5-2018

On Coding for Partial Erasure Channels

Carolyn Mayer
University of Nebraska - Lincoln, cmayer@huskers.unl.edu

ON CODING FOR PARTIAL ERASURE CHANNELS

by

Carolyn Mayer

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Mathematics

Under the Supervision of Professor Christine A. Kelley

Lincoln, Nebraska

May, 2018

ON CODING FOR PARTIAL ERASURE CHANNELS

Carolyn Mayer, Ph.D.

University of Nebraska, 2018

Adviser: Christine A. Kelley

Error correcting codes have been essential to the technology we use in everyday life in digital storage, wireless communication, barcodes, and much more. Different channel models are used for different types of communication (for example, if information is sent to one person or to many people) and different types of errors. Partial erasure channels were recently introduced to model applications in which some information remains after an erasure event. These remnants of information may be used to increase the chances of successful decoding. We introduce a new partial erasure channel in which partial erasures correspond to individual bit erasures in the binary expansion of a 2^k-ary symbol or p-ary symbols in the expansion of a p^k-ary symbol. We show how multilevel coding and multistage decoding may be used on partial erasure channels and investigate cases in which partial erasure channels may be decomposed into simpler channels. Further, we show that partial erasure channels do not always decompose into simple erasure channels, and that when they do, the erasure channels may not be independent. The rest of this work focuses on three areas: fountain codes on partial erasure channels, relay channels with partial erasures, and graph-based codes for distributed storage. We adapt a class of fountain codes for use on partial erasure channels and show an improvement in terms of the number of symbols that must be generated for the successful decoding of such codes. In a relay channel setting, we consider a simple three node system with a sender, receiver, and relay where at least one of the links is a partial erasure channel. When the sender-receiver link is a degraded version of the sender-relay link, we determine the

capacity of the channel. We also introduce a biregular hypergraph construction and find locality properties of codes based on these hypergraphs.

PREFACE

The results presented in Chapter 3 and Chapter 4 were published in *Advances in Mathematics of Communications*. (C. Mayer, K. Haymaker, and C. A. Kelley, "Channel Decomposition for Multilevel Codes Over Multilevel and Partial Erasure Channels" vol. 12, pp. 151–168, 2018. ©American Institute of Mathematical Sciences 2018). [1]

The material in Chapter 5 was presented at the 2017 IEEE International Symposium on Information Theory (ISIT) and appeared in the conference proceedings. (C. Mayer and C. A. Kelley, "LT Codes on Partial Erasure Channels," *2017 IEEE International Symposium on Information Theory (ISIT),* July 2017. ©2017, IEEE). [2]

Parts of Chapter 7 were presented at the 5^{th} International Castle Meeting on Coding Theory and Applications. (A. Beemer, C. Mayer, and C. A. Kelley, "Erasure Correction and Locality of Hypergraph Codes," In: Barbero Á., Skachek V., Ytrehus Ø. (eds) Coding Theory and Applications. ICMCTA 2017. *Lecture Notes in Computer Science,* vol 10495. Springer. 2017. ©Springer International Publishing AG 2017). [3]

Table of Contents

Chapter 1

Introduction

Modern applications have fueled the development of new types of erasure correcting codes over the past several decades. Examples of such applications include flash memory storage, wireless communication, packet-based internet transmission, and distributed storage. Using partial information has been a topic of interest in recent years, with research in areas such as *regenerating codes* in which encoding nodes store multiple symbols and erasures are recovered by downloading a subset of symbols from accessed nodes ([4, 5]) and *fractional decoding* for which a decoder may only download a fixed proportion of a codeword ([6, 7]). Applications in information storage have also motivated the recent introduction of partial erasure channels, which aim to model situations in which some information remains after an erasure event occurs [8, 9]. In this dissertation, we develop the theory of these channels further. In particular, we introduce a new partial erasure channel model, derive the capacity of this new channel, and examine the decomposition of partial erasure channels. We also look at coding applications such as fountain codes and relay channels in this context.

Graph-based codes have also been in the forefront of research in coding theory due to their efficient iterative decoders [10, 11, 12, 13, 14]. In Chapter 2, we give a brief overview of parts of classical coding theory necessary to understand the results we present. This includes the definition of a linear code and an explanation of mutual information and

capacity. We also review *Low-Density Parity-Check (LDPC) codes*, a family of graph-based codes that are widely used in applications today. Finally, we give a survey of the partial erasure channels introduced in [8, 9]: the *q-ary Partial Erasure Channel (QPEC)* and the *q-ary Multi-bit Channel (QMBC)*.

The goal of partial erasure channels is to handle cases in which some information remains after an erasure event. The QPEC has partial erasure sets of a fixed size, with each possible set of a given size containing the original symbol being equally likely [8]. The QMBC has partial erasures corresponding to bursts of bit erasures, each starting with the rightmost bit [9]. In Chapter 3, we present the *Multilevel Erasure Channel (MEC)*, a new partial erasure channel in which partial erasures correspond to bit erasures in the binary representation (or, more generally, p-ary symbols in the p-ary representation) of a symbol that is transmitted. The MEC may be applied as a natural model of erasures in multilevel flash memory storage, where erasures at the digit level may occur with different probabilities, depending on the system's storage architecture. In 1948, Claude Shannon introduced a mathematical theory of communication [15]. Part of this theory includes the study of the *capacity* of a channel, or how much information may be sent across the channel reliably. Along with the MEC channel model, we derive the capacity of the MEC.

In Chapter 4, we present results on multilevel coding and multistage decoding in the partial erasure setting. Multilevel coding and mutistage decoding were introduced in [16] as a tool for using side-information during decoding. The goal is to communicate more bits per channel use reliably by using this information. Using a multilevel coding scheme on the MEC, we are able to prove the capacity result presented in Chapter 3 and show that the MEC may be decomposed into simple independent Binary Erasure Channels (BEC). Applying a similar scheme on the QPEC, we see that while there are cases in which the QPEC decomposes into simple erasure channels, it does not always

do so. Furthermore, when the QPEC does decompose in simple erasure channels, they may not be independent. When the QPEC decomposes into simpler channels, this may simplify decoding analysis.

In Chapter 5 we show how *Luby Transform (LT)* codes may be used in a partial erasure setting. LT codes were introduced in [17] as a realization of *fountain codes*. Fountain codes are rateless in the sense that for a given set of information symbols, there is no limit on the number of encoding symbols produced. Such codes have many applications in settings where a message must be sent to multiple receivers or reception may be interrupted. Encoding symbols are generated until enough symbols are received for every receiver in the system to decode the message. We show that by taking advantage of the information left in a partial erasure, an LT code on the MEC requires fewer symbols to be generated than on the corresponding QEC.

In Chapter 6 we investigate relay networks with partial erasures. Relay channels were introduced in [18], and capacity bounds in various cases were derived in [19]. In the simple relay setting we consider, information is sent from the sender to the receiver both directly and through a relay node. Information from the sender and from the relay may be combined at the receiver to assist with decoding. We extend the cut-set bound of [19] to the setting in which the sender-relay link is a MEC and the other links are QECs. We also show that the cut-set bound may be achieved by giving an asymptotic analysis of how the codes may be chosen.

Finally, we present results on hypergraph codes in Chapter 7. Codes from regular hypergraphs with expansion-like properties were introduced and analyzed in [20, 21]. In distributed storage, codes in which erasures can be recovered by accessing a small number of code symbols rather than the full code word are of high interest. There are a variety of parameters that aim to measure this property. We consider the *locality* of a code (which measures the number of symbols that must be accessed to recover an erasure) and

the *availability* of a code (which is the number of disjoint repair sets there are for each symbol) [22, 23]. We introduce a construction for biregular a biregular hypergraph and consider the locality and availability of the resulting hypergraph code.

Chapter 2

Preliminaries

The fields of coding theory and information theory began with Shannon's *A Mathematical Theory of Communication* [15]. Shannon's paper introduced a model for the exchange of information as well as a framework for how to measure the amount of information conveyed between a sender and a receiver. In particular, Shannon introduced the notion of a channel and gave probabilistic proofs of the existence of good codes. We begin with a review of the basics of linear codes and channel coding. We also present background on Low-Density Parity-Check codes. We will end the section with an overview of partial erasure channels.

2.1 Linear Codes

There are several applications in which information must be sent or stored, such as internet communication, servers in a network, wireless communications and many, many more. When information is sent or stored, it is subject to various types of errors due to noise on the channel (for example, interfering signals or charge leakage in flash memories). Messages can be encoded with error-correcting codes to assist with information recovery if an error occurs.

Definition 1. A *code* $\mathcal{C}$ of length n over an alphabet A is a subset $\mathcal{C} \subseteq A^n$. If $\mathcal{C}$ is a

k-dimensional subspace of $\mathrm{GF}(q)^n$, then $\mathcal{C}$ is an $[n,k]_q$ *linear code*.

In an $[n,k]_q$ code, k *information symbols* are encoded to a length n word called a *codeword*. The higher the ratio of information symbols to code word symbols, the more efficient the code is in terms of symbols used. This efficiency is measured by the *rate* of the code.

Definition 2. The *rate* of an $[n,k]_q$ linear code is the ratio of transmitted symbols to message symbols, $r = \frac{k}{n}$.

Linear codes are commonly defined using a *generator matrix* or a *parity-check matrix*.

Definition 3. A *generator matrix* G of a linear code $\mathcal{C}$ is an $n \times k$ matrix such that

$$\mathcal{C} = \{xG | x \in \mathrm{GF}(q)^k\}.$$

Definition 4. A *parity-check matrix* H of a linear code $\mathcal{C}$ is an $(n-k) \times n$ matrix such that $Hc^T = \mathbf{0}$ if and only if c is a codeword in $\mathcal{C}$.

Note that a generator matrix is used to encode a message, and a parity-check matrix can be used to check if a given word is in the code. A generator matrix or parity-check matrix of a code is not unique, and certain generator or parity-check matrices may lead to simpler analysis of a code than others. Finding matrix representations that reveal code properties is part of the art of code design. The representation may also impact the performance of the decoder.

The *Hamming distance* between codewords can be measured by counting the number of coordinates in which the codewords differ. If two codewords are far apart, they are unlikely to be confused even if errors occurs.

Definition 5. The *minimum distance* of a code C is the smallest Hamming distance between distinct codewords. An $[n,k]_q$ code with minimum distance d is denoted $[n,k,d]_q$.

The minimum distance of a code guarantees a certain number of errors that can be corrected by decoding to the closest codeword. Note also, that if fewer than d errors have occurred, then we can detect that an error has occurred.

Theorem (Hamming '49)**.** *For a code with minimum distance d, there is a decoder that can detect that an error has occurred if there are up to $d-1$ errors and there is a decoder that can correct every pattern of up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

Codes with high minimum distance are desirable for their guaranteed error correction capabilities. However, the highest possible minimum distance of a linear code is limited by the blocklength and number of information symbols of the code.

Theorem (Singleton Bound)**.** *If C is an $[n,k,d]_q$ code, then $d \leq n-k+1$.*

Definition 6. If C is an $[n,k,d]_q$ code that meets the Singleton bound, then C is a *Maximum Distance Separable (MDS)* code.

2.2 Channel Coding

The process of transmitting or storing data may be studied using a model with three steps: encoding, transmission across a channel, and decoding. Figure 2.1 illustrates the steps most often considered when a message x is transmitted.

Definition 7. A *(communication) channel* is defined by an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, and transition probabilities $W(y \mid x) = \Pr(\text{output is } \mathbf{y} \mid \text{input is } \mathbf{x})$ for each $\mathbf{x} \in \mathcal{X},\ \mathbf{y} \in \mathcal{Y}$.

Figure 2.1: *Channel coding: A message x is encoded to a codeword c and sent through a channel. The received word r is decoded to an estimated message $\hat{x}$. Decoding is successful when $\hat{x} = x$.*

Definition 8. The probability of decoding error for a word $\mathbf{x}$, $P_{err}(\mathbf{x})$, is the probability that the message $\mathbf{x}$ is transmitted, but the received word $\mathbf{y}$ is not decoded to $\mathbf{x}$. Overall, the probability of decoding error, P_{err}, is the probability that the estimate of a message does not match the word sent.

A central goal in coding theory is to find codes that are efficient (have a high rate) and reliable (have a low P_{err}). The best possible rate for a given error tolerance depends on the channel.

2.2.1 Mutual Information and Capacity

As Shannon observed, given a channel model, it is natural to ask how much information can be reliably sent across the channel. In other words, we would like to know the highest rate a code can have if we want to send information across the channel with a low probability of error. The *entropy* of a random variable X measures the average uncertainty associated with X, or the average amount of information we gain if we learn the value of X.

Definition 9. The *entropy* (or *uncertainty*) of a random variable X taking on values $x_1, x_2, \ldots, x_n$ with probabilities $p_1, p_2, \ldots, p_n$, respectively, is

$$H(X) := -\sum_{i=1}^{n} p_i \log_2 p_i.$$

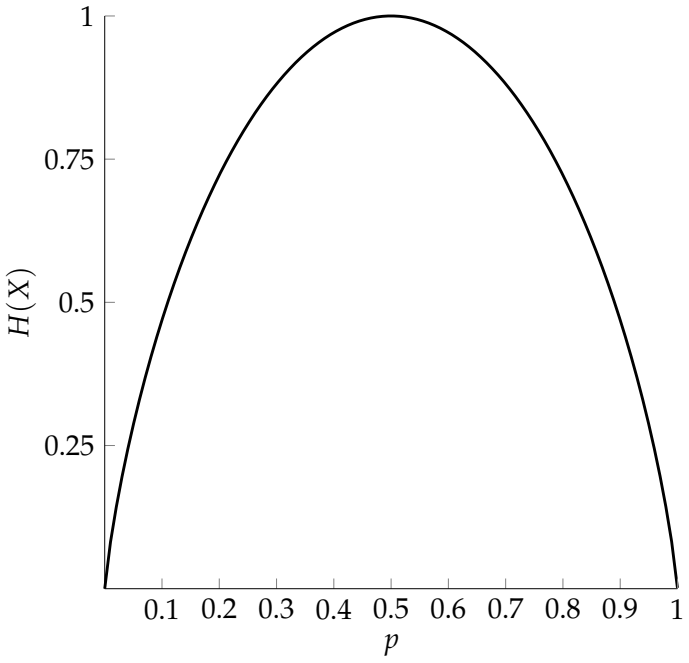


Figure 2.2: The entropy for a random variable X as a function of $p = \Pr(X = 0)$.

Example 1. Consider a binary random variable X taking value 0 with probability p and 1 with probability $(1 - p)$. The entropy function for X is

$$H(X) = -p \log_2 p - (1 - p) \log_2 (1 - p).$$

This is known as the *binary entropy function* and is show in Figure 2.2.1. Note that when the probability that $X = 0$ is either 0 or 1, there is no uncertainty. When $\Pr(X = 0) = \Pr(X = 1)$, we are completely uncertain about the outcome. Consider tossing a coin. If the coin is fair, we cannot predict whether a toss will come up heads or tails. As we look at increasingly biased coins, we become more certain about the outcome. For example, we are more confident in our prediction of the outcome of a toss if we know that heads occur with probability $\frac{99}{100}$ than we are about our prediction for a coin with heads occurring with probability $\frac{2}{3}$.

