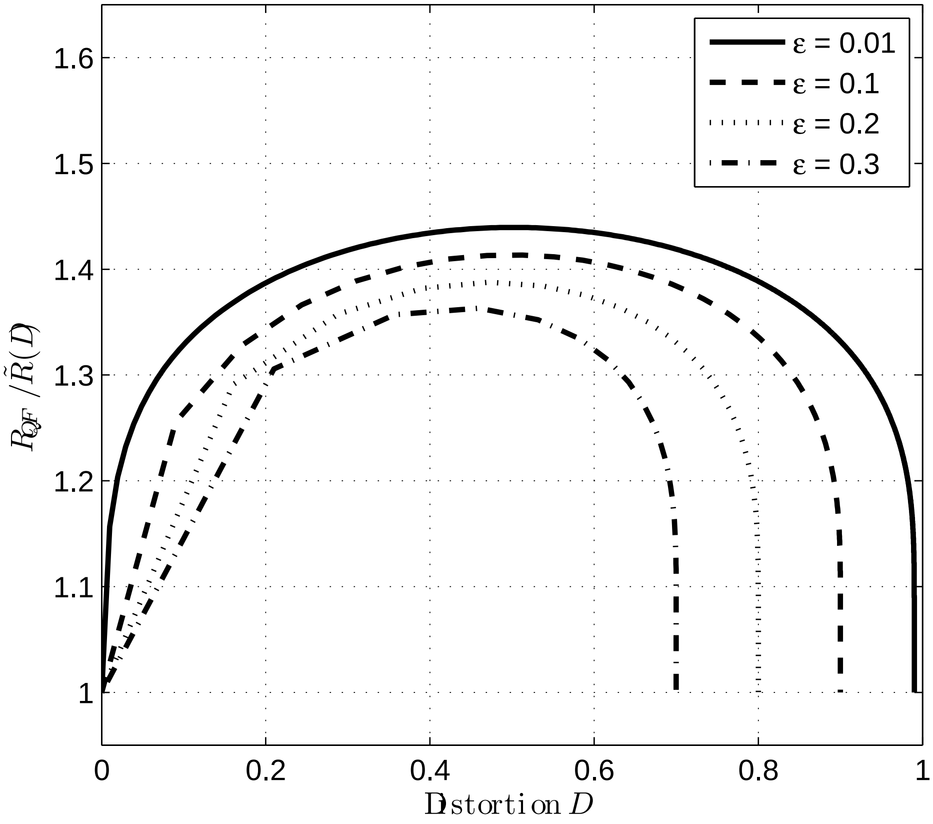


Figure 3.6. Comparison of quantized-feedback with the rate-distortion function

3.7 Hybrid-C: "Distortion-adapted" coding strategy

Both hybrid-A and hybrid-B protocols have the problem of a non-zero residual coding complexity even in the presence of full (i.e., undistorted) feedback. In particular, the fractional complexity $\mathtt{c}$ for these protocols equals the channel erasure rate ϵ when $\mathtt{f} = 1$. Consequently, these protocols perform poorly relative to time-sharing as ϵ is increased, for moderate to large values of the fractional feedback.

As pointed out in Section 3.5, this residual complexity results from the fact that the contribution of the erased bits to the coding complexity remains non-zero even when all erasures have been identified via feedback. More generally, for both hybrid-A and hybrid-B protocols, the average left degree of the encoding set (i.e., parity bits that participate in encoding) in each round remains constant irrespective of the size of the encoding set. Ideally, we would like the left degree to reduce to 1 when the encoding set is reduced to just the set of erased bits, as this corresponds to simply re-transmitting the erased bits in each round.

It turns out that the average left degree of the encoding set can be reduced by appropriately changing the Tornado code used. We illustrate this in the following for the hybrid-B protocol.

In the hybrid-B scheme, the encoding set for round $i+1$ consists of a subset of size $n_i(D+\epsilon)$ of the parities transmitted in round i, which also includes the $n_i\epsilon$ erasures in that round. The βn_i parities transmitted in round $i+1$ help recover these erasures, and are generated from the encoding set by suitably altering the structure of a Tornado code designed for erasure probability ϵ. However, among the parities of the encoding set, the *effective* erasure rate (fraction of erasures) is actually $\epsilon' = \epsilon/(D+\epsilon)$. Thus, the parities of round $i+1$ may alternatively be generated from the encoding set by using the parity layer of a new Tornado code designed for erasure probability ϵ', which helps recover any ϵ' fraction of erasures *within the encoding set.* Since $\epsilon' > \epsilon$, this implies that the new Tornado code must be of *lower* rate than the original code used in the coding-only scheme. For Tornado codes with appropriately chosen d.d. pairs, reducing the rate has the effect of also lowering the average left degree, as shown in the following.

Recall that the choice of the d.d. pair (λ, ρ) for a Tornado code is influenced by the target erasure probability ϵ, desired rate $R = 1 - \beta$ and overhead δ (here, $\delta = 1 - \epsilon/(1 - R)$). Further, the average left and right degrees are functions of the d.d. pair: $a_l = \frac{1}{\sum_i \lambda_i/i}$, and $a_r = \frac{1}{\sum_j \rho_j/j}$ [4].

For a Tornado code of rate R and overhead δ, it can be shown that the average right degree is lower bounded – *independent* of the actual d.d. pair (λ, ρ) – as $a_r \geq \frac{\log(1/\delta)}{\log(1/R)}$ [6]. Consequently, since $a_l = \beta a_r = (1 - R)a_r$, we have

$$a_l \geq \frac{1 - R}{\log(1/R)} \log(1/\delta). \tag{3.31}$$

The function $\frac{1-R}{\log(1/R)}$ is *increasing* in R. Thus, if there exists a family of d.d. pairs (λ, ρ) such that the above lower bound is met for all R and δ, then the average left degree a_l can be reduced by decreasing the code-rate R.

It is shown in [6] that the family of *right-regular* d.d. pairs comes close to satisfying this bound as $\delta \rightarrow 0$. With right-regular d.d. pairs, the right degrees take on only a single value, i.e., the right edge-degree distribution satisfies $\rho_j = \epsilon_{j,m}$ for some m (here, $\epsilon_{j,m}$ is the Kronecker-delta function, i.e., it is a time-sequence indexed by j that takes on value 1 for $j = m$, and 0 for all other j). The left-edge degree distribution λ_i is obtained from the coefficients of an appropriately-truncated Taylor series expansion of $1 - (1 - x)^{1/m}$ about $x = 0$. Specifically, it is shown in [6] that there exists a sequence of right-regular d.d. pairs $(\lambda^{(m)}, \rho^{(m)})$ of rate R, indexed by the right degree m, with the property that:

(i) the overhead δ_m satisfies $\lim_{m\to\infty} \delta_m = 0$,

(ii) the average left degree is given by $a_l^{(m)} = \frac{1-R}{\log(1/R)} \cdot \big(\log \frac{1}{\delta_m} + \Delta_m(R)\big)$, where $\lim_{m\to\infty} \Delta_m(R) \approx 1.78$.

From the above, it can be inferred that for every $0 < R < 1$, there exists a family of right-regular d.d. pairs $(\lambda_\delta, \rho_\delta)$, indexed by the overhead[1] δ, with average left degree

$$a_l = \frac{1-R}{\log(1/R)}\big(\log(1/\delta) + \Delta_\delta(R)\big), \tag{3.32}$$

where $\lim_{\delta\to 0} \Delta_\delta(R) \approx 1.78$.

Summarizing, Tornado codes designed with right-regular d.d. pairs are *asymptotically optimal* ([6]) in the sense that they satisfy the lower bound on a_l up to a constant additive term, whose contribution can be neglected as $\delta{\to}0$. Further, the average left degree a_l decreases with decreasing code-rate in the *limit* as $\delta{\to}0$.

In the following, we shall assume that Tornado codes based on *right-regular d.d. pairs* are employed in all protocols, including the coding-only scheme. However, our results also hold for any family of d.d. pairs that satisfies the lower bound (3.31) on the average left degree in a similar or tighter sense than right-regular d.d. pairs.

The hybrid-C scheme exploits the above property of right-regular distributions and is described as follows:

1. As in the hybrid-B scheme, s transmits $n_i = \beta^i k$ code bits in round i,

[1] Strictly speaking, there may not exist right-regular d.d. pairs for *every* (R, δ) pair. A solution is to employ *time-sharing* between two d.d. pairs that exist for (R, δ_1) and (R, δ_2), with $\delta_1 < \delta < \delta_2$.

$0 \leq i \leq N-1$, and d acknowledges $n_i(1-\epsilon-D)$ unerased bits (i.e., feeds back their addresses) among $n_i(1-\epsilon)$ non-erasures, using an optimum rate-distortion code with distortion D ($0 \leq D \leq 1-\epsilon$).

2. The feedback from round i is used in round $i+1$ as follows: the $n_i' = n_i(D+\epsilon)$ unacknowledged code bits of round i are used to generate $\beta' n_i'$ parities (akin to a parity layer of a Tornado code), using a right-regular d.d. pair designed for rate R' and overhead δ. Here $\beta' = \epsilon'/(1-\delta)$, $\epsilon' = \epsilon/(D+\epsilon)$ and $R' = 1-\beta'$.

   These $\beta' n_i'$ parities help recover up to $\epsilon' n_i' = \epsilon n_i$ erasures in round i. (It is easily verified that $\beta' n_i' = \beta n_i = \beta^{i+1} k$.)

3. In the final round $N-1$, the $\beta^{N-1}k$ parities generated similarly as above are communicated to d using the feedback-only protocol.

Thus, the size of the encoding set and the number of parities generated in each round i above is the same as in the hybrid-B protocol. The chief difference lies in the fact that the average left degree a_l' of the encoding set is now smaller, for δ small enough, as the effective rate R' at which each parity layer is encoded has decreased. The dependence of a_l' on R' and δ is given by (3.32).

The coding complexity for this protocol has the same form as hybrid-B, with the only difference being that the average left-degree is changed to a_l':

$$\mathsf{C}_C = \frac{k}{1-\beta} \cdot \Big((D+\epsilon)a_l' - \beta\Big). \tag{3.33}$$

The fractional complexity is evaluated in the limit as $\delta{\rightarrow}0$:

$$\begin{aligned}
\mathtt{c}_{\mathtt{C}} &= \lim_{\delta\rightarrow 0}\frac{\mathsf{C}_C}{\mathsf{C}_{FEC}} \\
&= \lim_{\delta\rightarrow 0}\frac{(D+\epsilon)a_l' - \beta}{a_l - \beta} \\
&= \lim_{\delta\rightarrow 0}\frac{(D+\epsilon)\cdot\frac{1-R'}{\log(1/R')}\big(\log(1/\delta)+\Delta_\delta(R')\big) - \beta}{\frac{1-R}{\log(1/R)}\big(\log(1/\delta)+\Delta_\delta(R)\big) - \beta} \qquad (3.34) \\
&= \frac{\log(1-\epsilon)}{\log(1-\epsilon')}. \qquad (3.35)
\end{aligned}$$

In going from (3.34) to (3.35), we have used the fact that $\log(1/\delta) \gg \Delta_\delta(\cdot)$, $R \approx 1-\epsilon$ and $R' \approx 1-\epsilon'$ when $\delta \ll 1$. It can be shown that $\mathtt{c}_{\mathtt{C}} < \mathtt{c}_{\mathtt{B}}$ for a fixed value of D, using the series expansion $\log(1-x) = \sum_{i=1}^{\infty} x^i/i$:

$$\begin{aligned}
\frac{\log(1-\epsilon)}{\log(1-\epsilon')} &= \frac{\epsilon}{\epsilon'}\cdot\left(\frac{1+\sum_{i=2}^{\infty}\epsilon^i/i}{1+\sum_{i=2}^{\infty}\epsilon'^i/i}\right) \\
&< \frac{\epsilon}{\epsilon'} \qquad \text{(because } \epsilon < \epsilon'\text{)} \\
&= D+\epsilon \qquad (= \mathtt{c}_{\mathtt{B}}). \qquad (3.36)
\end{aligned}$$

Since the feedback strategy is unchanged from the hybrid-B scheme, the fractional feedback remains the same for the hybrid-C protocol:

$$\mathtt{f}_{\mathtt{C}} = \frac{\tilde{R}(D)}{h(\epsilon)}. \qquad (3.37)$$

It is easily shown that $\lim_{D\rightarrow 0}\mathtt{c}_{\mathtt{C}} = 0$. Hence, $\mathtt{c}_{\mathtt{C}}{\rightarrow}0$ as $\mathtt{f}_{\mathtt{C}}$ approaches 1. Thus, there is *no* residual complexity in the limit of complete feedback, unlike the hybrid-B protocol.

A plot of $\mathtt{c}_\mathtt{C}$ versus $\mathtt{f}_\mathtt{C}$ for different values of ϵ is shown in Fig. 3.7. Also shown are the corresponding plots for hybrid-B. The hybrid-C scheme shows significant improvement over hybrid-B, especially with increasing ϵ and increasing $\mathtt{f}$. For $\epsilon < 0.01$, both schemes perform similarly, and for $\epsilon > 0.5$, hybrid-C performs poorly compared to time-sharing.

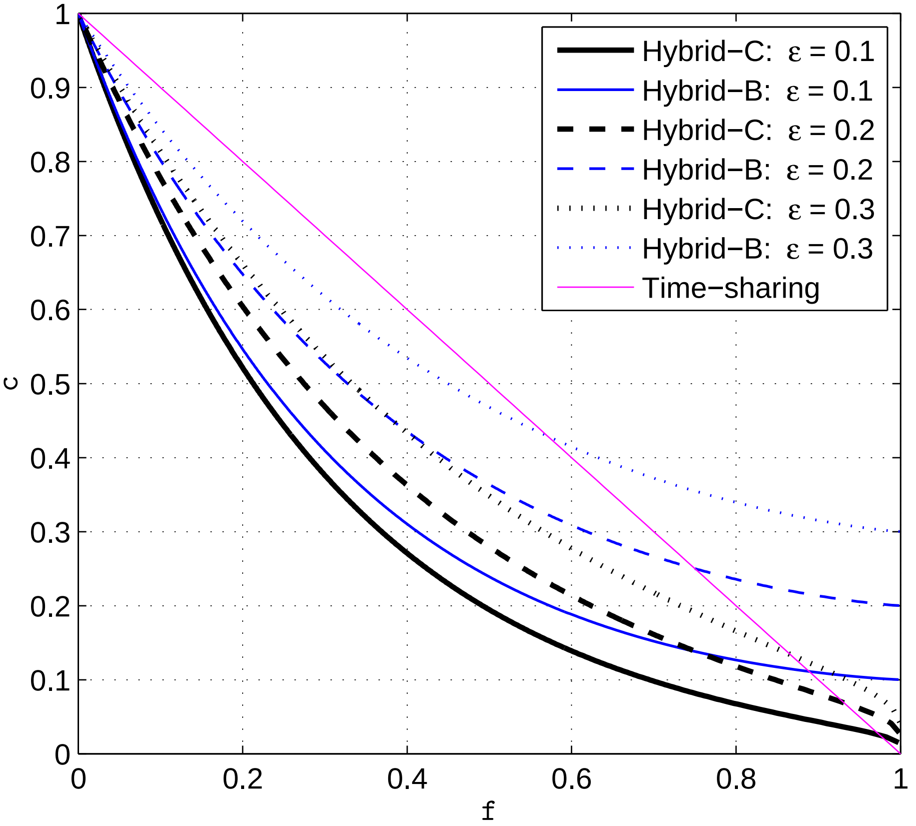


Figure 3.7. Performance of the hybrid-C protocol

3.8 Hybrid-D: Partial feedback of erasure locations

In the hybrid protocols considered so far, feedback is used to pinpoint the locations of a subset of *unerased* bits. In the following, we describe a protocol in which a portion of the *erasure* locations are reported to the source via feedback. The main idea here is to reduce coding complexity by simply *re-transmitting* bits known to be erased, as opposed to encoding them. Such a protocol might be appropriate in a scenario where the erasure probability is quite large and the simplicity of re-transmission is an attractive option.

As with non-erasure locations in the hybrid-B protocol, a rate-distortion theoretic formulation is possible for the problem of efficiently encoding and feeding back a subset of the erasure locations. As before, suppose the channel outcomes associated with m transmissions over a BEC are represented by an m-tuple $\mathbf{x} \in \{0,1\}^m$, where 0 denotes a non-erasure and 1 an erasure. Let these channel outcomes be encoded with some distortion as the m-tuple $\mathbf{y} \in \{1,\mathcal{U}\}^m$, where, as before, $\mathcal{U}$ represents an unacknowledged channel outcome (either erasure or non-erasure), while 1 represents an erasure. The vector $\mathbf{y}$ is fed back to s after appropriate compression. The following rules apply for encoding $\mathbf{y}$:

1. An element of $\mathbf{y}$ can equal 1 only if the corresponding element in $\mathbf{x}$ is also a 1, i.e., non-erasures cannot be represented as erasures.

   (As will be seen later, this condition is necessary to prevent re-transmission of unerased bits, which would lead to waste of forward channel resources resulting in a gap to capacity.)

   Consequently, all non-erasures are unacknowledged in $\mathbf{y}$.

2. The total number of erasures (1's) in $\mathbf{x}$ that are unacknowledged (reported

as $\mathcal{U}$) in $\mathbf{y}$ is upper-bounded by mD, where D is some fixed distortion parameter that takes on values in the interval $[0, \epsilon]$.

Formally, let $\mathbf{y}$ and $\mathbf{x}$ be related as $\mathbf{y} = g(\mathbf{x})$, where $g(\cdot)$ denotes the encoding function, i.e., $g: \{0,1\}^m \to \{1, \mathcal{U}\}^m$. The per-letter distortion measure is given by:

$$\mathcal{D}'(x, y) = \begin{cases} 0, & \text{if } x = 1, y = 1, \\ 1, & \text{if } x = 1, y = \mathcal{U}, \\ \infty, & \text{if } x = 0, y = 1, \\ 0, & \text{if } x = 0, y = \mathcal{U}. \end{cases} \tag{3.38}$$

Analogous to the hybrid-B case, we can define the sum-distortion measure between the vectors $\mathbf{x}$ and $\mathbf{y}$, and the average per-letter distortion $\bar{d}_g$ associated with encoding function $g(\cdot)$. Then, imposing the condition $\bar{d}_g \leq D$ on the encoding function $g(\cdot)$ enforces the encoding rules outlined above.

For $g(\cdot)$ satisfying the distortion constraint $\bar{d}_g \leq D$, a similar result as Theorem 1 holds, showing that the smallest *achievable* entropy for the encoded sequence $\mathbf{y} = g(\mathbf{x})$ is given by the rate-distortion function $\tilde{R}'(D)$:

$$\tilde{R}'(D) = \min_{p_{Y|X}(y|x):\ E(\mathcal{D}'(X,Y)) \leq D} I(X;Y). \tag{3.39}$$

(Here, X and Y are Bernoulli random variables with $Pr(X = 1) = \epsilon$, and the expectation $E(\mathcal{D}'(X,Y))$ is over the joint distribution of X and Y.)

Further, in a similar manner as how (3.25) was derived, it can be shown that

$$\tilde{R}'(D) \ = \ (1 - \epsilon + D) \log_2 \frac{1}{1 - \epsilon + D} \ - \ D \log_2 \frac{1}{D} \ + \ \epsilon \log_2 \frac{1}{\epsilon}. \tag{3.40}$$

In sum, by using an optimum rate-distortion code with rate $\tilde{R}'(D)$ at d, the encoded vector $\mathbf{y}$ can be fed back to s using $m \cdot \tilde{R}'(D)$ bits, such that at most mD erasures in $\mathbf{x}$ are unacknowledged in $\mathbf{y}$.

We now describe the hybrid-D protocol for a fixed distortion $D \in [0, \epsilon]$:

1. In round i, $0 \le i \le N-1$, s transmits $n_i = \gamma^i k$ code bits, where γ is defined below (round 0: message bits). Of the $n_i \epsilon$ erasures that occur, d feeds back the locations of $n_i(\epsilon - D)$ erased bits using an optimum rate distortion code as described above.

2. The feedback from round i is used in round $i+1$ as follows: the code bits of round i are partitioned into two sets F_i and C_i. The set F_i comprises the $n_i(\epsilon - D)$ acknowledged erasures, and C_i consists of the remaining $n'_i = n_i(1 - \epsilon + D)$ bits, which include $n_i D$ unacknowledged erasures. The bits in F_i are simply retransmitted in round $i+1$. However, the bits in C_i are encoded and transmitted as follows. The effective erasure rate among the bits in C_i is $\epsilon'' = D/(1-\epsilon+D)$. Let $\beta'' = \epsilon''/(1-\delta')$, for some $\delta' > 0$. Then, as in the hybrid-C scheme, the n'_i bits of C_i are encoded to produce $\beta'' n'_i$ Tornado parities using a right-regular d.d. pair designed for rate $R'' = 1 - \beta''$ and overhead δ'. So the total number of bits transmitted in round $i+1$ is:

$$\begin{aligned} n_{i+1} &= (\epsilon - D)n_i + \beta'' n'_i \\ &= (\epsilon - D)n_i + \frac{D}{(1-\epsilon+D)\cdot(1-\delta')} \cdot (1-\epsilon+D)n_i \\ &= \gamma \cdot n_i \qquad (3.41) \end{aligned}$$

where

$$\gamma = \epsilon - D + \frac{D}{1-\delta'}. \qquad (3.42)$$

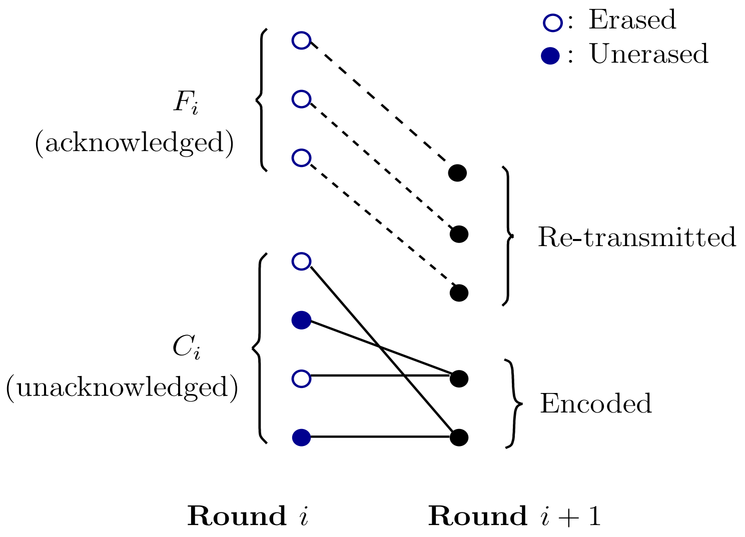


Figure 3.8. Hybrid-D scheme: Encoding process

3. In the final round $N-1$, s generates $\gamma^{N-1}k$ code bits as above, and these are communicated to d using the feedback-only protocol.

The encoding process is illustrated in Fig. 3.8.

The total number of bits transmitted by s is given by $k/(1-\gamma)$; since $\gamma \to \epsilon$ as $\delta' \to 0$, this scheme too approaches capacity. To equalize the total number of forward transmissions in this scheme and the preceding schemes, the overhead δ' must be chosen so that the condition $\gamma = \beta$ is satisfied, where $\beta = \epsilon/(1-\delta)$. This leads to:

$$\begin{aligned} \epsilon - D + \frac{D}{1-\delta'} &= \frac{\epsilon}{1-\delta}, \\ \text{or} \qquad \epsilon \cdot \left(\frac{1}{1-\delta} - 1\right) &= D \cdot \left(\frac{1}{1-\delta'} - 1\right), \end{aligned} \tag{3.43}$$

from which it follows that $\delta' \geq \delta$ since $D \leq \epsilon$. Further, in the limit of small δ and δ', we can use the approximation $\frac{1}{1-x} - 1 \approx x$ above to obtain $\delta' \approx \frac{\epsilon}{D} \cdot \delta$.

Since a right regular d.d. pair is used for code construction, the average left degree a_l'', rate R'' of the d.d. pair and overhead δ' are related according to (3.32). Unlike the hybrid-C scheme, the average left degree a_l'' in this case is *larger* than in the coding-only protocol, because $R'' > R$.

The average right degree a_r'' is given by $a_r'' = a_l''/\beta''$. The encoding (and decoding) complexity associated with round $i+1$ is $\beta'' n_i'(a_r''-1)$. Summing over all the rounds yields

$$\begin{aligned}\sum_{i=0}^{N-2} \beta'' n_i'(a_r''-1) &= \sum_{i=0}^{N-2} \beta''(1-\epsilon+D)\cdot n_i(a_r''-1)\\ &= \sum_{i=0}^{N-2}(1-\epsilon+D)\cdot\gamma^i k\cdot(a_l''-\beta'')\\ &= (1-\epsilon+D)\cdot k\cdot\frac{1-\gamma^{N-1}}{1-\gamma}\cdot(a_l''-\beta''). \qquad (3.44)\end{aligned}$$

Neglecting γ^{N-1} for large N and setting $\gamma=\beta$, the coding complexity for the hybrid-D scheme is given by:

$$\mathsf{C}_D = \frac{k}{1-\beta}\cdot(1-\epsilon+D)\cdot\Big(a_l''-\beta''\Big). \qquad (3.45)$$

The fractional complexity is again evaluated in the limit as $\delta\to 0$, noting that δ

and δ' are related according to (3.43):

$$\begin{aligned}
\mathtt{c_D} &= \lim_{\delta\to 0}\frac{\mathsf{C}_D}{\mathsf{C}_{FEC}},\\
&= (1-\epsilon+D)\cdot\lim_{\delta\to 0}\frac{a_l''-\beta''}{a_l-\beta},\\
&= (1-\epsilon+D)\cdot\lim_{\delta\to 0}\frac{\frac{1-R''}{\log(1/R'')}\big(\log(1/\delta')+\Delta_{\delta'}(R'')\big)-\beta''}{\frac{1-R}{\log(1/R)}\big(\log(1/\delta)+\Delta_{\delta}(R)\big)-\beta},\\
&= (1-\epsilon+D)\cdot\lim_{\delta\to 0}\frac{\frac{1-R''}{\log(1/R'')}\big(\log(1/\delta)+\log(D/\epsilon)+\Delta_{\delta'}(R'')\big)-\beta''}{\frac{1-R}{\log(1/R)}\big(\log(1/\delta)+\Delta_{\delta}(R)\big)-\beta},\\
&= (1-\epsilon+D)\cdot\frac{\epsilon''}{-\log(1-\epsilon'')}\cdot\frac{-\log(1-\epsilon)}{\epsilon},\\
&= \frac{D}{\epsilon}\cdot\frac{\log(1-\epsilon)}{\log(1-\epsilon'')}. \qquad (3.46)
\end{aligned}$$

Here, we have used the approximation $\delta' \approx \frac{\epsilon}{D}\delta$ for small δ, and the fact that $\epsilon'' = \frac{D}{1-\epsilon+D}$.

The amount of feedback needed in round i ($0 \le i \le N-2$) is given by $n_i\cdot\tilde{R}'(D)$. Neglecting the feedback in round $N-1$ (for large N), the total amount of feedback required for the hybrid-D scheme is given by:

$$\mathsf{F}_D = \frac{k}{1-\beta}\cdot\tilde{R}'(D). \qquad (3.47)$$

(Here, we have again set $\gamma = \beta$.)

Evaluating the fractional feedback as $\delta\to 0$ yields

$$\mathtt{f_D} = \frac{\tilde{R}'(D)}{h(\epsilon)}. \qquad (3.48)$$

In Fig. 3.9, plots of $\mathtt{c_D}$ versus $\mathtt{f_D}$ are shown for different values of ϵ. This protocol performs poorly for a large range of ϵ (up to 0.8). However, in contrast with earlier schemes, the performance of the hybrid-D scheme is seen to *improve* with increasing ϵ. For $\epsilon > 0.9$, the performance is better than the time-sharing and hybrid-C schemes for a wide range of $(\mathtt{c}, \mathtt{f})$ values.

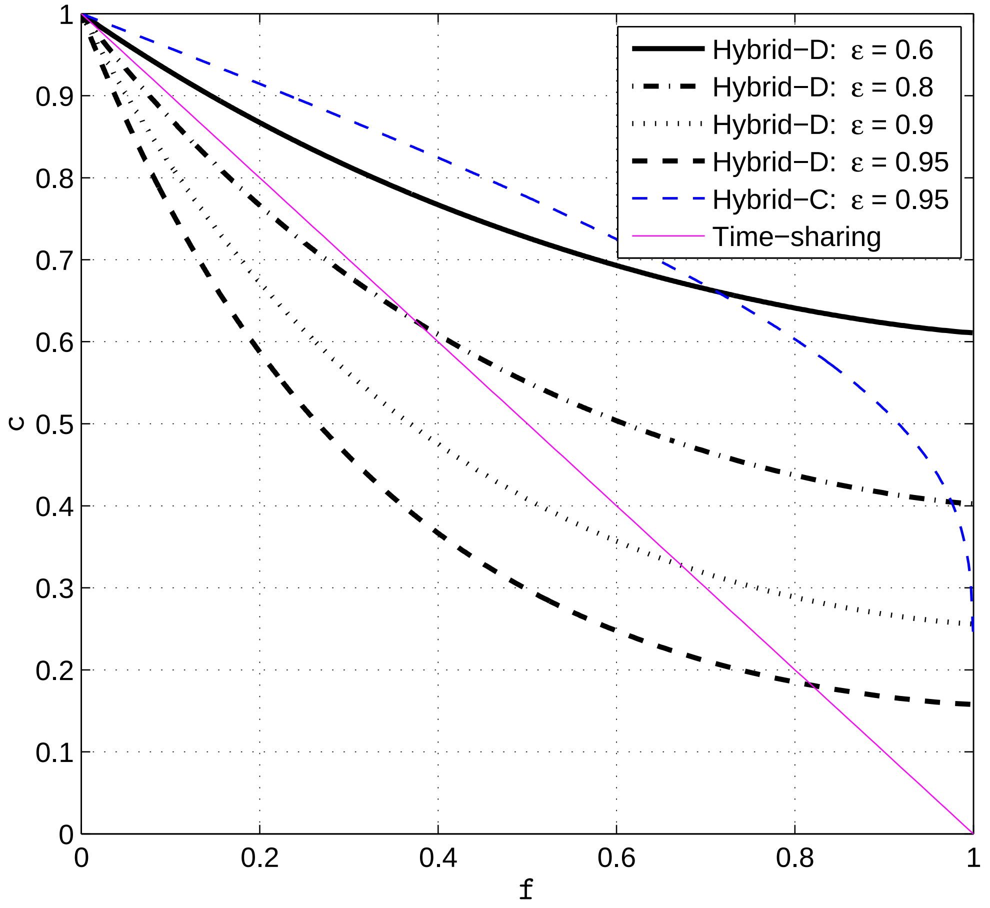


Figure 3.9. Performance of the hybrid-D protocol

Observe the relatively large "residual" fractional complexity in Fig. 3.9 when $\mathtt{f_D} = 1$, i.e., when the distortion $D = 0$. This arises from the behavior of the average left degree a_l'' in the hybrid-D scheme, as $D \rightarrow 0$ (which results in $\epsilon'' \rightarrow 0$ and the effective code-rate $R'' \rightarrow 1$). The exact value of $\mathtt{c_D}$ in the limit as $D \rightarrow 0$ can be shown to be $\frac{1-\epsilon}{\epsilon} \log \frac{1}{1-\epsilon}$.