We may also consider *conditional entropy*, the amount of uncertainty about a random variable given that another random variable's value is known.

Definition 10. The *conditional entropy* of a random variable X taking on values in finite set $\mathcal{X}$ given a random variable Y taking values in a finite set $\mathcal{Y}$ is

$$H(X|Y) := -\sum_{y\in\mathcal{Y}} \Pr(y) \sum_{x\in\mathcal{X}} \Pr(X = x|Y = y) \log(\Pr(X = x|Y = y)).$$

More information and an extension to more variables can be found in [24].

When analyzing channels, we are often interested in the amount of information about the input revealed by the output. This can be measured using *mutual information*.

Definition 11. The *mutual information* between two random variables X and Y taking on values in finite sets $\mathcal{X}$ and $\mathcal{Y}$, respectively, is defined by

$$I(X;Y) := H(X) - H(X|Y) = -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \Pr(X = x, Y = y) \log\left(\frac{\Pr(X = x)}{\Pr(X = x|Y = y)}\right).$$

Intuitively, this tells us that the amount of information passed between X and Y is the difference between the amount of information we gain by learning the value of X and the amount of information we gain by learning about X provided we know the value of Y. Mutual information follows a chain rule that is often useful in calculations.

Theorem (Chain Rule of Mutual Information)**.**

$$I(X_1, X_2, \cdots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y|X_1, X_2, \ldots, X_{i-1})$$

and

$$I(X_1, X_2, \ldots, X_n; Y|Z) = \sum_{i=1}^{n} I(X_i; Y|X_1, X_2, \ldots, X_{i-1}, Z).$$

If we can maximize the mutual information between the input random variable X and output random variable Y, we maximize the amount of information we can send across the channel reliably. Consider the following theorem of Shannon.

Theorem (Shannon, 1948)**.** *For each channel, there is a* capacity *C such that for all $R < C$, and $\varepsilon > 0$, there is a code of rate R and a corresponding decoder that achieves probability of decoding error $P_{err} < \varepsilon$. For all $R > C$, there is some ε such that no such code and decoder exist.*

The capacity of a channel $\mathcal{C}$, denoted $\text{Cap}(\mathcal{C})$, is an upper bound on the rate at which information may be transmitted with arbitrarily low probability of error. The capacity of a channel can be found as the maximum over channel input distributions p_x of the information conveyed about the input X by the output Y. That is

$$\text{Cap}(\mathcal{C}) = \max_{\{p_x\}} I(X;Y) = \max_{\{p_x\}} (H(Y) - H(Y|X)).$$

Determining the capacity of different channels has been a main interest of coding theory since the field's inception [19, 24, 25, 26, 27, 28].

2.2.2 Erasure Channels

Different channels have different types of errors. These channels may be used to model a variety of applications, and the channel used depends on the type of noise for a given application. Some types of errors include *bit flips* (a bit changes parity), *deletions* (parts of a codeword are removed), *insertions* (additional symbols are added by the channel), and others. We will focus on *erasures* – bits or symbols that have been erased, but whose locations are known. For example, the loss of a server in a distributed storage system may be modeled using erasures. If a server goes down, we may not know what information was stored on the server, but we do know which server is not responding. Another

example is packet-based communication (such as sending information across the internet) in which a packet may be lost during transmission.

The Binary Erasure Channel (BEC) with erasure probability ε is a binary-input channel model in which a bit may be erased with probability ε or sent without error with probability $1-\varepsilon$ (see Figure 2.3). When an erasure occurs, the channel outputs an *erasure*, which may be represented by a symbol "?" to indicate that bit has been erased.

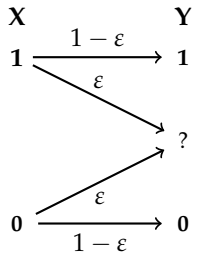


Figure 2.3: The BEC with erasure probability ε.

The q-ary Erasure Channel (QEC) is a generalization of the BEC in which a q-ary input symbol is either erased or sent without error. The transition probability of the QEC with erasure probability ε is given by

$$\Pr(Y = y \mid X = x) = \begin{cases} 1-\varepsilon & y = x \\ \varepsilon & y = ? \\ 0 & \text{else} \end{cases}.$$

Note that when a symbol is sent, it will still either be received intact (with probability $1-\varepsilon$) or it will be erased, and ? will be received (with probability ε). On the QEC, any symbol received is known to be correct. Also note that if each input symbol is equally likely to occur, then a received erasure symbol is equally likely to have been any of the possible inputs. Figure 2.2.2 shows the transition probabilities for the 3-ary erasure

channel.

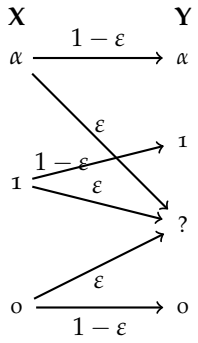


Figure 2.4: The 3-ary erasure channel with erasure probability ε.

The capacity of the QEC is $(1-\varepsilon)$ q-ary symbols per channel use [29].

2.3 Low-Density Parity-Check (LDPC) Codes

There are many different techniques for designing codes. Since Shannon first presented the problem, one goal of code design is to find capacity-approaching codes. We will focus on *Low-Density Parity-Check* codes in this section. Other notable codes include *Turbo codes* [12], *Expander codes* [30], and *Polar codes* [31], but these families will not be discussed here. *Luby Transform (LT) codes* will be discussed in Chapter 5.

Introduced by Gallager, *Low-Density Parity-Check (LDPC)* codes are a family of error-correcting codes defined by sparse parity-check matrices [10]. The sparsity of the matrix allows for an efficient decoding process due to the small number of variable nodes involved in each check equation. LDPC codes may be represented graphically using a graph that is now referred to as the *Tanner graph* [11]. For a binary code with an $(n-k)\times n$ parity-check matrix H, the Tanner graph is a bipartite graph with n *variable nodes* and $(n-k)$ *check* or *constraint nodes* (typically simple parity checks). There is an

edge between constraint node c_i and variable node v_j in the Tanner graph when $h_{ij} = 1$ in H.

A parity-check matrix of an LDPC code is the adjacency matrix of the Tanner graph. Note that a different matrix representation will result in a different Tanner graph for the same code. A sparse parity-check matrix leads to a sparse graph representation, making LDPC codes amenable to efficient graph-based iterative decoding algorithms [11].

Example 2. Suppose a binary code $\mathcal{C}$ has parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Then $\mathcal{C}$ may be represented by the Tanner graph in Figure 2.5, where circles denote

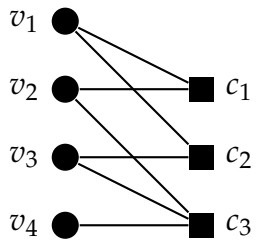


Figure 2.5: A Tanner graph with 4 variable nodes (circles) and 3 check nodes (squares).

variable nodes and squares denote constraint nodes. By the definition of the parity-check matrix, a vector $\mathbf{v} = (v_1, v_2, v_3, v_4)$ is a codeword in $\mathcal{C}$ if and only if the following

parity-check equations are satisfied.

$$v_1 + v_2 = 0$$
$$v_1 + v_3 = 0$$
$$v_2 + v_3 + v_4 = 0.$$

Each row of the parity check matrix (and so each parity-check equation) corresponds to a constraint node in the Tanner graph, and each column corresponds to a variable node.

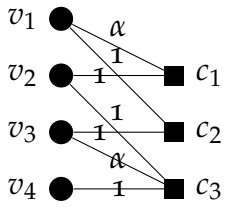
Tanner also introduces *generalized* LDPC codes in which the constraint nodes are *subcodes* with blocklength equal to the degree of the constraint node rather than simple parity checks (note that the subcodes are not subcodes of the overall code) [11]. When the subcode is the same at every constraint node, Tanner provides a lower bound on the minimum distance of the generalized LDPC code in terms of the minimum distance of its subcode and the girth of the Tanner graph. When the subcodes are simple parity checks, the minimum distance of each subcode is 2.

Irregular LDPC codes (LDPC codes whose Tanner graphs are not regular) were introduced in [32]. In these irregular LDPC codes, the variable and check node degree distributions are important design properties that affect the *decoding threshold* of the graph (the highest channel erasure probability for which the probability of decoding error will tend towards zero as the iterations of decoding progress) [32, 14]. In [14], irregular LDPC codes were shown to be capacity-approaching through the optimization of the degree distribution for long LDPC codes.

Nonbinary LDPC codes are defined similarly to their binary counterparts. Their parity-check matrices have entries in GF(q) and are sparse in the number of nonzero entries. Each nonzero entry is viewed as a weight on the corresponding edge of the

Tanner graph, and acts as a coefficient in the corresponding check equation. An example of a LDPC code over GF(3) is shown in Figure 2.6.

$$H = \begin{bmatrix} \alpha & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & \alpha & 1 \end{bmatrix}$$



$$\begin{aligned} \alpha v_1 + v_2 &\equiv 0 \pmod 3 \\ v_1 + v_3 &\equiv 0 \pmod 3 \\ v_2 + \alpha v_3 + v_4 &\equiv 0 \pmod 3. \end{aligned}$$

Figure 2.6: *Left: A parity-check matrix H for a code C over* GF(3). *Center: The graph corresponding to H. Right: The equations that must be satisfied if $v = (v_1, v_2, v_3, v_4) \in C$.*

2.4 Partial Erasure Channels

In some applications, it is reasonable to consider erasure events that leave remnants of information behind. For example, NAND flash memories and phase change memory are susceptible to retention errors and read errors [33, 34]. In these cases, multiple bits are stored in a cell and some bits may not be determined accurately during a readout of cell voltages, leading to partial erasures. Partial erasure channels were recently introduced to model such situations [8, 9].

2.4.1 q-ary Partial Erasure Channel

Motivated by applications in non-volatile memory multi-level read channels, the q-ary Partial Erasure Channel (QPEC) was introduced in [8]. The QPEC is a generalization of the QEC on which *partial erasures* occur instead of full erasures. The QPEC is parameterized by erasure probability ε and erasure size M. Based on the information remaining if a symbol is erased, the receiver has a set of M symbols as candidates for the symbol that was sent.

In a partial erasure of an input symbol x, a set of M symbols (for some fixed $2 \leq M \leq q$) containing x is received as the output. Each of the $\binom{q-1}{M-1}$ sets of size M containing x are equally likely to occur. The transition probability of the QPEC with erasure probability ε is given by

$$\Pr(Y = y \mid X = x) = \begin{cases} 1-\varepsilon & y = x \\ \frac{\varepsilon}{\binom{q-1}{M-1}} & y = ?_x^{(i)} \end{cases},$$

where $i = 1, 2, \ldots, \binom{q-1}{M-1}$, and each $?_x^{(i)}$ is a distinct M-set containing x. Figure 2.7 shows an example of the QPEC with $q = 4$ and $M = 2$. If the symbol $0 \in \mathrm{GF}(4)$ is sent, either 0 or one of the three possible sets containing 0 and one other symbol are received. Note that on the QPEC, unless $M = q$, it is not possible to have a full erasure. More specifically, $q - M$ symbols are known to have not been sent whenever an erasure occurs on the QPEC.

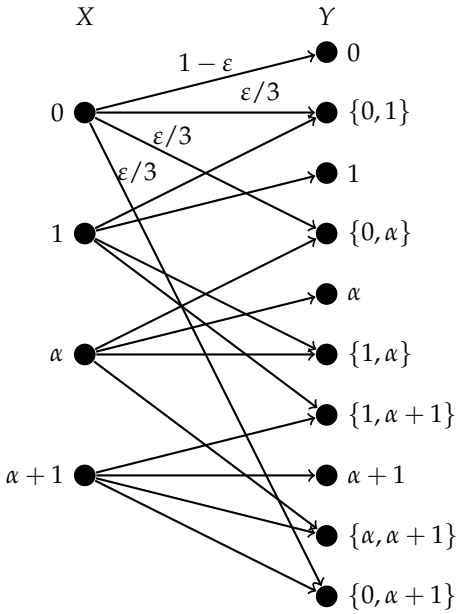


Figure 2.7: The 4-ary Partial Erasure Channel (QPEC) with $M = 2$.

Cohen and Cassuto [8] show that the capacity of the QPEC is $1 - \varepsilon \log_q M$. Observe that if $M = q$, then the QPEC is equivalent to the QEC. If $q = 2$, then $M = 2$ as well, and QPEC is equivalent to the BEC.

In [8], Cohen and Cassuto give an iterative message-passing decoding algorithm for LDPC codes over GF(q) on the QPEC. In the algorithm, the subsets of possible values of a variable node are passed along the edges. Check node calculations involve taking sums of subsets of GF(q), and variable node calculations are performed by taking the intersection of the incoming subsets. Decoding succeeds when the intersection at each variable node is a set with cardinality one. Moreover, the codeword estimate that results when all variable node subsets have size one is guaranteed to be the original codeword. The decoder fails if the size of an erasure subset at one or more variable nodes never reaches one. Since calculations involve taking Minkowski sums of sets of up to q symbols, the analysis of this decoder is complex.

In the traditional q-ary erasure setting, density evolution is the process of tracking the probability of a message from check-to-variable (CTV) node or variable-to-check (VTC) node being in error as iterations progress [10, 35]. If the probability of CTV error is below a predetermined threshold after a fixed number of iterations, the decoding process is declared a success. In the case of the QPEC, density evolution may be approximated by tracking the probability that the message sets have cardinality m [8]. Assume a regular LDPC code with check node degree d_c and the variable node degree d_v. The Dirac delta function, which is 1 if $m = 1$ and 0 otherwise, is denoted by δ_{m-1}.

Essentially, when determining what to send to a message node v, a check node finds the sum of messages sent from all its neighbors other than v. The probability of a CTV message with size m in iteration l is denoted by P_m^l. This probability is determined by the sizes of the incoming message sets from the preceding VTC messages. As shown in [8],

the CTV message sizes have the following probabilities:

$$P_m^l = \sum_{\{|S_j|\}_{j=1}^{d_c-1}} \left(\prod_{j=1}^{d_c-1} Q_{|S_j|}^{l-1} \right) P_m(\{|S_j|\}_{j=1}^{d_c-1}),$$

where $P_m(\{|S_j|\})$ denotes the probability of a CTV message of size m being output on the l^{th} iteration, given input message sizes in $\{|S_j^{l-1}|\}_{j=1}^{d_c-1}$.

When determining what message to send to a check node c, a message node essentially finds the intersection of messages from its neighboring check nodes other than c. The probability of a VTC message with size m in iteration l is denoted by Q_m^l. As shown in [8], the VTC message sizes have the following probabilities:

$$Q_m^l = \varepsilon \sum_{\{|S_j|\}_{j=1}^{d_v-1}} \left(\prod_{j=1}^{d_v-1} P_{|S_j|}^l \right) Q_{m,i_v}(\{|S_j|\}) + (1-\varepsilon)\delta_{m-1},$$

where $Q_{m,i_v}(\{|S_j^{l-1}|\})$ denotes the probability of a VTC message with size m, given that the incoming CTV messages are as indicated above. Note that if $m = 1$, then the variable node has been decoded correctly.