Note that an alternative to the coding approach in the hybrid-D protocol is to employ a similar approach as in the hybrid A and B protocols, wherein the Tornado code is *not* adapted according to the distortion in the feedback. In other words, we could use the same Tornado code as in the coding-only protocol (of rate $R = 1 - \epsilon/(1-\delta)$), such that erasures acknowledged in each round are simply omitted from the encoding process in the next round, and the erased bits are themselves re-transmitted separately. While this method can potentially reduce the encoding/decoding complexity, it requires the total number of transmissions from the source to be much larger than $k/(1-\beta)$. Consequently, this scheme *cannot* achieve capacity and is hence not considered here.

3.9 Hybrid-E: A generalization of hybrid-C and hybrid-D schemes

In this protocol, we generalize the feedback strategies used in hybrid-C and hybrid-D protocols, so that a subset of *both* erasure and non-erasure locations are conveyed to the source. The rate distortion formulation for this case involves *two* distortion measures.

As before, let the channel outcomes of m transmissions be represented by $\mathbf{x} \in \{0,1\}^m$ (0: non-erasure, 1: erasure). Let these channel outcomes be encoded as $\mathbf{y} = h(\mathbf{x}) \in \{0, 1, \mathcal{U}\}^m$, where unacknowledged channel outcomes ($\mathcal{U}$) occur in *addition* to erasures (1) and non-erasures (0). The vector $\mathbf{y}$ is fed back to s after appropriate compression. The rules in encoding $\mathbf{y}$ are:

1. Non-erasures cannot be reported as erasures, and vice versa.

2. The total number of non-erasures and erasures that are unacknowledged (reported as $\mathcal{U}$) in $\mathbf{y}$ are upper-bounded by mD_0 and mD_1, respectively. Here $D_0 \in [0, 1-\epsilon]$ and $D_1 \in [0, \epsilon]$ are fixed distortion parameters.

Formally, the per-letter distortion measures are given by:

$$\mathcal{D}_0(x, y) = \begin{cases} 1, & \text{if } x = 0, y = \mathcal{U}, \\ \infty, & \text{if } x = 1, y = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.49}$$

and

$$\mathcal{D}_1(x, y) = \begin{cases} 1, & \text{if } x = 1, y = \mathcal{U}, \\ \infty, & \text{if } x = 0, y = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.50}$$

Let $\bar{d}_{0,h}$ and $\bar{d}_{1,h}$ denote the corresponding average per-letter distortions associated with the encoding function $h(\cdot)$. Then, the conditions $\bar{d}_{0,h} \leq D_0$ and $\bar{d}_{1,h} \leq D_1$ on $h(\cdot)$ enforce the above encoding rules.

The rate-distortion function in this case is defined as

$$\tilde{R}(D_0, D_1) = \min_{p_{Y|X}(y|x):\ E(\mathcal{D}_i(X,Y)) \leq D_i,\ i \in \{0,1\}} I(X; Y). \tag{3.51}$$

where X and Y are Bernoulli random variables with $Pr(X = 1) = \epsilon$.

For the case of multiple (vector) distortion criteria, it can be shown that a result similar to Theorem 1 holds [37], i.e., the smallest achievable entropy for the encoded sequence $\mathbf{y} = h(\mathbf{x})$ is given by $m\tilde{R}(D_0, D_1)$. Hence, this is the smallest number of bits needed to feed back m channel outcomes, such that at most mD_0 non-erasures and mD_1 erasures are unacknowledged. Further, it can be shown that the rate-distortion function evaluates to:

$$\begin{aligned}\tilde{R}(D_0, D_1) &= h(\epsilon) \ + \ (D_0 + D_1)\log_2 \frac{1}{D_0 + D_1} \ - \ D_0 \log_2 \frac{1}{D_0} \ - \ D_1 \log_2 \frac{1}{D_1} \\ &= h(\epsilon) - (D_0 + D_1) \cdot h\Big(\frac{D_0}{D_0 + D_1}\Big) \qquad (3.52)\end{aligned}$$

(See Appendix B for a derivation.)

We now describe the hybrid-E protocol for a fixed distortion pair (D_0, D_1), with $D_0 \in [0, 1 - \epsilon]$ and $D_1 \in [0, \epsilon]$:

1. In round i, $0 \leq i \leq N - 1$, s transmits $n_i = \tilde{\gamma}^i k$ code bits (round 0: message bits), where $\tilde{\gamma}$ is defined below. Of the n_i channel outcomes, d feeds back the locations of $n_i(1 - \epsilon - D_0)$ unerased bits and $n_i(\epsilon - D_1)$ erased bits, using an optimum rate distortion code of rate $\tilde{R}(D_0, D_1)$ as described above.

2. In round $i + 1$, the feedback from round i is used to partition the code bits of round i into three sets E_i, R_i and U_i. The set E_i consists of the $n_i(\epsilon - D_1)$ acknowledged erasures, R_i comprises the $n_i(1 - \epsilon - D_0)$ acknowledged non-erasures, and the remaining $n'_i = n_i(D_0 + D_1)$ unacknowledged bits make up U_i.

   The bits in E_i are simply retransmitted in round $i + 1$. The bits in R_i do not participate in round $i + 1$, as they have already been received by d. The bits in U_i are encoded and transmitted as follows.

   The effective erasure rate among the bits in U_i is $\tilde{\epsilon} = D_1/(D_0 + D_1)$. Let $\tilde{\beta} = \tilde{\epsilon}/(1 - \tilde{\delta})$, for some $\tilde{\delta} > 0$. Then, as in the hybrid C and D schemes, the n'_i bits of U_i are encoded to produce $\tilde{\beta} n'_i$ Tornado parities using a right-regular d.d. pair designed for rate $\tilde{R} = 1 - \tilde{\beta}$ and overhead $\tilde{\delta}$.

So the total number of bits transmitted in round $i+1$ is:

$$\begin{aligned} n_{i+1} &= (\epsilon - D_1)n_i + \tilde{\beta}n'_i \\ &= (\epsilon - D_1)n_i + \frac{D_1}{(D_0 + D_1)\cdot(1-\tilde{\delta})} \cdot (D_0 + D_1)n_i \\ &= \tilde{\gamma} \cdot n_i \end{aligned} \tag{3.53}$$

where

$$\tilde{\gamma} = \epsilon - D_1 + \frac{D_1}{1 - \tilde{\delta}}. \tag{3.54}$$

(Note the similarity between the above expression and that for γ in (3.42) for the hybrid-D scheme.)

3. In the final round $N-1$, s generates $\tilde{\gamma}^{N-1}k$ code bits as above, and these are communicated to d using the feedback-only protocol.

As in the hybrid-D scheme, in order to utilize the same total number of transmissions as all preceding hybrid schemes, we set $\tilde{\gamma} = \beta$. This gives rise to a similar condition between $\tilde{\delta}$ and δ as (3.43). Again, for small δ, this simplifies to $\tilde{\delta} \approx \frac{\epsilon}{D_1}\delta$.

The average left degree $\tilde{a}_l$, rate $\tilde{R}$ and overhead $\tilde{\delta}$ are related according to (3.32). Further, the average left and right degrees are related as $\tilde{a}_l = \tilde{\beta}\tilde{a}_r$. Then, similar to the case of the hybrid-D scheme, it can be shown that the coding complexity of the hybrid-E scheme is given by:

$$\mathsf{C}_E = \frac{k}{1-\beta} \cdot (D_0 + D_1) \cdot \left(\tilde{a}_l - \tilde{\beta}\right), \tag{3.55}$$

and the fractional complexity can be shown to be:

$$\begin{aligned}\mathtt{c_E} &= \lim_{\delta\to 0}\frac{\mathsf{C}_E}{\mathsf{C}_{FEC}},\\ &= \frac{D_1}{\epsilon}\cdot\frac{\log(1-\epsilon)}{\log(1-\tilde{\epsilon})},\\ &= \frac{D_1}{\epsilon}\cdot\frac{\log(1-\epsilon)}{\log(1-D_1/(D_0+D_1))}. \end{aligned}\tag{3.56}$$

Likewise, the total amount of feedback required for the hybrid-E scheme (when N is large) is:

$$\mathsf{F}_E = \frac{k}{1-\beta}\cdot\tilde{R}(D_0,D_1). \tag{3.57}$$

and evaluating the fractional feedback as $\delta\rightarrow 0$ yields

$$\mathtt{f_E} = \frac{\tilde{R}(D_0,D_1)}{h(\epsilon)}. \tag{3.58}$$

Note that $\mathtt{c_E}$ and $\mathtt{f_E}$ each depend on both D_0 and D_1. It remains to determine the optimal (D_0, D_1) pair for each value of $\mathtt{f_E}$, so that the corresponding $\mathtt{c_E}$ is minimized.

In order to perform the above optimization, we employ the following change of variables, viz, $p \triangleq D_1/(D_0+D_1)$ and $D = D_0 + D_1$. Consequently, $D_1 = Dp$ and $D_0 = D(1-p)$. Then, the rate-distortion function in (3.52) can be re-written in terms of D and p as:

$$\tilde{R}(D,p) = h(\epsilon) - D\cdot h(p). \tag{3.59}$$

Further, the fractional complexity and fractional feedback in terms of (D, p) are

$$\mathtt{c}_{\mathtt{E}} = \frac{\log \frac{1}{1-\epsilon}}{\epsilon} \cdot \frac{D \cdot p}{\log \frac{1}{1-p}}, \tag{3.60}$$

$$\mathtt{f}_{\mathtt{E}} = 1 - \frac{D \cdot h(p)}{h(\epsilon)}. \tag{3.61}$$

Now, the problem of determining the optimal (D, p) pair, that minimizes $\mathtt{c}_{\mathtt{E}}$ for a fixed value of $\mathtt{f}_{\mathtt{E}}$, may be cast as follows.

Minimize $D \cdot g(p)$ subject to $D \cdot h(p) = \kappa$, where

$$g(p) = \frac{p}{\log \frac{1}{1-p}} \tag{3.62}$$

and κ is some constant in the interval $[0, h(\epsilon)]$.

This problem is solved in Appendix C. We present the solution here.

Let p^* denote the unique value of p that minimizes $f(p) \triangleq g(p)/h(p)$; this is given by $p^* \approx 0.692$. Also, let $\tilde{D}(\kappa)$ denote the unique solution of the equation $D \cdot h(\epsilon/D) = \kappa$, and $\hat{D}(\kappa)$ the unique solution of $D \cdot h((1-\epsilon)/D) = \kappa$, when each exists. Then, the optimal (D, p) is given by:

1. For $\epsilon \le p^*$,

$$(D, p) = \begin{cases} \left(\dfrac{\kappa}{h(p^*)},\ p^*\right), & \text{for } 0 \le \kappa \le \frac{h(p^*)}{p^*}\epsilon, \\ \left(\tilde{D}(\kappa),\ \dfrac{\epsilon}{\tilde{D}(\kappa)}\right), & \text{for } \frac{h(p^*)}{p^*}\epsilon < \kappa \le h(\epsilon). \end{cases} \tag{3.63}$$

2. For $\epsilon > p^*$,

$$(D,p) = \begin{cases} \left(\dfrac{\kappa}{h(p^*)},\ p^*\right), & \text{for } 0 \leq \kappa \leq \frac{h(p^*)}{1-p^*}(1-\epsilon), \\ \left(\hat{D}(\kappa),\ 1 - \dfrac{1-\epsilon}{\hat{D}(\kappa)}\right), & \text{for } \frac{h(p^*)}{1-p^*}(1-\epsilon) < \kappa \leq h(\epsilon). \end{cases} \tag{3.64}$$

In interpreting the above results, we first note that the two extreme values of κ, viz., $\kappa = 0$ and $\kappa = h(\epsilon)$, correspond to $\mathtt{f_E} = 1$ and $\mathtt{f_E} = 0$, respectively (since $D \cdot h(p) = \kappa$).

Now, suppose $\epsilon < p^*$. Then, from (3.63), when $\kappa < \frac{h(p^*)}{p^*}\epsilon$, it is optimal to split the total distortion D, between erasure and non-erasure locations, in the *fixed* proportion $(p^*, 1-p^*)$. For fixed p, note that both $\mathtt{f_E}$ and $\mathtt{c_E}$ are proportional to D. Consequently, $\mathtt{c_E}$ varies *linearly* with $\mathtt{f_E}$ for this range of κ. When $\kappa > \frac{h(p^*)}{p^*}\epsilon$, from (3.63), it is optimal to set the distortion in the erasure-locations to the *maximum* possible value, i.e., $D_1 = Dp = \epsilon$; the distortion in the non-erasure locations is then given by $D_0 = D - \epsilon$. Thus, for this range of κ, *none* of the erasure locations are conveyed to the source. Consequently, the hybrid-E protocol essentially *coincides* with the hybrid-C protocol in this regime.

Likewise, when $\epsilon > p^*$, the total distortion is again split in the fixed proportion $(p^*, 1-p^*)$, between erasures and non-erasures, for $\kappa < \frac{h(p^*)}{1-p^*}(1-\epsilon)$. Hence, $\mathtt{c_E}$ again varies linearly with $\mathtt{f_E}$ in this regime. For $\kappa > \frac{h(p^*)}{1-p^*}(1-\epsilon)$, it is optimal to set the distortion $D_0 = D(1-p) = 1-\epsilon$, i.e., to *not* feed back any of the non-erasure locations. Hence, the hybrid-E protocol coincides with the hybrid-D protocol in this regime.

The variation of $\mathtt{c_E}$ with $\mathtt{f_E}$ is illustrated in Figures 3.10(a) and 3.10(b), for the two cases $\epsilon < p^*$ and $\epsilon > p^*$, respectively.

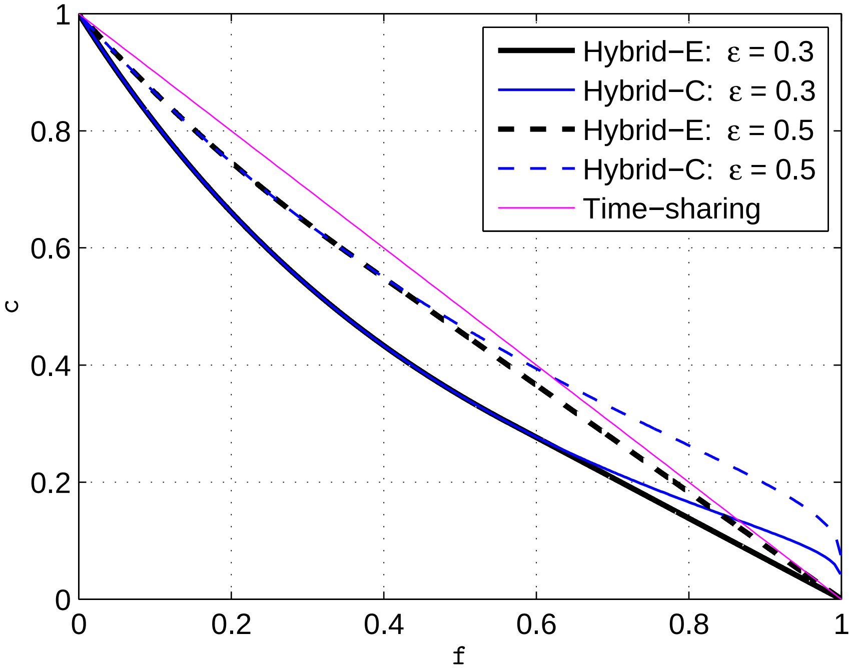
It is seen that, for $\epsilon < p^*$, the curve for the optimized hybrid-E protocol is also the *convex hull* of the curves for the hybrid-C and time-sharing protocols. In particular, for some threshold value $\mathtt{f}_0$, the hybrid-E curve follows hybrid-C when $\mathtt{f}_\mathtt{E} < \mathtt{f}_0$; on the other hand, for $\mathtt{f}_\mathtt{E} > \mathtt{f}_0$, the variation of $\mathtt{c}_\mathtt{E}$ with $\mathtt{f}_\mathtt{E}$ is *linear*. This is consistent with our earlier analysis. Further, it is easy to see that the linear portion of the hybrid-E curve can be replicated by simply *time-sharing* between the hybrid-C scheme, with $\mathtt{f}_\mathtt{C} = \mathtt{f}_0$, and the feedback-only protocol.

Likewise, when $\epsilon > p^*$, the resulting hybrid-E curve is the convex hull of the hybrid-D and time-sharing curves. Again, for some threshold $\mathtt{f}_1$, the hybrid-E and hybrid-D curves coincide for $\mathtt{f}_\mathtt{E} < \mathtt{f}_1$, beyond which the hybrid-E curve is linear. This linear portion can be replicated by time-sharing between the hybrid-D scheme with $\mathtt{f}_\mathtt{D} = \mathtt{f}_1$ and the feedback-only protocol.

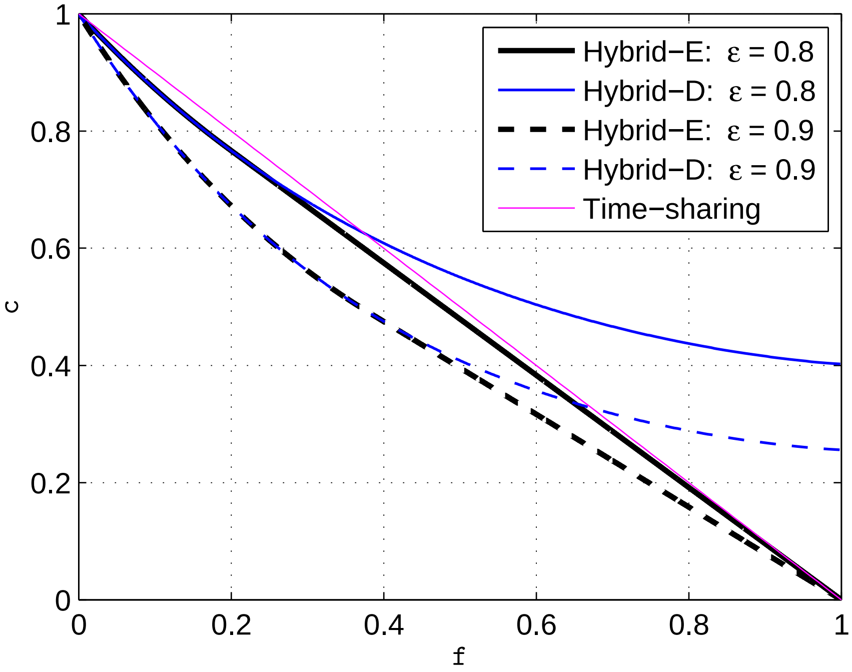
Although not shown, the hybrid-E curve coincides with the time-sharing scheme for $\epsilon = p^*$. Therefore, while in essence, the hybrid-E protocol represents a single unified approach to the hybrid-C and hybrid-D schemes, its performance is "achievable" using either one of the two schemes.

3.10 Summary

In this chapter, we introduced different types of hybrid protocols based on Tornado codes. The main idea behind these protocols is to use a limited amount of feedback in order to reduce encoding and decoding complexity. The first three hybrid strategies – A, B and C – accomplished this by feeding back to the sender a subset of the non-erasure locations among prior transmissions, so that these unerased bits may be omitted from the coding process in the current round. The hybrid-D scheme used feedback to convey a subset of erasure locations, so that the



(a) $\epsilon < p^*$



(b) $\epsilon > p^*$

Figure 3.10: Performance of the hybrid-E protocol.

corresponding erased bits may be *re-transmitted* without being subject to coding. The final hybrid-E scheme employed a combination of both kinds of feedback.

The next chapter deals with some practical aspects of these protocols, related to both coding and feedback.

CHAPTER 4

HYBRID ARQ PROTOCOLS – SOME PRACTICAL CONSIDERATIONS

4.1 Introduction

In this chapter, we investigate some practical aspects of the hybrid protocols introduced in Chapter 3. The main issues addressed here are: 1) the performance of Tornado codes with non-zero overhead, 2) incomplete decoding of Tornado codes, resulting from stopping sets, and 3) practical feedback mechanisms, including the design of low-complexity lossy compression (rate-distortion) schemes. Towards this end, our goal is to provide preliminary solutions to some of the problems that may arise in practice, and guidelines that can assist in making these hybrid protocols more amenable to implementation.

We conclude this chapter with another hybrid scheme that is a variation of the hybrid-B scheme – a variation that adapts the *feedback* strategy according to the actual structure of the Tornado code used.

4.2 Tornado codes with finite non-zero overhead

In Chapter 3, the performance of the hybrid schemes C,D and E was evaluated in the limit as the overhead δ of the underlying Tornado code went to 0. However, this requires that the average left and right degrees a_l and a_r of the code bits grow unbounded, leading to unbounded encoding and decoding complexity.

Therefore, to keep the complexity manageable, in practice, we need to design Tornado codes with a small but *non-zero* overhead. In this section, we explore how this choice influences the complexity-feedback tradeoff curves derived in Chapter 3. In particular, we illustrate the results for the hybrid-C scheme.

Recall that, while the results for the hybrid-C (as well as D and E) schemes in Chapter 3 were derived assuming right-regular d.d. pairs for the underlying Tornado code, they also hold for any family of d.d. pairs that is optimal in the sense of achieving the lower bound (3.31) on the average left degree a_l. However, the results presented in this section specifically make use of right-regular d.d. pairs.

For the hybrid-C scheme, recall that right-regular d.d. pairs need to be designed for erasure probability ϵ' and overhead δ. Given ϵ', we start by constructing a sequence of right-regular d.d. pairs $(\lambda^{(m)}, \rho^{(m)})$ of overhead δ_m, indexed by the right degree m, using a modification[1] of Algorithm 1 in [6]. From this construction, it is observed that $\{\delta_m\}_{m \geq 1}$ form a decreasing sequence, which is in agreement with the results of [6]. In order to obtain the desired overhead δ, we can time-share between two right-regular d.d. pairs $(\lambda^{(m')}, \rho^{(m')})$ and $(\lambda^{(m'+1)}, \rho^{(m'+1)})$, chosen such that their overheads satisfy $\delta_{m'} \geq \delta \geq \delta_{m'+1}$. This is illustrated in Appendix D. Note that the resulting d.d. pair is not strictly right-regular, in the sense that the right degrees can be either m' or $m' + 1$.

It is now possible to evaluate the complexity of the hybrid-C scheme using the average left and right degrees of code bits obtained via the above approach. For a fixed value of $\delta = 0.01$, Fig. 4.1 illustrates the corresponding fractional complexity $\mathtt{c}_\mathtt{C}$ versus fractional feedback $\mathtt{f}_\mathtt{C}$ curves. In comparison with the case when $\delta = 0$,

[1] When specialized to right-regular d.d. pairs, the algorithm described in [6] takes as inputs m and the rate R', and outputs the d.d. pair $(\lambda^{(m)}, \rho^{(m)})$ along with the overhead δ. Our modification is to input the erasure probability ϵ' instead of the rate.

it is seen that these curves are very similar and even marginally better as ϵ is increased. The marginal improvement comes from the fact that, when computing c_C in the limit $\delta{\rightarrow}0$, we neglected the contribution of β in comparison with a_l and a_l' in (3.34). For $\delta = 0.01$, not neglecting the contribution of β reduces the value of c_C further.

Here, it should be kept in mind that, for each value of δ (i.e., 0 and 0.01), the fractional complexity c_C is evaluated relative to the coding-only protocol using a baseline Tornado code that has the *same* overhead δ. Hence, the complexity-feedback curves for the two values of δ are obtained with respect to two *different* baseline coding-only schemes.

In sum, the "promised" tradeoff curves in chapter 3 for $\delta{\rightarrow}0$ are achievable even for small non-zero values of δ such as $\delta = 0.01$.

4.2.1 Comparison with other capacity achieving block codes

As outlined earlier, besides Tornado codes, there are other classes of c.a. block codes for the erasure channel, such as systematic irregular-repeat-accumulate (SIRA) codes ([7]), non-systematic irregular-repeat-accumulate (NSIRA) codes ([9]), and accumulate-repeat-accumulate (ARA) codes ([10]). In particular, ARA and NSIRA codes have the property that their encoding and decoding complexity remains *bounded* as their gap to capacity δ (similar to the overhead parameter for Tornado codes) approaches 0.

It is therefore of interest to compare the complexity of Tornado codes for small values of δ, with the asymptotically constant complexity (as $\delta{\rightarrow}0$) of ARA and NSIRA codes. For ARA and NSIRA codes, we choose degree-3 check-regular (CR-3) and degree-3 bit-regular (BR-3) codes, as outlined in Table I of [10]; these are

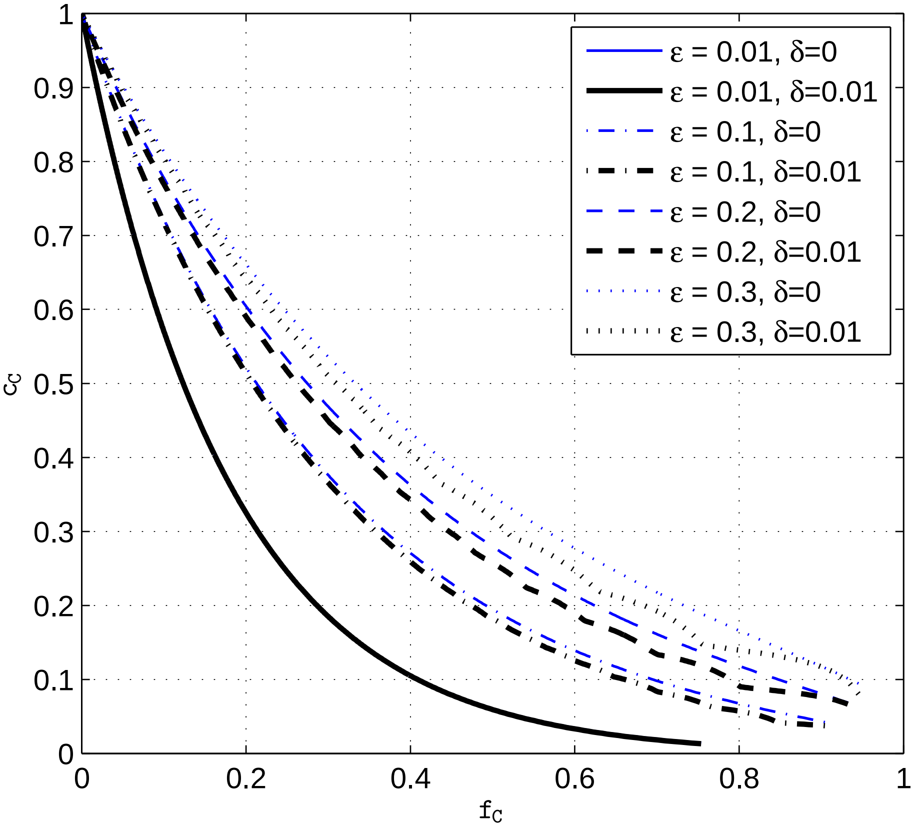


Figure 4.1. Performance of the hybrid-C protocol using a right-regular Tornado code with overhead $\delta = 0.01$

representative of the corresponding c.a. classes of codes.

The coding complexity in [10] is computed as the number of *edges* in the Tanner graph representing the code (normalized by the number of message bits), which is related but not identical to the metric we use, i.e., the number of XOR operations needed to encode and decode. Hence, we need to re-compute the complexities of ARA and NSIRA codes in terms of the number of XORs needed per codeword, per message bit; the results are outlined in Table 4.1, along with the range of ϵ

for which each code achieves capacity.

The coding complexities of right-regular Tornado codes, for $\delta \neq 0$, as well as the asymptotic complexities of NSIRA and ARA codes (as $\delta \rightarrow 0$) are compared in Fig. 4.2. As expected, the coding complexity for Tornado codes increases as δ is reduced. For $\delta = 0.001$, the complexity is significantly higher than that of NSIRA and ARA codes. As δ is increased to about $0.01 \sim 0.02$, this gap is reduced, and the complexities are comparable over a wide range of ϵ. The widest gap occurs for small ϵ (< 0.08), when the complexity of Tornado codes with $\delta = 0.02$ is about a factor of 1.5 times larger than that of the NSIRA BR-3 codes.

TABLE 4.1

COMPLEXITY OF ARA AND NSIRA CODES, MEASURED AS THE TOTAL NUMBER OF XOR OPERATIONS NORMALIZED BY THE MESSAGE SIZE k

Code Ensemble	Type	Range of ϵ	Complexity
NSIRA	Check-regular 3	$(0, 0.95)$	$\dfrac{3}{1-\epsilon}$
NSIRA	Bit-regular 3	$(0, \frac{1}{13})$	3
ARA	Check-regular 3	$(0.616, 1)$	$1 + \dfrac{3\epsilon}{1-\epsilon}$
ARA	Bit-regular 3	$(0, 0.348)$	4

In summary, there is a natural complexity penalty associated with using Tornado codes as opposed to other c.a. block codes with bounded complexity, especially for small overheads (~ 0.001) and for small ϵ (< 0.1). If larger overheads (between 0.01 and 0.05) are permissible in practice for this range of ϵ, then this "complexity-gap" for Tornado codes may be significantly reduced.

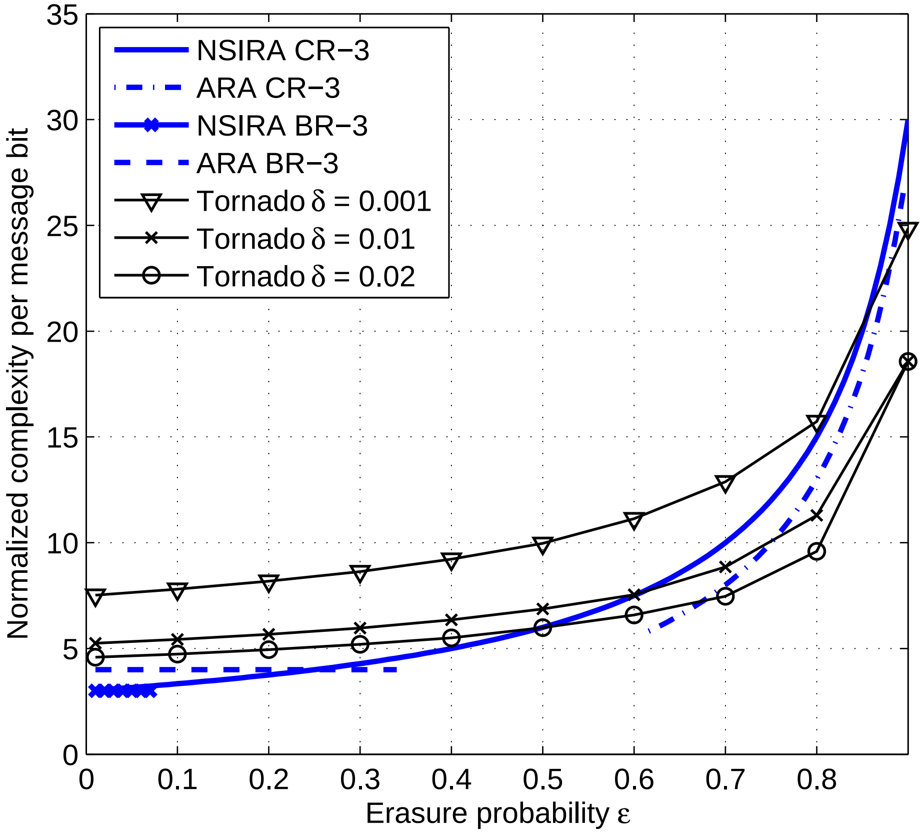


Figure 4.2. Coding complexity of NSIRA, ARA and Tornado codes, as a function of the erasure probability ϵ

4.3 Stopping sets in Tornado codes

It is well-known [4] that the decoding algorithm for Tornado codes (i.e., the belief-propagation algorithm) fails to recover a set of erased left nodes belonging to a bipartite graph in the cascade, if this set constitutes a *stopping set* [22], i.e., a set of left nodes whose every neighbor on the right is connected to at least two left nodes in this set. In particular, it is shown in [4] that stopping sets among the erased bits in a Tornado code occur with positive probability (independent of the blocklength), if the minimum left degree of the bipartite graph is less than 3.