2.4.2 q-ary Multi-bit Channel

Cohen, Raviv, and Cassuto introduce the *q-ary Multi-bit Channel* (initially called the q-ary Bit Measurement Channel) in [9] to model the premature termination of reads in multi-level memories. On the q-ary Multi-bit Channel (QMBC), erasure events correspond to bursts of erasures starting with the rightmost bit in the binary representation of a 2^k-ary symbol. Henceforth, for $x \in \mathrm{GF}(2^k)$, we will use x to refer both to the symbol and its binary expansion.

On the QMBC, the probability of an i-bit long erasure is given by ε_i for $i = 0, 1, \ldots, k$

and the probability that some erasure event occurs is $1 - \varepsilon_0 = \sum_{i=1}^{k} \varepsilon_i$. The transition probabilities for the symbol 000 over the 8-ary QMBC are shown in Figure 2.8 The capacity

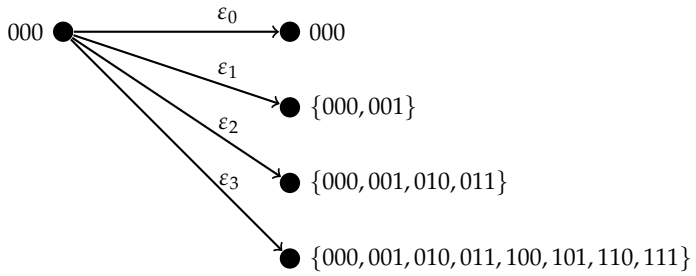


Figure 2.8: Transition probabilities of 000 on the 8-ary QMBC.

of the QMBC is $1 - \sum_{j=1}^{k} \frac{j\varepsilon_j}{k}$ measured in q-ary symbols per channel use [9].

Chapter 3

Multilevel Erasure Channel

While the idea of a partial erasure is addressed in [8, 9], these channel models do not encompass all plausible types of partial erasures. In particular, the QPEC assumes that all partial erasure sets have the same cardinality and the QMBC requires that all erasures are bursts starting from the rightmost bit. Some applications may have erasure events resulting in partial information with a different structure.

In flash memory storage, binary representations of q-ary symbols can be stored with bits on separate "pages," each prone to different erasure rates [36]. For example, using the notation of [36], in triple level cell (TLC) flash, the bits of an 8-ary symbol stored on separate pages are called the least significant bit (LSB), center significant bit (CSB) and most significant bit (MSB). An example of this is shown in Figure 3.1. In the figure, the MSB is shown stored in the leftmost page, the LSB is shown in the rightmost page, and the CSB is shown in the center page. For a typical mapping scheme (of encoded bits to signal voltage levels), the MSB should have the lowest probability of erasure. To address partial erasures corresponding to bit erasures within a symbol, we introduce the Multilevel Erasure Channel (MEC) in [1]. We note that this is a natural model that may capture the behavior of information stored in this case as well as in many other applications.

$\alpha + 1 \in \mathrm{GF}(8) \to 011 \to$

MSB	CSB	LSB
0	1	1

Figure 3.1: Storing a symbol in $\mathrm{GF}(8)$ across three pages.

3.1 Channel Model

We now introduce the MEC, a channel in which partial erasures correspond to one or more erasures at the bit or digit level. A 2^k-ary symbol x sent across the MEC with erasure probability ε will either be received without error with probability $1 - \varepsilon$, or an erasure event will occur with probability ε. An erasure event may consist of any combination of bits in the binary representation of x being erased. For $i = 1, \ldots, k$, we will use γ_i to denote the probability that bit i is erased. For each $j = 1, \ldots, k$, there are $\binom{k}{j}$ sets of symbols whose binary representations differ from the binary representation of x in at most j selected bits. Using similar notation to [8], we define the super-symbol $?_x^B$ to be the set of 2^k-ary symbols differing from x in at most the bits given by a nonempty index set $B \subseteq [k]$.

For a 2^k-ary symbol x, the transition probability of the MEC is given by

$$\Pr(Y = y \mid X = x) = \begin{cases} 1 - \varepsilon & y = \{x\}, \\ \prod_{i \in B} \gamma_i \prod_{i \notin B} (1 - \gamma_i) & y = ?_x^B, B \subseteq [k] \end{cases},$$

where

$$\varepsilon = \sum_{B \subseteq [k]} \left(\prod_{i \in B} \gamma_i \prod_{i \notin B} (1 - \gamma_i) \right).$$

When each bit has the same probability, γ, of being erased, we say the channel is

constrained. The constrained MEC has the following transition probability.

$$\Pr(Y = y \mid X = x) = \begin{cases} 1 - \varepsilon & y = \{x\} \\ \gamma^j (1-\gamma)^{k-j} & y = ?_x^B \text{ with } |B| = j, \end{cases}$$

for $j = 1, \ldots, k$, and where

$$\varepsilon = \sum_{j=1}^{k} \binom{k}{j} \left(\gamma^j (1-\gamma)^{k-j} \right) = 1 - (1-\gamma)^k.$$

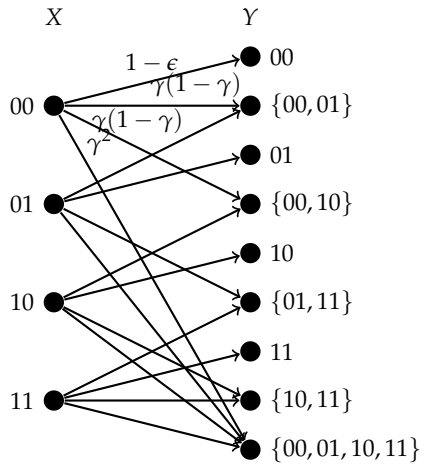


Figure 3.2: 4-ary constrained MEC with erasure probability ε and bit error probability γ.

The 4-ary constrained MEC is shown in Figure 3.2. Note that if the symbol $00 = 0 \in \mathrm{GF}(4)$ is sent, then the possible outcomes are (i) no erasure occurs ($\{00\}$ is received), (ii) the rightmost bit is erased ($\{00, 01\}$ is received), (iii) the leftmost bit is erased ($\{00, 10\}$ is received), or (iv) both bits are erased ($\mathrm{GF}(4)$ is received). The possible outcomes for the other input symbols are similar.

Example 3. For an 8-ary symbol the transition probability of the constrained MEC in

which every bit has probability γ of being erased is given by

$$\Pr(Y = y \mid X = x) = \begin{cases} 1-\varepsilon & y = \{x\} \\ \gamma(1-\gamma)^2 & y = ?_x^B \text{ with } |B| = 1 \\ \gamma^2(1-\gamma) & y = ?_x^B \text{ with } |B| = 2 \\ \gamma^3 & y = ?_x^B \text{ with } |B| = 3, \end{cases}$$

where

$$\varepsilon = \sum_{j=1}^{3} \binom{3}{j} \left(\gamma^j (1-\gamma)^{3-j}\right) = 1 - (1-\gamma)^3.$$

For example, assume $x = (0,0,0)$ is transmitted, corresponding to the symbol $0 \in \mathrm{GF}(8)$. Then the possible output sets and their transition probabilities are

$$\Pr(Y = y \mid X = 000) = \begin{cases} (1-\gamma)^3 & y = \{000\} \\ \gamma(1-\gamma)^2 & y = \{000,001\}, \{000,010\}, \text{ or } \{000,100\} \\ \gamma^2(1-\gamma) & y = \{000,001,010,011\}, \{000,001,100,101\}, \cdot \\ & \quad \text{or } \{000,010,100,110\} \\ \gamma^3 & y = \mathrm{GF}(8) \end{cases}$$

Figure 3.3 shows a comparison of the transition probabilities for an input 00 over the 4-ary QEC, QPEC, QMBC, and MEC. Note that the size and number of outputs varies for the different channels.

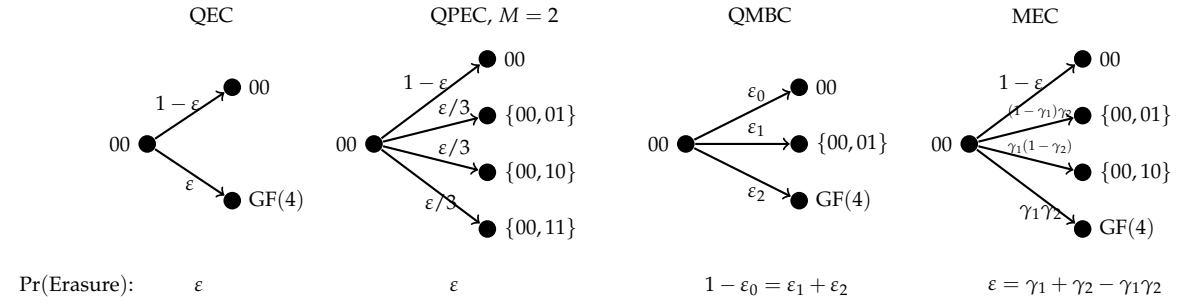


Figure 3.3: Transition probabilities seen by $x = 00 \in \mathrm{GF}(4)$ for the QEC, QPEC with $M = 2$, QMBC, and MEC. For ease of comparison, we use the binary representation of each symbol.

3.2 Capacity

With the introduction of a new channel, a natural goal is to determine the capacity of the channel. We first show that the constrained MEC is uniformly dispersive.

Definition 12. A channel with possible outputs $y_1, \ldots, y_t$ is said to be *uniformly dispersive* if the set

$$\mathcal{A}(x) := \{\Pr(Y = y_1 \mid X = x), \ldots, \Pr(Y = y_t \mid X = x)\}$$

is identical for each input symbol x [37].

Lemma 1 ([1])**.** *The constrained MEC is uniformly dispersive.*

Proof. For any input x on the constrained MEC, there are $\binom{k}{j}$ possible outputs of size 2^j for each $j = 0, \ldots, k$. For each output y of size 2^j, $\Pr(Y = y \mid X = x) = \gamma^j(1-\gamma)^{k-j}$. Moreover, an output set of size 2^j is obtained by choosing which bits to vary in $\binom{k}{j}$ ways and choosing the remaining bits in 2^{k-j} ways. Thus, there are a total of $\sum_{j=0}^{k}\binom{k}{j}2^{k-j} = 3^k$ total possible channel outputs. Of these 3^k outputs, $3^k - \sum_{j=0}^{k}\binom{k}{j} = 3^k - 2^k$ have probability 0 of

occurring given a fixed input x. Therefore,

$$\mathcal{A}(x) := \{ \underbrace{0, \ldots, 0}_{3^k - 2^k \text{ times}} \} \cup \bigcup_{j=0}^{k} \{ \underbrace{\gamma^j (1-\gamma)^{k-j}, \ldots, \gamma^j (1-\gamma)^{k-j}}_{\binom{k}{j} \text{ times}} \}$$

regardless of x, as desired. □

Using the fact that $1 - \varepsilon = (1-\gamma)^k$ and that the constrained MEC is a uniformly dispersive channel, for any input distribution,

$$H(Y|X) = -\sum_{j=0}^{k} \binom{k}{j} \gamma^j (1-\gamma)^{k-j} \log(\gamma^j (1-\gamma)^{k-j}).$$

Thus to determine capacity, it is enough to maximize $H(Y)$. For the uniform input distribution $\{p_x\}$ where $Pr(X = x) = 1/q$ for each $x \in \mathrm{GF}(q)$, we have

$$H(Y) = -\sum_{j=0}^{k} \binom{k}{j} \gamma^j (1-\gamma)^{k-j} \log(2^{j-k} \gamma^j (1-\gamma)^{k-j}).$$

It follows that given the uniform input distribution and $q = 2^k$,

$$I(X;Y) = k(1-\varepsilon)^{1/k}$$

measured in bits per channel use, or

$$I(X;Y) = k \log_q(2)(1-\varepsilon)^{1/k} = (1-\varepsilon)^{1/k}$$

measured in q-ary symbols per channel use. A plot of $I(X;Y)$ as a function of ε for various k is shown in Figure 3.2.

We now show that the uniform input distribution is a capacity-achieving input

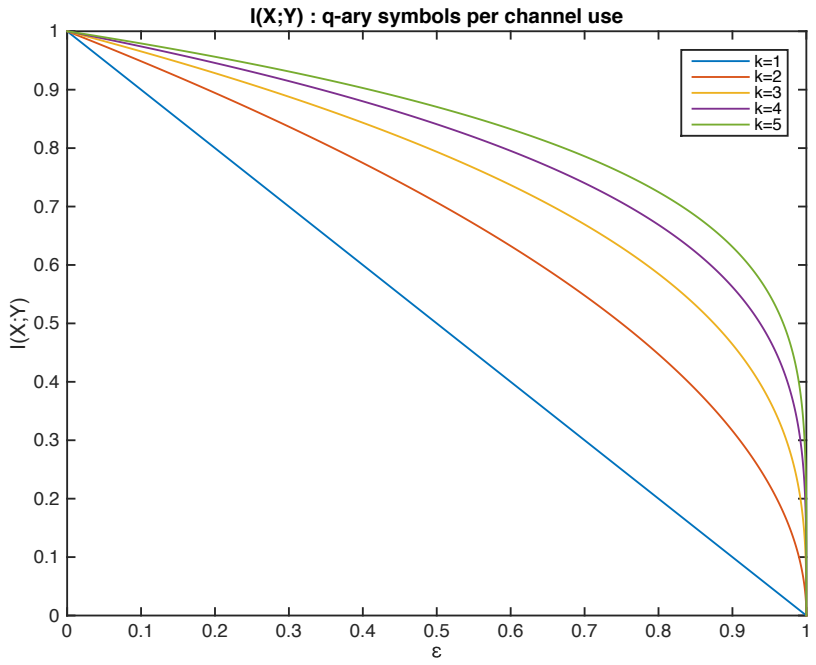


Figure 3.4: $I(X, Y)$ as a function of ε on the constrained MEC.

distribution when $k = 2$ to prove the following theorem.

Theorem 1 ([1]). *The capacity on the constrained multilevel erasure channel with $k = 2$ is*

$$\text{Cap}(\text{C-MEC}) = (1 - \varepsilon)^{1/2}.$$

Proof. Let $\Pr(\cdot)$ be a capacity achieving input distribution for the MEC. For each input $x_i \in \text{GF}(4)$, let Y_i be the set of possible outputs. Then

$$\begin{aligned} I(x_i; Y) &= \sum_{y \in Y_i} \Pr(y \mid x_i) \log\left(\frac{\Pr(y \mid x_i)}{\Pr(y)}\right) \\ &= (1-\varepsilon)\log(1-\varepsilon) + 2(1-\gamma)\gamma\log((1-\gamma)\gamma) \\ &\quad + \gamma^2 \log(\gamma^2) - \sum_{y \in Y_i} \left(\Pr(y \mid x_i)\log(\Pr(y))\right). \end{aligned}$$

For $x_i = 0$,

$$\sum_{y \in Y_0} (\Pr(y \mid 0) \log(\Pr(Y = y))) = (1-\varepsilon)\log(\Pr(0)) + 2\gamma(1-\gamma)\log(\gamma(1-\gamma)) +$$
$$\gamma(1-\gamma)\log(\Pr(0)+\Pr(1)) + \gamma(1-\gamma)\log(\Pr(0)+\Pr(\alpha)) + \gamma^2\log(\gamma^2)$$

and the sum is similar for each $x_i \in \mathrm{GF}(4)$. By the Karush-Kuhn-Tucker (KKT) conditions [37], a capacity achieving input distribution for a channel with capacity C must satisfy $I(x;Y) = C$ for all x with $\Pr(x) > 0$. If each $\sum_{y \in Y_i} (\Pr(y \mid x_i) \log(\Pr(Y = y)))$ is equal, then we have the following system of equations.

$$\log(\Pr(0)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(0)+\Pr(\alpha)) = \log(\Pr(1)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(1)+\Pr(\alpha+1))$$
$$\log(\Pr(0)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(0)+\Pr(1)) = \log(\Pr(\alpha)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(\alpha)+\Pr(\alpha+1))$$
$$\log(\Pr(1)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(0)+\Pr(1)) = \log(\Pr(\alpha+1)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(\alpha)+\Pr(\alpha+1))$$
$$\log(\Pr(\alpha)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(0)+\Pr(\alpha)) = \log(\Pr(\alpha+1)) + \frac{\gamma}{(1-\gamma)}\log(\Pr(1)+\Pr(\alpha+1)).$$

Assuming each input symbol occurs with probability > 0, the uniform input distribution is the unique distribution satisfying the system of equations. Therefore for $k = 2$, $\mathrm{Cap}(\text{C-MEC}) = k\log_{2^k}(2)(1-\varepsilon)^{1/k}$ as claimed. □

More generally, for $k > 2$, we have

Theorem 2. *[1] The capacity of the constrained multilevel erasure channel is*

$$\mathrm{Cap}(\text{C-MEC}) = (1-\varepsilon)^{1/k},$$

2^k-ary symbols/channel use.

The proof of Theorem 2 uses multilevel coding and will be presented in Chapter 4.

Note that the multilevel erasure channel model and these results may be extended to GF(p^k), where p is prime. In this setting, partial erasures correspond to erasures of p-ary symbols within the p-ary representation of a q-ary symbol. The proof of Theorem 2, presented in Chapter 4, shows that Theorem 2 holds over GF(p^k) when the capacity is measured in p^k-symbols per channel use.

Chapter 4

Multilevel Coding and Multistage Decoding

The iterative decoder proposed by [8] for LDPC codes on the QPEC may also be used for LDPC codes on the MEC. However, analysis of the decoder performance is difficult, as it relates to the subset sum problem in group theory involving a large alphabet size [38]. In search of a simpler decoding method, we investigate a multilevel coding and multistage decoding scheme on partial erasure channels. We show that the MEC may be decomposed into k independent binary or p-ary erasure channels. We also show that whether the QPEC decomposes into simple erasure channels depends on M and q.

4.1 Multilevel Coding

Imai and Hirakawa introduced multilevel coding in [16] as a tool for using side-information during decoding. They initially considered an application of multilevel coding to multi-phase modulation. To encode $q = p^k$-ary information symbols $m_1, \ldots, m_K$ into a length n codeword, the Kk p-ary symbols in the p-ary representations of $m_1, \ldots, m_K$ are divided into k groups with $\ell_1, \ell_2, \ldots, \ell_k$ p-ary information symbols in each group. The information symbols within each group are encoded to a length N p-ary codeword using a component code specific to the group. In particular, group i uses an (N, ℓ_i) p-ary component code $\mathcal{C}_i$. Thus each group is encoded into N p-ary symbols.

The first encoded p-ary symbols from each of the k groups are combined to form the first q-ary symbol for transmission on the MEC, where the p-ary symbol from group i forms the i^{th} coordinate of the p-ary representation of the q-ary symbol. Similarly, the second encoded p-ary symbol from each of the groups are combined to form the second q-ary symbol, and so on. Thus, a length N q-ary codeword is obtained via k p-ary codewords of length N. From the above, we have $Kk = \ell_1 + \cdots + \ell_k$, and the overall code rate of the multi-level code is

$$R = \frac{K}{N} = \frac{\ell_1 + \ell_2 + \cdots + \ell_k}{Nk} = \frac{R_1 + R_2 + \cdots + R_k}{k},$$

where $R_i = \frac{\ell_i}{N}$ is the code rate of the i^{th} p-ary component code $\mathcal{C}_i$.

Example 4. A $[7, 4]_9$ linear code $\mathcal{C}$ may be formed using multilevel coding as follows. Let k_1, k_2, k_3, k_4 be information symbols to be encoded. Note that each of the four 9-ary information symbol may be written as three ternary symbols, $k_i = (k_i^1, k_i^2, k_i^3)$ for $i = 1, 2, 3, 4$. The 12 resulting ternary symbols may be divided into 3 groups of possibly different sizes, ℓ_1, ℓ_2, ℓ_3. An example of such groups is shown in Figure 4.1. Each group of ternary symbols is encoded using a blocklength 7 code, and a 9-ary code of blocklength 7 is formed by combining the j^{th} ternary symbol from each codeword for $j = 1, 2, \ldots, 7$.

Note that multilevel coding differs from concatenated coding ([39]), as the initial message symbols are not encoded. We do not have an inner code and an outer code in this setting.

4.2 Multistage Decoding

In the multilevel decoding scheme described in the previous section, the code $\mathcal{C}_i$ is a code over the i^{th} coordinate of each p^k-ary codeword, for $i = 1, 2, \ldots, k$. Since each coordinate

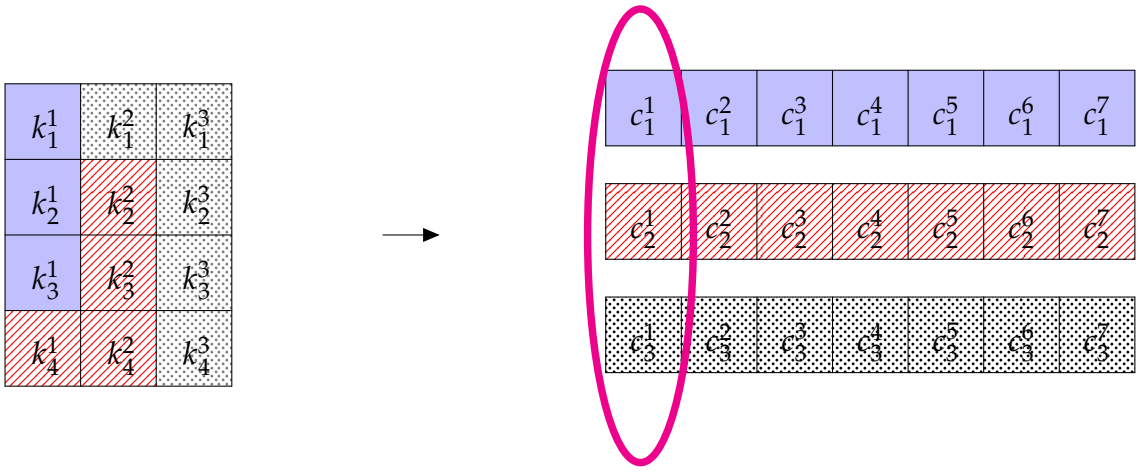


Figure 4.1: **Left:** Groups of ternary information symbols with $\ell_1 = 3, \ell_2 = 4, \ell_3 = 5$. $\mathcal{C}_1$ is a $[7,3]_3$ code encoding k_1^1, k_2^1, k_3^1. $\mathcal{C}_2$ is a $[7,4]_3$ code encoding $k_4^1, k_2^2, k_3^2, k_4^2$. $\mathcal{C}_3$ is a $[7,5]_3$ code encoding the remaining 5 symbols. **Right:** The codes from each group are combined to a length seven 9-ary code word. The ternary expansion of the i^{th} code symbol is given by $c_1^i c_2^i c_3^i$ for $i = 1, \ldots, 7$. The first 9-ary code symbol is circled above.

is encoded with its own code, each coordinate may be decoded separately. However, if the symbols in each coordinate are not independent, information about symbols in each coordinate may be gained by considering previously decoded coordinates. That is, there may be a benefit to decoding coordinates in series rather than in parallel.

In addition to a multilevel coding scheme, Imai and Hirakawa present a multistage decoding strategy [16]. Let X_i be a random variable representing the i^{th} coordinate of a codeword from a multilevel coding scheme, for $i = 1, \ldots, k$. Let X be a random variable representing the channel input and Y be a random variable representing the channel output. Note that $X = (X_1, X_2, \ldots, X_k)$. Then the mutual information between the channel input and channel output is given by

$$I(X;Y) = I((X_1, X_2, \ldots, X_k); Y).$$

Using the chain rule of mutual information on the righthand side, the mutual infor-

mation can be rewritten as

$$I(X;Y) = I(X_1;Y) + I(X_2;Y|X_1) + I(X_3;Y|X_1,X_2) + \cdots + I(X_k;Y|X_1,\ldots,X_{k-1}).$$

In terms of multistage decoding, this can be interpreted as follows. The decoder uses the code $\mathcal{C}_1$ to decode Y and obtain an estimate for X_1. After finding an estimate for X_1, the decoder uses the code $\mathcal{C}_2$ as well as the estimate for X_1 to decode Y and obtain an estimate for X_2. This process continues until the decoder has found an estimate for each coordinate. The estimate for X_i is found using code $\mathcal{C}_i$ and the estimates for $X_1,\ldots,X_{i-1}$ for each $i = 2,\ldots,k$. If the X_i are not independent then as the stages of decoding progress, the channels typically improve due to access to more side information.

Intuitively, it makes sense to assign the strongest code to the weakest channel so the code $\mathcal{C}_1$ will have the lowest code rate R_1, and in general, the codes will be organized so that $R_1 \leq R_2 \leq \cdots \leq R_k$ where R_i is the rate of code $\mathcal{C}_i$. To achieve capacity using multistage decoding, the code rates should be chosen so that $R_i = I(X_i;Y|X_1,\ldots,X_{i-1})$ for each $i = 1,\ldots,k$ [16, 40].

Note that if X_i and X_j are independent for all $i \neq j$ even when Y is known, then

$$I(X;Y) = I(X_1;Y) + I(X_2;Y) + I(X_3;Y) + \cdots + I(X_k;Y)$$

and decoding the coordinates in parallel is as reliable as decoding them in series.

A multilevel coding and multistage decoding scheme may be used to simplify the decoding process on the MEC.

4.3 Multistage Decoding on the Multilevel Erasure Channel

Using a multilevel coding scheme, we can show that the multilevel erasure channel decomposes into independent p-ary erasure channels.

Example 5. Consider the 4-ary MEC. The information rate $I(X;Y)$ for this channel can be written as $I(X_1, X_2; Y) = I(X_1; Y) + I(X_2; Y|X_1)$. Assuming the uniform distribution on the inputs, the effective channel for X_1 is a binary erasure channel with erasure probability γ, shown in Figure 4.2. To see this, observe that if $X_1 = 0$, the outputs $\{00, 10\}, \{00, 01, 10, 11\}, \{01, 11\}$ from input symbols $\{00\}, \{01\}$ lead to uncertainty in the value of X_1 at the output. The probability that $X_1 = 0$ and one of these outputs is received is

$$\frac{1}{4}\gamma(1-\gamma) + \frac{2}{4}\gamma^2 + \frac{1}{4}\gamma(1-\gamma) = \frac{1}{2}\gamma.$$

Allowing for $X_1 = 0$ or $X_1 = 1$, the probability is $2 \cdot \frac{\gamma}{2} = \gamma$.

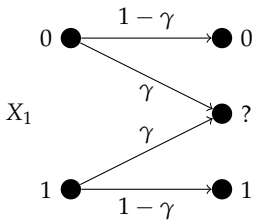


Figure 4.2: Subchannel for X_1 on 4-ary multilevel erasure channel with parameters ε, γ.

Without any side knowledge of X_1, the variable X_2 also has effective channel that is a BEC with erasure probability γ.

We now consider the case when bit X_1 is known and examine the resulting channel is seen by X_2. Without loss of generality, suppose X_1 is known to equal 0. Figure 4.3 shows the 4-ary subchannel seen by bit X_2.

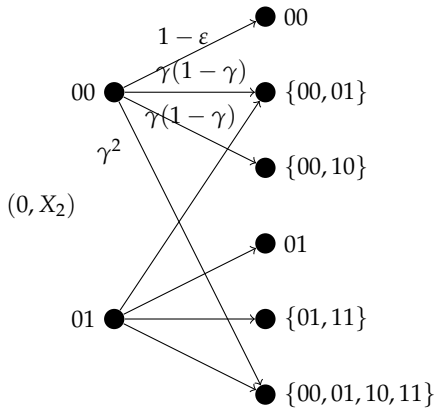


Figure 4.3: Subchannel for X_2 on 4-ary multilevel erasure channel with parameters ε, γ.

The outputs in Figure 4.3 leading to uncertainty in X_2 are $\{00, 01\}$ and $\{00, 10, 01, 11\}$. The probability of receiving one of these outputs when $X_1 = 0$ is $\gamma(1-\gamma) + \gamma^2 = \gamma$. The effective erasure probability for the binary erasure channel seen by X_2 conditioned on X_1 is therefore γ, which is the same as the case in which we remove the conditioning on X_1. Thus, X_2 can in fact be decoded independent of knowing X_1.

Theorem 3. *For the MEC over* $\mathrm{GF}(p^k)$ *with erasure probability* ε *and p-ary symbol erasure probabilities* $\gamma_1, \gamma_2, \ldots, \gamma_k$ *for* $X_1, \ldots, X_k$*, respectively, the conditional mutual information*

$$I(X_i; Y | X_1, \ldots, X_{i-1}) = I(X_i; Y)$$

for $i = 1, 2, \ldots, k$ *and where* $X_0 = \{\}$*. That is, the MEC can be decomposed into k independent p-ary erasure channels, each with erasure probability* γ_i*, and mutual information rate* $I(X; Y) = \sum_{i=1}^{k} I(X_i; Y) = \sum_{i=1}^{k} (1 - \gamma_i)$ *p-ary symbols/channel use.*

Proof. We show that the channels representing $I(X_i; Y)$ and $I(X_i; Y \mid X_1, \ldots, X_{i-1})$ each have transition probability γ_i. To see this, consider sending a p^k-ary symbol across the channel. The probability of receiving an output with uncertainty in the i^{th} p-ary symbol

is then

$$\gamma_i \sum_{B \subseteq [k] \setminus i} \left(\prod_{j \in B} \gamma_j \prod_{j \in ([k] \setminus i) \setminus B} (1 - \gamma_j) \right) = \gamma_i,$$

where the sum is over all subsets that do not include the i^{th} p-ary symbol. Now suppose that a p^k-ary symbol is sent across the channel and p-ary symbols $1, \ldots, i-1$ are known. The probability of receiving an output with uncertainty in the i^{th} p-ary symbol is then

$$\gamma_i \sum_{B \subseteq [k] \setminus [i]} \left(\prod_{j \in B} \gamma_j \prod_{j \in ([k] \setminus [i]) \setminus B} (1 - \gamma_j) \right) = \gamma_i,$$

where the sum is over all subsets excluding the first i positions. Therefore $I(X_i; Y) = I(X_i; Y \mid X_1, \ldots, X_{i-1})$. □

Using the fact that the MEC may be decomposed into k independent p-ary erasure channels, we may now prove that the constrained p^k-ary MEC has capacity $(1-\varepsilon)^{\frac{1}{k}}$ measured in p^k-ary symbols per channel use.