Since the d.d. pairs derived in [4] support a left degree of 2, the resulting Tornado codes do contain stopping sets. This is countered in [4] by "artificially" increasing the minimum left degree (over that specified by the d.d. pair) by supplementing the bipartite graph with a small additional set of right nodes along with additional edges from each left node to this set. This, however, has the effect of increasing coding complexity significantly for finite non-zero values of the overhead δ.

The right-regular d.d. pairs used in Section 4.2 also support a minimum left degree of 2, and consequently give rise to stopping sets in Tornado codes constructed using them. As an alternative to the above technique, used in [4], which increases the coding complexity, we propose using feedback to recover undecoded erasures resulting from stopping sets. In particular, it follows from [4, Proposition 2] that for sufficiently large blocklengths, the fraction of unrecovered erasures due to stopping sets can be set to an arbitrarily small constant value. Consequently, these erased bits can be communicated to the destination using simple ARQ, at the cost of a small increase in the amount of feedback.

The amount of feedback needed to recover erased bits forming a stopping set can be further reduced using selective-feedback techniques, as outlined in [21]. In particular, re-transmitting an appropriately chosen (erased) bit within the stopping set can help the decoder to "break-through" the stopping set-barrier and continue decoding till the next (smaller) stopping set is encountered. Thus, the amount of feedback needed for re-transmission requests can be reduced (significantly) from a number equal to the *size* of the initial stopping set, to simply the number of times the decoder *encounters* smaller stopping sets, starting from the same initial set.

4.4 Towards practical rate-distortion codes

In Chapter 3, we have assumed the existence of optimum rate-distortion codes that achieve the rate-distortion curve, when formulating the feedback strategies for hybrid B, D and E schemes. In theory, the existence of codes with performance approaching the rate-distortion curve is guaranteed by non-constructive, random-coding arguments [36]. In practice, deterministic code constructions with low-complexity encoding and decoding algorithms are necessary for implementation.

The problem of designing such "practical" rate-distortion codes to achieve (optimal) lossy compression for different sources and distortion criteria has been studied by several researchers in the past. In particular, low-complexity channel codes, such as low-density generator matrix (LDGM) and low-density parity-check (LDPC) codes have been useful in constructing good rate-distortion codes. For the memoryless binary symmetric source and Hamming distortion criterion, LDGM and LDPC codes have been used to achieve the rate-distortion curve [38], [39]. These results have also been generalized to memoryless asymmetric binary sources with a bounded distortion criterion in [40], [41], [42]. Recall, however, that our rate-distortion formulations deal with asymmetric binary sources with *unbounded* distortion criteria.

We describe in the following a simple, but sub-optimal, code construction based on sparse-bipartite graphs that may be substituted for the rate-distortion code used in the feedback of the hybrid B (or hybrid C) protocol. This scheme consists of a simple pre-coding operation, with encoding/decoding complexity that is linear in blocklength, followed by a standard lossless compressor. Though this does not achieve the rate-distortion function, it is seen to improve on the quantized-feedback strategy of the hybrid-A scheme in certain regimes. In particular, for small ϵ (~ 0.01), this scheme achieves compression rates up to within 10% of the rate-distortion bound (far superior to the quantized-feedback scheme) for a wide range of distortions.

Consequently, this demonstrates the existence of low-complexity compression schemes that can be used to obtain complexity-feedback performance close to that of the hybrid-B protocol, in certain regimes.

4.4.1 A bipartite-graph based lossy compression scheme

As formulated in Section 3.6, we wish to encode the vector representing the channel outcomes $\mathbf{x} \in \{0,1\}^m$ (0: non-erasure, 1: erasure) as the distorted vector $\mathbf{y} \in \{0, \mathcal{U}\}^m$ (0: non-erasure, $\mathcal{U}$: unacknowledged channel outcome), such that the number of non-erasures in $\mathbf{x}$ reported as $\mathcal{U}$ in $\mathbf{y}$ is at most mD, and no erasure in $\mathbf{x}$ is reported as a non-erasure in $\mathbf{y}$. The vector $\mathbf{y}$ is communicated by the destination d to the source s, conveying partial information about the channel outcomes.

To accomplish this, we first encode $\mathbf{x}$ as $\mathbf{z} \in \{0,1\}^{m\hat{R}}$, for some $\hat{R} \in (0,\infty)$, and then establish a mapping from $\mathbf{z}$ to $\mathbf{y}$. Consequently, it suffices to communicate $\mathbf{z}$ to s, instead of $\mathbf{y}$. The relationships between $\mathbf{x}$ and $\mathbf{z}$, and between $\mathbf{y}$ and $\mathbf{z}$ are determined by a randomly constructed bi-regular bipartite graph $\mathcal{G}$.

The graph $\mathcal{G}$ consists of m **left** nodes and $m\hat{R}$ **right** nodes, where each left node is of degree l and each right node of degree r. Every left node is connected to l right nodes chosen uniformly at random from the set of $m\hat{R}$ right nodes; likewise, each right node is connected to r randomly chosen left nodes. As with Tornado codes, note that l and r are related by $l = \hat{R} \cdot r$.

In order to generate $\mathbf{z}$ from $\mathbf{x}$, we first map all the bits in $\mathbf{x}$ to left nodes in $\mathcal{G}$, and all bits in $\mathbf{z}$ to the right nodes. Now, each bit in $\mathbf{z}$ is evaluated as the **logical-OR** of all the bits of $\mathbf{x}$ that it is connected to on the left, i.e., a bit of $\mathbf{z}$ is marked as a 1 if it is connected to at least one bit in $\mathbf{x}$ that is a 1.

To obtain $\mathbf{y}$ from $\mathbf{z}$, we simply replace $\mathbf{x}$ with $\mathbf{y}$ for the left nodes in $\mathcal{G}$ (in the same order), such that $\mathbf{y}$ is now connected to $\mathbf{z}$ in the same way as $\mathbf{x}$. Now, a position in $\mathbf{y}$ is marked 0 if it is connected to at least one bit of $\mathbf{z}$ that is 0; else, that position in $\mathbf{y}$ is set to $\mathcal{U}$. The encoding process is illustrated in Fig. 4.3.

Let the notation a_i be used for the i^{th} element of a vector $\mathbf{a}$. Then, the above encoding process for $\mathbf{z}$ and $\mathbf{y}$ ensures that if $x_i = 1$, then $y_i = \mathcal{U}$, as the common connections to x_i and y_i in $\mathbf{z}$ all evaluate to 1.

Let q denote the probability that a bit in $\mathbf{z}$ is a 1. Therefore, $q = 1 - (1 - \epsilon)^r$. Likewise, let $p = Pr(y_i = \mathcal{U})$, for any i. It follows that $y_i = \mathcal{U}$ under one of two conditions: either (i) $x_i = 1$, as explained above, or (ii) $x_i = 0$ and all its connections in $\mathbf{z}$ evaluate to 1. Let the l connections of x_i in $\mathbf{z}$ be denoted $\zeta(x_i, \mathbf{z})$.

Thus, we have:

$$
\begin{aligned}
p &= \epsilon + (1-\epsilon) \cdot Pr\Big(\bigcap_{j \,\in\, \zeta(x_i,\mathbf{z})} z_j = 1 \;\Big|\; x_i = 0 \Big) \\
&= \epsilon + (1-\epsilon) \cdot \prod_{j \,\in\, \zeta(x_i,\mathbf{z})} Pr\Big(z_j = 1 \;\Big|\; x_i = 0 \Big) \qquad (4.1) \\
&= \epsilon + (1-\epsilon) \cdot \big(1 - (1-\epsilon)^{r-1} \big)^{l} \qquad (4.2)
\end{aligned}
$$

In writing (4.1), we have assumed that the bits in $\mathbf{x}$ that are connected to $\zeta(x_i, \mathbf{z})$ are all distinct, except for x_i itself. In graph-theoretic terms, this is equivalent to x_i not being part of a cycle of length 4. It is possible to construct such a graph for which this is true with high probability, by increasing m with l and r held constant; this results in a *sparse* bipartite graph.

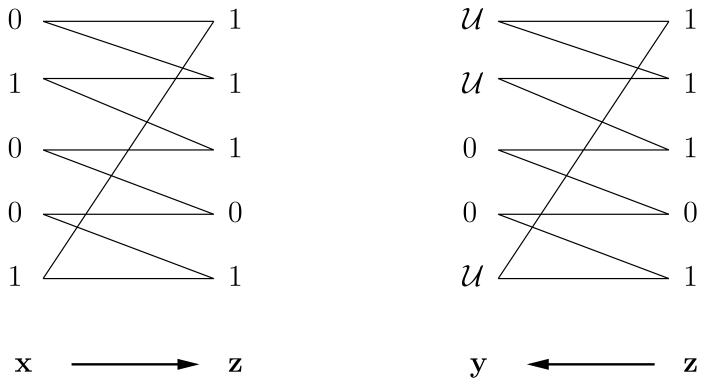


Figure 4.3. Encoding process for the graph-based lossy compression scheme

Since the fraction of erasures in $\mathbf{x}$ is ϵ, the number of non-erasures in $\mathbf{x}$ that are reported as $\mathcal{U}$ in $\mathbf{y}$ is $m(p-\epsilon)$. Thus, the distortion D is given by:

$$\begin{aligned} D &= p-\epsilon \\ &= (1-\epsilon)\cdot\left(1-(1-\epsilon)^{r-1}\right)^{l} \\ &= (1-\epsilon)\cdot\left(1-(1-\epsilon)^{r-1}\right)^{\hat{R}r}. \end{aligned} \tag{4.3}$$

Hence, the distortion may be varied by changing the right degree r. Figure 4.4 shows the variation of the *normalized* distortion $D/(1-\epsilon)$, as a function of the right degree r for different values of $\hat{R}$ and ϵ. It is seen that, in order to span the entire range $0 \leq D \leq (1-\epsilon)$, larger values of r are required for larger $\hat{R}$ (for a given ϵ) and smaller ϵ (for a given $\hat{R}$).

Now, in order to communicate $\mathbf{z}$ to the decoder (in this case, the source s), we need at most $\hat{R}m \cdot h(q)$ bits on average, since the entropy of $\mathbf{z}$ is bounded as:

$$\begin{aligned} H(\mathbf{z}) &\leq \sum_{i=1}^{\hat{R}m} H(z_i) \\ &= \hat{R}m \cdot h(q). \end{aligned} \tag{4.4}$$

Henceforth, we shall assume that this upper bound is tight. Thus, the *rate* of compression $R_{GC}(D)$ achieved by this graph-based construction (for a distortion D, given by (4.3)) is $\hat{R}\cdot h(q)$. We now compare this rate against the rate-distortion function $\tilde{R}(D)$ in (3.25), for different values of ϵ.

In Fig. 4.5, the ratio $R_{GC}(D)/\tilde{R}(D)$ is plotted as a function of the distortion D, for different values of $\hat{R}$ and ϵ. Also shown is the performance of the quantized-feedback strategy used in the hybrid-A protocol (reproduced from Fig. 3.6). It is observed that, for the values of ϵ considered, the graph-based scheme performs

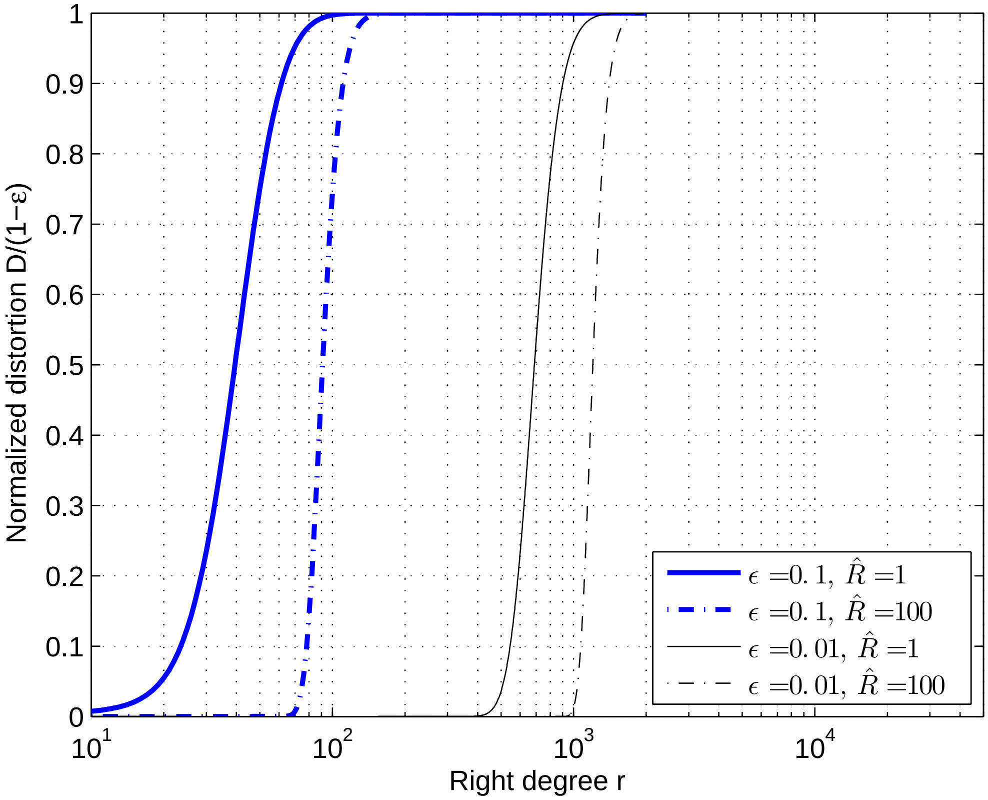
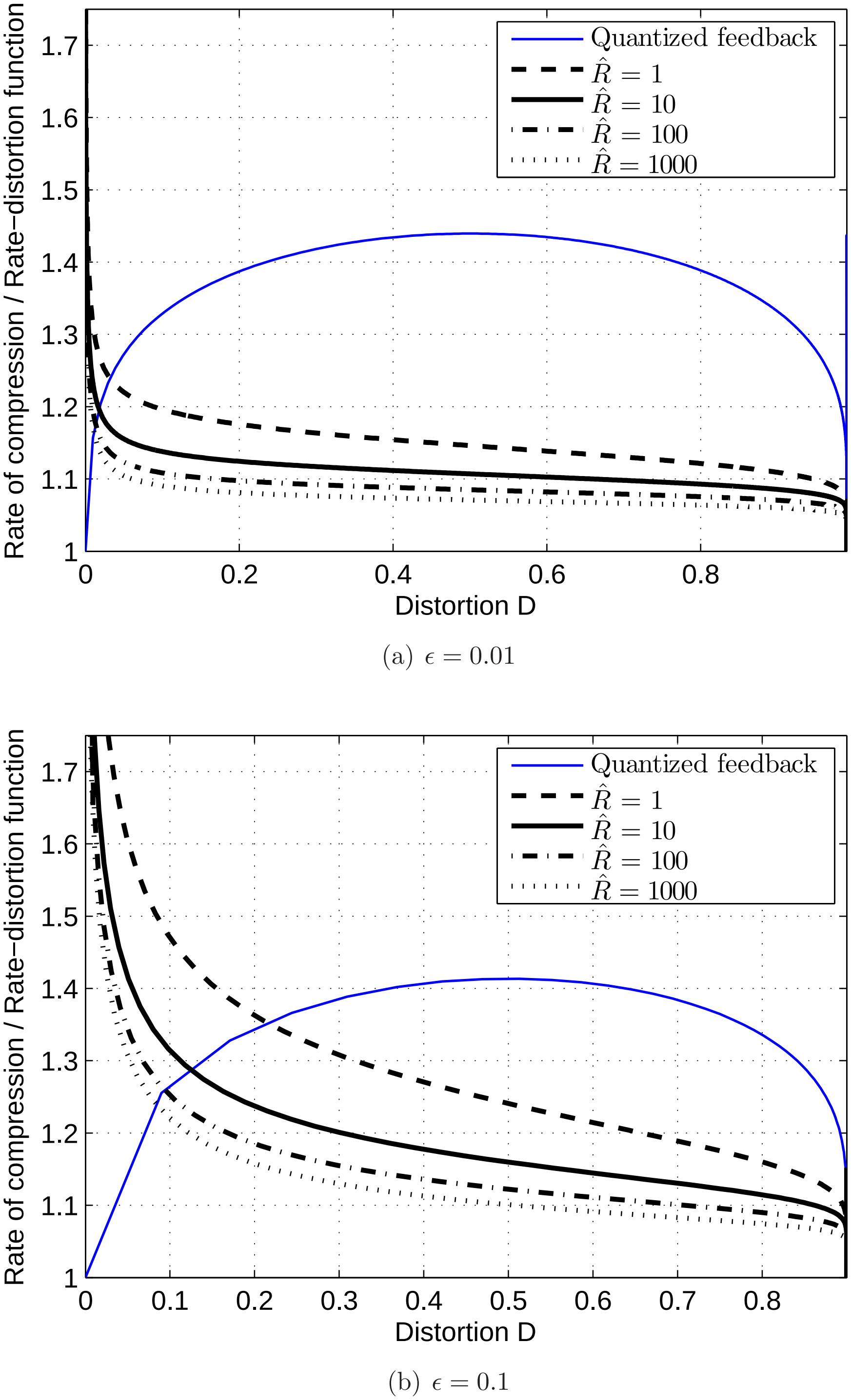


Figure 4.4. Distortion introduced by the graph-based scheme as a function of the right degree r.

better than the quantized feedback strategy, except at lower distortions. Further, increasing $\hat{R}$ is seen to yield better performance. Overall, the graph-based scheme is seen to perform better at lower ϵ, in contrast with the quantized feedback scheme, which achieves better compression at higher ϵ.

It is of interest to see how the above results translate into complexity-versus-feedback performance. For this, consider the hybrid-$\tilde{B}$ protocol which is derived from the hybrid-B protocol by simply replacing the rate-distortion code in its feedback with the graph-based compression scheme described here. The performance



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

Figure 4.5. Ratio of the rate $R_{GC}(D)$ of the graph-based compression scheme to the rate-distortion function $\tilde{R}(D)$.

of this protocol vis-a-vis the hybrid A and B protocols is demonstrated in Fig. 4.6.

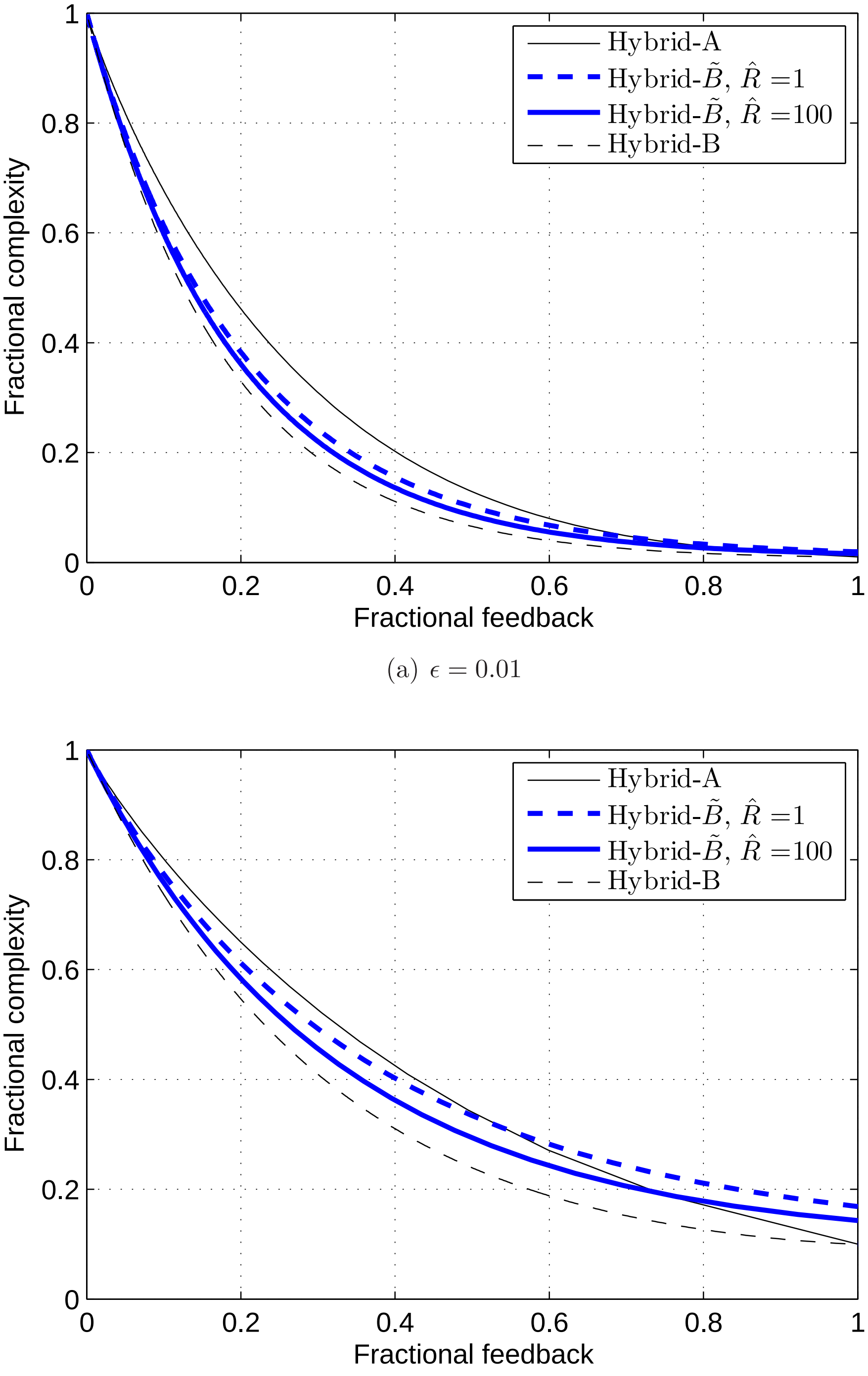
It is seen that for smaller values of fractional feedback, hybrid-$\tilde{B}$'s performance is intermediate to that of hybrid-A and hybrid-B, whereas for larger fractional feedback, it is poorer than both. This is consistent with our observations regarding Fig. 4.5. Further, for a fixed value of $\hat{R}$, the performance gains of hybrid-$\tilde{B}$ over hybrid-A are *bigger* for small ϵ (~ 0.01); recall that the small-ϵ regime also corresponds to a bigger gap between hybrid-A and hybrid-B schemes (Section 3.6). Although not shown, for $\epsilon > 0.2$, the performance of hybrid-$\tilde{B}$ is generally poorer than hybrid-A.

Thus far, we have ignored the complexity of the final lossless compressor in the graph-based scheme, as well as in the hybrid-A and feedback-only protocols. In practice, the same class of lossless compression schemes could be chosen for all protocols to ensure uniformity. Popular examples of such schemes include Huffman coding, arithmetic coding, Lempel-Ziv compression, etc. [43]. Alternatively, if very low compression complexity is desired, simple run-length encoding schemes could also be used, at the cost of higher feedback overhead.

4.5 Adapting the feedback strategy based on the Tornado code

In this section, we discuss a variant of the hybrid-B scheme wherein the feedback is adapted based on the Tornado code used to encode at the source. In a way, this is the dual to the approach in the hybrid-C scheme, where we adapted the Tornado code to the amount of distortion in the fedback information.

In the hybrid-B protocol, the distortion is "uniformly spread" across all non-erasure locations, i.e., the rate-distortion code does not prioritize certain unerased



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

Figure 4.6. Complexity-feedback tradeoff curves obtained using the graph-based compression scheme.

bits over others in acknowledging their locations. However, note that Tornado codes typically have irregular d.d pairs, i.e., either the left nodes or the right nodes or both sets can have more than one possible degree in the Tornado code graph. In particular, when the left-node degree distribution spans more than one degree, it may be advantageous to feed back unerased bits in a given round that have *higher* left degrees, with *lesser* distortion. This is because, bits with higher left degrees contribute more to the encoding complexity in the next round than those with smaller left degrees. Thus, intuitively it makes sense to omit as many large left-degree bits from the encoding process as possible.

Suppose the left edge-degree distribution is given by $\lambda := \{\lambda_i\}_{i=1}^{\infty}$, and the left node-degree distribution by $L := \{L_i\}_{i=1}^{\infty}$. Recall that L_i denotes the fraction of left *nodes* in each bipartite code graph that are of degree i, whereas λ_i represents the fraction of *edges* that originate from left nodes of degree i. It is easily shown [4] that L and λ are related as:

$$L_i \;=\; \frac{\lambda_i/i}{\sum_j \lambda_j/j}. \tag{4.5}$$

Note that the parities transmitted in every round of the hybrid-B protocol have the same left node-degree distribution L. The new *code-adapted* feedback strategy is described as follows:

1. Choose a *threshold* parameter l_T for the left degrees, and a pair of distortions (D_L, D_H) for lower and higher left-degrees, that are non-negative and satisfy $D_L + D_H \leq (1 - \epsilon)$.

2. **Sort** the n_i parities transmitted in round i according to *increasing* left degrees (from the perspective of encoding in round $i + 1$) as $c_1, c_2, \cdots, c_{n_i}$,

i.e., $LDEG(c_1) \leq LDEG(c_2) \leq \cdots \leq LDEG(c_{n_i})$, where $LDEG(\cdot)$ denotes the left-degree. Let $x_1, x_2 \cdots x_{n_i}$ denote the corresponding (sorted) channel outcomes, using the conventional notation, i.e., $x_i = 0$ denotes a non-erasure and $x_i = 1$ an erasure of c_i.

3. Let $m_i \triangleq \max_{1 \leq j \leq n_i}\{LDEG(c_j) \leq l_T\}$. The channel transmissions in round i are partitioned into two vectors $\mathbf{c_L} \triangleq [c_1, \cdots, c_{m_i}]$ and $\mathbf{c_H} \triangleq [c_{m_i+1}, \cdots, c_{n_i}]$. Thus, $\mathbf{c_L}$ represents the set of code bits with low left degrees ($\leq l_T$) and $\mathbf{c_H}$ the set of bits with high left degrees ($> l_T$)

   The corresponding channel outcomes are given by $\mathbf{x_L} \triangleq [x_1, \cdots, x_{m_i}]$ and $\mathbf{x_H} \triangleq [x_{m_i+1}, \cdots, x_{n_i}]$. The vectors $\mathbf{x_L}$ and $\mathbf{x_H}$ are separately encoded and communicated by d to s with distortions D_L and D_H in the non-erasure locations, respectively, using optimum rate-distortion codes of rates $\tilde{R}(D_L)$ and $\tilde{R}(D_H)$, as described in Section 3.6.

4. The encoding in round $i+1$ is done in the same manner as the hybrid-B protocol, with the acknowledged unerased bits in round i omitted from the encoding process.

We refer to this variation of the hybrid-B protocol as the hybrid-C2 protocol. The coding complexity and amount of feedback needed for this protocol are computed as follows.

The fraction of Tornado code bits with left degree no bigger than l_T is given by $\theta_{l_T} = \sum_{j=1}^{l_T} L_j$. Consequently, we have $m_i = n_i \theta_{l_T}$. Let a_l^L and a_l^H denote the average left degrees within the sets $\mathbf{c_L}$ and $\mathbf{c_H}$, respectively. It follows that:

$$a_l^L = \frac{1}{\theta_{l_T}} \sum_{j=1}^{l_T} j \cdot L_j, \tag{4.6}$$

and

$$a_l^H \;=\; \frac{1}{1-\theta_{l_T}} \sum_{j=l_T+1}^{\infty} j \cdot L_j \;. \tag{4.7}$$

Thus, the overall average left degree a_l of the Tornado code is given by:

$$a_l \;=\; \theta_{l_T} \cdot a_l^L \;+\; (1-\theta_{l_T}) \cdot a_l^H. \tag{4.8}$$

Note that $(D_L+\epsilon)m_i$ bits in $\mathbf{c_L}$ and $(D_H+\epsilon)(n_i-m_i)$ bits belonging to $\mathbf{c_H}$ participate in encoding in round $i+1$. The average right degree for round $i+1$ parities is therefore given by:

$$\begin{aligned} a'_r &= \frac{(D_L+\epsilon)m_i a_l^L \;+\; (D_H+\epsilon)(n_i-m_i)a_l^H}{\beta n_i} \\ &= \frac{(D_L+\epsilon)\theta_{l_T} a_l^L \;+\; (D_H+\epsilon)(1-\theta_{l_T})a_l^H}{\beta} \\ &= \frac{D_L\theta_{l_T} a_l^L \;+\; D_H(1-\theta_{l_T})a_l^H \;+\; \epsilon a_l}{\beta}. \end{aligned} \tag{4.9}$$

Since the coding complexity for round $i+1$ is given by $n_{i+1} \cdot (a'_r - 1)$, the overall coding complexity for the hybrid-C2 protocol is:

$$\begin{aligned} \mathsf{C}_{C2} &= \frac{\beta k}{1-\beta} \cdot (a'_r - 1) \\ &= \frac{k}{1-\beta} \cdot \Big(\; D_L\theta_{l_T} a_l^L \;+\; D_H(1-\theta_{l_T})a_l^H \;+\; \epsilon a_l \;-\; \beta \; \Big). \end{aligned} \tag{4.10}$$

The amount of feedback required in round i is:

$$m_i \cdot \tilde{R}(D_L) + (n_i - m_i) \cdot \tilde{R}(D_H) \;=\; n_i \cdot \big(\theta_{l_T}\tilde{R}(D_L) + (1-\theta_{l_T})\tilde{R}(D_H)\big). \tag{4.11}$$

Hence, the total amount of feedback required by the hybrid-C2 protocol is:

$$\mathsf{F}_{C2} \;=\; \frac{k}{1-\beta}\cdot\big(\theta_{l_T}\tilde{R}(D_L) \;+\; (1-\theta_{l_T})\tilde{R}(D_H)\big). \tag{4.12}$$

As a result, the fractional complexity and fractional feedback for the hybrid-C2 protocol are:

$$\mathtt{c}_{\mathtt{C2}} \;=\; \frac{D_L\theta_{l_T}a_l^L \;+\; D_H(1-\theta_{l_T})a_l^H \;+\; \epsilon a_l \;-\; \epsilon}{a_l-\epsilon}, \tag{4.13}$$

$$\mathtt{f}_{\mathtt{C2}} \;=\; \frac{\theta_{l_T}\tilde{R}(D_L) \;+\; (1-\theta_{l_T})\tilde{R}(D_H)}{h(\epsilon)}, \tag{4.14}$$

where we have assumed that the overhead $\delta \ll 1$, and hence $\beta \approx \epsilon$.