Proof of Theorem 2. Theorem 3 shows that $I(X;Y) = \sum_{i=1}^{k} (1 - \gamma_i)$ p-ary symbols/channel use can be acheived using multilevel codes on the p^k-ary MEC. In the case of the constrained MEC, $\gamma_i = \gamma$ for all i, so

$$I(X;Y) = k - k\gamma = k(1-\gamma)$$

p-ary symbols/channel use is achievable. Recall that $1 - \gamma = (1-\varepsilon)^{1/k}$. Thus in p^k-ary symbols/channel use we have

$$I(X;Y) = (1-\gamma) = (1-\varepsilon)^{1/k}.$$

We now show that for any input distribution $\{p_x\}$ of X where $p_x = \Pr(X = x)$,

$I(X;Y) \leq k - k\gamma$ in bits/channel use.

Write the MEC output Y as a vector $(b_1, \ldots, b_k)$, where b_i takes the value of the i^{th} p-ary symbol if it is known, and $b_i =?$ otherwise. For example, for the 3^2-ary MEC, the output set $\{00, 01, 02\}$ will be written as $(0, ?)$. Next, define a random vector $E = a = (a_1, \ldots, a_k)$ corresponding to Y as follows. Let

$$a_i = \begin{cases} 1 & \text{if } b_i =? \\ 0 & \text{if } b_i \neq 0 \end{cases}.$$

E may be regarded as a random variable representing an outcome of the MEC. Moreover, by the chain rule of entropy,

$$H(Y) = H(Y, E) = H(E) + H(Y|E)$$

where the first equality follows from the fact that E is a function of Y.

Observe that

$$\Pr\left(E = (a_1, \ldots, a_k) \text{ and } a_i = 1 \text{ for exactly } j \text{ indices}\right) = \gamma^j (1-\gamma)^{k-j}.$$

Therefore

$$\begin{aligned} H(E) &= -\sum_{j=0}^{k} \binom{k}{j} \gamma^i (1-\gamma)^{k-j} \log(\gamma^j (1-\gamma)^{k-j}) \\ &= -\sum_{x} \Pr(x) \sum_{B \subseteq [k]} \Pr(Y =?_x^B | X = x) \log(\Pr(Y =?_x^B | X = x)) \\ &= \sum_{x} \Pr(x) H(Y|X = x) \\ &= H(Y|X). \end{aligned}$$

Since $I(X;Y) = H(Y) - H(Y|X)$, we have

$$I(X;Y) = H(E) + H(Y|E) - H(Y|X) = H(E) + H(Y|E) - H(E) = H(Y|E).$$

Consider an output $y = (b_1, \ldots, b_k)$. Note that $\Pr(Y = y|E = a) = 0$ if the positions in Y that are erased are not identical to the set $\{i|a_i = 1\}$.

We have

$$\begin{aligned} H(Y|E) &= \sum_{a \in \mathrm{GF}(p)^k} \Pr(E = a) H(Y|E = a) \\ &= \sum_{j=0}^{k} \sum_{a^j \in \mathrm{GF}(p)^k} \gamma^j (1-\gamma)^{k-j} H(Y|E = a^j) \end{aligned}$$

where a^i denotes a vector in $\mathrm{GF}(p)^k$ with weight i for $i = 0, \ldots, k$.

Observe that each $H(Y|E = a^j)$ is at most $k - j$ (measured in p^k-ary symbols per channel use) since j components of Y are erased and there are p^{k-j} possible values of Y conditioned on $E = a^j$. So the maximum entropy for $H(Y|E = a^j)$ is at most $\log_p(p^{k-j}) = k - j$. Thus,

$$H(Y|E) \leq \sum_{j=0}^{k} (k-j) \binom{k}{j} \gamma^j (1-\gamma)^{k-j} = k(1-\gamma).$$

This proves that $I(X;Y) \leq k - k\gamma$. Since we have already shown that the mutual information $I(X;Y) = k - k\gamma$ is achievable using multilevel codes and multistage decoding (assuming a uniform distribution on the input X), the capacity of the MEC is indeed $k - k\gamma$ p-ary symbols/channel use, which is $(1-\varepsilon)^{1/k}$ p^k-ary symbols/channel use.

□

4.4 Multistage Decoding on the q-ary Partial Erasure Channel

While the MEC has a natural decomposition into simple p-ary erasure channels, it is not obvious that the structure of the QPEC also lends itself to decomposition. However, under certain conditions, multilevel coding and multistage decoding allow the QPEC to decompose into simpler channels.

Theorem 4. *For the QPEC with erasure probability ε, $q = 2^k$ and $M = 2$, the mutual information for the subchannels with multistage decoding is given by*

$$I(X_i; Y|X_1, X_2, \ldots, X_{i-1}) = 1 - \varepsilon_i \,,$$

where $\varepsilon_i = \frac{2^{k-i}}{2^k-1}\varepsilon$ for $i = 1, \ldots, k$. Thus, each subchannel is a binary erasure channel with erasure probability ε_i. Moreover, multistage decoding of binary codes optimized for each subchannel achieves the capacity on the QPEC.

Proof. Note that for the QPEC with $q = 2^k$ and $M = 2$, the probability of any specified partial erasure set is $\frac{\varepsilon}{2^k-1}$. For any given input symbol, there are a total of 2^{k-i} possible erasure sets with bits $1, \ldots, i-1$ known and uncertainty in the i^{th} bit. To see this, note that once $i-1$ bits are fixed, there are $k-1-(i-1) = k-i$ bits other than the (erased) i^{th} bit for which to choose values.

The probability of receiving an output with uncertainty in the i^{th} bit when bits $1, \ldots, i-1$ are known is thus

$$\varepsilon_i := \frac{2^{k-i}}{2^k-1}\varepsilon$$

for each $i = 1, \ldots, k$. Thus $I(X_i; Y|X_1, X_2, \ldots, X_{i-1}) = 1 - \varepsilon_i$.

To see that capacity on the QPEC is achieved, note that

$$\sum_{i=1}^{k}(1-\varepsilon_i) = k - \sum_{i=1}^{k}\frac{2^{k-i}\varepsilon}{2^k-1} = k - \varepsilon = k\left(1-\frac{\varepsilon}{k}\right).$$

which is $\left(1-\frac{\varepsilon}{k}\right) = (1-\varepsilon\log_q(M))$ 2^k-ary symbols per channel use. □

Theorem 4 generalizes as follows.

Theorem 5. *For the QPEC with erasure probability ε, $q = p^k$ and $M = p$, the mutual information for the subchannels with multistage decoding is given by*

$$I(X_i; Y|X_1, X_2, \ldots, X_{i-1}) = 1 - \varepsilon_i,$$

where

$$\varepsilon_i = \frac{\sum_{j=1}^{p-1}\binom{(p-1)p^{k-i}}{j}\binom{p^{k-i}-1}{p-1-j}}{\binom{p^k-1}{p-1}}\varepsilon$$

for $i = 1, \ldots, k$. Thus, each subchannel is a p-ary erasure channel with erasure probability ε_i.

Proof. Note that for the QPEC with $q = p^k$ and $M = p$, the probability of any specified partial erasure set is $\frac{\varepsilon}{\binom{p^k-1}{p-1}}$. Suppose x is the message symbol sent. The partial erasure set will contain x as well as $p-1$ other symbols, at least one of which differs from the symbol sent in the i^{th} coordinate. There are $(p-1)p^{k-i}$ symbols that match with x in the first $i-1$ spots and differ from x in the i^{th} spot and $p^{k-i}-1$ symbols (other than x) that match x in the first i spots.

Thus

$$\varepsilon_i = \frac{\sum_{j=1}^{p-1}\binom{(p-1)p^{k-i}}{j}\binom{p^{k-i}-1}{p-1-j}}{\binom{p^k-1}{p-1}}\varepsilon.$$

□

Note that although the QPEC decomposes into QECs under certain parameters, these subchannels are not independent.

Example 6. Consider the 4-ary QPEC with $M = 2$. Assuming a uniform input distribution,

$$\begin{aligned}
I(X_1;Y) =& \sum_{x_1,x_2}\sum_{y}\Pr(y|x_1,x_2)\Pr(x_1)\Pr(x_2)\log_2\left(\frac{\sum_{x_2'}\Pr(y|x_1,x_2')\Pr(x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')\Pr(x_1')\Pr(x_2')}\right)\\
=&4\sum_{y}\Pr(y|00)\frac{1}{2}\cdot\frac{1}{2}\log_2\left(\frac{\sum_{x_2'}\Pr(y|0,x_2')\frac{1}{2}}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')\frac{1}{2}\cdot\frac{1}{2}}\right)\\
=&\Pr(00|00)\log_2\left(\frac{2\sum_{x_2'}\Pr(00|0,x_2')}{\sum_{x_1'x_2'}\Pr(00|x_1'x_2')}\right)+\Pr(\{00,01\}|00)\log_2\left(\frac{2\sum_{x_2'}\Pr(\{00,01\}|0,x_2')}{\sum_{x_1'x_2'}\Pr(\{00,01\}|x_1'x_2')}\right)\\
&+\Pr(\{00,10\}|00)\log_2\left(\frac{2\sum_{x_2'}\Pr(\{00,10\}|0,x_2')}{\sum_{x_1'x_2'}\Pr(\{00,10\}|x_1'x_2')}\right)\\
&+\Pr(\{00,11\}|00)\log_2\left(\frac{2\sum_{x_2'}\Pr(\{00,11\}|0,x_2')}{\sum_{x_1'x_2'}\Pr(\{00,11\}|x_1'x_2')}\right)\\
=&(1-\varepsilon)+\frac{\varepsilon}{3}\left(\log_2(2)+\log_2(1)+\log_2(1)\right)\\
=&(1-\frac{2}{3}\varepsilon).
\end{aligned}$$

Similarly, without knowledge of X_1,

$$I(X_2;Y) = (1-\frac{2}{3}\varepsilon).$$

However, with knowledge of X_1,

$$\begin{aligned}
I(X_2;Y|X_1) =& \sum_{x_1,x_2}\sum_{y}\Pr(y|x_1,x_2)\Pr(x_1)\Pr(x_2)\log_2\left(\frac{\Pr(y|x_1,x_2)}{\sum_{x_2'}\Pr(y|x_1x_2')\Pr(x_2')}\right)\\
=&4\sum_{y}\Pr(y|00)\frac{1}{4}\log_2\left(\frac{\Pr(y|00)}{\sum_{x_2'}\Pr(y|0x_2')\frac{1}{2}}\right)\\
=&(1-\varepsilon)+\frac{\varepsilon}{3}\log_2\left(\frac{\Pr(\{00,01\}|00)}{\sum_{x_2'}\Pr(\{00,01\}|0x_2')\frac{1}{2}}\right)\\
&+\frac{\varepsilon}{3}\log_2\left(\frac{\Pr(\{00,10\}|00)}{\sum_{x_2'}\Pr(\{00,10\}|0x_2')\frac{1}{2}}\right)+\frac{\varepsilon}{3}\log_2\left(\frac{\Pr(\{00,11\}|00)}{\sum_{x_2'}\Pr(\{00,11\}|0x_2')\frac{1}{2}}\right)\\
=&(1-\varepsilon)+\frac{\varepsilon}{3}\left(\log_2(1)+\log_2(2)+\log_2(2)\right)\\
=&(1-\frac{1}{3}\varepsilon).
\end{aligned}$$

As $I(X_2;Y) \neq I(X_2;Y|X_1)$, we see that the subchannels are not independent.

In other cases, the QPEC does not decompose into simple QECs.

Example 7. Consider the 4-ary partial erasure channel with $M = 3$. Using the uniform input distribution, the mutual information of the first subchannel may be found as

follows.

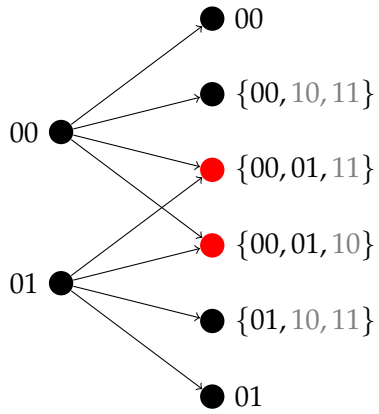
$$
\begin{aligned}
I(X_1;Y) &= \sum_{x_1,x_2}\sum_{y}\Pr(y|x_1,x_2)\Pr(x_1)\Pr(x_2)\log_2\left(\frac{\sum_{x_2'}\Pr(y|x_1,x_2')\Pr(x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')\Pr(x_1')\Pr(x_2')}\right)\\
&= 4\sum_{y}\Pr(y|00)\frac{1}{2}\cdot\frac{1}{2}\log_2\left(\frac{\sum_{x_2'}\Pr(y|0,x_2')\frac{1}{2}}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')\frac{1}{2}\cdot\frac{1}{2}}\right)\\
&= (1-\varepsilon)\log_2\left(2\frac{(1-\varepsilon)}{(1-\varepsilon)}\right)+\sum_{y=?_{00}}\frac{\varepsilon}{\binom{4-1}{3-1}}\log_2\left(2\frac{\sum_{x_2'}\Pr(y|0,x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\sum_{y=?_{00}}\log_2\left(2\frac{\sum_{x_2'}\Pr(y|0,x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\sum_{01\notin y}\log_2\left(\frac{2\sum_{x_2'}\Pr(y|0,x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)+\frac{\varepsilon}{3}\sum_{10\notin y}\log_2\left(\frac{2\sum_{x_2'}\Pr(y|0,x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&\qquad+\frac{\varepsilon}{3}\sum_{11\notin y}\log_2\left(\frac{2\sum_{x_2'}\Pr(y|0,x_2')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\left(\log_2\left(\frac{2\cdot\frac{\varepsilon}{3}}{3\frac{\varepsilon}{3}}\right)+\log_2\left(\frac{4\cdot\frac{\varepsilon}{3}}{3\frac{\varepsilon}{3}}\right)+\log_2\left(\frac{4\cdot\frac{\varepsilon}{3}}{3\frac{\varepsilon}{3}}\right)\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\left(\log_2\left(\frac{2}{3}\right)+\log_2\left(\frac{4}{3}\right)+\log_2\left(\frac{4}{3}\right)\right)\\
&= (1-\varepsilon)+\frac{5}{3}\varepsilon\log_2(2)-\varepsilon\log_2(3)\\
&= 1+\frac{2}{3}\varepsilon-\varepsilon\log_2(3).
\end{aligned}
$$