In practice, the choice of (D_L, D_H, l_T) can be optimized[2] for each value of $\mathtt{f}_{\mathtt{C2}}$, so as to minimize the corresponding $\mathtt{c}_{\mathtt{C2}}$. The resulting complexity-feedback performance depends on the actual d.d. pair chosen for implementing the Tornado code. In the following, we demonstrate the performance gains that are achievable vis-a-vis the "uniform distortion" strategy used in the hybrid-B protocol, if *right-regular* d.d. pairs are used to construct the Tornado code.

Figure 4.7 illustrates the "optimized" performance of the hybrid-C2 protocol, compared with the hybrid B and C protocols. For these complexity versus feedback curves, the overhead of the right-regular d.d. pairs was chosen to be $\delta = 0.01$. Not surprisingly, it is seen that the hybrid-C2 scheme performs uniformly better than the hybrid B scheme for all values of ϵ and $\mathtt{f}$. For small ϵ (~ 0.01), the hybrid-C2 scheme also outperforms hybrid-C. As ϵ is increased (~ 0.2), the performance gain relative to the hybrid-C scheme comes only in the *low-feedback* regime. Consequently, with right-regular d.d. pairs, the performance of the hybrid-C2 and

[2] Note that, for fixed $\mathtt{f}_{\mathtt{C2}}$, only two of these three parameters can be chosen independently.

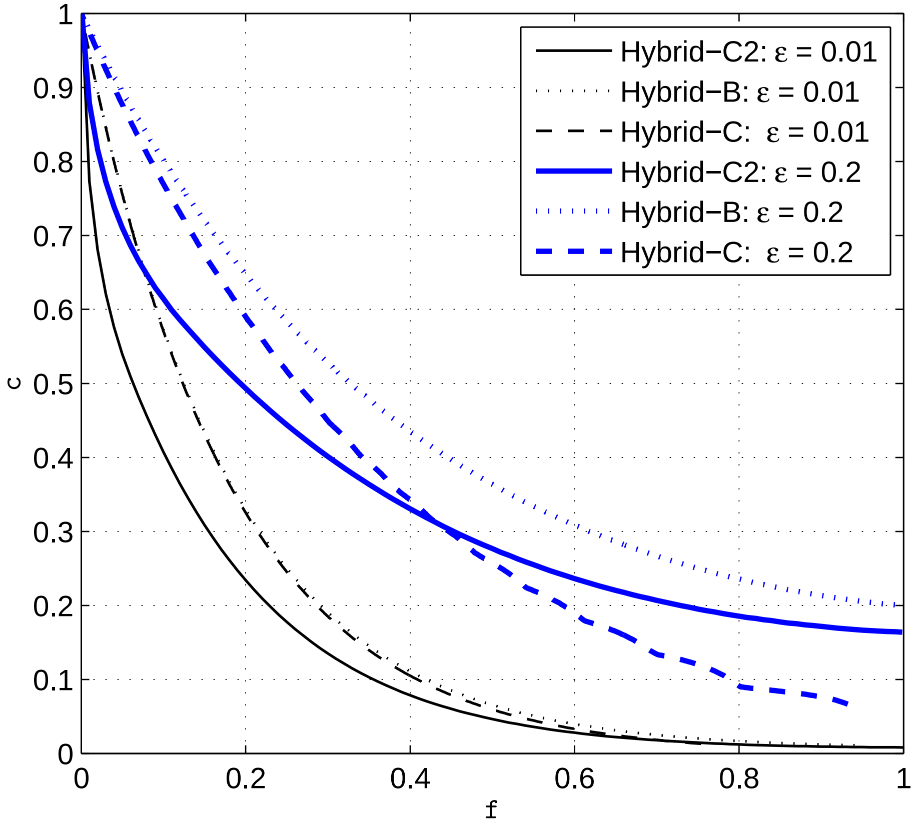


Figure 4.7. Performance of the hybrid-C2 protocol with right-regular d.d. pairs designed for $\delta = 0.01$

hybrid-C schemes are seen to be *complementary* to each other – the former works better at lower feedback and the latter at higher feedback.

CHAPTER 5

CAPACITY RESULTS FOR TWO WIRELESS RELAY NETWORKS WITH ERASURE LINKS

5.1 Introduction

In this chapter, we consider the problem of reliable communication in two classes of wireless relay networks – the multiple-access relay channel (MARC) and the multiple-relay channel (MRC) – in the context when individual links in these networks behave as memoryless erasure channels. In the M-source MARC (Fig. 5.1) , sources $s_1, \ldots, s_M$ convey information to the destination d with the aid of a relay r. In the M-relay MRC (Fig. 5.2), relays $r_1, \ldots, r_M$ assist a single source s in conveying information to a single destination d.

Since we deal with wireless networks, the medium is shared by the transmitting nodes. We assume that this sharing is done in an *orthogonal* fashion, i.e., transmissions from different nodes are non-interfering. Our model of wireless erasure networks is influenced by similar models adopted in prior work, cf. [31] and [32].

We enumerate cut-set outer bounds on the capacity regions for these networks and demonstrate their achievability (under a side-information assumption), using practical codes for the point-to-point erasure channel. The result is an *explicit* characterization of the network's capacity (region) - a characterization that of-

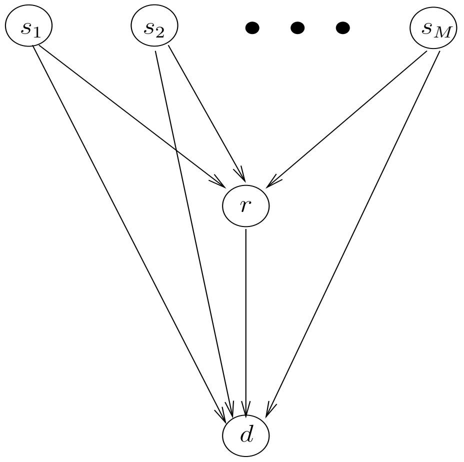


Figure 5.1. The M-source multiple-access relay channel

fers useful insights into the networks' optimal mode of operation under different scenarios.

Recall that, the capacity results of [31] are derived for *arbitrary* wireless erasure networks, and hence can be specialized to the MARC and MRC configurations considered here. However, as noted earlier, in [31] the available bandwidth is allocated equally among all transmitting nodes. In contrast, we do not pre-determine the allocation of the wireless medium – rather, this extra degree of freedom is exploited in selecting the bandwidth-allocation strategy that minimizes the total usage of the wireless medium.

Consequently, the region of all achievable rates obtained here is a superset of the region that follows from the results of [31]. Besides characterizing the optimal bandwidth allocation strategy, our approach also allows us to obtain useful insights regarding the *utility* of the relays under different channel conditions, for both the MARC and the MRC. Finally, as noted before, our achievability results are demonstrated using low-complexity capacity-achieving (c.a.) codes designed for

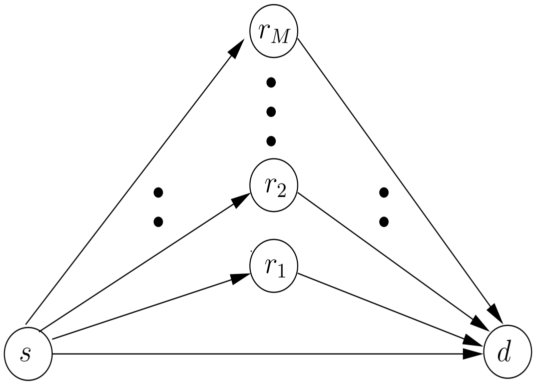


Figure 5.2. The M-relay multiple-relay channel

the BEC (such as LDPC codes), in contrast with the random coding arguments employed in [31].

5.1.1 Description of the channel model

As noted above, we assume that the transmitting nodes share the wireless medium orthogonally. Specifically, a **time-division-multiplexing** (TDM) strategy is adopted wherein medium access is restricted to non-overlapping *time slots*, with at most one node transmitting in any given slot. Further, we assume that each time slot carries exactly one bit.

The TDM assumption is merely for convenience of exposition and the results may be suitably "re-derived" for other orthogonal medium access protocols. Also, though we have assumed that the nodes transmit bits rather than packets, our results can easily be generalized to packetized transmission by treating packets as binary strings of fixed-length, and allowing one packet transmission per time-slot. Further, bit-level XORs in the encoding process may be replaced with packet-level XORs.

The link between any two nodes in the network is a **memoryless** binary erasure channel (BEC) of known erasure probability. The erasure random processes of different links are assumed to be independent of one other. Also, as noted above, all the outgoing links of a given node constitute a **broadcast** channel, i.e., the same bit is transmitted on all the outgoing links.

Further, we assume that the destination d in both networks has access to perfect **side-information** about the *erasure outcomes* on all links that occur over the course of communication. For example, in the MRC, d knows exactly which subset of relays received each transmission from s without erasures. In practice, this side-information can be conveyed to d by the intermediate nodes (relays), which can encode this information into the header of each received packet and forward it to d, as outlined in [31]. For a sufficiently large packet-size, this constitutes negligible overhead.

Finally, as depicted in Figure 5.1, the sources in the MARC cannot listen to each others' transmissions. Likewise, the relay's transmissions are "inaudible" to the sources. The same holds true for the MRC; further, each relay's transmissions cannot be heard by the other relays, as conveyed by Figure 5.2. Effectively, the relays in the MRC operate in a parallel, independent fashion.

5.2 The erasure multiple-access relay channel

This section derives the capacity region of the erasure MARC with M sources ($\mathcal{C}_{MARC}$) under the assumptions formulated in Section 5.1.1. Specifically, a closed-form description for $\mathcal{C}_{MARC}$ is obtained as a function of the channel parameters.

We first establish an outer bound $\mathcal{C}_{OB}$ to the capacity region (i.e., $\mathcal{C}_{MARC} \subseteq \mathcal{C}_{OB}$) via standard cut-set arguments. The resulting cut-set bounds are cast as a problem in *linear programming*, which yields a closed-form description of the boundary of $\mathcal{C}_{OB}$ as its solution. We then describe a coding scheme that achieves all rate-combinations in $\mathcal{C}_{OB}$, making use of capacity-achieving (c.a.) codes for the point-to-point erasure channel.

5.2.1 Preliminaries

Consider the M-source erasure MARC in Figure 5.1, wherein each link is a memoryless BEC with the following erasure probabilities:

- ϵ_{id} for the link from source s_i to the destination d (for $i = 1, 2, \ldots M$);
- ϵ_{ir} for the link from s_i to the relay r (for $i = 1, 2, \ldots M$);
- ϵ_{rd} for the link from r to d.

A *code* for the M-source erasure MARC consists of:

- one encoder at each source, with the encoder at s_i mapping k_i information bits onto n_i channel bits – i.e., $f_i : \{0, 1\}^{k_i} \rightarrow \{0, 1\}^{n_i}$ for $i = 1, 2, \ldots, M$.
- one encoder at the relay that accepts the (potentially erased) bits from each source and produces an n_r-bit codeword - i.e., $f_r : \{0, 1, E\}^{n_1+n_2+\cdots+n_M} \rightarrow \{0, 1\}^{n_r}$.
- a decoder at the destination that estimates the bits produced at each source – i.e., $g : \{0, 1, E\}^n \rightarrow \{0, 1\}^{k_1} \times \{0, 1\}^{k_2} \times \cdots \times \{0, 1\}^{k_M}$, where $n = n_1 + n_2 + \cdots + n_M + n_r$ is the total number of bits received by the destination. (Note: In keeping with the assumption about side information, we assume that $g(\cdot)$ may depend on the locations of the erasures on the source-to-relay channels.)

The rate of information transfer from s_i to d is $R_i \triangleq k_i/n$, and so associated with each code is a rate M-tuple $(R_1, R_2, \cdots R_M)$. The capacity region can then be formally defined as follows.

Definition 1. *A rate M-tuple $(R_1, R_2, \cdots R_M)$ is* **achievable** *on the erasure MARC if there exists a sequence of codes with increasing n such that, for n sufficiently large, the rate of information transfer from s_i to d can be made arbitrarily close to R_i for each $i = 1, \ldots M$, and the probability of decoder failure can be made arbitrarily small.*

The **capacity region** *$\mathcal{C}_{MARC}$ of the erasure MARC is the set of all achievable rate-tuples.*

The capacity region may alternatively be represented using the coordinates $(R, \boldsymbol{\theta})$, where $R = \sum_{i=1}^{M} R_i$, $\boldsymbol{\theta}= (\theta_1, \cdots, \theta_M)$ and $\theta_i = R_i/R$, for $i = 1, \cdots, M$. In terms of $\{k_i\}_{i=1}^{M}$ and n, we have $R = \sum_{i=1}^{M} k_i/n$, and $\theta_i = k_i/\sum_{j=1}^{M} k_j$. Here, R is the "sum-rate" of the code, and $\boldsymbol{\theta}$ is the "rate-allocation" vector describing how the sum-rate is divided among the sources. Note that $\boldsymbol{\theta}$ can be any real vector satisfying (i) $\theta_i \geq 0$, (ii) $\sum_{i=1}^{M} \theta_i = 1$. Thus, the region $\mathcal{C}_{MARC}$ is equivalently described by the set of all valid rate-allocation vectors $\boldsymbol{\theta}$ and the range of achievable sum-rates R associated with each $\boldsymbol{\theta}$. In the ensuing sections, we derive an outer bound $\mathcal{C}_{OB}$ to $\mathcal{C}_{MARC}$ by computing an upper bound for the sum-rate R for each rate-allocation vector $\boldsymbol{\theta}$.

We now develop the cut-set bounds used for the remainder of the section. Define an $\mathcal{S} - \mathcal{D}$ *cut* of the MARC to be a partition of the nodes of the MARC into two sets $\mathcal{S}$ and $\mathcal{D}$ $(= \mathcal{S}^c)$, such that at least one source is contained in $\mathcal{S}$ and the destination d is contained in $\mathcal{D}$. Let $k_{\mathcal{S}}$ denote the total information content of all messages originating in $\mathcal{S}$, i.e., $k_{\mathcal{S}} = \sum_{i:s_i \in \mathcal{S}} k_i$. Further, let the random variable $N_{\mathcal{S}}$ denote the total number of bits transmitted *successfully* from $\mathcal{S}$ to $\mathcal{D}$ – i.e., the total number of bits transmitted by nodes in $\mathcal{S}$ that are received *unerased* by at least one node in $\mathcal{D}$. Finally, let $p_e^{(n)}(\mathcal{S})$ denote the probability that d is unable to decode all the messages originating in $\mathcal{S}$. Then, we have the following result.

Lemma 1. *Given $\{k_i\}_{i=1}^M$ and an $\mathcal{S} - \mathcal{D}$ cut of the MARC, if $\{n_i\}_{i=1}^M$ and n_r are chosen such that $\mathbb{E}(N_{\mathcal{S}}) < k_{\mathcal{S}}$, then there exists a constant $\beta > 0$ such that $p_e^{(n)}(\mathcal{S}) \geq \beta$, where β is independent of $\{k_i\}_{i=1}^M$, $\{n_i\}_{i=1}^M$ and n_r.*

Proof. See Appendix E. □

Thus, in order to achieve arbitrarily small probabilities of decoding failure, it is necessary to choose $\{n_i\}_{i=1}^M$ and n_r such that $\mathbb{E}(N_{\mathcal{S}}) \geq k_{\mathcal{S}}$ for every $\mathcal{S} - \mathcal{D}$ cut.

5.2.2 Outer bound on the capacity region of the two-source erasure MARC

Prior to solving for $\mathcal{C}_{OB}$ for the M-source MARC, we illustrate this procedure for the two-source case, shown in Figure 5.3. Let k_1 and k_2 denote the information content at sources s_1 and s_2, and let n_1, n_2 and n_r denote the number of bits transmitted by s_1, s_2 and r, respectively.

We characterize $\mathcal{C}_{OB}$ via an *upper bound* on the sum-rate $R = (k_1 + k_2)/n$ for each rate allocation $\boldsymbol{\theta}$. This corresponds to computing a *lower bound* on the total number of transmissions $n = n_1 + n_2 + n_r$ required for fixed values of k_1 and k_2. Note that, given k_1 and k_2, (n_1, n_2, n_r) are subject to the cut-set bounds implied by Lemma 1.

The cuts that we consider for the two-source MARC are shown in Figure 5.3. For instance, cut C_1 defines the partition $\mathcal{S} = \{s_1\}$ and $\mathcal{D} = \{s_2, r, d\}$. For this

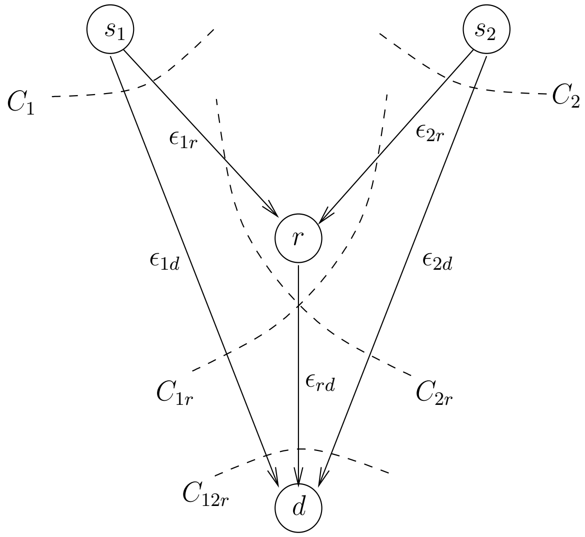


Figure 5.3. The two-source erasure MARC

case, $\mathbb{E}(N_{\mathcal{S}}) = n_1(1-\epsilon_{1d}\epsilon_{1r})$[1]. From Lemma 1, in order for d to decode s_1's message with high probability, it is necessary that $n_1(1-\epsilon_{1d}\epsilon_{1r}) \;\geq\; k_1$. Similar bounds can be derived by applying Lemma 1 to the remaining cuts, as given below:

$$\begin{array}{lcrl}
C_1 : & n_1(1-\epsilon_{1r}\epsilon_{1d}) & \geq & k_1 \\
C_2 : & n_2(1-\epsilon_{2r}\epsilon_{2d}) & \geq & k_2 \\
C_{1r} : & n_1(1-\epsilon_{1d}) + n_r(1-\epsilon_{rd}) & \geq & k_1 \\
C_{2r} : & n_2(1-\epsilon_{2d}) + n_r(1-\epsilon_{rd}) & \geq & k_2 \\
C_{12r} : & n_1(1-\epsilon_{1d}) + n_2(1-\epsilon_{2d}) + n_r(1-\epsilon_{rd}) & \geq & k_1 + k_2
\end{array}$$

Note that, in addition to the cuts in Figure 5.3, the cut C_{12} is also possible, resulting in $\mathcal{S} = \{s_1, s_2\}$ and $\mathcal{D} = \{r, d\}$. However, the corresponding cut-set bound $n_1(1-\epsilon_{1r}\epsilon_{1d}) + n_2(1-\epsilon_{2r}\epsilon_{2d}) \;\geq\; k_1 + k_2$ is already implied by the bounds

[1] Note that $1-\epsilon_{1d}\epsilon_{1r}$ is the probability that no bit is erased simultaneously on both $s_1 \rightarrow r$ and $s_1 \rightarrow d$ (cf. the model proposed in [31])

derived from C_1 and C_2. Hence, we do not explicitly consider C_{12}. Analogously, for the M-source case, we shall not consider cuts that result in $\mathcal{S}$ containing only multiple sources (and no relay).

In addition to the cut-set bounds and the implied non-negativity of n_1 and n_2, we also have the non-negativity constraint on n_r:

$$N_r: \quad n_r \;\geq\; 0$$

We now minimize $n = n_1 + n_2 + n_r$ subject to the above necessary conditions. This has the form of a standard problem in linear programming (LP).

In particular, let $\mathcal{N} \subset \mathbb{R}^3$ denote the set of (n_1, n_2, n_3) that satisfy the above constraints; geometrically, $\mathcal{N}$ is a convex 3-dimensional polytope. It follows from LP theory that the minimum value of n is attained at one of the *vertices* of $\mathcal{N}$ [44]. Each vertex of $\mathcal{N}$ is completely determined by specifying three adjacent *faces* of $\mathcal{N}$; there are a total of 6 faces for $\mathcal{N}$, each corresponding to one of the constraints being met with equality. Thus, there are $\binom{6}{3} = 20$ possible combinations that *may* intersect to give a vertex; of these, it can be verified that only the following four[2] actually form vertices of $\mathcal{N}$: $C_1C_2C_{12r}$, $C_1C_{1r}C_{12r}$, $C_2C_{2r}C_{12r}$, and $C_{1r}C_{2r}N_r$.

Thus, it suffices to evaluate $n_1 + n_2 + n_r$ at these four vertices to determine the minimum. Note that the optimum vertex, i.e., the optimum (n_1, n_2, n_r), is a function of the channel parameters (i.e., erasure probabilities); in Section 5.2.3, we illustrate a procedure that permits us to characterize this function *explicitly* for the M-source case, using standard tools from LP theory. The results are outlined below for the two-source case:

[2] The notation '$C_iC_jN_k$' means that the constraints labeled C_i, C_j and N_k are met with equality.

- $\epsilon_{rd} < \min(\epsilon_{1d}, \epsilon_{2d})$: The minimum is attained at $C_1C_2C_{12r}$ and the resulting values of n_1, n_2 and n_r are

$$\begin{aligned} n_1^* &= \frac{k_1}{1-\epsilon_{1r}\epsilon_{1d}}, \\ n_2^* &= \frac{k_2}{1-\epsilon_{2r}\epsilon_{2d}}, \\ n_r^* &= \frac{\epsilon_{1d}(1-\epsilon_{1r})}{1-\epsilon_{rd}} n_1^* + \frac{\epsilon_{2d}(1-\epsilon_{2r})}{1-\epsilon_{rd}} n_2^*. \end{aligned} \tag{5.1}$$

- $\epsilon_{2d} < \epsilon_{rd} < \epsilon_{1d}$: The minimum is attained at $C_1C_{1r}C_{12r}$ and

$$\begin{aligned} n_1^* &= \frac{k_1}{1-\epsilon_{1r}\epsilon_{1d}}, \\ n_2^* &= \frac{k_2}{1-\epsilon_{2d}}, \\ n_r^* &= \frac{\epsilon_{1d}(1-\epsilon_{1r})}{1-\epsilon_{rd}} n_1^*. \end{aligned} \tag{5.2}$$

  For the case $\epsilon_{1d} < \epsilon_{rd} < \epsilon_{2d}$, the minimum is attained at $C_2C_{2r}C_{12r}$ and the corresponding n_1^*, n_2^* and n_r^* are obtained by symmetry.

- $\epsilon_{rd} > \max(\epsilon_{1d}, \epsilon_{2d})$ The minimum is attained at $C_{1r}C_{2r}N_r$:

$$\begin{aligned} n_1^* &= \frac{k_1}{1-\epsilon_{1d}}, \\ n_2^* &= \frac{k_2}{1-\epsilon_{2d}}, \\ n_r^* &= 0. \end{aligned} \tag{5.3}$$

Note that for the last case (wherein the $r \to d$ link is worse than either of the source-destination links), the minimum value of n is attained by setting $n_r = 0$, i.e., by not using the relay at all. Similarly, when $\epsilon_{2d} < \epsilon_{rd} < \epsilon_{1d}$, the optimal choice for n_r depends only on k_1 (via n_1), suggesting that the relay should assist only that source with a worse link to d than r itself. This principle holds in general, as will be shown in Section 5.2.3.

We conclude this section with a description of $\mathcal{C}_{OB}$ for the case when $\epsilon_{rd} < \min(\epsilon_{1d}, \epsilon_{2d})$, in terms of the sum-rate R and rate allocation $\boldsymbol{\theta}$. Let $n^* = n_1^* + n_2^* + n_r^*$. Then $R \leq (k_1 + k_2)/n^*$, and, noting that $\theta_i = k_i/(k_1 + k_2)$ $(i = 1, 2)$, we have:

$$\begin{aligned} R &\leq \frac{k_1 + k_2}{\sum_{i=1}^{2} \frac{k_i}{1 - \epsilon_{ir}\epsilon_{id}} \cdot \left(1 + \frac{\epsilon_{id}(1 - \epsilon_{ir})}{1 - \epsilon_{rd}}\right)} \\ &= \frac{1}{\sum_{i=1}^{2} \frac{\theta_i}{1 - \epsilon_{ir}\epsilon_{id}} \cdot \left(1 + \frac{\epsilon_{id}(1 - \epsilon_{ir})}{1 - \epsilon_{rd}}\right)} \end{aligned} \tag{5.4}$$

Thus, $\mathcal{C}_{OB}$ consists of all $(R, \boldsymbol{\theta})$ that satisfy (5.4).

5.2.3 Outer bound for the M-source MARC

We now derive $\mathcal{C}_{OB}$ for the general M-source case. Let the set of sources be denoted $\mathsf{S} = \{1, 2, \cdots, M\}$. From Lemma 1, it follows that the following cut-set bounds must be satisfied, if d is to decode all messages with high probability:

$$n_r(1 - \epsilon_{rd}) + \sum_{j \in \mathcal{I}} n_j(1 - \epsilon_{jd}) \geq \sum_{j \in \mathcal{I}} k_j, \quad \forall\, \mathcal{I} \subseteq \mathsf{S}, \tag{5.5}$$

$$n_i(1 - \epsilon_{ir}\epsilon_{id}) \geq k_i, \quad \forall\, i \in \mathsf{S}. \tag{5.6}$$

Let $\mathsf{S}' \subseteq \mathsf{S}$ comprise the sources with a *poorer* link to d than the relay-to-destination link, i.e., $\mathsf{S}' \triangleq \{j \in \mathsf{S} : \epsilon_{jd} > \epsilon_{rd}\}$. Then, the following result holds.

Lemma 2. *The minimization of* $n = n_r + \sum_{i \in \mathsf{S}} n_i$ *subject to* (5.5) *and* (5.6) *yields* *the following solution:*

$$n_i^* = \begin{cases} \dfrac{k_i}{1-\epsilon_{id}\epsilon_{ir}}, & i \in \mathsf{S}', \\ \dfrac{k_i}{1-\epsilon_{id}}, & i \notin \mathsf{S}', \end{cases} \tag{5.7}$$

$$\begin{aligned} n_r^* &= \frac{1}{1-\epsilon_{rd}} \cdot \sum_{i \in \mathsf{S}'} n_i^* \epsilon_{id}(1-\epsilon_{ir}) \\ &= \frac{1}{1-\epsilon_{rd}} \cdot \sum_{i \in \mathsf{S}'} k_i \cdot \frac{\epsilon_{id}(1-\epsilon_{ir})}{1-\epsilon_{id}\epsilon_{ir}}. \end{aligned} \tag{5.8}$$

Proof. For convenience, we assume that the sources are numbered in *decreasing* order of source-destination erasure probabilities, i.e, if $i < j$ then $\epsilon_{id} \geq \epsilon_{jd}$. Also, let $M' = \max\{j : \epsilon_{jd} > \epsilon_{rd}\}$, so $\mathsf{S}' = \{1, 2, \cdots, M'\}$.

We first demonstrate that (5.7) and (5.8) are *feasible* - i.e., they satisfy (5.5) and (5.6). It is easily seen that (5.6) is not violated by n_i^* chosen according to (5.7). Further, for any $\mathcal{I} \subseteq \mathcal{S}$, let $\mathcal{I}' = \mathcal{I} \cap \mathsf{S}'$. Then,

$$\begin{aligned} n_r^*(1-\epsilon_{rd}) + \sum_{j \in \mathcal{I}} n_j^*(1-\epsilon_{jd}) &= \sum_{i \in \mathsf{S}'} n_i^* \epsilon_{id}(1-\epsilon_{ir}) + \sum_{j \in \mathcal{I}} n_j^*(1-\epsilon_{jd}) \\ &\geq \sum_{i \in \mathcal{I}'} n_i^* \epsilon_{id}(1-\epsilon_{ir}) + \sum_{j \in \mathcal{I}} n_j^*(1-\epsilon_{jd}) \\ &= \sum_{i \in \mathcal{I}'} n_i^*(1-\epsilon_{id}\epsilon_{ir}) + \sum_{j \in \mathcal{I} \backslash \mathcal{I}'} n_j^*(1-\epsilon_{jd}) \\ &= \sum_{i \in \mathcal{I}'} k_i + \sum_{j \in \mathcal{I} \backslash \mathcal{I}'} k_j \end{aligned} \tag{5.9}$$

which shows that (5.5) is satisfied.

The constraints (5.5) and (5.6) describe a convex $(M+1)$-dimensional polytope $\mathcal{P}_{MARC}$. From LP theory, the minimum of $n = \sum_{i \in \mathsf{S}} n_i + n_r$ occurs at one of the vertices of $\mathcal{P}_{MARC}$, where a vertex is specified by setting $M+1$ of the constraints in (5.5) and (5.6) to hold with equality.

Consider the following subset of $M+1$ constraints:

$$\begin{aligned} n_i(1-\epsilon_{ir}\epsilon_{id}) &\geq k_i, \qquad 1 \leq i \leq M', &(5.10)\\ n_r(1-\epsilon_{rd}) + \sum_{j=1}^{M'+m} n_j(1-\epsilon_{jd}) &\geq \sum_{j=1}^{M'+m} k_j, \quad 0 \leq m \leq M-M'. &(5.11)\end{aligned}$$

We shall demonstrate that n is minimized when these constraints are met with equality. We first transform (5.10) and (5.11) to equalities by introducing non-negative *slack* variables $\{X_i\}_{1\leq i\leq M'}$, $\{Z_m\}_{0\leq m\leq M-M'}$:

$$\begin{aligned} n_i(1-\epsilon_{ir}\epsilon_{id}) &= k_i + X_i, \qquad 1 \leq i \leq M', &(5.12)\\ n_r(1-\epsilon_{rd}) + \sum_{j=1}^{M'+m} n_j(1-\epsilon_{jd}) &= \sum_{j=1}^{M'+m} k_j + Z_m, \quad 0 \leq m \leq M-M'. &(5.13)\end{aligned}$$

For this system of equations, observe that the coefficient matrix of the vector $(n_1, \cdots, n_{M'}, n_r, n_{M'+1}, \cdots, n_M)$ is *lower triangular*. Thus, (5.12) and (5.13) can be easily solved for $\{n_i\}_{i\in\mathsf{S}}$ and n_r, yielding:

$$n_i = \begin{cases} \dfrac{k_i}{1-\epsilon_{ir}\epsilon_{id}} + \dfrac{X_i}{1-\epsilon_{ir}\epsilon_{id}}, & 1 \leq i \leq M' \\ \dfrac{k_i}{1-\epsilon_{id}} + \dfrac{Z_{i-M'} - Z_{i-M'-1}}{1-\epsilon_{id}}, & M'+1 \leq i \leq M \end{cases} \qquad (5.14)$$

$$\begin{aligned} n_r &= \frac{1}{1-\epsilon_{rd}} \sum_{j=1}^{M'} \frac{k_j\epsilon_{jd}(1-\epsilon_{jr})}{1-\epsilon_{jr}\epsilon_{jd}} + \frac{Z_0}{1-\epsilon_{rd}} \\ &\quad - \sum_{j=1}^{M'} \frac{X_j}{1-\epsilon_{jr}\epsilon_{jd}} \cdot \frac{1-\epsilon_{jd}}{1-\epsilon_{rd}}. \qquad (5.15)\end{aligned}$$

Then, we have:

$$\begin{aligned} n_r + \sum_{i=1}^{M} n_i \ &= \ f(k_1, k_2, \cdots, k_M) \ + \ \sum_{i=1}^{M'} \frac{X_i(\epsilon_{id} - \epsilon_{rd})}{(1-\epsilon_{ir}\epsilon_{id})(1-\epsilon_{rd})} \\ &\quad + \sum_{j=M'}^{M-1} \frac{Z_{j-M'}(\epsilon_{jd} - \epsilon_{(j+1)d})}{(1-\epsilon_{jd})(1-\epsilon_{(j+1)d})} \ + \ \frac{Z_{M-M'}}{1-\epsilon_{Md}} \end{aligned} \tag{5.16}$$

where $f(\cdots)$ is independent of the slack variables. Since $\epsilon_{id} \geq \epsilon_{jd}$ for $i < j$ and $\epsilon_{id} > \epsilon_{rd}$ for $i \leq M'$, the coefficients of $\{X_i\}_{1 \leq i \leq M'}$ and $\{Z_m\}_{0 \leq m \leq M-M'}$ in the above equation are non-negative. Thus, $n_r + \sum_{i=1}^{M} n_i$ is minimized by setting $\{X_i\}_{1 \leq i \leq M'}$ and $\{Z_m\}_{0 \leq m \leq M-M'}$ to zero, which yields the solution in (5.7) and (5.8).

□

Corollary 1. *For the M-source MARC, $\mathcal{C}_{MARC} \subseteq \mathcal{C}_{OB}$, where $\mathcal{C}_{OB}$ consists of all $(R, \boldsymbol{\theta})$ satisfying*

$$R \ \leq \ \frac{1}{\displaystyle\sum_{i \in \mathsf{S}'} \alpha_i \theta_i + \sum_{j \notin \mathsf{S}'} \beta_j \theta_j} \tag{5.17}$$

where

$$\alpha_i \ = \ \frac{1}{1 - \epsilon_{id}\epsilon_{ir}} \cdot \left(1 + \frac{\epsilon_{id}(1-\epsilon_{ir})}{1-\epsilon_{rd}}\right), \ i \in \mathsf{S}', \tag{5.18}$$

$$\beta_j \ = \ \frac{1}{1-\epsilon_{jd}}, \quad j \notin \mathsf{S}'. \tag{5.19}$$

5.2.4 Achieving the outer bound

We now demonstrate that every $(R, \boldsymbol{\theta})$ in $\mathcal{C}_{OB}$ is achievable by fixing $\boldsymbol{\theta}$ and showing that the upper bound on the sum-rate R in (5.17) is achievable. More precisely, given $\boldsymbol{\theta}$ and the total number of message bits k (with $k_i = \theta_i k$), it suffices to construct a coding scheme with $\{n_i\}_{i \in \mathsf{S}}$ and n_r given by Lemma 2 that achieves reliable transfer of the k message bits to d with high probability for sufficiently large k.

The strategy employed here is to encode at the sources with capacity-achieving (c.a.) codes for the BEC, followed by compress-and-forward (CF) encoding at the relay to convey only the *useful* portion of the relay's observations to d. As we have seen, in practice, c.a. codes for the BEC such as Tornado codes ([4]) can achieve rates within a factor $1-\delta$ of the capacity, where $\delta > 0$ is a design parameter that can be made arbitrarily small. So these codes have the property that, if the message blocklength k is sufficiently large, any $k/(1-\delta) \approx k(1+\delta)$ unerased code bits suffice to recover the message with high probability. For convenience, we assume all "capacity approaching codes" operate *at* capacity, i.e., with $\delta = 0$. Further, we replace the (random) number of successful (unerased) transmissions from any node to any group of nodes by the *expected value* of the same. These assumptions make the notation simpler and do not change the results, which address asymptotic rates.

The coding scheme is as follows. Each source s_i encodes and transmits its message consisting of k_i bits using a c.a. code of rate $\tilde{R}_i$ as follows:

- for $i \in \mathsf{S}'$, $\tilde{R}_i = 1 - \epsilon_{ir}\epsilon_{id}$, and $n_i = \dfrac{k_i}{1 - \epsilon_{ir}\epsilon_{id}}$,
- for $i \notin \mathsf{S}'$, $\tilde{R}_i = 1 - \epsilon_{id}$, and $n_i = \dfrac{k_i}{1 - \epsilon_{id}}$.

(Recall that S' indexes those sources with a worse direct link to the destination than that of the relay.)

For any $i \notin \mathsf{S}'$, the destination can (with high probability) decode s_i's message without assistance from the relay.

So assume $i \in \mathsf{S}'$; denote the number of code bits received *unerased* by r and d from s_i as n_{ir} and n_{id}, respectively. By assumption, $n_{ir} = n_i(1-\epsilon_{ir})$ and $n_{id} = n_i(1-\epsilon_{id})$. Since c.a. codes are assumed, any k_i unerased code bits from s_i suffice to decode the corresponding message with high probability. Hence, d needs an additional $k_i - n_{id} = \epsilon_{id} n_{ir}$ code bits to decode the message, a number equal to the number of code bits received by r from s_i but not by d. Let the vector $\boldsymbol{z_i}$ denote the code bits received by r. The side-information assumption implies that d knows a fraction $1-\epsilon_{id}$ of $\boldsymbol{z_i}$. The unknown portion of $\boldsymbol{z_i}$ can be communicated by r as follows.

As a first step, r compresses $\boldsymbol{z_i}$ by a factor ϵ_{id} such that any missing fraction ϵ_{id} of $\boldsymbol{z_i}$ can be recovered using the compressed bits; this can be accomplished in several ways – e.g., encoding $\boldsymbol{z_i}$ using the first layer of parities of a Tornado code results in a low-complexity compression scheme. The compressed version of $\boldsymbol{z_i}$ may now be communicated to d using a c.a. code of rate $1-\epsilon_{rd}$, designed for the $r \to d$ link. Thus, d can decode the missing portion of $\boldsymbol{z_i}$ and hence the original message.

For each $i \in \mathsf{S}'$, the number of bits transmitted by r in this process is $\epsilon_{id} n_{ir}/(1-\epsilon_{rd})$. Summing over all i in S' results in (5.8). Thus, we have demonstrated a coding technique to achieve all $(R, \boldsymbol{\theta})$ within $\mathcal{C}_{OB}$.

Note that, in the above scheme, the relay encodes each source in S' *independently.* A slightly modified encoding strategy at the relay is to compress the combined vector $[\boldsymbol{z_i}]_{i\in\mathsf{S}'}$ by a factor ϵ:

$$\epsilon = \frac{\sum_{i\in\mathsf{S}'} n_{ir}\epsilon_{id}}{\sum_{i\in\mathsf{S}'} n_{ir}}, \tag{5.20}$$

such that any missing fraction ϵ of the combined vector can be recovered from the compressed bits. These compressed bits can now be communicated to d and suffice to decode the missing portions of all $\boldsymbol{z_i}$. The obvious advantage of this approach is that the code bits received by the relay are jointly encoded into a longer codeword for the same total number of transmitted bits, ensuring lower probability of decoding error for finite blocklengths.

Thus, we have the following theorem.

Theorem 2 (Capacity region of the M-source erasure MARC)**.** *The capacity region $\mathcal{C}_{MARC}$ consists of all non-negative rate M-tuples $(R_1, \cdots, R_M)$ that satisfy*

$$\sum_{i \in \mathsf{S}'} \alpha_i R_i + \sum_{j \notin \mathsf{S}'} \beta_j R_j \leq 1, \tag{5.21}$$

for $\{\alpha_i\}_{i \in \mathsf{S}'}$ and $\{\beta_j\}_{j \notin \mathsf{S}'}$ defined in (5.18) *and* (5.19).

5.2.5 Comparison with earlier results

We now compare our results with those of [31] for the case of the two-source MARC. Recall that [31] derives the capacity region of arbitrary wireless networks with orthogonal erasure links of specified bandwidth; in particular, it is assumed that every link is a binary erasure channel[3] and that all links operate simultaneously. This effectively divides the available wireless spectrum *equally* among all transmitting nodes, in contrast with our optimal allocation strategy for the MARC based on the channel parameters.

So the rate region in [31] for the two-source MARC is contained within the region described in Theorem 2. In particular, let CR_1 denote the capacity region described by Theorem 2, and CR_2 the capacity region based on the results in [31,

[3] Alternatively, each link may be a packet erasure channel with a fixed packet size.

Theorem 2]; then CR_2 is the set of all rate pairs satisfying these constraints:

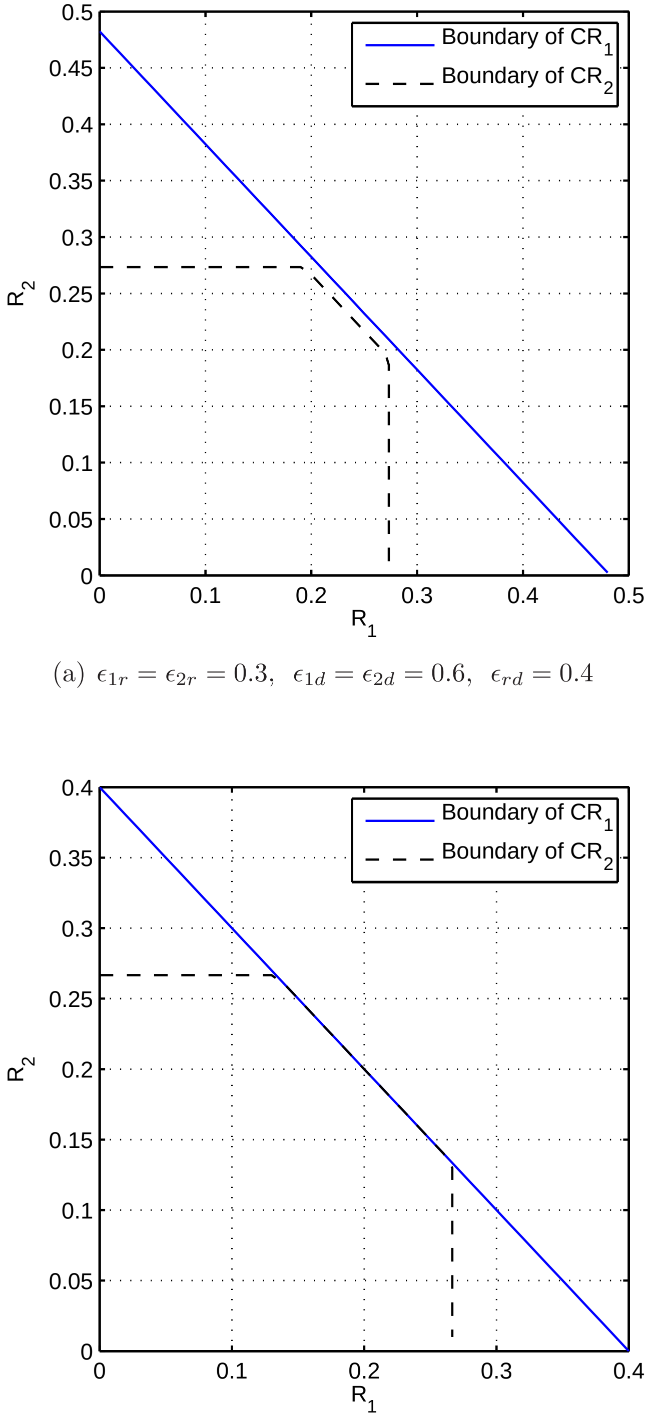
$$\begin{aligned}
R_1 &\leq \frac{1}{3} \cdot \min(1 - \epsilon_{1r}\epsilon_{1d}, 1 - \epsilon_{1d} + 1 - \epsilon_{rd}), \\
R_2 &\leq \frac{1}{3} \cdot \min(1 - \epsilon_{2r}\epsilon_{2d}, 1 - \epsilon_{2d} + 1 - \epsilon_{rd}), \\
R_1 + R_2 &\leq \frac{1}{3} \cdot \Big(1 - \epsilon_{1d} \ + \ 1 - \epsilon_{2d} \ + \ 1 - \epsilon_{rd}\Big),
\end{aligned}$$

where we have included a factor of 1/3 in each bound to take into account the fact that one-third of the total bandwidth is allocated to each of s_1, s_2 and r - i.e., each node has access to only a third of all channel uses.

Figure 5.4 compares CR_1 and CR_2 for two different cases. When $\epsilon_{1r} = \epsilon_{2r} = 0.3$, $\epsilon_{1d} = \epsilon_{2d} = 0.6$ and $\epsilon_{rd} = 0.4$, it is seen that CR_2 is strictly contained within CR_1 – the resulting “gap” implying that equally apportioning bandwidth among the three transmitters (at the two sources and the relay) is never optimal in this scenario. On the other hand, when ϵ_{rd} is increased to 0.6 to equal the source-destination erasure rates, the boundaries of CR_1 and CR_2 coincide in part; in this regime, it is optimal for the three transmitting nodes to share the medium equally.

More generally, for a symmetric two-source MARC with $\epsilon_{1d} = \epsilon_{2d} = \epsilon_{sd}$ and $\epsilon_{1r} = \epsilon_{2r} = \epsilon_{sr}$, it can be shown that equal allocation of bandwidth among the transmitting nodes can result in the relay being either “over-utilized” or “under-utilized.” In particular, the relay is over-utilized, i.e., allocated too many channel uses, if either of the following conditions hold:

- the $r \rightarrow d$ link is of poorer quality than the source-destination links, i.e., $\epsilon_{rd} > \epsilon_{sd}$; or
- the maximum rate at which r can communicate with d exceeds the maximum



(a) $\epsilon_{1r} = \epsilon_{2r} = 0.3, \;\; \epsilon_{1d} = \epsilon_{2d} = 0.6, \;\; \epsilon_{rd} = 0.4$

(b) $\epsilon_{1r} = \epsilon_{2r} = 0.3, \quad \epsilon_{1d} = \epsilon_{2d} = \epsilon_{rd} = 0.6$

Figure 5.4. A comparison of the rate regions CR_1 (with optimal bandwidth allocation) and CR_2 (without optimal bandwidth allocation) for the two-source MARC.

rate at which r can receive *useful* information from the two sources, i.e., $1 - \epsilon_{rd} > 2\epsilon_{sd}(1 - \epsilon_{sr})$.

Conversely, the relay is under-utilized if both $\epsilon_{rd} < \epsilon_{sd}$ and $1 - \epsilon_{rd} < 2\epsilon_{sd}(1 - \epsilon_{sr})$. In sum, equal bandwidth allocation can be optimal *only if* either $\epsilon_{rd} = \epsilon_{sd}$ or $1 - \epsilon_{rd} = 2\epsilon_{sd}(1 - \epsilon_{sr})$.

5.3 The erasure multiple relay channel

We now compute the capacity of the erasure multiple relay channel (MRC) with M relays. As with the MARC, an upper bound is first established using cut-set arguments; this bound is then shown to be achievable using capacity approaching codes for the BEC. Finally, the capacity result is discussed for specific examples of practical interest.

5.3.1 Preliminaries

Consider the M-relay erasure MRC in Figure 5.2. Each link is a memoryless BEC with erasure probabilities given by:

- ϵ_{sd} for the link from the source s to the destination d;
- ϵ_{si} for the link from the source s to relay r_i (for $i = 1, 2, \ldots M$);
- ϵ_{id} for the link from relay r_i to the destination d (for $i = 1, 2, \ldots M$).

We shall see that the relay-destination erasure probabilities play a crucial role in determining the best communication strategy. So, for convenience of exposition, we assume that the relays are numbered in **non-decreasing order** of their relay-destination probabilities, i.e., for $i < j$, we have $\epsilon_{id} \leq \epsilon_{jd}$.

We now define a code for this channel. An encoder $f_s : \{0,1\}^k \to \{0,1\}^{n_s}$ maps k message bits at the source onto n_s channel bits; each relay r_i observes a (possibly erased) version of those channel bits and implements an encoder $f_i : \{0,1,E\}^{n_s} \to \{0,1\}^{n_i}$. Then, the destination implements a decoder $g : \{0,1,E\}^n \to \{0,1\}^k$ to recover the k-bit message based on the $n = n_s + \sum_{i=1}^{M} n_i$ received bits[4]. The rate of this code is k/n, and a rate $R > 0$ is said to be achievable if there exists a sequence of codes with increasing n with rates approaching R and vanishing probability of decoder failure. The capacity C_{MRC} is the supremum of all achievable rates.

Proceeding to the cut-set bounds, define an $\mathcal{S}-\mathcal{D}$ cut of the MRC as a partition of its nodes into sets $\mathcal{S}$ and $\mathcal{D}$ $(= \mathcal{S}^c)$, such that $s \in \mathcal{S}$ and $d \in \mathcal{D}$. Let $N_{\mathcal{S}}$ denote a random variable indicating the total number of bits transmitted successfully (without erasure) from $\mathcal{S}$ to $\mathcal{D}$, and $p_e^{(n)}$ the probability that d is unable to decode the k message bits. Then, Lemma 1 extends in a straightforward manner to the MRC, yielding this result.

Lemma 3. *Given an $\mathcal{S} - \mathcal{D}$ cut of the MRC, if n_s and $\{n_i\}_{i=1}^{M}$ are chosen such that $\mathbb{E}(N_{\mathcal{S}}) < k$, then there exists $\beta > 0$, independent of k, n_s and $\{n_i\}_{i=1}^{M}$, such that $p_e^{(n)} \geq \beta$.*

5.3.2 Upper bound on the capacity of the erasure MRC

We now derive an upper bound on C_{MRC} using Lemma 3. For fixed k, we compute a lower bound on the total number of transmissions n required for reliable communication, resulting in an upper bound on any achievable rate.

Let the set of relays be denoted $\mathcal{R} = \{1, 2, \cdots, M\}$. From Lemma 3, to achieve an arbitrarily small probability of decoding failure, it is necessary that n_s

[4] Once again, the side information assumption means that $g(\cdot)$ is also a function of the erasure locations on the source-to-relay links.

and $\{n_i\}_{i=1}^M$ satisfy $\mathbb{E}(N_{\mathcal{S}}) \geq k$ for *every* $\mathcal{S} - \mathcal{D}$ cut of the MRC. This translates to the following set of conditions:

$$n_s\Big(1 - \epsilon_{sd} \prod_{i \in \mathcal{I}^c} \epsilon_{si}\Big) + \sum_{j \in \mathcal{I}} n_j(1 - \epsilon_{jd}) \;\geq\; k, \quad \forall\, \mathcal{I} \subseteq \mathcal{R}. \tag{5.22}$$

In addition to the above, there are M non-negativity constraints:

$$n_m \geq 0, \quad 1 \leq m \leq M. \tag{5.23}$$

As before, minimizing $n = n_s + \sum_{i \in \mathcal{R}} n_i$ subject to (5.22) and (5.23) is a standard problem in LP. The convex polytope $\mathcal{P}_{MRC} \subset \mathbb{R}^{M+1}$ described by (5.22) and (5.23) has *at most* $\binom{2^M + M}{M+1}$ possible vertices – each determined by solving a subset of $M+1$ constraints with equality. In the following, we determine which of these vertices minimizes n as a function of the channel parameters.

In particular, consider the class of $(M+1)$-tuples $(n_s, n_1, \cdots, n_M) \in \mathbb{R}^{M+1}$ obtained as follows: for every $\ell = 0, 1 \ldots, M$, satisfy (5.23) with equality for $m \in \{\ell + 1, \cdots, M\}$ and satisfy (5.22) with equality for all $\mathcal{I}$ belonging to the set

$$\Big\{\{1, 2, \ldots, M\}, \{2, \ldots, M\}, \{3, \ldots, M\}, \ldots, \{\ell + 1, \ldots, M\}\Big\}.$$

The cuts corresponding to these choices of $\mathcal{I}$ are illustrated in Figure 5.5. The resulting set of points in $\mathbb{R}^{M+1}$ constitute $M+1$ vertices of $\mathcal{P}_{MRC}$, and the minimum value of n occurs at one of these vertices. We formalize this result in the following.

Definition 2. *For a given message length k, a* **channel allocation** *is an $(M+1)$-tuple of non-negative real numbers $(n_s, n_1, \cdots, n_M)$. The* **rate** *of a channel*

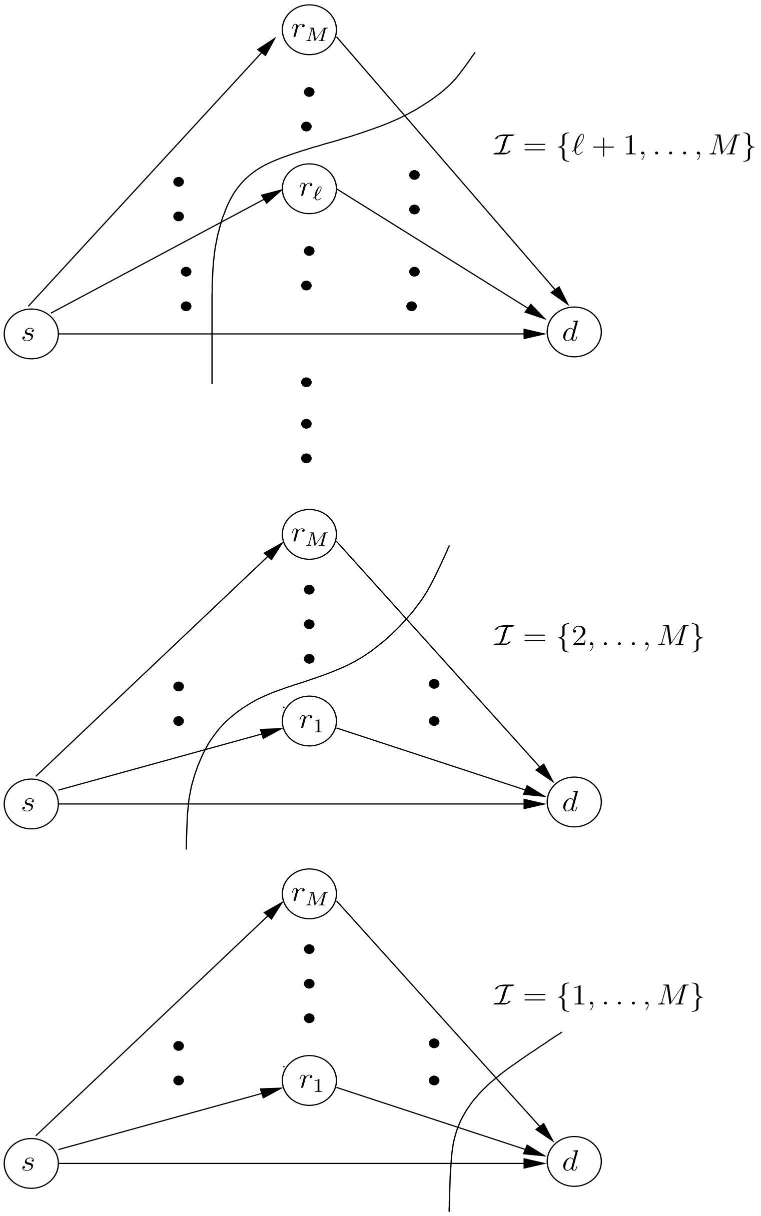

r_M
r_ℓ
s
d
$\mathcal{I} = \{\ell + 1, \ldots, M\}$
r_M
r_1
s
d
$\mathcal{I} = \{2, \ldots, M\}$
r_M
r_1
s
d
$\mathcal{I} = \{1, \ldots, M\}$

Figure 5.5. Cuts corresponding to the α_ℓ-channel allocation

allocation is the ratio $k/(n_s + \sum_{i=1}^{M} n_i)$.

For $0 \leq \ell \leq M$, *an* $\boldsymbol{\alpha_\ell}$**-channel allocation** *is defined as the* $(M+1)$*-tuple* $(n_s, n_1, \cdots, n_M)$ *satisfying these equalities:*

$$\begin{aligned} n_s\Big(1 - \epsilon_{sd} \prod_{j=1}^{m} \epsilon_{sj}\Big) + \sum_{j=m+1}^{M} n_j(1-\epsilon_{jd}) &= k, \quad 0 \leq m \leq \ell, & (5.24) \\ n_m &= 0, \quad \ell < m \leq M. & (5.25) \end{aligned}$$

Lemma 4. *For* $0 \leq \ell \leq M$, *the* α_ℓ*-channel allocation is given by:*

$$\begin{aligned} n_s &= \frac{k}{1 - \epsilon_{sd} \prod_{i=1}^{\ell} \epsilon_{si}}, \\ n_m(1-\epsilon_{md}) &= n_s \epsilon_{sd}(1-\epsilon_{sm}) \prod_{i=1}^{m-1} \epsilon_{si}, \quad 1 \leq m \leq \ell, \\ n_m &= 0, \quad m > \ell, \end{aligned} \tag{5.26}$$

with rate given by

$$C_\ell = \frac{1 - \epsilon_{sd} \prod_{i=1}^{\ell} \epsilon_{si}}{1 + \epsilon_{sd} \cdot \displaystyle\sum_{j=1}^{\ell} \Big(\frac{1-\epsilon_{sj}}{1-\epsilon_{jd}} \cdot \prod_{i=1}^{j-1} \epsilon_{si}\Big)}. \tag{5.27}$$

Further, every α_ℓ*-channel allocation is a vertex of* $\mathcal{P}_{MRC}$, *i.e., it satisfies (5.22) and (5.23).*

Proof. See Appendix F. □

(Note: We make the usual assumption that a sum over an empty set is equal to zero and a product over an empty set is unity, so, for example $C_0 = (1 - \epsilon_{sd} \cdot 1)/(1 + \epsilon_{sd} \cdot 0) \ = \ 1 - \epsilon_{sd}$.)

Any coding scheme employing the α_ℓ-channel allocation has the following properties:

1. The relays $r_{\ell+1}, \cdots r_M$ are *unused*.

2. For every $\mathcal{S} - \mathcal{D}$ cut where $\mathcal{D}$ is of the form $\mathcal{D} = \{r_1, \cdots, r_{m-1}, d\}$ where $m \leq \ell$, the average number of unerased bits transferred from r_m to d $\Big(= n_m(1 - \epsilon_{md}) \Big)$ equals the average number of bits that were received by r_m from s, but **not** by *any* of the nodes in $\mathcal{D}$.

The fact that the relays are ordered according to $\epsilon_{id} \leq \epsilon_{jd}$ for $i < j$ induces a particular ordering among the channel allocation rates C_ℓ, described in the following lemma.

Lemma 5. *Suppose ℓ^* is chosen as the smallest value of ℓ for which C_ℓ attains its maximum, i.e.,*

$$\ell^* \;=\; \min\left\{ \underset{0 \leq \ell \leq M}{\arg\max}\; C_\ell \right\} \tag{5.28}$$

Then the following inequalities hold:

$$C_\ell \leq C_{\ell+1} \leq (1 - \epsilon_{(\ell+1)d}), \qquad \textit{for } 0 \leq \ell \leq \ell^* - 1,\; 1 \leq \ell^* \leq M \tag{5.29}$$

$$C_\ell \geq C_{\ell+1} \geq (1 - \epsilon_{(\ell+1)d}), \qquad \textit{for } \ell^* \leq \ell \leq M - 1,\; 0 \leq \ell^* \leq M - 1 \tag{5.30}$$

Further, when $\ell^* \neq 0$, we have the following strict inequality:

$$C_{\ell^*-1} \;<\; C_{\ell^*} \;<\; (1 - \epsilon_{\ell^* d}) \tag{5.31}$$

This lemma is proved in Appendix G. It is useful in proving the following theorem.

Theorem 3. *Let ℓ^* be chosen as in (5.28). Then, the channel allocation that minimizes $n_s + \sum_{i=1}^{M} n_i$ subject to (5.22) and (5.23) is given by the α_{ℓ^*}-channel allocation.*

Proof. By definition, the α_{ℓ^*}-channel allocation forms the solution to this subset of (5.22) and (5.23) met with equality:

$$\begin{aligned} n_s\Big(1 - \epsilon_{sd}\prod_{j=1}^{m}\epsilon_{sj}\Big) + \sum_{j=m+1}^{M} n_j(1-\epsilon_{jd}) &\geq k, \quad 0 \leq m \leq \ell^*, &(5.32)\\ n_m &\geq 0, \quad \ell^* < m \leq M. &(5.33)\end{aligned}$$

We re-write (5.32) and (5.33) as equalities by introducing non-negative slack variables $\{Z_m\}_{m=0}^{M}$:

$$\begin{aligned} n_s\Big(1 - \epsilon_{sd}\prod_{i=1}^{m}\epsilon_{si}\Big) + \sum_{j=m+1}^{M} n_j(1-\epsilon_{jd}) &= k + Z_m, \quad 0 \leq m \leq \ell^*, &(5.34)\\ n_m &= Z_m, \quad \ell^* < m \leq M. &(5.35)\end{aligned}$$

It is now possible to solve (5.34) and (5.35) for n_s, $\{n_m\}_{m=1}^{M}$ in terms of $\{Z_m\}_{m=0}^{M}$; then, rewriting the objective function in terms of these slack variables leads to

$$n_s + \sum_{j=1}^{M} n_j = c_k + \sum_{m=0}^{M} \gamma_m Z_m \tag{5.36}$$

where, for $\ell^* = 0$ we have $c_k = k/(1-\epsilon_{sd})$ and

$$\gamma_m = \begin{cases} \dfrac{1}{1-\epsilon_{sd}}, & \text{for } m = 0,\\ \dfrac{\epsilon_{md} - \epsilon_{sd}}{1-\epsilon_{sd}}, & \text{for } 1 \leq m \leq M, \end{cases} \tag{5.37}$$

while for $\ell^* \geq 1$ we have $c_k = k/C_{\ell^*}$ and

$$\gamma_m = \begin{cases} \dfrac{1}{1-\epsilon_{1d}}, & \text{for } m = 0, \\ \dfrac{1}{1-\epsilon_{(m+1)d}} - \dfrac{1}{1-\epsilon_{md}}, & \text{for } 1 \leq m \leq (\ell^* - 1), \\ \dfrac{1}{C_{\ell^*}} - \dfrac{1}{1-\epsilon_{\ell^* d}}, & \text{for } m = \ell^*, \\ 1 - \dfrac{1-\epsilon_{md}}{C_{\ell^*}}, & \text{for } (\ell^*+1) \leq m \leq M. \end{cases} \tag{5.38}$$

A detailed evaluation of c_k and $\{\gamma_m\}_{m=0}^{M}$ for the two cases is provided in Appendix H.

When $\ell^* = 0$, observe that $\gamma_0 > 0$. Also, $\{\gamma_m\}_{m=1}^{M}$ are non-negative, because $C_0 \geq C_m \geq 1 - \epsilon_{md}$ for $1 \leq m \leq M$, from (5.30); since $C_0 = 1 - \epsilon_{sd}$, we have $\epsilon_{sd} \leq \epsilon_{md}$.

Likewise, when $\ell^* \geq 1$,

- $\gamma_0 > 0$,
- $\{\gamma_m\}_{m=1}^{\ell^*-1}$ are non-negative, as $\epsilon_{md} \leq \epsilon_{(m+1)d}$ by the ordering of the relays,
- $\gamma_{\ell^*} > 0$ from (5.31),
- $\{\gamma_m\}_{m=\ell^*+1}^{M}$ are non-negative, as $C_{\ell^*} \geq C_\ell \geq (1 - \epsilon_{\ell d})$ for $\ell \geq \ell^* + 1$, from (5.30).