The second subchannel has mutual information

$$
\begin{aligned}
I(X_2;Y|X_1) &= \sum_{x_1,x_2}\sum_{y}\Pr(y|x_1,x_2)\Pr(x_1)\Pr(x_2)\log_2\left(\frac{\Pr(y|x_1,x_2)}{\sum_{x_2'}\Pr(y|x_1x_2')\Pr(x_2')}\right)\\
&= 4\sum_{y}\Pr(y|00)\frac{1}{4}\log_2\left(\frac{\Pr(y|00)}{\sum_{x_2'}\Pr(y|0x_2')\frac{1}{2}}\right)\\
&= \sum_{y}\Pr(y|00)\log_2\left(\frac{2\Pr(y|00)}{\sum_{x_2'}\Pr(y|0x_2')}\right)\\
&= (1-\varepsilon)+\sum_{y=?_{00}}\Pr(y|00)\log_2\left(\frac{2\Pr(y|00)}{\sum_{x_2'}\Pr(y|0x_2')}\right)\\
&= (1-\varepsilon)+\sum_{y=?_{00}}\frac{\varepsilon}{3}\log_2\left(\frac{\frac{2}{3}\varepsilon}{\sum_{x_2'}\Pr(y|0x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\sum_{01\notin y}\log_2\left(\frac{\frac{2}{3}\varepsilon}{\sum_{x_2'}\Pr(y|0x_2')}\right)+\frac{\varepsilon}{3}\sum_{10\notin y}\log_2\left(\frac{\frac{2}{3}\varepsilon}{\sum_{x_2'}\Pr(y|0x_2')}\right)\\
&\quad+\frac{\varepsilon}{3}\sum_{11\notin y}\log_2\left(\frac{\frac{2}{3}\varepsilon}{\sum_{x_2'}\Pr(y|0x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\left(\log_2\left(\frac{\frac{2}{3}\varepsilon}{\frac{\varepsilon}{3}}\right)+\log_2\left(\frac{\frac{2}{3}\varepsilon}{2\cdot\frac{\varepsilon}{3}}\right)+\log_2\left(\frac{\frac{2}{3}\varepsilon}{2\cdot\frac{\varepsilon}{3}}\right)\right)\\
&= 1-\varepsilon+\frac{\varepsilon}{3}\left(\log_2(2)+\log_2(1)+\log_2(1)\right)\\
&= 1-\varepsilon+\frac{1}{3}\varepsilon\\
&= 1-\frac{2}{3}\varepsilon.
\end{aligned}
$$

The sum of these mutual informations is $1+\frac{2}{3}\varepsilon-\varepsilon\log_2(3)+1-\frac{2}{3}\varepsilon=(2-\varepsilon\log_2 3)$ bits per channel use, or $(1-\varepsilon\log_4 3)$ 4-ary symbols per channel use. If binary codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with rates $1+\frac{2}{3}\varepsilon-\varepsilon\log_2(3)$ and $1-\frac{2}{3}\varepsilon$ respectively are used for the components, then the capacity of the QPEC may be achieved. However, the form of the mutual information rate of the first component channel suggests that this QPEC does not decompose into simple binary erasure channels.

In particular, to see that the first subchannel is not a simple BEC, let $\varepsilon' = 1 - I(X_1; Y) = \varepsilon \log_2(3) - \frac{2}{3}\varepsilon$. Assuming a uniform input distribution, if X_1 is known (without loss of generality, we will assume $X_1 = 0$), then we have the following channel.



The only output sets leading to uncertainty in X_1 are the two sets containing 00 and 01. If X_1 is known to equal 0, such a set occurs with probability $\frac{2}{3}\varepsilon$. However, $1 - \frac{2}{3}\varepsilon \neq 1 - \varepsilon'$. Thus the first subchannel for the QPEC is not an erasure channel with erasure probability ε'.

We can also show that the subchannels of the QPEC with $q = 2^2$ and $M = 3$ are not independent. To do this, it is sufficient to show $I(X_2; Y|X_1) \neq I(X_2; Y)$. Without

knowledge of the first component, we have

$$\begin{aligned}
I(X_2;Y) &= \sum_{x_1,x_2}\sum_{y}\Pr(y|x_1,x_2)\Pr(x_1)\Pr(x_2)\log_2\left(\frac{\sum_{x_1'}\Pr(y|x_1',x_2)\Pr(x_1')}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')\Pr(x_1')\Pr(x_2')}\right)\\
&= 4\sum_{y}\Pr(y|00)\frac{1}{2}\cdot\frac{1}{2}\log_2\left(\frac{\sum_{x_1'}\Pr(y|x_1',0)\frac{1}{2}}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')\frac{1}{2}\cdot\frac{1}{2}}\right)\\
&= (1-\varepsilon)\log_2\left(2\frac{(1-\varepsilon)}{(1-\varepsilon)}\right)+\sum_{y=?_{00}}\frac{\varepsilon}{\binom{4-1}{3-1}}\log_2\left(2\frac{\sum_{x_1'}\Pr(y|x_1',0)}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\sum_{y=?_{00}}\log_2\left(2\frac{\sum_{x_1'}\Pr(y|x_1',0)}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\sum_{10\notin y}\log_2\left(\frac{2\sum_{x_1'}\Pr(y|x_1',0)}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&\qquad+\frac{\varepsilon}{3}\sum_{11\notin y}\log_2\left(\frac{2\sum_{x_1'}\Pr(y|x_1',0)}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)+\frac{\varepsilon}{3}\sum_{01\notin y}\log_2\left(\frac{2\sum_{x_1'}\Pr(y|x_1',0)}{\sum_{x_1'x_2'}\Pr(y|x_1'x_2')}\right)\\
&= (1-\varepsilon)+\frac{\varepsilon}{3}\left(\log_2\left(\frac{2\cdot\frac{\varepsilon}{3}}{3\frac{\varepsilon}{3}}\right)+\log_2\left(\frac{4\cdot\frac{\varepsilon}{3}}{3\frac{\varepsilon}{3}}\right)+\log_2\left(\frac{4\cdot\frac{\varepsilon}{3}}{3\frac{\varepsilon}{3}}\right)\right)\\
&= 1+\frac{2}{3}\varepsilon-\varepsilon\log_2(3).
\end{aligned}$$

Indeed, this is different from $I(X_2;Y|X_1)$ and so the subchannels are not independent.

Even when the QPEC does not decompose into simple binary or p-ary erasure channels, we can find a simple expression for the mutual information rate for the last subchannel in the decomposition of the p^k-ary partial erasure channel.

Theorem 6. *For the QPEC with erasure probability ε, $q = p^k$ and any M, the last (k^{th}) subchannel has mutual information*

$$\begin{aligned}
&I(X_k;Y \mid X_1,\ldots,X_{k-1})\\
&\quad= (1-\varepsilon)+\frac{\varepsilon}{\binom{q-1}{M-1}}\sum_{t=0}^{\min\{M-1,p-1\}}\binom{q-p}{M-t-1}\binom{p-1}{t}\log_p\left(\frac{p}{t+1}\right).
\end{aligned}$$

Proof. Define the symbol $?_{\mathbf{0}}$ to be the set of all M-sets containing the all-zeros vector.

$$I(X_k; Y \mid X_1, \ldots, X_{k-1}) = \sum_y \Pr(y \mid 0, \ldots, 0) \log_p \left(\frac{p \cdot \Pr(y \mid 0, \ldots, 0)}{\sum_{x'_k} \Pr(y \mid 0, \ldots, 0, x'_k)} \right),$$

where the above equality exploits the symmetry across the values of $X_1, \ldots, X_k$ in the mutual information calculation.

$$I(X_k; Y \mid X_1, \ldots, X_{k-1}) = (1-\varepsilon) + \frac{\varepsilon}{\binom{q-1}{M-1}} \sum_{y \in ?_{\mathbf{0}}} \log_p \left(\frac{p \cdot \frac{\varepsilon}{\binom{q-1}{M-1}}}{\sum_{x'_k} \Pr(y \mid 0, \ldots, 0, x'_k)} \right)$$

Note that there are $\binom{q-1-(p-1)}{M-1-t}\binom{p-1}{t}$ erasure sets of $\mathbf{0}$ containing exactly t symbols of the form $0 \cdots 0 x'_k$, with $x'_k \in \{1, \ldots, p-1\}$. There are $t+1$ input symbols of the form $0 \cdots 0 \tilde{x}_k$ with $\tilde{x}_k \in \{0, \ldots, p-1\}$ for which each such erasure set is possible. Therefore

$$\begin{aligned}
& I(X_k; Y \mid X_1, \ldots, X_{k-1}) \\
=& (1-\varepsilon) + \frac{\varepsilon}{\binom{q-1}{M-1}} \sum_{t=0}^{\min\{M-1, p-1\}} \binom{q-p}{M-t-1} \binom{p-1}{t} \log_p \left(\frac{p \cdot \frac{\varepsilon}{\binom{q-1}{M-1}}}{(t+1)\frac{\varepsilon}{\binom{q-1}{M-1}}} \right) \\
=& (1-\varepsilon) + \frac{\varepsilon}{\binom{q-1}{M-1}} \sum_{t=0}^{\min\{M-1, p-1\}} \binom{q-p}{M-t-1} \binom{p-1}{t} \log_p \left(\frac{p}{t+1} \right).
\end{aligned}$$

□

When $p = 2$, Theorem 6 informs us that the k^{th} subchannel is a simple BEC with

erasure probability given by

$$\begin{aligned}\varepsilon_k &= 1 - \left((1-\varepsilon) + \frac{\varepsilon}{\binom{2^k-1}{M-1}} \sum_{t=0}^{1} \binom{2^k-2}{M-t-1} \binom{1}{t} \log_2 \left(\frac{2}{t+1} \right) \right) \\ &= 1 - \left((1-\varepsilon) + \frac{\varepsilon}{\binom{2^k-1}{M-1}} \binom{2^k-2}{M-1} \right) \\ &= 1 - \left((1-\varepsilon) + \varepsilon \frac{2^k - 1 - (M-1)}{2^k - 1} \right) \\ &= \left(\frac{M-1}{2^k-1} \right) \varepsilon.\end{aligned}$$

The QPEC is also known to decompose when $q = p^2$.

Theorem 7 ([1]). *For the QPEC with ε, $q = p^2$ and $2 \leq M \leq q$, the subchannel mutual information is given by:*

$$I(X_1; Y) = (1-\varepsilon) + \frac{\varepsilon}{\binom{p^2-1}{M-1}} \sum_{t=0}^{\min\{p-1, M-1\}} \binom{p-1}{t} \binom{p^2-p}{M-1-t} \log_p \left(\frac{p(t+1)}{M} \right)$$

$$I(X_2; Y | X_1) = (1-\varepsilon) + \frac{\varepsilon}{\binom{p^2-1}{M-1}} \sum_{t=0}^{\min\{p-1, M-1\}} \binom{p-1}{t} \binom{p^2-p}{M-1-t} \log_p \left(\frac{p}{t+1} \right).$$

Further classification of when the QPEC breaks into simpler channels is an open question. One area of interest is describing the channels the QPEC decomposes into when the channel does not decompose into simple p-ary erasure channels.

Chapter 5

LT Codes on Partial Erasure Channels

Fountain codes, a class of codes designed to transmit data on channels with erasures, were introduced in [41, 42]. Fountain codes received their name from the property that a stream, or digital fountain, of encoded symbols is generated, and receivers collect symbols until they have enough intact packets to decode. Similar to multiple people gathering water droplets from a fountain, the receivers do not need to all collect the same symbols, provided they all receive a sufficient quantity of symbols.

An example of a modern and practical application of fountain codes explained in [43] is broadcasting updates for in-car navigation systems. It is not reasonable to assume that every car that needs an update will be able to receive a transmission at the same time. Nor is it reasonable to assume that a single car will be able to receive a full transmission uninterrupted. A car in a garage or going through a tunnel may miss parts of a transmission. If a fountain code is used, then the cars' navigation systems need only collect enough packets to decode, rather than waiting to receive a specific intact sequence of packets. Updates may be completed more quickly this way, as if some packets are missed, the receiver may just use the next received packets instead of waiting for the transmission to start over.

Luby Transform (LT) codes [17] were the first practical realization of fountain codes. An extension of LT codes, Raptor codes were introduced in [44, 45] and have linear time

encoding and decoding. In Raptor codes, input symbols are precoded prior to being encoded by an LT code, providing an additional method of information recovery in the event that the LT decoder terminates before completion. Researchers have since extended the ideas of LT and Raptor codes to other channels, such as the Binary Symmetric Channel (BSC) and binary Additive White Gaussian Noise (AWGN) Channel [46, 45, 47]. In this section, we consider LT codes on partial erasure channels. This is the first time fountain codes have been considered on these channels.

5.1 Classical LT Codes

We now describe the encoding process and the decoding algorithm for classical Luby Transform codes. The encoding process gives rise to an encoding graph that may be used to visualize the message passing decoding algorithm. As in [17, 48], encoding symbols for an LT code on the BEC are generated as follows:

1. Choose a degree d independently for each encoding bit according to a given degree distribution.

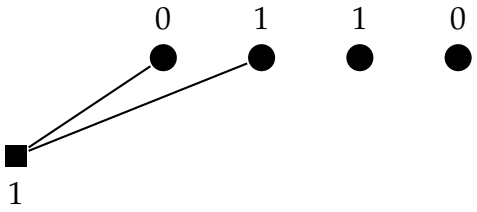
2. Randomly choose d of the k information bits and take the XOR of the chosen bits.

Example 8. Consider an LT Code with four message symbols, 0 1 1 0, and degree distribution

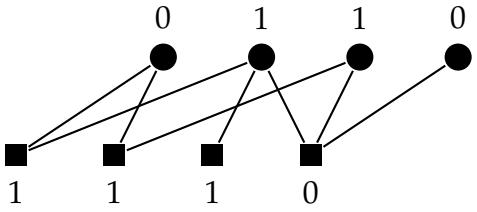
$$p(\text{degree } d) = \begin{cases} \frac{1}{3} & \text{if } d = 1 \\ \frac{1}{2} & \text{if } d = 2 \\ \frac{1}{6} & \text{if } d = 3 \end{cases}.$$

The encoding graph begins with a node for each message symbol (shown below as circular nodes).

Suppose the encoder picks degree 2 from the distribution and randomly selects the first two message nodes. The encoding symbol then takes the value $0 \oplus 1 = 1$. The resulting encoding graph is shown below.



To generate the next symbols, the encoder might then choose degrees 2, 1, and 3 next and (after randomly selecting sets of two, one, and three message nodes to add) end up with the following graph.



If more encoding symbols are needed, the encoder will continue picking degrees and adding (mod 2) the appropriate number of message symbols.

Natural questions about LT encoding include how the degree distribution should be chosen and how many encoding nodes should be generated for each user to receive all the information with a high probability.

Using the standard terminology for LT codes, decoding is performed as follows using the structure of the encoding graph with any erasures discarded.

Decoding Process ([17])

1. Each message node starts in an *uncovered* state.
2. At the first iteration, each degree one encoding node is *released* to *cover* its neighboring message node. These encoding nodes are removed from the decoding graph.
3. The set of covered message nodes not yet *processed* forms a *ripple*.
4. *Process* a randomly selected message node from the ripple. Each neighboring encoding node's symbol is replaced by the XOR of the encoding symbol with the message node bit. Remove the message node and its incident edges.
5. Release any degree one encoding nodes. Continue processing nodes in the ripple and releasing degree one encoding nodes.

The number of encoded symbols that must be received for reliable decoding depends on the degree distribution chosen for the encoding process, and several works have further studied this aspect of the code design [47, 49, 50, 51, 52].

In Luby's seminal work, the *All-At-Once*, *Ideal Soliton*, and *Robust Soliton* degree distributions are discussed to illustrate the impact a distribution has on how many encoding symbols are needed for the resulting LT code to be decoded with a high probability of success [17].

Definition 13 ([17])**.** The *All-At-Once distribution* is defined by $\rho_{all}(1) = 1$. That is, every encoding symbol has degree 1.

With the All-At-Once distribution, generating an encoding symbol has a low cost (the number of symbol operations is small), but a large number of encoding symbols must be

generated to ensure that every message symbol is covered (specifically, $k \ln(k/\delta)$ symbols must be generated to have all input symbols covered with a probability of $1 - \delta$) [17].

Definition 14 ([17])**.** The *Ideal Soliton distribution*, for an encoding graph with n message nodes is defined by

$$\rho_{ideal}(i) = \begin{cases} \frac{1}{n} & \text{for } i = 1 \\ \frac{1}{i(i-1)} & \text{for } i = 2, \ldots, n \end{cases}.$$

and $n \ln(n/\delta)$ encoding symbols must be received. An expected n encoding symbols are needed to cover n message symbols.

A graph built using the Ideal Soliton distribution has an expected ripple size of 1 at every intermediate step of decoding. While constantly having an expected ripple size of 1 is efficient, the Ideal Soliton distribution is not practical. Any deviation from the (theoretical) expected behavior using the Ideal Soliton distribution will cause decoding failure. The Robust Soliton distribution is a modification of the Ideal Soliton distribution that has a higher constant expected ripple size. As a result, the Robust Soliton is more tolerant of deviations from expected behavior without having an unreasonably large overhead.

Definition 15 ([17])**.** Let $\rho_{Ideal}(\cdot)$ denote the Ideal Soliton distribution. Let δ be the allowable probability of decoder failure and $r = c \cdot \ln(n/\delta)\sqrt{n}$. Define

$$\tau(i) = \begin{cases} \frac{r}{in} & \text{for } i = 1, \ldots, \frac{n}{r} - 1 \\ \frac{r \ln(r/\delta)}{n} & \text{for } i = \frac{n}{r} \\ 0 & \text{for } i = \frac{n}{r} + 1, \ldots, n \end{cases}.$$

The *Robust Soliton distribution* is defined as

$$\rho_{robust}(i) = \frac{\rho_{ideal}(i) + \tau(i)}{\sum_{j=1}^{n} \rho_{ideal}(j) + \tau(j)}$$

and has expected ripple size r.

For LT codes on partial erasure channels, the classical encoding process may be used with the following modifications. For the QPEC with $q = p^k$, the sum of information symbols is taken $(\text{mod } q)$, and for the MEC and QMBC, the bitwise sum of information symbols is taken.

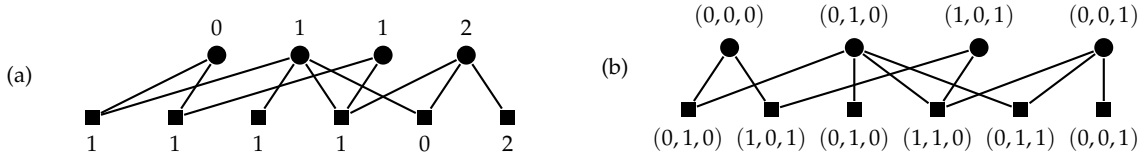


Figure 5.1: Encoding graphs (a) on the QPEC with $q = 3$ and $M = 2$ and (b) on the MEC or QMBC with $q = 8$. Circular nodes represent message symbols and square nodes represent encoding symbols.

Figure 5.1 shows possible encoding graphs on the QPEC and on the MEC or QBMC with degree distribution $p(1) = \frac{1}{3}$, $p(2) = \frac{1}{2}$, and $p(3) = \frac{1}{6}$. While the structure of the encoding graph will still be used to decode, the classical decoding process must be modified to make use of information left in partial erasures.

5.2 Two-phase Decoder for Partial Erasure LT Codes

We now introduce a two-phase decoding process for LT codes on partial erasure channels. "Phase I" mirrors the original LT process on the QEC [17], and "Phase II" allows for further decoding when the Phase I algorithm gets stuck.

Decoding: Phase I

1. Each message node starts in an *uncovered* state.
2. At the first iteration, each degree one encoding node is *released* to *cover* its neighboring message node (i.e. the message node is assigned the value or intersection of sets of values associated with the released encoding node neighbors). These encoding nodes are removed from the decoding graph.
3. The set of covered message nodes not yet *processed* forms a *ripple*.
4. If a message node in the ripple has an associated set of cardinality one, then the message symbol represented by the node is the value of the symbol in the set. Randomly select a message node in the ripple with an associated set of cardinality one, if such a node exists. *Process* the selected message node by subtracting its associated value from each of its neighboring encoding nodes. Remove the message node and its incident edges.
5. Release any degree one encoding nodes. Continue processing nodes in the ripple with an associated set of cardinality one and releasing degree one encoding nodes.

An example of Phase I decoding for the QPEC is shown in Figure 5.2. If all message symbols have been recovered at the end of Phase I, decoding is complete. If there are still message symbols to be recovered, note that there are two possibilities: either the ripple is empty and a decoder failure results, or the ripple is non-empty, all message nodes in the ripple are covered by sets of size greater than 1, and all of the encoding nodes remaining have degree greater than 1. Note that the first case (an empty ripple) is what characterizes decoder failure for LT codes on the QEC.

We will use $\mathcal{N}(e)$ to denote the neighborhood of an encoding node e, and $\mathcal{N}_R(e)$ to denote the subset of $\mathcal{N}(e)$ contained in the ripple. Let E_R be the set of encoding

nodes with $\mathcal{N}(e) = \mathcal{N}_R(e)$; that is, E_R contains all of the encoding nodes that have all of their neighbors in the ripple. Denote by I_m (resp., I_e), the set of symbols associated with message node m (resp., encoding node e). For sets U and V, their sum is the set $\{u + v : u \in U \text{ and } v \in V\}$, and their difference is defined similarly.

Decoding: Phase II

1. For each encoding node $e \in E_R$, send I_m along each incident edge $x_m x_e$ for each $m \in \mathcal{N}(e)$.

2. For each edge $x_m x_e$ in the previous step return $I_e - \sum_{m' \in (\mathcal{N}(e) \setminus m)} I_{m'}$ along $x_e x_m$.

3. Update I_m to be
$$I_m \cap \bigcap_{e \in E_R} \left(I_e - \sum_{m' \in (\mathcal{N}(e) \setminus m)} I_{m'} \right).$$

4. If $|I_m| = 1$ for any m, process x_m and return to Phase I.

Recall that in each partial erasure channel model, the transmitted symbol must be contained in the set of symbols associated with a partially erased node. That is, I_e contains the original sum of the values of the neighboring symbols of e. If a message node m is processed, then the single element in I_m associated with the message node is the correct information symbol represented by that node. Note that at the start of decoding, $|I_e| > 1$ only if the encoding symbol e is partially erased, and $|I_e|$ may increase or decrease as decoding iterations proceed. Moreover, $I_m = \emptyset$ at the start of decoding for each message node m, and once $I_m \neq \emptyset$, the size $|I_m|$ decreases monotonically with increasing decoding iterations.

When all message nodes are processed (i.e. message symbols recovered), decoding is complete. In the LT process [17], there is a one-to-one correspondence between an

encoding symbol covering a message symbol and recovering a message symbol. However, this correspondence no longer holds on the QPEC, MEC, and QMBC.

Example 9. LT Decoding on the QPEC

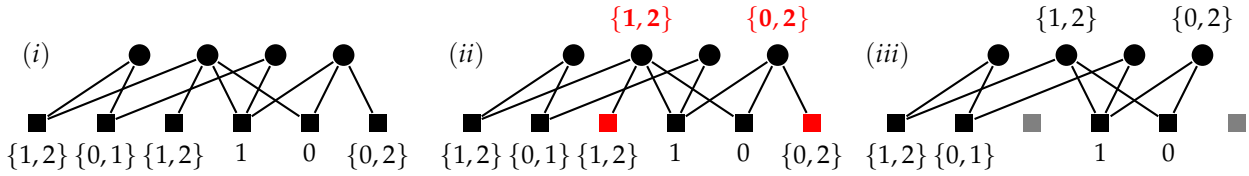


Figure 5.2: Phase I decoding on the QPEC ($q = 3, M = 2$), starting with a possible decoding graph (i) for the encoding graph in Figure 5.1(a). After Step (iii), Phase II begins.

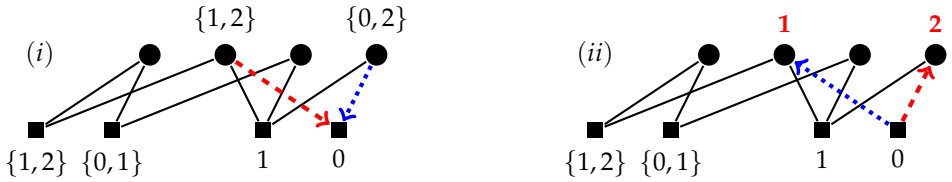


Figure 5.3: Phase II decoding following the Phase I decoding in Fig. 5.2. After Step (ii), the decoder returns to Phase I.

Fig. 5.2 shows Phase I decoding on the QPEC with $q = 3$ and $M = 2$, and Fig. 5.3 shows an example of the Phase II decoding process after Phase I decoding ends in Step (iii) of Fig. 5.2. In Step (i) (resp., Step (ii)) of Fig. 5.3, $\{1,2\}$ (resp., $0 - \{1,2\} = \{1,2\}$) is sent along the dashed edge. At the fourth message node, $\{1,2\}$ is intersected with $\{0,2\}$, and the symbol 2 is recovered. Simultaneously, in Step (i) (resp., Step (ii)) of Fig. 5.3 , $\{0,2\}$ (resp., $\{0,1\}$) is sent along the dotted edge, leading to the recovery of the symbol 1 at the second message node. Decoding now returns to Phase I as in Figure 5.4. Note that decoding would be unsuccessful if the second encoding node had started with the partial erasure $\{1,2\}$ instead of $\{0,1\}$.

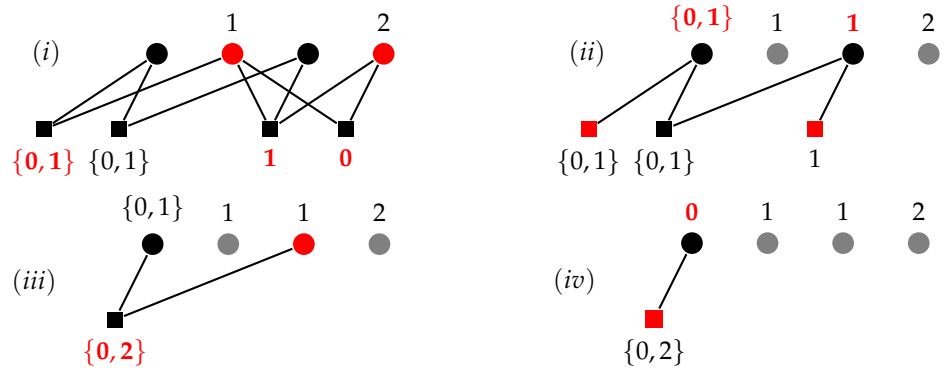


Figure 5.4: Successful Phase I decoding following Phase II decoding in Fig. 5.3.

For the MEC and QMBC, if bitwise sums are taken during the encoding process, then we may decode bits separately. To do this, create k copies of the decoding graph, $G_1, G_2, \ldots, G_k$. In G_i, discard message symbols in which bit i was erased. The usual LT decoding algorithm may be applied to each graph. Figure 5.5 shows a possible overall decoding graph on the MEC along with the corresponding component graph for its first bit.

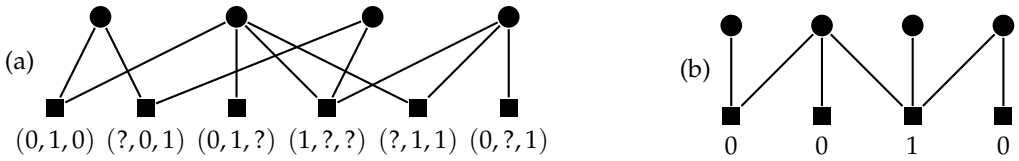


Figure 5.5: (a) A possible decoding graph on the MEC from the encoding graph in Figure 5.1(b), and in (b) the corresponding decoding graph for the first bit.

Proposition 1. *On the MEC or QMBC, the probability of decoding success is the same using the single overall graph with two-phase decoder or with component graphs each with the standard LT decoder.*

Proof. If the graph starts with no degree one encoding nodes, then failure occurs both when using a single graph and when using the multiple graphs. Otherwise, if failure

occurs using the single decoding graph then either: (i) in Phase I, the ripple empties before each message symbol is recovered, or (ii) in Phase II, no message node cardinality is reduced to one after any number of iterations.

In case (i), message nodes previously in the ripple are successfully recovered and processed, but no more encoding nodes are reduced to cardinality one. The degrees of the encoding nodes are the same in the individual bit graphs of the multiple graph case. After the same message symbols are recovered in the multiple graph case, no encoding node will have low enough degree to be released.

In case (ii), some bits are not recovered for message nodes in the ripple. Thus, for at least one of the individual bit graphs, a message node from the ripple in the single graph case will either not reach the ripple (if adjacent encoding symbols are erased), or its processing will not reduce an encoding node to degree one. Thus, if a failure occurs on the single graph, then a failure would also occur if individual component decoding graphs were used instead.

As the encoding symbols are formed by taking bitwise sums of message symbol, any message bit that cannot be recovered using component graphs is also not be recoverable using a single graph. □

For LT codes on the MEC, we are thus able to use a standard LT decoder for each component rather than using the more complex two-phase decoder.

5.3 Number of Encoding Symbols

The number of symbols that must be received in order to guarantee a given level of decoding performance depends on the degree distribution used for encoding. Let δ be the allowable failure probability. For example, for n message symbols on the QEC, the

All-At-Once distribution requires $n \cdot \ln(n/\delta)$ received encoding symbols, and the Robust Soliton distribution requires $n + O(\ln^2(n/\delta)\sqrt{n})$ received encoding symbols [17].