Since $\gamma_m \geq 0$ for all m for both cases, $n_s + \sum_{j=1}^{M} n_j$ is minimized when $Z_m = 0$ for all m, which yields the α_{ℓ^*}-channel allocation. From Lemma 4, the α_{ℓ^*}-channel allocation satisfies (5.22) and (5.23), proving the theorem. □

Corollary 2. *The capacity of the M-relay erasure MRC is bounded as $C_{MRC} \leq C_{\ell^*}$.*

5.3.3 Achievability of the upper bound

We now show that the upper bound in Corollary 2 is achievable. Specifically, for any ℓ ($0 \leq \ell \leq M$), we construct a coding scheme using the α_ℓ-channel allocation that achieves reliable communication at a rate arbitrarily close to C_ℓ for sufficiently large k.

As with the MARC results in Section 5.2.4, we employ capacity approaching codes for the BEC and make the same simplifying assumptions – i.e., c.a. codes operate *at* capacity, and the random number of erasures experienced (jointly) by a node (group of nodes) is equal to its expected value.

The coding scheme is as follows. The k message bits are encoded at s using a c.a. code of rate $1 - \epsilon_{sd} \prod_{i=1}^{\ell} \epsilon_{si}$, consistent with the α_ℓ-channel allocation. Of the $n_s = k/(1 - \epsilon_{sd} \prod_{i=1}^{\ell} \epsilon_{si})$ transmitted bits, the number received unerased by at least one node among d and the first m ($\leq \ell$) relays is $n_s(1 - \epsilon_{sd} \prod_{i=1}^{m} \epsilon_{si})$.

Since c.a. codes are used, any $k = n_s(1 - \epsilon_{sd} \prod_{i=1}^{\ell} \epsilon_{si})$ code bits transmitted by s suffice to decode the message. Therefore, if every relay r_m (for $m \leq \ell$) communicates its received bits to d, then d can decode the message correctly, This can be accomplished efficiently via an *inductive* encoding procedure at the relays as follows.

We describe the encoding process at r_m ($m \leq \ell$) under the hypothesis that the preceding relays ($r_1, \ldots, r_{m-1}$) have successfully communicated to d the bits they received (unerased) from s. (This is trivially true for $m = 1$.) Denote the $n_s(1 - \epsilon_{sm})$ code bits received by r_m from s as the vector $\boldsymbol{z_m}$. The inductive hypothesis, together with the side information that d has about the erasure locations on all links, implies that d knows a fraction $(1 - \epsilon_{sd} \prod_{i=1}^{m-1} \epsilon_{si})$ of $\boldsymbol{z_m}$. Therefore, r_m only needs to communicate the remaining fraction $\epsilon_{sd} \prod_{i=1}^{m-1} \epsilon_{si}$ of $\boldsymbol{z_m}$ to d. This can be accomplished with a compress-and-forward strategy at r_m: (i) the vector $\boldsymbol{z_m}$ is first compressed by a factor $\epsilon_{sd}\prod_{i=1}^{m-1}\epsilon_{si}$, making possible the recovery of *any* unknown fraction $\epsilon_{sd}\prod_{i=1}^{m-1}\epsilon_{si}$ of $\boldsymbol{z_m}$ (for example, using a Tornado code), (ii) the compressed bits are then communicated perfectly to d over the $r_m \to d$ link using a c.a. code for the BEC of rate $1-\epsilon_{md}$. Thus, d now has decoded all the code bits received by relays $(r_1, \ldots, r_m)$. The number of bits transmitted by r_m in this process is $n_m = \dfrac{n_s(1-\epsilon_{sm})\epsilon_{sd}\prod_{i=1}^{m-1}\epsilon_{si}}{1-\epsilon_{md}}$, as required by the α_ℓ-channel allocation.

By induction, it follows that d can recover all the received code bits at relays $(r_1, \cdots, r_\ell)$, which suffice to decode the source's message. Thus, we have constructed a reliable coding scheme using the α_ℓ-channel allocation, and so the following result is obtained.

Theorem 4. *The capacity of the M-relay erasure MRC is given by*

$$C_{MRC} \;=\; C_{\ell^*} \;=\; \max_{0\le\ell\le M} C_\ell. \tag{5.39}$$

5.3.4 Unique optimality of the α_{ℓ^*} channel allocation

We have demonstrated that the α_{ℓ^*}-channel allocation can be used to construct a c.a. coding scheme for the erasure MRC. It is natural to ask if this allocation is *unique*, i.e., whether there exist other channel allocations with the same rate for which reliable coding schemes can be constructed. This is easily answered by inspecting (5.36). Clearly, if $\gamma_m > 0$ for all m, the value of $n = n_s + \sum_{j=1}^{M}$ in (5.36) is minimized *only when* $Z_m = 0$ for all m, leading to the unique optimality of the α_{ℓ^*}-channel allocation.

Therefore, from the observations in the proof of Theorem 3, it follows that the following two conditions suffice to guarantee unique optimality:

1. $\epsilon_{md} \; < \; \epsilon_{(m+1)d}, \quad$ for $1 \leq m \leq \ell^* - 1$ (when $\ell^* \geq 1$), i.e., there must exist a *unique* ordering of the relays $r_1, \cdots, r_{\ell^*-1}$ according to decreasing reliability of the relay-destination links.

2. $1 - \epsilon_{md} \; < \; C_{\ell^*}, \quad$ for $\ell^* + 1 \leq m \leq M$, i.e., the relay-destination capacities of all unused relays in the α_{ℓ^*}-channel allocation must be *strictly smaller* than the MRC capacity.

These conditions are also *necessary* for uniqueness; when they do not hold, it is possible to construct c.a. coding schemes with channel allocations different from the α_{ℓ^*}-channel allocation.

5.3.5 Interpreting the capacity result for the MRC

Recall that, in the context of the multiple access relay channel (MARC), the relay r assisted a source s_i *if and only if* r had a better channel to d than s_i. Does an analogous result hold for the MRC? The answer is, "Not quite" - as the following lemma[5] reveals.

Lemma 6. *For any relay r_i with $\epsilon_{id} > \epsilon_{sd}$, it holds that $n_i = 0$ under the α_{ℓ^*}-channel allocation, i.e., the optimal channel allocation does not make use of relays with a worse channel to d than the source. However, the converse is not true, i.e., if $\epsilon_{id} < \epsilon_{sd}$, it is not necessary that $n_i > 0$ under the α_{ℓ^*}-channel allocation.*

Proof. Note that $n_i > 0$ in the α_{ℓ^*}-channel allocation only if $i \leq \ell^*$. So, given $\epsilon_{id} > \epsilon_{sd}$, we must show that $i > \ell^*$. Suppose this were *not* true - i.e., that $i \leq \ell^*$. It then follows from Lemma 5 that $C_i \leq 1 - \epsilon_{id}$ and $C_i \geq C_j$ for all $j < i$: in particular, $C_i \geq C_0$. From (5.27), note that $C_0 = 1 - \epsilon_{sd}$. Hence, we have

[5] In this section we assume that the α_{ℓ^*}-channel allocation is uniquely optimal. In particular, we assume there is a unique ordering of the relays satisfying $\epsilon_{1d} < \epsilon_{2d} < \cdots < \epsilon_{Md}$

$1 - \epsilon_{id} \geq C_i \geq C_0 = 1 - \epsilon_{sd}$, which contradicts our assumption that $\epsilon_{id} > \epsilon_{sd}$. Therefore, it must hold that $i > \ell^*$ and $n_i = 0$.

For the converse, consider a two-relay MRC wherein the first relay has both incoming and outgoing links that are lossless - i.e., $\epsilon_{s1} = \epsilon_{1d} = 0$. Then, for any $\epsilon_{sd} > 0$ and any $\epsilon_{2d} > 0$ – and, in particular, for $0 < \epsilon_{2d} < \epsilon_{sd}$ – it is easily verified that $\ell^* = 1$. Thus, it is shown by example that $\epsilon_{id} < \epsilon_{sd}$ does not imply $n_i > 0$ under the α_{ℓ^*}-channel allocation. □

In fact, from Lemma 5 and the capacity result of Theorem 4, we have the following:

1. For $M' \leq \ell^*$, the capacity of the MRC *restricted* to only the first M' relays is $C_{M'}$. For $M' > \ell^*$, the capacity is C_{ℓ^*}.

2. For any relay r_i, if $1 - \epsilon_{id} > C_{i-1}$, then $i \leq \ell^*$; conversely, if $1 - \epsilon_{id} < C_{i-1}$, then $i > \ell^*$.

These observations are summarized in the following lemma, which generalizes Lemma 6.

Lemma 7. *For $1 \leq i \leq M$, relay r_i participates in the optimal coding scheme for the M-relay MRC if and only if its link to the destination has a* **higher** *capacity than the MRC consisting of all the previous $i - 1$ relays.*

5.4 Some MRC examples

It is instructive to compute the *range* of possible values for C_{MRC} for a given value of ϵ_{sd}. The best case is the one in which all the links (except the direct $s \rightarrow d$ link) are perfect - i.e., have an erasure probability of zero - and the worst case corresponds to a situation where all the links (except the direct $s \rightarrow d$ link) are useless - i.e., have an erasure probability of one. The corresponding capacities yield these bounds:

$$1 - \epsilon_{sd} \;\leq\; C_{MRC} \;\leq\; \frac{1}{1+\epsilon_{sd}}. \tag{5.40}$$

For small ϵ_{sd}, note that these bounds are fairly close.

5.4.1 Relays "close" to the source

Suppose that all the *source-relay* links are *perfect*, i.e., $\epsilon_{si} = 0$, which might reflect relays positioned close to the source. Further, assume there is at least one relay with a better link to d than s, i.e., $\epsilon_{1d} < \epsilon_{sd}$. Then all relays possess identical information about the source's message, and, in the optimum channel allocation, only the relay with the *best* link to d (i.e., r_1) is used; indeed, it can be verified that $n_s = k$, $n_1 = \epsilon_{sd}k/(1-\epsilon_{1d})$ and $n_j = 0$ for $j \geq 2$, which yields a capacity of

$$C_{MRC} \;=\; \frac{1-\epsilon_{1d}}{1+\epsilon_{sd}-\epsilon_{1d}}. \tag{5.41}$$

5.4.2 Relays "close" to the destination

Now suppose that all the *relay-destination* links are perfect, i.e., $\epsilon_{id} = 0$, and all source-relay links have the same erasure probability: $\epsilon_{si} = \epsilon < 1$, as shown in Figure 5.6(a). This might represent a situation in which all the relays are located close to d. Evaluating C_ℓ for this case yields:

$$C_\ell \;=\; \frac{1-\epsilon_{sd}\epsilon^\ell}{1-\epsilon_{sd}\epsilon^\ell+\epsilon_{sd}}. \tag{5.42}$$

Since C_ℓ *increases* with ℓ, we conclude that $\ell^* = M$ (i.e, *all* relays are used), and the capacity is given by

$$C_{MRC} \;=\; \frac{1-\epsilon_{sd}\epsilon^M}{1-\epsilon_{sd}\epsilon^M+\epsilon_{sd}}. \tag{5.43}$$

As the number of relays increases, C_{MRC} increases monotonically and converges to the following value:

$$C_\infty \;\triangleq\; \lim_{M\to\infty} C_{MRC} \;=\; \frac{1}{1+\epsilon_{sd}} \tag{5.44}$$

Note that this is equal to the upper bound in (5.40). Figure 5.6(b) shows the variation of C_{MRC} with ϵ for several values of M.

This result can be generalized, as follows:

1. Remove the assumption that the source-relay channels are identical; instead just assume that $0 < \epsilon_{si} < 1$ for each i. Then, C_ℓ still increases with ℓ, and so $\ell^* = M$ and *all* the relays should be used. Further, for the asymptotic case when $M \to \infty$, if it is true that $\limsup_{i\to\infty} \epsilon_{si} < 1$, then the limiting value of C_{MRC} is still given by (5.44).

2. Now, suppose further that the relay-destination channels are *not* lossless, but identical with erasure probability $\epsilon' < \epsilon_{sd}$. Then, it can *still* be shown that C_ℓ increases with ℓ and so it is optimal to use *all* the relays. However, as $M \to \infty$ with $\limsup_{i\to\infty} \epsilon_{si} < 1$, the asymptotic capacity now behaves as

$$C_\infty \;=\; \frac{1}{1+\epsilon_{sd}/(1-\epsilon')}. \tag{5.45}$$

In summary, if all relays have statistically *identical* links to d which are better than the direct link, then it is optimum to use *all* relays. Moreover, having a large (infinite) number of relays with *perfect* links to d is equivalent (capacity-wise) to having a single relay with both a perfect link from s and a perfect link to d.

5.4.3 A two-relay MRC

Now consider the special case shown in Figure 5.7(a). The links $r_1 \rightarrow d$ and $s \rightarrow r_2$ are lossless, while $\epsilon_{s1} = \epsilon_{2d} = \epsilon$. This is representative of a situation in which r_1 is located close to d and r_2 is close to s.

In this "anti-symmetric" setting, it is interesting to characterize the roles played by the two relays in the optimal strategy. For this, we evaluate $\{C_\ell\}_{\ell=0}^{2}$ and determine ℓ^* by finding their maximum. The result is given by

$$\ell^* = \begin{cases} 1, & \text{for } \epsilon > \epsilon^*, \\ 2, & \text{otherwise,} \end{cases} \tag{5.46}$$

where,

$$\epsilon^* \;=\; \frac{1+\epsilon_{sd}}{2\epsilon_{sd}} - \sqrt{\left(\frac{1+\epsilon_{sd}}{2\epsilon_{sd}}\right)^2 - 1}. \tag{5.47}$$

Thus, there exists a threshold value for ϵ beyond which it is optimal to use only relay r_1. In this particular case, it can be shown that the threshold ϵ^* lies in the range: $0.75\epsilon_{sd} \;\leq\; \epsilon^* \;\leq\; \epsilon_{sd}$. (The lower bound is tight for $\epsilon_{sd} = 2/3$, and the upper bound is tight for both $\epsilon_{sd} = 0$ and 1.)

In Figure 5.7(b), the channel capacity is plotted as a function of ϵ for two different values of ϵ_{sd}. For both cases, a "cusp" appears in the capacity at $\epsilon = \epsilon^*$, reflecting the transition from using both relays to using only one. For $\epsilon_{sd} = 0.75$, the values of C_ℓ for $\ell = 1, 2$ are also plotted; the capacity is the "envelope" of these two curves.

Now, let ζ_i denote the *fraction* of the total information delivered to d from relay r_i in the optimal coding scheme, i.e.,

$$\zeta_i = \frac{\#\text{ bits received by } d \text{ from } r_i}{k}, \quad (i = 1, 2) \tag{5.48}$$

Then, for the case when $\epsilon^* > 1/2$, an interesting fact is that

$$\begin{aligned} \zeta_1 &> \zeta_2, & &\text{for } \epsilon < 1/2, \\ \zeta_1 &< \zeta_2, & &\text{for } 1/2 < \epsilon < \epsilon^*, \\ \zeta_1 &> \zeta_2 = 0, & &\text{for } \epsilon > \epsilon^*. \end{aligned}$$

In other words, while r_1 may be used in more situations than r_2, it is possible that r_2 is actually more "useful" than r_1 in certain cases.

5.5 A note on erasure feedback

Suppose that, prior to the transmission of a bit in a given time slot, the *state* of the transmission up to that point – specifically, a history of all the erasures that have occurred – is available to all the nodes in the network. It turns out that such feedback does not change the capacity results for either the MARC or the MRC, a result that follows because the cut-set bounds for both channels are essentially *unchanged* in the presence of erasure feedback. Consequently, since the capacity (region) in the absence of feedback already coincides with the outer bound (region) described by the cut-set bounds, the presence of feedback does not improve capacity.

We now demonstrate that the cut-set bounds derived without feedback also hold when there is erasure feedback. For reasons of simplicity in exposition, we present the complete proof for the MRC and only sketch the analogous argument for the MARC.

We first describe a generic coding scheme for the MRC with feedback.

Definition 3. *An* (n, k) **feedback-code** *for the* M*-relay erasure MRC with feedback consists of:*

- *An* **encoder** *that maps* k *message bits onto* n *code bits, transmitted over* n *time slots shared without overlap by the source and* M *relays. The encoding operation in the* i^{th} *time slot consists of two operations:*

  1. *First, one node is scheduled to transmit based on the erasure history of all previous transmissions. The scheduling function is given by* $S_i : \Omega_\epsilon^{i-1} \to \{s, r_1, \cdots, r_M\}$*, where* Ω_ϵ *describes the set of all possible erasure outcomes for a single transmission in the network. (So* Ω_ϵ *contains* $2^{M+1} + M$ *elements –* 2^{M+1} *possible erasure outcomes associated with a transmission from* s*, and a single erasure outcome for each transmission from* M *different relays.)*
  2. *The scheduled node then determines the code bit that is transmitted during the* i^{th} *time slot. The encoding function at* s *is given by* $F_i^s : \{0,1\}^k \times \Omega_\epsilon^{i-1} \to \{0,1\}$*. The encoding function at relay* r_j *is given by* $F_i^j : \{0,1,E,\phi\}^{i-1} \times \Omega_\epsilon^{i-1} \to \{0,1\}$*, where* E *and* ϕ *respectively denote the erasure and absence of an incoming transmission.*

- *A* **decoder** *at the destination that estimates the source's message via the function* $g : \{0,1,E\}^n \times \Omega_\epsilon^n \to \{0,1\}^k$*.*

The rate of this code is $R \triangleq k/n$*.*

As before, a rate R is achievable for the MRC with feedback if there exists a sequence of feedback-codes with increasing n with rates approaching R and vanishing probability of decoder failure, and the capacity $C_{MRC}^{(fb)}$ is the supremum of all achievable rates. Also, we retain the definitions of an $\mathcal{S} - \mathcal{D}$ cut of the MRC and the associated random variable $N_{\mathcal{S}}$ from Lemma 3.

Lemma 8. *Given an $\mathcal{S} - \mathcal{D}$ cut of the MRC and an (n, k) feedback-code, if there exists $\delta > 0$ such that $\mathbb{E}(N_{\mathcal{S}}) < k(1 - \delta)$, then the probability of decoder failure is lower bounded as $p_e^{(n)} > \delta/2$.*

Proof. Knowing that a bit has been erased conveys no information about the value of the erased bit, so the number of bits received by $\mathcal{D}$ from $\mathcal{S}$ in n transmissions must *still* be at least k for successful decoding to be possible at d. In particular, if $N_{\mathcal{S}} < k$, then the decoder fails with probability at least 1/2. Consequently, similar to the case without feedback, we have:

$$p_e^{(n)} \geq \frac{1}{2} \cdot \mathbb{P}(N_{\mathcal{S}} < k). \tag{5.49}$$

Suppose there exists $\delta > 0$ such that $\mathbb{E}(N_{\mathcal{S}}) < k(1-\delta)$. From Markov's inequality, we have

$$\mathbb{P}\Big(N_{\mathcal{S}} \ \geq \ \frac{\mathbb{E}(N_{\mathcal{S}})}{1-\delta}\Big) \leq 1 - \delta, \tag{5.50}$$

and so $\mathbb{P}(N_{\mathcal{S}} < k) \geq \mathbb{P}\Big(N_{\mathcal{S}} < \frac{\mathbb{E}(N_{\mathcal{S}})}{1-\delta}\Big) > \delta$ which results in $p_e^{(n)} > \delta/2$. □

Since the number of transmissions from each node depends on the previous erasure events, it follows that $n_s, n_1, \cdots, n_M$ are *random variables.* Let $\bar{n}_s \triangleq \mathbb{E}(n_s)$ and $\bar{n}_j \triangleq \mathbb{E}(n_j)$, $1 \leq j \leq M$. Also, let $\lambda_{i,\eta}$ denote the marginal probability that node η is selected for transmission in the i^{th} time slot, and $q_{i,\mathcal{S}}$ the marginal probability that a bit is transmitted unerased from $\mathcal{S}$ to $\mathcal{D}$ in the i^{th} time slot.

Then, it follows that

$$q_{i,\mathcal{S}} = \sum_{\eta \in \mathcal{S}} \lambda_{i,\eta}(1 - \epsilon_{\eta,\mathcal{D}}), \tag{5.51}$$

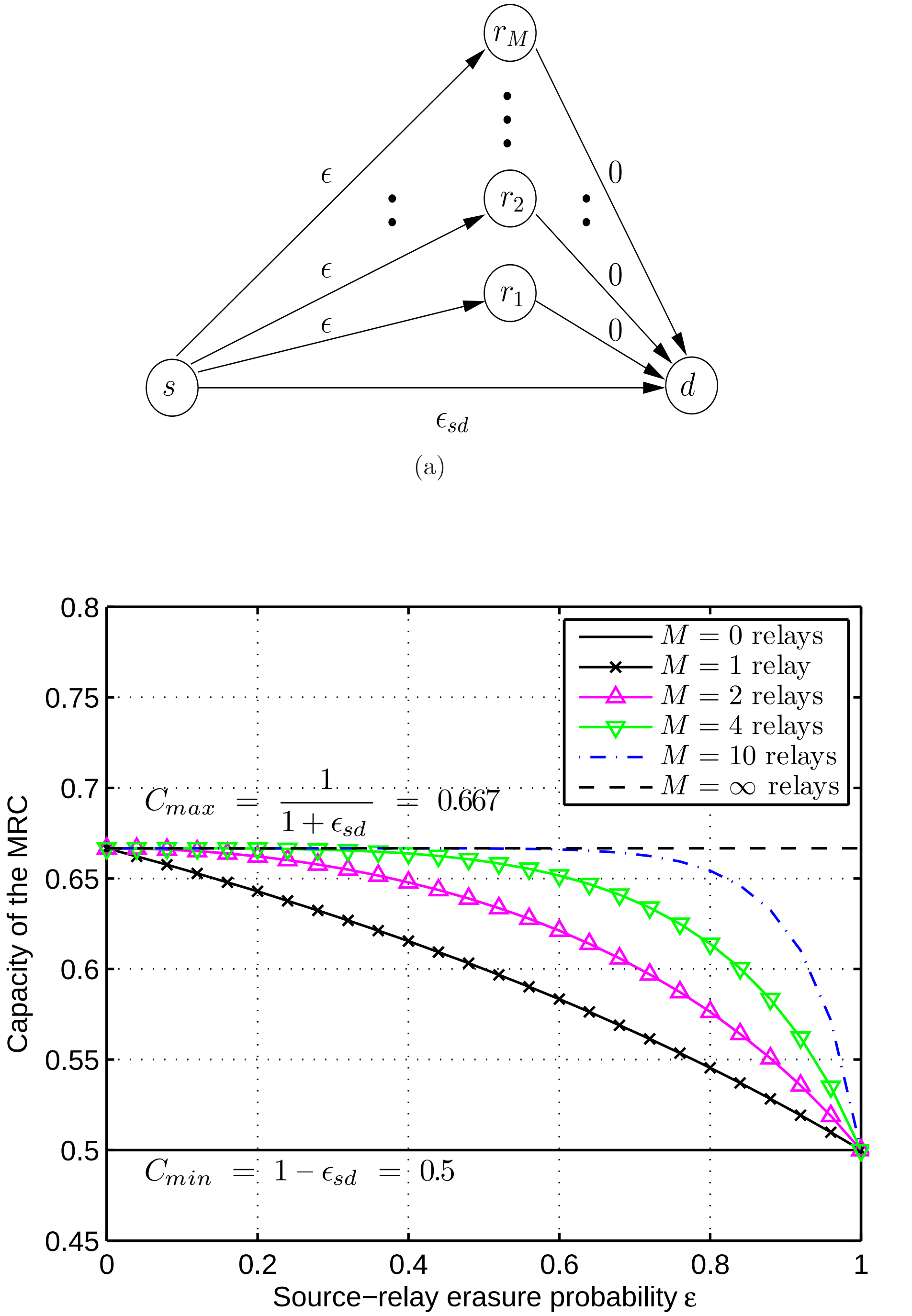
where $\epsilon_{\eta,\mathcal{D}}$ is the probability that a transmission from η is not received by any node in $\mathcal{D}$. This equality holds because the event that a node is scheduled to transmit in the i^{th} slot depends only on the erasure events of *preceding* time slots. Then,

$$\begin{aligned} \mathbb{E}(N_{\mathcal{S}}) &= \sum_{i=1}^{n} q_{i,\mathcal{S}} && (5.52) \\ &= \sum_{i=1}^{n} \sum_{\eta \in \mathcal{S}} \lambda_{i,\eta}(1 - \epsilon_{\eta,\mathcal{D}}) && (5.53) \\ &= \sum_{\eta \in \mathcal{S}} \bar{n}_{\eta}(1 - \epsilon_{\eta,\mathcal{D}}). && (5.54) \end{aligned}$$

Lemma 8 implies that for any $\delta > 0$, in order for $p_e^{(n)} \leq \delta/2$ we require $\mathbb{E}(N_{\mathcal{S}}) \geq k'$ for all $\mathcal{S} - \mathcal{D}$ cuts, where $k' = k(1 - \delta)$. From (5.54), it is seen that this condition leads to the same inequalities (5.22) and (5.23), with n_s and $\{n_j\}_{j=1}^{M}$ replaced by their expectations $\bar{n}_s$ and $\{\bar{n}_j\}_{j=1}^{M}$, and k replaced by k'. Since $\bar{n}_s + \bar{n}_1 + \cdots + \bar{n}_M = n$, minimizing n subject to these constraints yields the result that capacity with feedback is bounded as $C_{MRC}^{(fb)} \leq C_{\ell^*}/(1 - \delta)$. Since $p_e^{(n)} \to 0$ as $n \to \infty$, this upper bound must hold for all $\delta > 0$, and we have $C_{MRC}^{(fb)} \leq C_{\ell^*}$. Therefore, from our earlier achievability result, $C_{MRC}^{(fb)} = C_{\ell^*}$ – i.e., the capacity of the erasure MRC *does not increase* in the presence of universal feedback of erasure location information.

For the MARC, similar reasoning holds. In particular, one can define a feedback-code for the MARC, and Lemma 1 can be extended in the same fashion as Lemma 8 to show that for any $\mathcal{S} - \mathcal{D}$ cut of the MARC, if $E(N_{\mathcal{S}}) < k_{\mathcal{S}}(1 - \delta)$

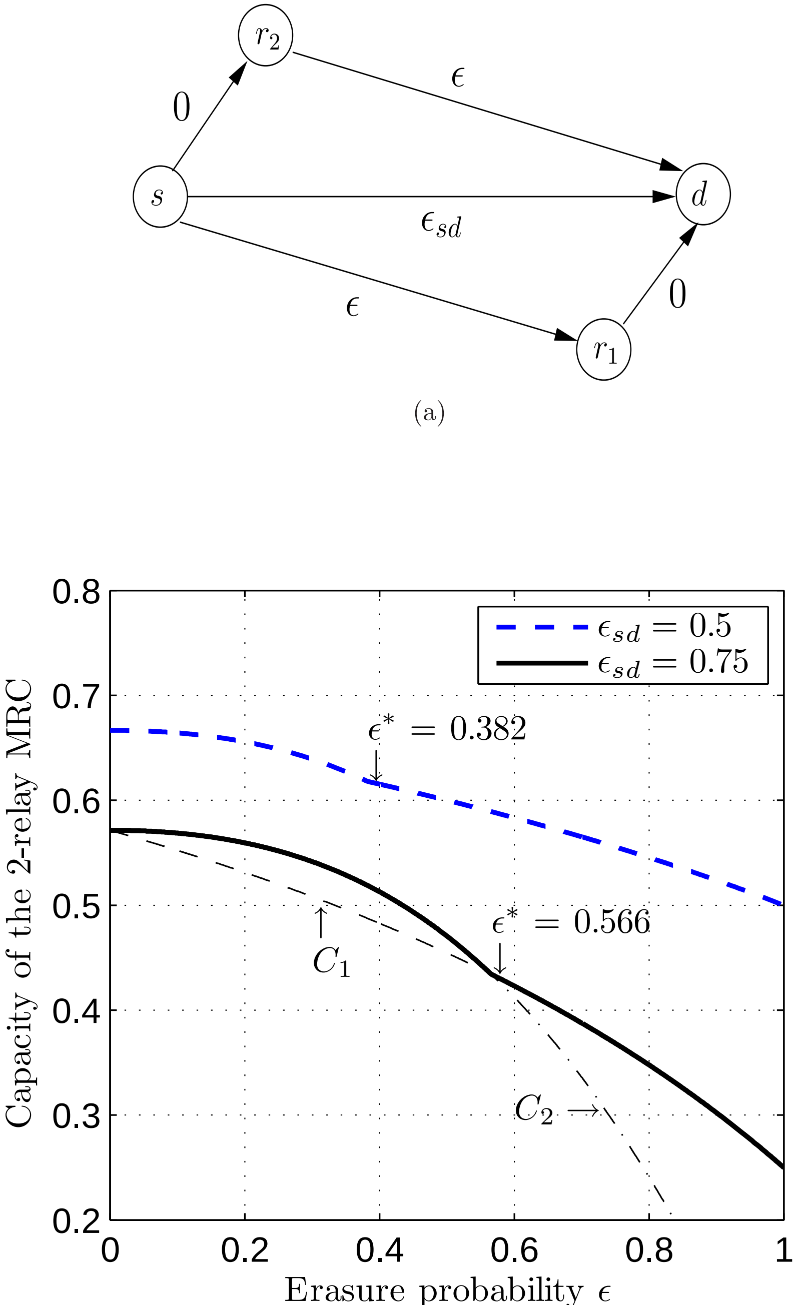
for some fixed $\delta > 0$, then the probability of decoder failure is bounded as $p_e^{(n)}(\mathcal{S}) > \delta/2$. Enforcing the condition $E(N_{\mathcal{S}}) \geq k_{\mathcal{S}}(1-\delta)$ on the feedback-code yields the bounds (5.5) and (5.6) with $\{n_i\}_{i=1}^{M}, n_r$ replaced by their expected values and $\{k_i\}_{i=1}^{M}$ replaced by $\{k_i'\}_{i=1}^{M}$ where $k_i' = k_i(1-\delta)$. Consequently, minimizing n subject to $E(N_{\mathcal{S}}) \geq k_{\mathcal{S}}(1-\delta)$ yields an upper bound on the sum rate R (for fixed $\boldsymbol{\theta}$) that is equal to the upper bound (5.17) (without feedback) multiplied by a factor $1/(1-\delta)$. Letting $\delta \rightarrow 0$ yields the same outer bound region as in Corollary 1. Thus, the capacity region of the erasure MARC also does not increase in the presence of universal feedback of erasure location information.



(a)

(b)

Figure 5.6. (a) An MRC with M relays close to the destination; (b) Plot of capacity versus ϵ for $\epsilon_{sd} = 0.5$ and different values of M.



(a)

(b)

Figure 5.7. (a) A two-relay MRC with “anti-symmetric” relay positions; (b) Capacity as a function of ϵ for $\epsilon_{sd} = 0.5$ and 0.75. The transition point ϵ^* indicates the value of ϵ at which optimal encoding transitions from using both relays to using only only one.

CHAPTER 6

CONCLUDING REMARKS

In this dissertation, we have addressed two problems pertaining to reliable communication over packet erasure networks, viz., the design of hybrid ARQ protocols that allow us to trade-off between coding complexity and feedback, and the characterization of the capacity region for two kinds of wireless erasure relay networks – the MARC and the MRC.

6.1 Complexity-versus-feedback tradeoffs

For the point-to-point erasure channel, different hybrid ARQ protocols were developed and their feedback-complexity performance analyzed vis-a-vis a benchmark time-sharing scheme. The underlying theme in these protocols is to feed back the indices (locations) of a portion of the unerased/erased bits (packets), in order to simultaneously reduce feedback requirements (relative to feedback-only) and coding complexity (relative to coding-only). This is done based on the following simple observations:

- Transmitted bits that are acknowledged to have been received successfully (i.e., unerased) can be omitted in encoding future parities (used in the hybrid A,B and C protocols)

- Likewise, bits known to have been erased can simply be re-transmitted, instead of being encoded into parities (used in the hybrid D protocol).

A combination of both forms of feedback, i.e., erasure and non-erasure locations, is used in the hybrid-E protocol.

The performance of the different hybrid protocols leads to the following observations:

- Using optimum rate-distortion codes to minimize the amount of feedback is most beneficial at small erasure probabilities ($\sim$ 0.01 to 0.1); at larger erasure rates, simpler sub-optimal feedback schemes can be used without much penalty (hybrid-B versus hybrid-A).

- Protocols that use feedback for acknowledging a subset of non-erasures provide better tradeoffs with *decreasing* erasure probability (hybrid A,B,C and C2). In contrast, using feedback for acknowledging *erasures* leads to better performance as the erasure probability is *increased* (hybrid D).

- Adapting the code structure based on the amount of information fed back, and conversely adapting the feedback strategy according to the code structure can bring about significant performance gains (hybrid-C and hybrid-C2 versus hybrid-B).

The protocols designed in this dissertation make use of Tornado codes that are designed for a particular channel erasure probability ϵ, assumed to be known. In practice, this information may not be available *a priori*. In that case, one possible way of adapting these protocols is to have the destination feed back the *observed* erasure rate (in each round); based on this, the transmitter can either choose the appropriate Tornado code from a “bank” of codes designed for different erasure rates, or generate the code itself directly on-the-fly[1].

The following issues present interesting avenues for further research on these protocols:

- **Non-ideal feedback:** The protocols designed here assume that the feedback link is *noiseless*. In practice, this is not the case, and very low-rate channel codes might be needed in order to make the feedback link appear “virtually noiseless”. Alternatively, it is of interest to design hybrid protocols that are robust or exhibit “graceful degradation” when the feedback is noisy: e.g., as over a packet erasure channel. Characterizing the performance of such hybrid protocols relative to a feedback-only protocol in the same situation is also of interest.

- **Extension to codes with bounded complexity:** As noted in Chapter 4, Tornado codes must be designed with a non-zero overhead in order to have similar complexity to ARA and NSIRA codes (which have bounded complexity at zero overhead). Since this leads to inefficient use of the forward channel, it may be of interest to investigate (i) if codes with *bounded* complexity exist, similar to ARA and NSIRA codes, that can be used with these hybrid protocols, or (ii) if variations of these hybrid protocols exist that can make use of ARA and NSIRA codes.

- **Joint design of coding and feedback:** As observed with the hybrid C and C2 protocols, improved performance is obtained by respectively adapting the

[1] If the code is generated randomly on-the-fly, then a description of the code needs to be communicated to the decoder (possibly within the header of each packet), which incurs additional transmission overhead. However, if the encoder and decoder make use of the same pseudo-random number generator (PRNG) to generate the code, the transmitter only needs to convey the initial seed used by the PRNG to the decoder, which typically requires much lesser overhead.

coding and feedback strategies. This poses the natural question of whether it is possible to *jointly* design both the coding and the feedback schemes, in order to achieve even better complexity-feedback tradeoffs.

- **Unknown channel statistics:** As mentioned above, the channel erasure probability may not be known *a priori*. More generally, the erasure process itself could have some unknown correlation; in contrast, our hybrid protocols are designed for *memoryless* channels. Hence, in such situations, it is of interest to design *universal* hybrid protocols, wherein both the coding as well as feedback strategies can adapt to unknown/varying channel statistics.

6.2 Capacity results for the MARC and the MRC

In this dissertation, we characterized the limits of reliable communication for two wireless network configurations – the MARC and the MRC – under the assumption that individual links behave as memoryless erasure channels. In particular, the optimal channel allocation strategy for each network was derived as a function of the network's parameters and a capacity-achieving coding scheme was constructed based on this strategy. Also, the capacity regions were shown to be *unchanged* in the presence of global feedback of erasure locations.

The results demonstrate that, for the MARC, the relay assists a source s if and only if the $r \rightarrow d$ link has larger capacity than the $s \rightarrow d$ link. With regard to the single-source MRC, a given relay r is used in the optimal coding scheme if and only if the capacity of the $r \rightarrow d$ link is larger than the capacity of a "reduced" MRC consisting of only those relays with better links to d than r.

Since the MARC and the MRC constitute fundamental configurations, the results presented here are potentially useful in the design of wireless network architectures. For instance, depending on how inter-node distances in a network affect the reliability of transmission, the capacity results presented here could play an important role in determining the geometry or layout of a network – such as the placement/identification of relay nodes in a sensor or ad-hoc network.

The following constitute possible avenues for further investigation:

- **Modeling erasures:** As mentioned above, the capacity results presented here could have architectural implications for wireless networks. For further work in this direction, a good understanding of how a given physical noisy channel may be mapped to a higher layer erasure model is required.

- **Role of relays:** Along the above lines, relays in a network may have two roles: (i) improving network-layer reliability in terms of delivering a sequence of packets (addressed in this dissertation), (ii) improving physical-layer reliability in delivering each individual packet, perhaps using cooperative diversity techniques [45], [46]. It is interesting to investigate if there exists an optimal tradeoff between these two roles, which can maximize the overall throughput.

- **Lack of side-information:** The capacity results presented in this dissertation assume the existence of perfect side information at the destination about the erasure outcomes on all links in the network. To make this possible, it is necessary to encode this erasure information into the headers of packets at the relay(s), prior to forwarding them. This may constitute a significant overhead if the packet sizes are small. In this case, solving for the capacity region and designing practical codes in the *absence* of this side information becomes an important problem.

APPENDIX A

COMPUTING THE RATE-DISTORTION FUNCTION $\tilde{R}(D)$ FOR THE HYBRID-B PROTOCOL

Recall that $\tilde{R}(D)$ is defined as:

$$\tilde{R}(D) = \min_{p_{Y|X}(y|x):\ E(\mathcal{D}(X,Y)) \leq D} I(X;Y), \tag{A.1}$$

Since $\mathcal{D}(1,0) = \infty$, the condition $E(\mathcal{D}(X,Y)) \leq D$ is satisfied for finite D only if $Pr(Y = 0|X = 1) = 0$. Hence, the distribution $p_{Y|X}(y|x)$ is completely characterized by $p \triangleq Pr(Y = \mathcal{U}|X = 0)$. Noting that $Pr(X = 1) = \epsilon$, we have

$$\begin{aligned} Pr(Y = \mathcal{U}) &= \sum_{i=0}^{1} Pr(X = i) \cdot Pr(Y = \mathcal{U}|X = i) \\ &= (1 - \epsilon)p + \epsilon \end{aligned} \tag{A.2}$$

Therefore, the entropy of Y is given by $H(Y) = h((1 - \epsilon)p + \epsilon)$, where $h(z) = -z\log(z) - (1 - z)\log(1 - z)$ is the binary entropy function in *nats*. Further, we have for the conditional entropy: $H(Y|X) = (1 - \epsilon) \cdot h(p)$. Hence,

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= h((1 - \epsilon)p + \epsilon)\ -\ (1 - \epsilon) \cdot h(p) \end{aligned} \tag{A.3}$$

It is easily shown that $\frac{d}{dz}h(z) = \log(\frac{1-z}{z})$. So,

$$\begin{aligned}
\frac{d}{dp}I(X;Y) &= (1-\epsilon)\cdot\log\Big(\frac{1-(1-\epsilon)p-\epsilon}{(1-\epsilon)p+\epsilon}\Big) \;-\; (1-\epsilon)\cdot\log\Big(\frac{1-p}{p}\Big) \\
&= (1-\epsilon)\cdot\log\Big(\frac{(1-\epsilon)p}{(1-\epsilon)p+\epsilon}\Big) \qquad \text{(A.4)} \\
&< 0,
\end{aligned}$$

since the argument to $\log(\cdot)$ is smaller than 1. Hence, $I(X;Y)$ is a decreasing function of p.

The expected distortion is given by:

$$\begin{aligned}
E(\mathcal{D}(X,Y)) &= \sum_{x,y} Pr(X=x)\cdot Pr(Y=y|X=x)\cdot\mathcal{D}(x,y) \\
&= (1-\epsilon)p \qquad \text{(A.5)}
\end{aligned}$$

Therefore, the condition $E(\mathcal{D}(X,Y)) \leq D$ translates to $p \leq D/(1-\epsilon)$ (< 1). Consequently, $I(X;Y)$ is minimized subject to $E(\mathcal{D}(X,Y)) \leq D$ when $p = D/(1-\epsilon)$. Hence, the rate-distortion function (in nats) is given by:

$$\begin{aligned}
\tilde{R}(D) &= h(D+\epsilon) \;-\; (1-\epsilon)h\Big(\frac{D}{1-\epsilon}\Big) \\
&= (D+\epsilon)\log\frac{1}{D+\epsilon} \;-\; D\log\frac{1}{D} \;+\; (1-\epsilon)\log\frac{1}{1-\epsilon}, \qquad \text{(A.6)}
\end{aligned}$$

from which we obtain $\tilde{R}(D)$ in bits, i.e., (3.25), by changing the natural logarithm to $\log_2(\cdot)$.

APPENDIX B

COMPUTING THE RATE-DISTORTION FUNCTION $\tilde{R}(D_0, D_1)$ FOR THE HYBRID-E PROTOCOL

Recall that $\tilde{R}(D_0, D_1)$ is defined as:

$$\tilde{R}(D_0, D_1) = \min_{p_{Y|X}(y|x):\ E(\mathcal{D}_i(X,Y)) \leq D_i,\ i \in \{0,1\}} I(X;Y). \tag{B.1}$$

From the fact that $\mathcal{D}_0(1,0) = \mathcal{D}_1(0,1) = \infty$, the expected distortions remain bounded only if $Pr(Y = 0|X = 1) = Pr(Y = 1|X = 0) = 0$. Hence, the distribution $p_{Y|X}(y|x)$ is completely characterized by $p \triangleq Pr(Y = \mathcal{U}|X = 0)$ and $q \triangleq Pr(Y = \mathcal{U}|X = 1)$. Noting that $Pr(X = 1) = \epsilon$, we have

$$\begin{aligned} Pr(Y = \mathcal{U}) &= \sum_{i=0}^{1} Pr(X = i) \cdot Pr(Y = \mathcal{U}|X = i) \\ &= p(1 - \epsilon) + q\epsilon \end{aligned} \tag{B.2}$$

We shall evaluate the mutual information using the expansion $I(X;Y) = H(X) - H(X|Y)$. Note that $H(X) = h(\epsilon)$, where $h(z) = -z \log(z) - (1 - z) \log(1 - z)$ is the binary entropy function in nats.

Now, it is easily seen that $Pr(X = 0|Y = 0) = Pr(X = 1|Y = 1) = 1$. Further, it can be verified that $Pr(X = 1|Y = \mathcal{U}) = \frac{q\epsilon}{p(1-\epsilon)+q\epsilon}$. Hence,

$$\begin{aligned} H(X|Y) &= \sum_y Pr(Y=y)\cdot H(X|Y=y) \\ &= Pr(Y=\mathcal{U})\cdot h\Big(\frac{q\epsilon}{p(1-\epsilon)+q\epsilon}\Big) \qquad \text{(B.3)} \end{aligned}$$

Thus,

$$I(X;Y) = h(\epsilon) - \big(p(1-\epsilon)+q\epsilon\big)\cdot h\Big(\frac{q\epsilon}{p(1-\epsilon)+q\epsilon}\Big) \qquad \text{(B.4)}$$

Now, let $a = p(1-\epsilon)$ and $b = q\epsilon$. Then, minimizing $I(X;Y)$ is equivalent to maximizing the following function

$$(a+b)\cdot h\Big(\frac{a}{a+b}\Big) = a\log\frac{1}{a} + b\log\frac{1}{b} - (a+b)\log\frac{1}{a+b} \qquad \text{(B.5)}$$

with respect to a and b.

It is easily verified that $\frac{d}{dz}(z\log\frac{1}{z}) = \log\frac{1}{z} - 1$. Consequently, we have:

$$\begin{aligned} \frac{\partial}{\partial a}\Big[(a+b)\cdot h\Big(\frac{a}{a+b}\Big)\Big] &= \log\frac{a+b}{a} \qquad \text{(B.6)} \\ &> 0 \quad \text{for all } b \end{aligned}$$

By symmetry, the partial derivative with respect to b is also positive for all a. Therefore, $I(X;Y)$ is minimized by setting a and b to their maximum possible values.

It is easily verified that the expected distortions are given by:

$$E(\mathcal{D}_0(X,Y)) \;=\; p(1-\epsilon) \;=\; a \tag{B.7}$$

$$E(\mathcal{D}_1(X,Y)) \;=\; q\epsilon \;=\; b \tag{B.8}$$

Thus, the constraints on the expected distortions translate to $a \leq D_0$ and $b \leq D_1$. Substituting the maximum values of a and b, we obtain the smallest value of $I(X;Y)$. Hence, the rate-distortion function (in nats) is given by:

$$\begin{aligned} \tilde{R}(D_0, D_1) &= h(\epsilon) - (D_0 + D_1) \cdot h\Big(\frac{D_0}{D_0 + D_1}\Big) \qquad \text{(B.9)} \\ &= h(\epsilon) \;+\; (D_0 + D_1) \log \frac{1}{D_0 + D_1} \;-\; D_0 \log \frac{1}{D_0} \;-\; D_1 \log \frac{1}{D_1} \end{aligned}$$

The corresponding value of $\tilde{R}(D_0, D_1)$ in bits is obtained by changing the natural logarithm to $\log_2(\cdot)$.

APPENDIX C

OPTIMAL (D, p) PAIR FOR THE HYBRID-E PROTOCOL

Prior to solving for the optimal (D, p) pair, we first outline the region of valid (D, p) pairs. Since we have $0 \leq D_1 \leq \epsilon$ and $0 \leq D_0 \leq 1 - \epsilon$, the following constraints apply to (D, p):

1. For $\epsilon < 1/2$, the set of valid (D, p) is described as follows:

$$\begin{aligned} & \{(D,p): \ 0 \leq D \leq \epsilon, \quad 0 \leq p \leq 1\} \\ \cup \ & \{(D,p): \ \epsilon \leq D \leq 1-\epsilon, \ 0 \leq p \leq \epsilon/D\} \\ \cup \ & \{(D,p): \ 1-\epsilon \leq D \leq 1, \ 1 - \tfrac{1-\epsilon}{D} \leq p \leq \tfrac{\epsilon}{D}\} \end{aligned}$$

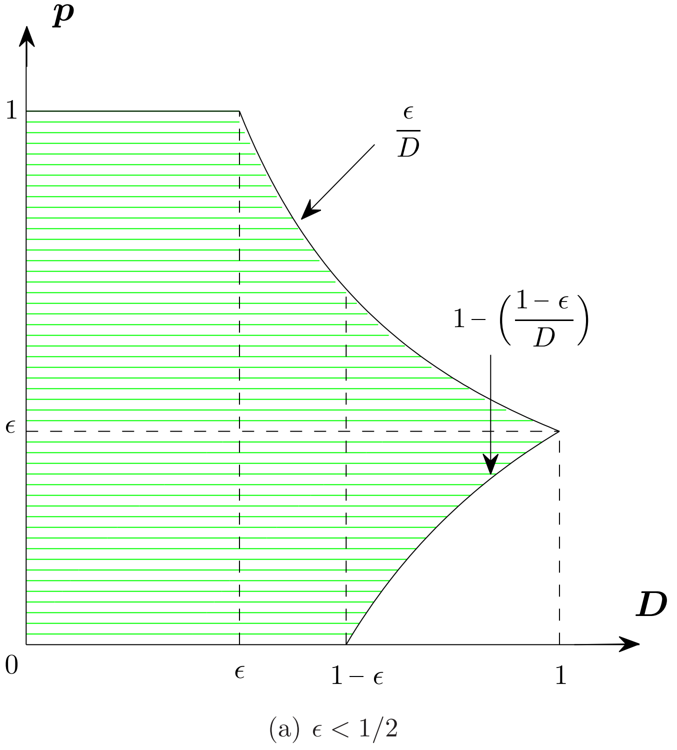
2. Similarly, for $\epsilon \geq 1/2$, the set of valid (D, p) is given by:

$$\begin{aligned} & \{(D,p): \ 0 \leq D \leq 1-\epsilon, \ 0 \leq p \leq 1\} \\ \cup \ & \{(D,p): \ 1-\epsilon \leq D \leq \epsilon, \ 1 - \tfrac{1-\epsilon}{D} \leq p \leq 1\} \\ \cup \ & \{(D,p): \ \epsilon \leq D \leq 1, \quad 1 - \tfrac{1-\epsilon}{D} \leq p \leq \tfrac{\epsilon}{D}\} \end{aligned}$$

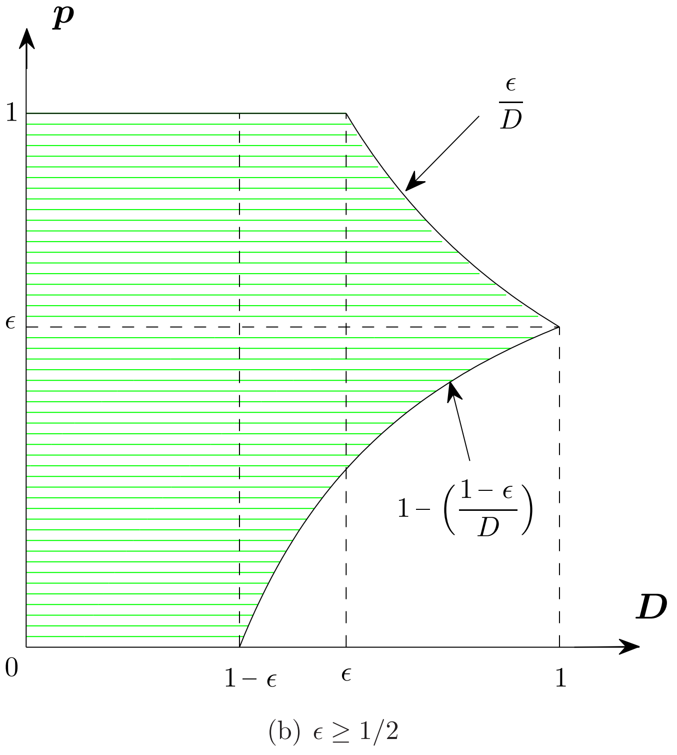
These regions are plotted in Fig. C.1.

Hence, we need to determine the valid (D, p) pair that minimizes $D \cdot g(p)$ subject to $D \cdot h(p) = \kappa$, where $g(p) \ = \ p/\log \frac{1}{1-p}$ and $\kappa \in [0, h(\epsilon)]$.

This is equivalent to the following problem: find p which minimizes $f(p) \triangleq g(p)/h(p)$, subject to $\left(D = \dfrac{\kappa}{h(p)}, p\right)$ being a valid pair. We solve this problem as



(a) $\epsilon < 1/2$



(b) $\epsilon \geq 1/2$

Figure C.1: Region of valid (D, p) pairs.

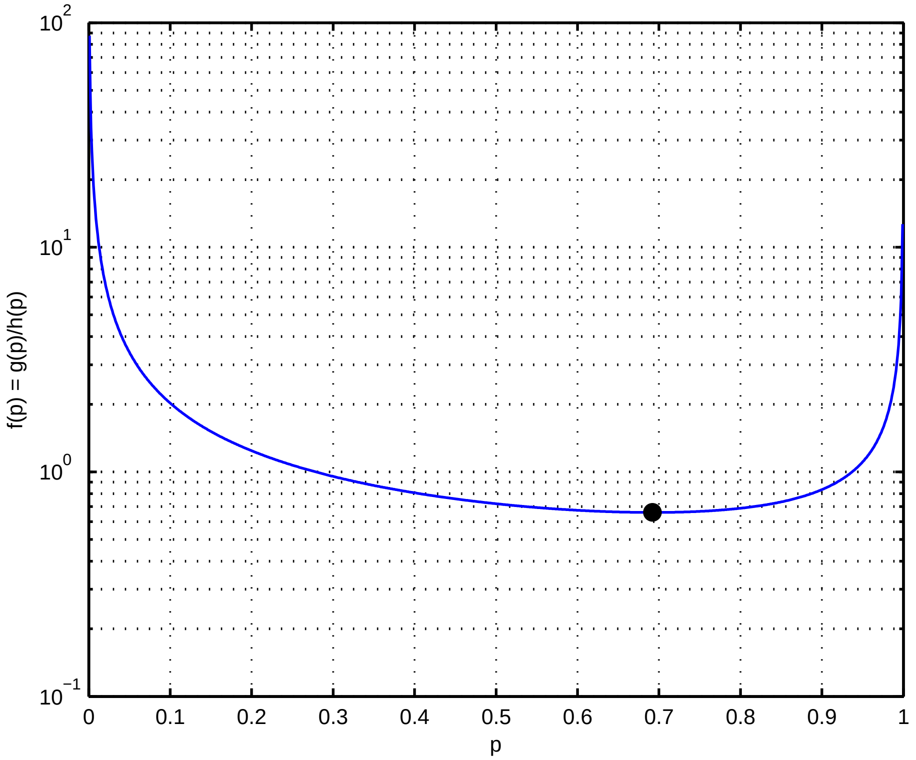


Figure C.2. Plot of $f(p)$ versus p.

follows.

Let p^* denote the value of p that minimizes $f(p)$ over the entire interval $0 \leq p \leq 1$, i.e.,

$$p^* = \underset{0 \leq p \leq 1}{\arg\min} f(p) \tag{C.1}$$

Figure C.2 plots $f(p)$ as a function of p – as seen, $f(p)$ is convex with a single minimum at $p = p^* \approx 0.692$. Therefore, whenever $(D = \kappa/h(p^*), p^*)$ is valid, it is the optimal choice. Also note that, for a given value of κ, we need to pick $D \geq \kappa$; otherwise $D \cdot h(p) = \kappa$ is infeasible (since $h(p) \leq 1$).

In the following, for each value of κ, we first characterize the valid (D, p) pairs that satisfy $D \cdot h(p) = \kappa$ and then identify the optimal pair in this set.

Case 1: $\epsilon < 1/2$

For this case, we shall make use of the fact that if we choose $p \geq \epsilon$, then it always satisfies any lower bound that applies. (Note that the greatest lower bound on p for any D is given by $\max(0, 1-(1-\epsilon)/D)$; since $1-(1-\epsilon)/D \leq 1-(1-\epsilon)$, the lower bound is no larger than ϵ.)

Likewise, if we choose $p \leq \epsilon/D$, then p always satisfies any applicable upper bound.

- **Case 1.1:** $0 \leq \kappa \leq \frac{h(p^*)}{p^*}\epsilon$

  We claim that $D = \kappa/h(p^*)$ and $p = p^*$ is the optimal choice. As noted earlier, it suffices to show that this is a valid (D, p) pair. Since $p^* > \epsilon$, any lower bound on p is satisfied. Any upper bound is also satisfied because $\epsilon/D = \epsilon h(p^*)/\kappa \geq p^*$, from the upper bound on κ.

- **Case 1.2:** $\frac{h(p^*)}{p^*}\epsilon < \kappa \leq 2\epsilon$

  This is a valid interval because $h(p^*)/p^* \approx 1.29$. For this case, $p \leq \epsilon/D$ is *necessary* as $D \geq \kappa > \epsilon$. A valid (D, p) pair that satisfies $Dh(p) = \kappa$ is given by $D = \kappa$, $p = 1/2$: any lower bound on p is obviously satisfied; the upper bound is also met because $\epsilon/D = \epsilon/\kappa \geq 1/2$, as $\kappa \leq 2\epsilon$. Also, note that $p \geq p^*$ is infeasible, as this leads to $D = \kappa/h(p) \geq \kappa/h(p^*) > \frac{h(p^*)}{p^*}\frac{\epsilon}{h(p^*)} = \epsilon/p^*$, which violates $p \leq \epsilon/D$.

  From Fig. C.2, observe that $f(p)$ *decreases* as p is increased from $1/2$ to p^*, and *increases* as p is reduced below $1/2$. Therefore, the optimum choice of p is given by its maximum *valid* value beyond $1/2$. Now, as p is increased from $1/2$, $D = \kappa/h(p)$ too increases, lowering the upper bound ϵ/D – thus, the maximum valid value of p is obtained when it coincides with its upper bound, i.e., $p = \epsilon/D$. Therefore let $\tilde{D}(\kappa)$ denote the *unique* solution[1] to $D \cdot h(\epsilon/D) = \kappa$. Let $\tilde{p}(\kappa) = \epsilon/\tilde{D}(\kappa)$. Thus, $(\tilde{D}(\kappa), \tilde{p}(\kappa))$ is the optimal pair.

- **Case 1.3:** $2\epsilon < \kappa \leq h(\epsilon)$

  This is also a valid interval because $2\epsilon \leq h(\epsilon)$ for $\epsilon \leq 1/2$, with equality if and only if $\epsilon = 0$ or $\epsilon = 1/2$.

  Again, as in the previous case, $p \leq \epsilon/D$ is necessary, as $D > 2\epsilon$. In particular, this implies that $p < 1/2$. Consequently, from the reasoning for the previous case, the optimum choice of p is given by its maximum valid value below 1/2. For this case, suppose we choose $p = \epsilon$, $D = \kappa/h(\epsilon)$ – this is a valid choice as $\kappa \leq D = \kappa/h(\epsilon) \leq 1$ and $p = \epsilon \leq \epsilon \cdot \frac{h(\epsilon)}{\kappa} = \epsilon/D$.

  For $\epsilon \leq p < 1/2$, $h(p)$ is an increasing function of p. Since $D = \kappa/h(p)$, the upper bound on p: $\epsilon/D = \epsilon \cdot h(p)/\kappa$ also increases with p. However, note that $h(p)$ tends to "flatten" out as p approaches 1/2 – therefore, ϵ/D eventually increases at a *slower* rate than p. Thus, the maximum value of p is achieved when it meets the upper bound: this condition is given by $\epsilon/D = \epsilon \cdot h(\epsilon/D)/\kappa$, which yields $D \cdot h(\epsilon/D) = \kappa$. As before, denote the solution to this equation by $\tilde{D}(\kappa)$, and let $\tilde{p}(\kappa) = \epsilon/\tilde{D}(\kappa)$. Then, the optimal choice of D and p is again given by $(\tilde{D}(\kappa), \tilde{p}(\kappa))$.

Case 2: $1/2 \leq \epsilon \leq p^*$

Here again, we note that $1 - \frac{1-\epsilon}{D} \leq \epsilon$. Hence, for any D, $p = \epsilon$ satisfies the corresponding lower bound.

[1] For $\kappa \leq D \leq 1$, we have: $\kappa h(\epsilon/\kappa) \leq D h(\epsilon/D) \leq h(\epsilon)$. Thus, $D h(\epsilon/D) = \kappa$ has a solution for $D \in [\kappa, 1]$. The uniqueness follows from the fact that $D \cdot h(\epsilon/D)$ is a strictly *increasing* function of D, for $D \geq \epsilon$. Since $\kappa > \epsilon$, $D \cdot h(\epsilon/D)$ is strictly increasing for $D \in [\kappa, 1]$.

- **Case 2.1:** $0 \leq \kappa \leq \frac{h(p^*)}{p^*}\epsilon$

  Suppose, we choose $p = p^*$ (thus, the lower bound is always met). Then, $D = \kappa/h(p^*)$. It is easily verified that $\epsilon/D \geq p^* = p$. Consequently, this choice of (D, p) is always valid and hence optimal.

- **Case 2.2:** $\frac{h(p^*)}{p^*}\epsilon < \kappa \leq h(\epsilon)$

  Note that $h(x)/x$ is a *decreasing* function of x for $x \geq 1/2$; therefore, the above interval is valid. Further, for this case, $D \geq \kappa \implies D > \epsilon$; therefore, p is upper bounded as $p \leq \epsilon/D$.

  Suppose we set $p = \epsilon$. Then, $D = \kappa/h(\epsilon)$. Clearly, $\epsilon/D \geq \epsilon$ and the upper bound is always met. Thus, this is a valid pair. Now, owing to the behavior of $f(p)$, it follows that the optimum value of p is determined by its maximum *valid* value above ϵ. Further, if p is increased above ϵ, $D = \kappa/h(\epsilon)$ also increases – therefore, the maximum valid value of p is reached when $p = \epsilon/D$. As before, let $\tilde{D}(\kappa)$ denote the solution to $D \cdot h(\epsilon/D) = \kappa$, and let $\tilde{p}(\kappa) = \epsilon/\tilde{D}(\kappa)$. Then, the optimal choice of (D, p) is once again given by $(\tilde{D}(\kappa), \tilde{p}(\kappa))$.

Case 3: $\epsilon > p^*$

- **Case 3.1:** $0 \leq \kappa \leq \frac{h(p^*)}{1-p^*}(1-\epsilon)$

  Let $p = p^*$ and $D = \kappa/h(p^*)$. Therefore, $D \leq \frac{1-\epsilon}{1-p^*}$, and hence $1-(1-\epsilon)/D \leq p^* = p$. Further $p^* < \epsilon \leq \epsilon/D$. Thus, the lower and upper bounds on p are always met. Hence, this choice of (D, p) is optimal.

- **Case 3.2:** $\frac{h(p^*)}{1-p^*}(1-\epsilon) < \kappa \leq h(\epsilon)$

  It can be verified that $h(x)/x$ is a decreasing function of x for *all* x. Thus,

$\frac{h(1-p^*)}{1-p^*} < \frac{h(1-\epsilon)}{1-\epsilon}$, which implies that the above interval is valid. Since $D \geq \kappa$ and $\frac{h(p^*)}{1-p^*} > 1$, it follows that $D \geq 1-\epsilon$, and hence, p *must* satisfy $p \geq 1-\frac{1-\epsilon}{D}$. Further, since we need $D = \kappa/h(p)$, the lower bound on p implies:

$$\begin{aligned} p &\geq 1 - \frac{(1-\epsilon)h(p)}{\kappa} && \text{(C.2)} \\ &> 1 - \frac{(1-p^*)h(p)}{h(p^*)} && \text{(from the lower bound on } \kappa\text{)(C.3)} \\ \text{i.e.,} \quad \frac{h(p)}{1-p} &> \frac{h(p^*)}{1-p^*} && \text{(C.4)} \end{aligned}$$

Since $h(x)/x$ is monotone decreasing, the above implies that $(1-p) < (1-p^*)$ or $p > p^*$.

Suppose we choose $p = \epsilon$ $(> p^*)$ and $D = \kappa/h(\epsilon)$. Then, it follows that $\epsilon/D \geq \epsilon$, and we also have $1-(1-\epsilon)/D < \epsilon$; thus, both upper and lower bounds on p are met and this is a valid choice. Now, the optimum choice of p is given by its smallest valid value above p^*. As we decrease p below ϵ, since $p > p^* > 1/2$ at all times, $h(p)$ increases. Consequently, with $D = \kappa/h(p)$, the lower bound $1-(1-\epsilon)/D$ is also reduced. However, since $h(p)$ "flattens" out near $p = 1/2$, the smallest valid value of p is achieved when it meets the lower bound. This yields $p = 1-(1-\epsilon)/D$ with $D \cdot h(p) = \kappa$, or equivalently, $D \cdot h((1-\epsilon)/D) = \kappa$. As in the earlier cases, it can be shown that this equation has a unique solution, denoted $\hat{D}(\kappa)$; let $\hat{p}(\kappa) = 1-(1-\epsilon)/\hat{D}(\kappa)$. Thus, $(\hat{D}(\kappa), \hat{p}(\kappa))$ is the optimal choice.

Summarizing, the optimal (D, p) is given by:

1. For $\epsilon \le p^*$,

$$(D, p) = \begin{cases} \left(\dfrac{\kappa}{h(p^*)},\ p^*\right) & \text{for } 0 \le \kappa \le \frac{h(p^*)}{p^*}\epsilon \\ \left(\tilde{D}(\kappa),\ \dfrac{\epsilon}{\tilde{D}(\kappa)}\right) & \text{for } \frac{h(p^*)}{p^*}\epsilon < \kappa \le h(\epsilon) \end{cases} \tag{C.5}$$

2. For $\epsilon > p^*$,

$$(D, p) = \begin{cases} \left(\dfrac{\kappa}{h(p^*)},\ p^*\right) & \text{for } 0 \le \kappa \le \frac{h(p^*)}{1-p^*}(1-\epsilon) \\ \left(\hat{D}(\kappa),\ 1 - \dfrac{1-\epsilon}{\hat{D}(\kappa)}\right) & \text{for } \frac{h(p^*)}{1-p^*}(1-\epsilon) < \kappa \le h(\epsilon) \end{cases} \tag{C.6}$$

APPENDIX D

ACHIEVING THE DESIRED OVERHEAD δ BY TIME-SHARING BETWEEN DIFFERENT D.D. PAIRS

Suppose we wish to encode k message bits into βk Tornado parities, such that the resulting overhead is δ, i.e., $\beta = \epsilon/(1-\delta)$ for erasure probability ϵ. Further, let (λ_1, ρ_1) and (λ_2, ρ_2) denote two d.d. pairs that achieve overheads δ_1 and δ_2, respectively, with $\delta_1 > \delta > \delta_2$.

A straightforward way to achieve the desired overhead is to appropriately time-share between the two d.d. pairs. The message is split into two sets containing αk and $(1-\alpha)k$ bits, which are then separately encoded using the d.d. pairs (λ_1, ρ_1) and (λ_2, ρ_2) to produce $\beta_1 \alpha k$ and $\beta_2(1-\alpha)k$ parities, respectively. Here, $\beta_1 = \epsilon/(1-\delta_1)$ and $\beta_2 = \epsilon/(1-\delta_2)$. Thus, to achieve an overhead δ, we need to choose α such that $\alpha\beta_1 + (1-\alpha)\beta_2 = \beta$. This translates to

$$\frac{1}{1-\delta} \;=\; \frac{\alpha}{1-\delta_1} + \frac{1-\alpha}{1-\delta_2} \tag{D.1}$$

which results in

$$\alpha = \frac{\delta - \delta_2}{\delta_1 - \delta_2} \cdot \frac{1-\delta_1}{1-\delta}. \tag{D.2}$$

Note that the overall average left and right degrees of the message and parities are simply the time-shared averages of the corresponding values for the individual d.d. pairs.

APPENDIX E

CUTSET BOUNDS FOR THE MARC

We prove Lemma 1. If $N_{\mathcal{S}}$ (the number of bits transmitted successfully from $\mathcal{S}$ to $\mathcal{D}$) is strictly less than $k_{\mathcal{S}}$ (the number of information bits originating in $\mathcal{S}$), then the block error probability is at least 1/2, owing to the many-to-one nature of the decoding rule. Consequently,

$$p_e^{(n)}(\mathcal{S}) \geq \frac{1}{2} \cdot \mathbb{P}(N_{\mathcal{S}} < k_{\mathcal{S}}) \tag{E.1}$$

In the following, we show that if $\mathbb{E}(N_{\mathcal{S}}) < k_{\mathcal{S}}$, then $\mathbb{P}(N_{\mathcal{S}} < k_{\mathcal{S}}) \geq \beta$, where $\beta > 0$ is independent of $\{k_i\}_{i=1}^{M}$, $\{n_i\}_{i=1}^{M}$ and n_r, thus proving the lemma.

Let $\{n_i\}_{i=1}^{M}$ and n_r be such that $\mathbb{E}(N_{\mathcal{S}}) < k_{\mathcal{S}}$. For every node $\gamma \in \mathcal{S}$, denote the number of bits transmitted by γ as n_γ; each of those transmitted bits has a probability p_γ of being successfully received by at least one node in $\mathcal{D}$, and so if we let N_γ denote a random variable indicating the number of successful transmissions then $\{N_\gamma\}_{\gamma\in\mathcal{S}}$ are independently distributed *binomial* random variables with parameters (n_γ, p_γ). Further, $N_{\mathcal{S}} = \sum_{\gamma\in\mathcal{S}} N_\gamma$.

Assume that none of the random variables in $\{N_\gamma\}_{\gamma\in\mathcal{S}}$ are degenerate[1] - i.e., $n_\gamma > 0$ and $0 < p_\gamma < 1$ for each $\gamma \in \mathcal{S}$. For a non-degenerate binomial random variable N with parameters (n, p), let m denote the median of N, i.e, $m \triangleq \min\{i :$

[1] If N_γ *is* degenerate for some $\gamma \in \mathcal{S}$, then $N_\gamma = \mathbb{E}(N_\gamma)$ with probability one, and the proof of the Lemma is still valid.

$\mathbb{P}(N \leq i) > 1/2\}$. It is well known that the mean and median of such a random variable differ by less than one (cf. [47]), i.e., $|\mathbb{E}(N) - m| < 1$; in particular, $\lfloor np \rfloor + 1 \geq m$. Thus,

$$\begin{aligned}
\mathbb{P}\big(N \leq \mathbb{E}(N)\big) &\geq \mathbb{P}(N \leq \lfloor np \rfloor) && \text{(E.2)} \\
&= \mathbb{P}(N \leq \lfloor np \rfloor + 1) \cdot \frac{\lfloor np \rfloor + 1}{n - \lfloor np \rfloor} \cdot \frac{1-p}{p} && \text{(E.3)} \\
&\geq \mathbb{P}(N \leq \lfloor np \rfloor + 1) \cdot \frac{np}{n(1-p)+1} \cdot \frac{1-p}{p} && \text{(E.4)} \\
&\geq \mathbb{P}(N \leq \lfloor np \rfloor + 1) \cdot \frac{1-p}{2-p} && \text{(E.5)} \\
&> \frac{1}{2} \cdot \frac{1-p}{2-p}. && \text{(E.6)}
\end{aligned}$$

And so

$$\begin{aligned}
\mathbb{P}(N_{\mathcal{S}} < k_{\mathcal{S}}) &\geq \mathbb{P}\big(N_{\mathcal{S}} \leq \mathbb{E}(N_{\mathcal{S}})\big) && \text{(E.7)} \\
&\geq \prod_{\gamma \in \mathcal{S}} \mathbb{P}\big(N_\gamma \leq \mathbb{E}(N_\gamma)\big) && \text{(E.8)} \\
&> \prod_{\gamma \in \mathcal{S}} \frac{1}{2} \cdot \frac{1-p_\gamma}{2-p_\gamma} \qquad (> 0) && \text{(E.9)}
\end{aligned}$$

which proves the lemma.

APPENDIX F

THE α_ℓ-CHANNEL ALLOCATION FOR THE MRC

We prove Lemma 4. It is straightforward to verify that (5.26) is the solution to (5.24) and (5.25):

1. $\{n_i\}_{i=\ell+1}^{M} = 0$ from (5.25),
2. n_s is directly obtained from (5.24) with $m = \ell$,
3. for $1 \leq i \leq \ell$, n_i can be computed from the pair of equations of (5.24) with $m = i$ and $m = (i-1)$, by subtracting both sides of one equation from the other.

The rate C_ℓ can then be computed as the ratio $k/(n_s + \sum_{i=1}^{M} n_i)$. We now show that (5.26) satisfies (5.22).

For any $\mathcal{I} \subseteq \mathcal{R}$, let $\mathcal{J} = \mathcal{I} \cap \{1, 2, \cdots, \ell\} = \{i_1, i_2, \cdots i_J\}$, where $J = |\mathcal{J}|$ and $i_1 < i_2 < \cdots < i_J$. Also, let $\mathcal{J}' = \mathcal{I}^c \cap \{1, 2, \cdots, \ell\}$. Then, it follows that

$$n_s(1 - \epsilon_{sd} \prod_{j \in \mathcal{I}^c} \epsilon_{sj}) + \sum_{j \in \mathcal{I}} n_j(1 - \epsilon_{jd}) \;\geq\; n_s(1 - \epsilon_{sd} \prod_{j \in \mathcal{J}'} \epsilon_{sj}) + \sum_{j \in \mathcal{J}} n_j(1 - \epsilon_{jd}). \quad \text{(F.1)}$$

When $\mathcal{J} = \phi$, we have $\mathcal{J}' = \{1, 2, \cdots, \ell\}$; therefore, it follows from (5.24) with $m = \ell$ and (5.25) that the right-hand-side (RHS) of (F.1) is at least k. Likewise, when $\mathcal{J}' = \phi$, we have $\mathcal{J} = \{1, 2, \cdots, \ell\}$ and it again follows from (5.24) with

$m = 0$ and (5.25) that the RHS of (F.1) is no smaller than k. Thus, in both cases, (5.22) is satisfied.

Now consider the case when neither $\mathcal{J}$ nor $\mathcal{J}'$ is empty. In this case:

$$\begin{aligned}
\prod_{j \in \mathcal{J}'} \epsilon_{sj} &= \prod_{j=1}^{i_1-1} \epsilon_{sj} \cdot \prod_{j \in \mathcal{J}', j > i_1} \epsilon_{sj} && \text{(F.2)} \\
&= \prod_{j=1}^{i_1-1} \epsilon_{sj} \cdot \big(\epsilon_{si_1} + (1 - \epsilon_{si_1})\big) \cdot \prod_{j \in \mathcal{J}', j > i_1} \epsilon_{sj} && \text{(F.3)} \\
&= \prod_{j=1}^{i_2-1} \epsilon_{sj} \cdot \prod_{j \in \mathcal{J}', j > i_2} \epsilon_{sj} \; + \; \prod_{j=1}^{i_1-1} \epsilon_{sj} \cdot (1 - \epsilon_{si_1}) \prod_{j \in \mathcal{J}', j > i_1} \epsilon_{sj} && \text{(F.4)} \\
&\leq \prod_{j=1}^{i_2-1} \epsilon_{sj} \cdot \prod_{j \in \mathcal{J}', j > i_2} \epsilon_{sj} \; + \; (1 - \epsilon_{si_1}) \prod_{j=1}^{i_1-1} \epsilon_{sj} && \text{(F.5)}
\end{aligned}$$

Likewise,

$$\begin{aligned}
\prod_{j=1}^{i_2-1} \epsilon_{sj} \cdot \prod_{j \in \mathcal{J}', j > i_2} \epsilon_{sj} &= \prod_{j=1}^{i_2-1} \epsilon_{sj} \cdot \big(\epsilon_{si_2} + (1 - \epsilon_{si_2})\big) \cdot \prod_{j \in \mathcal{J}', j > i_2} \epsilon_{sj} && \text{(F.6)} \\
&\leq \prod_{j=1}^{i_3-1} \epsilon_{sj} \cdot \prod_{j \in \mathcal{J}', j > i_3} \epsilon_{sj} \; + \; (1 - \epsilon_{si_2}) \prod_{j=1}^{i_2-1} \epsilon_{sj} && \text{(F.7)}
\end{aligned}$$

Therefore, by induction,

$$\prod_{j \in \mathcal{J}'} \epsilon_{sj} \;\leq\; \prod_{j=1}^{\ell} \epsilon_{sj} \; + \; \sum_{m=1}^{J} (1 - \epsilon_{si_m}) \prod_{j=1}^{i_m-1} \epsilon_{sj}. \tag{F.8}$$

This implies that

$$n_s(1 - \epsilon_{sd} \prod_{j \in \mathcal{J}'} \epsilon_{sj}) \;\geq\; n_s(1 - \epsilon_{sd} \prod_{j=1}^{\ell} \epsilon_{sj}) \; - \; \sum_{m=1}^{J} n_s \epsilon_{sd} (1 - \epsilon_{si_m}) \prod_{j=1}^{i_m-1} \epsilon_{sj} \tag{F.9}$$

Further, noting the values of $\{n_j\}_{j\le\ell}$ from (5.26), we have

$$\begin{aligned} \sum_{j\in\mathcal{J}} n_j(1-\epsilon_{jd}) &= \sum_{m=1}^{J} n_{i_m}(1-\epsilon_{i_m d}) && \text{(F.10)} \\ &= \sum_{m=1}^{J} n_s\epsilon_{sd}(1-\epsilon_{si_m})\prod_{j=1}^{i_m-1}\epsilon_{sj}. && \text{(F.11)} \end{aligned}$$

Consequently,

$$\begin{aligned} n_s(1-\epsilon_{sd}\prod_{j\in\mathcal{J}'}\epsilon_{sj})+\sum_{j\in\mathcal{J}} n_j(1-\epsilon_{jd}) &\ge n_s(1-\epsilon_{sd}\prod_{j=1}^{\ell}\epsilon_{sj}), \\ &= k, && \text{(F.12)} \end{aligned}$$

where the last equality follows from (5.26). Thus, (F.1) and (F.12) imply that (5.26) satisfies (5.22), concluding the proof.

APPENDIX G

PROPERTIES OF THE α_ℓ-CHANNEL ALLOCATION RATES

We prove Lemma 5. We first prove the following propositions:

(a) $C_{\ell-1} \ \leq \ C_\ell \iff C_{\ell-1} \ \leq \ 1-\epsilon_{\ell d}, \quad 1 \leq \ell \leq M$

(b) $C_\ell \ \leq \ 1-\epsilon_{\ell d} \iff C_{\ell-1} \ \leq \ 1-\epsilon_{\ell d}, \quad 1 \leq \ell \leq M$

(c) $C_{\ell-1} \ \leq \ 1-\epsilon_{\ell d} \implies C_{\ell-1} \ \leq \ 1-\epsilon_{(\ell-1)d}, \quad 2 \leq \ell \leq M$

Define A_ℓ, B_ℓ and D_ℓ as follows:

$$A_\ell \ = \ 1-\epsilon_{sd}\prod_{i=1}^{\ell}\epsilon_{si}, \quad 0 \leq \ell \leq M \tag{G.1}$$

$$B_\ell \ = \ 1+\sum_{j=1}^{\ell}\Big(\frac{\epsilon_{sd}(1-\epsilon_{sj})}{1-\epsilon_{jd}}\cdot\prod_{i=1}^{j-1}\epsilon_{si}\Big), \quad 0 \leq \ell \leq M \tag{G.2}$$

$$D_\ell \ = \ \epsilon_{sd}(1-\epsilon_{s\ell})\cdot\prod_{i=1}^{\ell-1}\epsilon_{si} \quad 1 \leq \ell \leq M \tag{G.3}$$

Then, it follows that

$$C_\ell \ = \ \frac{A_\ell}{B_\ell}, \quad 0 \leq \ell \leq M \tag{G.4}$$

$$A_\ell \ = \ A_{\ell-1}+D_\ell, \quad 1 \leq \ell \leq M \tag{G.5}$$

$$B_\ell \ = \ B_{\ell-1}+\frac{D_\ell}{1-\epsilon_{\ell d}}, \quad 1 \leq \ell \leq M. \tag{G.6}$$

Now, we have

$$\begin{aligned} C_{\ell-1} \;\le\; C_\ell \;\Longleftrightarrow\; & \frac{A_{\ell-1}}{B_{\ell-1}} \;\le\; \frac{A_\ell}{B_\ell} \\ \Longleftrightarrow\; & \frac{A_{\ell-1}}{B_{\ell-1}} \;\le\; \frac{A_{\ell-1}+D_\ell}{B_{\ell-1}+\frac{D_\ell}{1-\epsilon_{\ell d}}} \\ \Longleftrightarrow\; & \frac{A_{\ell-1}}{B_{\ell-1}} \;\le\; 1-\epsilon_{\ell d}, \quad \text{for } 1 \le \ell \le M \qquad \text{(G.7)} \end{aligned}$$

which proves (a). Note from (G.5) and (G.6) that

$$\begin{aligned} B_\ell \;&=\; B_{\ell-1} + \frac{A_\ell - A_{\ell-1}}{1-\epsilon_{\ell d}} && \text{(G.8)} \\ &=\; \frac{A_\ell \;+\; B_{\ell-1}\cdot\big((1-\epsilon_{\ell d}) - C_{\ell-1}\big)}{1-\epsilon_{\ell d}} \quad \text{for } 1 \le \ell \le M && \text{(G.9)} \end{aligned}$$

Thus,

$$C_\ell \;=\; (1-\epsilon_{\ell d}) \cdot \frac{A_\ell}{A_\ell \;+\; B_{\ell-1}\cdot\big((1-\epsilon_{\ell d}) - C_{\ell-1}\big)} \qquad \text{(G.10)}$$

which yields (b). Proposition (c) follows from the fact that $\epsilon_{(\ell-1)d} \;\le\; \epsilon_{\ell d}$.

The proofs of (a), (b) and (c) also yield the following propositions:

(d) $C_{\ell-1} \;\ge\; C_\ell \Longleftrightarrow C_{\ell-1} \;\ge\; 1-\epsilon_{\ell d}, \quad 1 \le \ell \le M$

(e) $C_\ell \;\ge\; 1-\epsilon_{\ell d} \Longleftrightarrow C_{\ell-1} \;\ge\; 1-\epsilon_{\ell d}, \quad 1 \le \ell \le M$

(f) $C_\ell \;\ge\; 1-\epsilon_{\ell d} \Longrightarrow C_\ell \;\ge\; 1-\epsilon_{(\ell+1)d}, \quad 1 \le \ell \le M-1$

(Note that (d) and (e) are *not* the exact contrapositives of (a) and (b).)

Now, from (a) and (c), it follows that:

$$C_{\ell-1} \;\le\; C_\ell \;\Longrightarrow\; C_{\ell-1} \;\le\; 1-\epsilon_{(\ell-1)d}, \quad 2 \le \ell \le M \qquad \text{(G.11)}$$

From (a) and (b), it follows that:

$$C_\ell \;\le\; 1-\epsilon_{\ell d} \iff C_{\ell-1} \;\le\; C_\ell, \quad 1 \le \ell \le M \tag{G.12}$$

By applying (G.11) and (G.12) recursively, it follows that:

$$\begin{aligned} C_{\ell-1} \;\le\; C_\ell \implies\; & C_{j-1} \;\le\; C_j \;\le\; (1-\epsilon_{jd}), \\ & 1 \le j \le \ell, \quad 1 \le \ell \le M \end{aligned} \tag{G.13}$$

Using the fact that $C_{\ell^*-1} < C_{\ell^*}$ for $\ell^* \ge 1$, we have (5.29).

Likewise, from (f) and (d), we have:

$$C_\ell \;\ge\; 1-\epsilon_{\ell d} \implies C_\ell \;\ge\; C_{\ell+1}, \quad 1 \le \ell \le M-1 \tag{G.14}$$

From (d) and (e), we have:

$$C_{\ell-1} \;\ge\; C_\ell \iff C_\ell \;\ge\; 1-\epsilon_{\ell d}, \quad 1 \le \ell \le M \tag{G.15}$$

Again, applying (G.14) and (G.15) recursively yields

$$\begin{aligned} C_{\ell-1} \;\ge\; C_\ell \implies\; & C_{j-1} \;\ge\; C_j \;\ge\; (1-\epsilon_{jd}), \\ & \ell \le j \le M, \quad 1 \le \ell \le M \end{aligned} \tag{G.16}$$

Finally, since $C_{\ell^*} \ge C_{\ell^*+1}$ for $\ell^* \le M-1$, we have (5.30).

For the case when $\ell^* \ne 0$, we know from the definition of ℓ^* that $C_{\ell^*-1} < C_{\ell^*}$. Consequently, from (G.7), it follows that $C_{\ell^*-1} < (1-\epsilon_{\ell^* d})$. Using this in (G.10), we have $C_{\ell^*} < (1-\epsilon_{\ell^* d})$, leading to (5.31).

APPENDIX H

EVALUATING THE SLACK COEFFICIENTS $\{\gamma_m\}_{m=0}^M$ IN THE PROOF TO THEOREM 3

We evaluate the coefficients $\{\gamma_m\}_{m=0}^M$ in (5.36) by solving for $n_s, \{n_m\}_{m=1}^M$ from (5.34) and (5.35). We solve separately for the two cases $\ell^* = 0$ and $\ell^* > 0$.

Case 1: $\ell^* = 0$

For this case, $\{n_m\}_{m\geq 1}$are given by (5.35) to be $n_m = Z_m$ for $m = 1, 2, \ldots, M$. Therefore, setting $m = 0$ in (5.34), we have:

$$n_s = \frac{k + Z_0 - \sum_{j=1}^M Z_j(1 - \epsilon_{jd})}{1 - \epsilon_{sd}} \tag{H.1}$$

Thus, the objective function may be written as

$$n_s + \sum_{m=1}^M n_m \;=\; \frac{k}{1 - \epsilon_{sd}} + \frac{Z_0}{1 - \epsilon_{sd}} + \sum_{m=1}^M Z_m \cdot \frac{\epsilon_{md} - \epsilon_{sd}}{1 - \epsilon_{sd}}. \tag{H.2}$$

This leads to (5.37).

Case 2: $\ell^* \geq 1$

As in the previous case, $\{n_m\}_{m\geq \ell^*+1}$ are directly given by (5.35). Setting $m = \ell^*$ in (5.34), we have:

$$n_s \;=\; \frac{1}{1 - \epsilon_{sd}\prod_{i=1}^{\ell^*} \epsilon_{si}} \cdot \Big(k + Z_{\ell^*} - \sum_{j=\ell^*+1}^M Z_j(1 - \epsilon_{jd})\Big). \tag{H.3}$$

We can solve for $\{n_{m'}\}_{m'=1}^{\ell*}$ in terms of n_s by subtracting (5.34) with $m = m'$ from (5.34) with $m = m' - 1$:

$$n_{m'} \;=\; \frac{1}{1-\epsilon_{m'd}} \cdot \left(n_s \epsilon_{sd} (1-\epsilon_{sm'}) \cdot \prod_{i=1}^{m'-1} \epsilon_{si} + Z_{m'-1} - Z_{m'} \right), \quad 1 \le m' \le \ell^*. \tag{H.4}$$

Thus, the objective function may be written as

$$\begin{aligned} n_s + \sum_{m=1}^{M} n_m \;&=\; n_s \cdot \left(1 + \sum_{m=1}^{\ell^*} \frac{\epsilon_{sd}(1-\epsilon_{sm})}{1-\epsilon_{md}} \cdot \prod_{i=1}^{m-1} \epsilon_{si} \right) \;+\; \sum_{m=1}^{\ell^*} \frac{Z_{m-1} - Z_m}{1-\epsilon_{md}} \\ &\quad + \sum_{m=\ell^*+1}^{M} Z_m \end{aligned} \tag{H.5}$$

Since $\ell^* \ge 1$, we can write:

$$\sum_{m=1}^{\ell^*} \frac{Z_{m-1} - Z_m}{1-\epsilon_{md}} \;=\; \frac{Z_0}{1-\epsilon_{1d}} + \sum_{m=1}^{\ell^*-1} Z_m \Big(\frac{1}{1-\epsilon_{(m+1)d}} - \frac{1}{1-\epsilon_{md}} \Big) - \frac{Z_{\ell^*}}{1-\epsilon_{\ell^* d}} \tag{H.6}$$

From this, on substituting for n_s from (H.3) and recalling the definition of C_ℓ from (5.27), we finally have after some simplification:

$$\begin{aligned} n_s + \sum_{m=1}^{M} n_m \;&=\; \frac{k}{C_{\ell^*}} \;+\; \frac{Z_0}{1-\epsilon_{1d}} \;+\; \sum_{m=1}^{\ell^*-1} Z_m \Big(\frac{1}{1-\epsilon_{(m+1)d}} - \frac{1}{1-\epsilon_{md}} \Big) \\ &\quad + Z_{\ell^*} \Big(\frac{1}{C_{\ell^*}} - \frac{1}{1-\epsilon_{\ell^* d}} \Big) \;+\; \sum_{m=\ell^*+1}^{M} Z_m \Big(1 - \frac{1-\epsilon_{md}}{C_{\ell^*}} \Big) \end{aligned} \tag{H.7}$$

This leads to (5.38).

BIBLIOGRAPHY

1. P. Elias, “Coding for two noisy channels,” in *Proc. 3rd London Symp. Inform. Theory.* London, U.K.: Academic Press, 1956, pp. 61–76.

2. T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, 1991.

3. I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Ind. Appl. Math.*, vol. 8, pp. 300–304, Jun. 1960.

4. M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient erasure correcting codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

5. R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

6. P. Oswald and A. Shokrollahi, “Capacity-achieving sequences for the erasure channel,” *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3017–3028, Dec. 2002.

7. H. Jin, A. Khandekar, and R. J. McEliece, “Irregular repeat-accumulate codes,” in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sep. 2000, pp. 1–8.

8. I. Sason and R. Urbanke, “Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel,” *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1247–1256, Jun. 2004.

9. H. D. Pfister, I. Sason, and R. Urbanke, “Capacity-achieving ensembles for the binary erasure channel with bounded complexity,” *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2352–2379, Jul. 2005.

10. H. D. Pfister and I. Sason, “Accumulate-repeat-accumulate codes: Capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity,” *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2088–2115, Jun. 2007.

11. M. Luby, “LT codes,” in *Proc. of the 43rd Annual IEEE Symp. on Foundations of Comp. Sc.*, Vancouver, Canada, Nov. 2002, pp. 271–280.

12. A. Shokrollahi, “Raptor codes,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

13. D. Bertsekas and R. Gallager, *Data networks (2nd ed.).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

14. C. Huitema, “The case for packet level FEC,” in *Proc. IFIP 5th Int. Workshop on Protocols for High-Speed Networks*, Sophia Antipolis, France, Oct. 1996, pp. 109–120.

15. L. Rizzo and L. Vicisano, “A reliable multicast data distribution protocol based on software fec techniques,” in *Proc. 4th IEEE Workshop on High-Performance Communication Systems*, Chalkidiki, Greece, Jun. 1997, pp. 116–125.

16. J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data,” in *Proc. of ACM SIGCOMM*, Vancouver, Canada, Aug. 1998, pp. 56–67.

17. J. W. Byers, M. Luby, and M. Mitzenmacher, “A digital fountain approach to asynchronous reliable multicast,” *IEEE J. Sel. Areas in Commun.*, vol. 20, no. 8, pp. 1528–1540, Aug. 2002.

18. J. Nonnenmacher, E. W. Biersack, and D. Towsley, “Parity-based loss recovery for reliable multicast transmission,” *IEEE/ACM Trans. Netw.*, vol. 6, no. 4, pp. 349–361, Aug. 1998.

19. A. Beimel, S. Dolev, and N. Singer, “RT oblivious erasure correcting,” *IEEE/ACM Trans. Networking*, vol. 15, no. 6, pp. 1321–1332, Dec. 2007.

20. X. Zhang. and Q. Du, “Adaptive low-complexity erasure-correcting code-based protocols for QoS-driven mobile multicast services over wireless networks,” *IEEE Trans. Vehicular Technology*, vol. 55, no. 5, pp. 1633–1647, Sep. 2006.

21. K. Nagasubramanian, *Code design for erasure channels with limited or noisy feedback.* M.S. thesis, Texas A & M University, Dec. 2007.

22. C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, “Finite length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

23. J. K. Sundararajan, D. Shah, and M. Medard, “ARQ for network coding,” in *Proc. IEEE Int. Symposium on Inform. Theory*, Toronto, Canada, Jul. 2008, pp. 1651–1655.

24. S. Lin, D. J. Costello, and M. J. Miller, “Automatic-repeat-request error-control schemes,” *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5–17, Dec. 1984.

25. E. C. van der Meulen, “Three-terminal communication channels,” *Adv. Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.

26. T. Cover and A. A. E. Gamal, “Capacity theorems for the relay channel,” *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 572–584, Sep. 1979.

27. G. Kramer and A. J. van Wijngaarden, “On the white Gaussian multiple-access relay channel,” in *Proc. IEEE Int. Symposium on Inform. Theory*, Sorrento, Italy, Jun. 2000, p. 40.

28. L. Sankaranarayanan, G. Kramer, and N. B. Mandayam, “Capacity theorems for the multiple-access relay channel,” in *Proc. of the 42nd Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Sep. 2004, pp. 1782–1791.

29. G. Kramer, M. Gastpar, and P. Gupta, “Cooperative strategies and capacity theorems for relay networks,” *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3037–3063, Sep. 2005.

30. L.-L. Xie and P. R. Kumar, “Multisource, multidestination, multirelay wireless networks,” *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3586–3595, Oct. 2007.

31. A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, “Capacity of wireless erasure networks,” *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 789–804, Mar. 2006.

32. D. S. Lun, M. Medard, R. Koetter, and M. Effros, “On coding for reliable communication over packet networks,” *Physical Communication*, vol. 1, no. 1, pp. 3–20, Mar. 2008.

33. R. Khalili and K. Salamatian, “A tighter cut-set bound for the multi-terminal erasure channel without side information,” in *Proc. IEEE Int. Symposium on Inform. Theory*, Seattle, WA, Jul. 2006, pp. 1876–1880.

34. ——, “On the capacity of erasure relay channel: multi-relay case,” in *Proc. IEEE Inform. Theory Workshop*, Rotorua, New Zealand, Aug.-Sept. 2005, pp. 103–107.

35. K. Salamatian and R. Khalili, “An information theory for the erasure channel,” in *Proc. of the 43rd Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Sep. 2005.

36. R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

37. R. M. Gray, *Conditional Rate-Distortion Theory.* Stanford University: Technical Report No. 6502-2, Information Systems Laboratory, Oct. 1972.

38. Y. Matsunaga and H. Yamamoto, “A coding theorem for lossy data compression by LDPC codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2225–2229, Sep. 2003.

39. E. Martinian and M. Wainwright, “Low density codes achieve the rate-distortion bound,” in *Proc. 2006 Data Compression Conf. (DCC 2006)*, Snowbird, UT, Mar. 2006, pp. 153–162.

40. S. Miyakei and J. Muramatsu, “Construction of a lossy source code using LDPC matrices,” in *Proc. IEEE Int. Symposium on Inform. Theory*, Nice, France, Jun. 2007, pp. 1106–1110.

41. A. Gupta and S. Verdu, “Nonlinear sparse-graph codes for lossy compression,” *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 1961–1975, May 2009.

42. J. Honda and H. Yamamoto, “Variable length lossy coding using an LDPC code,” in *Proc. IEEE Int. Symposium on Inform. Theory*, Seoul, Korea, Jul. 2009, pp. 1973–1977.

43. K. Sayood, *Introduction to data compression (3rd edn.).* San Francisco: Morgan-Kaufmann Publishers, 2006.

44. R. Webster, *Convexity.* New York: Oxford University Press, 1994.

45. J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior,” *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

46. J. N. Laneman and G. W. Wornell, “Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks,” *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.

47. K. Hamza, “The smallest uniform upper bound on the distance between the mean and the median of the binomial and Poisson distributions,” *Statistics and Probability Letters*, vol. 23, no. 1, pp. 21–25, 1995.