If each component has failure probability at most δ when decoding componentwise on the MEC, then the overall decoding process has failure probability at most $1-(1-\delta)^k$ (as $(1-\delta)^k$ is the probability that no component fails). In order to have overall probability of failure at most δ_{total}, we may set $\delta \leq 1-(1-\delta_{total})^{\frac{1}{k}}$ and determine how many symbols must be received by each component for the given degree distribution.

Proposition 2. *For the MEC with $\gamma_i = \gamma$ for $i = 1, \ldots, k$, an average of at most*

$$\sum_{t=N}^{\infty} t \cdot (1-\gamma)^{tk} \left[\left(\sum_{j=0}^{t-N} \binom{t}{j} \frac{\gamma^j}{(1-\gamma)^j} \right)^k - \left(\sum_{j=0}^{t-N-1} \binom{t-1}{j} \frac{\gamma^j}{(1-\gamma)^{j+1}} \right)^k \right]$$

symbols are required to be transmitted in order to decode successfully using component decoding graphs, where N is the number of encoding symbols required to be received by each component for the specified LT code degree distribution.

Proof. Let X_i be the number of symbols needed for channel i, and let $Z = \max(X_1, \ldots, X_k)$. Then

$$E(Z) = \sum_{t=N}^{\infty} t \Pr(Z = t) = \sum_{t=N}^{\infty} t \left(\prod_{i=1}^{k} \Pr(X_i \leq t) - \prod_{i=1}^{k} \Pr(X_i \leq t-1) \right).$$

If N encoding symbols must be received on each channel, then $\Pr(X_i \leq t)$ is the probability that at most $t-N$ of the first t generated symbols are erased, so

$$\Pr(X_i \leq t) = \sum_{j=0}^{t-N} \binom{t}{j} \gamma_i^j (1-\gamma_i)^{t-j},$$

γ	.005	.1	.25	.45
$E(Z)$	3.0590	3.9506	5.2425	7.7521
$N/(1-\gamma)^k$	3.0608	4.5725	9.4815	32.7846

Table 5.1: $E(Z)$ on the constrained MEC and $\frac{N}{(1-\gamma)^k}$ for $N = 3, k = 4$, and various γ.

and $E(\max(X_1, \ldots, X_k) =$

$$\sum_{t=N}^{\infty} t \left[\prod_{i=1}^{k} \sum_{j=0}^{t-N} \binom{t}{j} \gamma_i^j (1-\gamma_i)^{t-j} - \prod_{i=1}^{k} \sum_{j=0}^{t-N-1} \binom{t-1}{j} \gamma_i^j (1-\gamma_i)^{t-1-j} \right].$$

For the constrained MEC, this becomes $E(\max(X_1, \ldots, X_k)) =$

$$\sum_{t=N}^{\infty} t(1-\gamma)^{tk} \left[\left(\sum_{j=0}^{t-N} \binom{t}{j} \frac{\gamma^j}{(1-\gamma)^j} \right)^k - \left(\sum_{j=0}^{t-N-1} \binom{t-1}{j} \frac{\gamma^j}{(1-\gamma)^{j+1}} \right)^k \right].$$

□

Note that on the MEC, an expected $\frac{N}{1-\varepsilon}$ symbols must be generated before N are received with no (full or partial) erasures. On the constrained MEC, $1 - \varepsilon = (1-\gamma)^k$, and thus an expected $\frac{N}{(1-\gamma)^k}$ symbols must be generated before N are received with no (full or partial) erasures.

In Table 5.1, the second row shows the expected number of symbols to be generated on the constrained MEC for $N = 3$, $k = 4$, and various γ. The third row shows the expected number of symbols to be generated before 3 are received with no (full or partial) erasures. Note that as the bit erasure probability γ increases, partial erasures become more common. Since any partial erasure is treated as a full erasure on the QEC, more outputs are treated as erasures on the QEC than on the MEC.

Another estimate for the required number of symbols to be generated can be found by

finding the least integer M such that the expected number of symbols received given that M are sent is at least N for each component. The expected number of symbols received by the component with the most erasures (note that this might not be the component with the highest erasure probability) given that M are sent is

$$\sum_{(i_1,i_2,\cdots,i_k)} \left(\left(\prod_{j=1}^{k} \binom{M}{i_j} \right) \gamma^{\sum_{j=1}^{k} i_j} (1-\gamma)^{\sum_{j=1}^{k} (M-i_j)} \min_{j=1,\ldots,k} (M-i_j) \right),$$

where $(i_1, i_2, \cdots, i_k)$ is a k-tuple with integer entries between 0 and M.

If decoding is successful on the QEC, then decoding will also be successful on the MEC without generating additional encoding symbols. Thus the expected number of encoding symbols that must be generated for successful decoding on the MEC may be bounded above by the minimum of the expected number needed for the corresponding QEC and the number found in the Proposition 2.

Example 10. Consider an LT Code with $q = p^3$, $n = 10$ message symbols, and allowable probability of decoding failure $\delta_{total} = 0.0014$. Using the All-At-Once distribution, approximately 89 intact encoding symbols must be received on the QEC. In order for each component to have decoding failure at most $1 - (1 - \delta_{total})^{\frac{1}{k}}$, approximately 100 symbols must be received by each component on the MEC.

Figure 5.6 shows the upper bound given by Proposition 2 for the expected number of symbols that must be generated on the MEC for various γ as well as the expected number of encoding symbols that must be generated on the corresponding QEC. Note that the number of symbols needed for the QEC is also an upper bound on the number needed for the MEC.

Lemma 2. *The QMBC has no gain (in terms of number of symbols generated) over the corresponding QEC.*

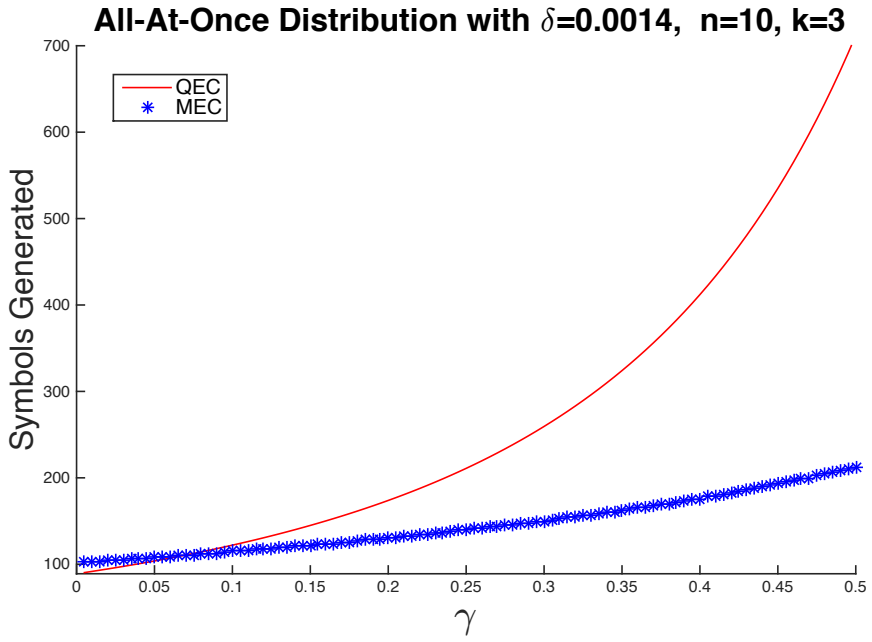


Figure 5.6: Upper Bound on Expected Number of Symbols Generated using the All-At-Once Distribution with $\delta_{total} = 0.0014$, n=10, and $k = 3$.

Proof. Note that if any erasure event occurs on the QMBC, then the rightmost bit is erased. Decoding is successful exactly when decoding of the rightmost bit of each symbol is successful. The effective channel for the rightmost bit is a BEC with erasure probability $\sum_{i=1}^{k} \varepsilon_i = 1 - \varepsilon_0$, and an expected $\frac{N}{\varepsilon_0}$ symbols must be generated. This is the same as the expected number of symbols to be generated before N are received with no (full or partial) erasures. □

5.4 Density Evolution Analysis

Since LT codes on the QPEC can not be decomposed and analyzed using component graphs as in the MEC or QMBC, we give an approximation of density evolution equations for the two-phase decoder on the QPEC. Density evolution is an asymptotic analysis that tracks how message probability density functions evolve as iterations progress, and

is based on the degree distribution of the graph [14, 53]. The analysis assumes that messages received at each node are independent (i.e. the graph is cycle free). Typically this analysis results in determining a threshold, called the *decoding threshold*, that gives the worst channel condition for which the code may be decoded with an arbitrarily small probability of error. A density evolution analysis of LDPC codes with an iterative subset sum decoding on the QPEC was done in [8]. In contrast to density evolution equations for LDPC codes with a standard belief propagation decoder, we adapt these density evolution equations for the encoding and message nodes for the second part of the two-phase decoder in the LT code framework.

Using the notation from [8], let $\mathcal{S}_1, \ldots, \mathcal{S}_{2^q-1}$ denote the nonempty subsets of GF(q), and let $\mathcal{I}_i$ be an ordered list of i (not necessarily distinct) indices from $1, \ldots, 2^q - 1$. Let $\chi_t(\mathcal{I}_i)$ indicate if the sets indexed in $\mathcal{I}_i$ lead to information set S_t at a message node. That is, $\chi_t(\mathcal{I}_i)$ is 1 if $\bigcap_{j \in \mathcal{I}_i} S_j$ is S_t, and 0 otherwise. Similarly, let $\eta_t(\mathcal{I}_i)$ indicate if the sets indexed in $\mathcal{I}_i$ lead to information set S_t at an encoding node. That is, $\eta_t(\mathcal{I}_i)$ is 1 if the sumset of sets in $\mathcal{I}_i$ is S_t, and 0 otherwise. Let $\theta_t = 1$ if $|\mathcal{S}_t| = 1$, and o otherwise. Denote by $p^{\ell}_{m \to e}(\mathcal{S}_t)$ (resp., $p^{\ell}_{e \to m}(\mathcal{S}_t)$) the probability that $\mathcal{S}_t$ is sent from a message node to an encoding node (resp., from an encoding node to a message node) in iteration ℓ.

We now derive density evolution equations for Phase II as an approximate analysis for the two-phase decoder. While the expected performance of LT codes is optimized for the Ideal Soliton distribution, in practice the performance is better when encoding is done using a Robust Soliton degree distribution with an expected ripple size greater than one, so that the ripple does not empty before decoding is successfully completed [17]. Let λ_i (resp., ρ_i) be the fraction of edges incident to a message node (resp., encoding node) of degree i. For the Ideal or Robust Soliton degree distribution,

$$\lambda_i = e^{-d_\rho/R} \frac{(d_\rho/R)^{i-1}}{(i-1)!}, \quad \rho_i = \frac{niE_i}{\sum_i niE_i},$$

where $R = \frac{\text{\# message symbols}}{\text{\# encoding symbols}}$ and E_i is the probability of an encoding node having degree i, d_ρ is the average degree of encoding nodes given by $\frac{\sum_i niE_i}{n}$ [54], and n is the number of encoding nodes (assumed to be large for the density evolution analysis). For $\ell \geq 1$, the probability that a message $\mathcal{S}$ is sent by a message node is the probability that there is at least one message node that receives messages from encoding nodes that intersect to give $\mathcal{S}$ in iteration $\ell - 1$. This can be found by summing over the degrees i in the message node distribution and finding the probability that a neighborhood of size i exists and will yield $\mathcal{S}$ at a message node. This yields

$$p^{\ell}_{m\to e}(\mathcal{S}_t) = \sum_{i\geq 2, \lambda_i \neq 0} \lambda_i \sum_{\mathcal{I}_{i-1}} \left(\prod_{j\in\mathcal{I}_{i-1}} p^{\ell-1}_{e\to m}(\mathcal{S}_j) \right) \cdot \chi_t(\mathcal{I}_{i-1}).$$

To find the probability that a message $\mathcal{S}$ is sent by an encoding node in iteration ℓ, we consider the probabilities of a message being sent by an intact encoding node or a (partially) erased encoding node as well as the probabilities of a message being sent by a degree one encoding node or an encoding node of higher degree. The degree one encoding nodes will not send updated messages. The higher degree encoding nodes will send messages that are sumsets of the messages received from neighboring message nodes.

$$p^{\ell}_{e\to m}(\mathcal{S}_t) = \rho_1 p^0_{e\to m}(t)(1-\theta_t) + \frac{1-\varepsilon}{q}\theta_t + \varepsilon \sum_{i\geq 2, \rho_i \neq 0} \rho_i \sum_{\mathcal{I}_{i-1}} \left(\prod_{j\in\mathcal{I}_{i-1}} p^{\ell}_{m\to e}(\mathcal{S}_j) \right) \cdot \eta_t(\mathcal{I}_{i-1}).$$

Recall that each received encoding node will either be partially erased (and so initially have a set of cardinality M) or be intact. We assume that the inputs have a uniform distribution. The probability that a given set of cardinality 1 is received is $\frac{1}{q}(1-\varepsilon)$ and the probability that a given set of cardinality M is received is $\sum_{x\in M} \frac{1}{q} \frac{\varepsilon}{\binom{q-1}{M-1}} = \frac{\varepsilon}{\binom{q}{M}}$. The initial conditions for the analysis (when $\ell = 0$) are thus

$$p^0_{e\to m}(S_t) = \begin{cases} \frac{\varepsilon}{\binom{q}{M}} & \text{if } |\mathcal{S}_t| = M \\ (1-\varepsilon)\frac{1}{q} & \text{if } |\mathcal{S}_t| = 1 \\ 0 & \text{else} \end{cases} .$$

At the end of iteration ℓ, the probability of decoding failure is given by $1 - \sum_{t:|S_t|=1} p^\ell_{m\to e}(S_t)$, i.e. failure occurs if the cardinality of the information set sent by some message node is never one. The threshold of the density process is defined as

$$\epsilon^* = \arg\max_{\varepsilon} \left\{ \left(1 - \sum_{|S_t|=1} p^\ell_{m\to e}(\mathcal{S}_t) \right) \to 0 \text{ as } \ell \to \infty \right\},$$

and gives a lower bound on the decoding threshold for the two-phase decoder. Note that any degree one encoding nodes continue to send their received message throughout the analysis. While degree one encoding nodes are essential to the LT process, their presence results in a non-vanishing term in the $p^\ell_{e\to m}(\mathcal{S}_t)$ equation (specifically, the first term in the equation). Thus density evolution analysis for LT codes are hindered by an error floor [54]. This may be overcome by using an outer code that may be used to decode if LT decoding fails, as is done with Raptor codes [45].

Figure 5.4 shows the asymptotic performance of LT codes on the QPEC with $q = 4$, $M = 2$ or 3, and $\varepsilon = 0.1$. The plot shows the error rate decreasing as a function of R. Due to the complexity of the density evolution analysis on the QPEC, the density evolution equations were run for 10 iterations using a modification of the Ideal Soliton distribution without high degree encoding nodes. The performance curves are similar for these parameters due to the degree one encoding node terms. Optimizing the degree distribution for LT codes on partial erasure channels, and in particular on the QPEC, is a subject for future work.

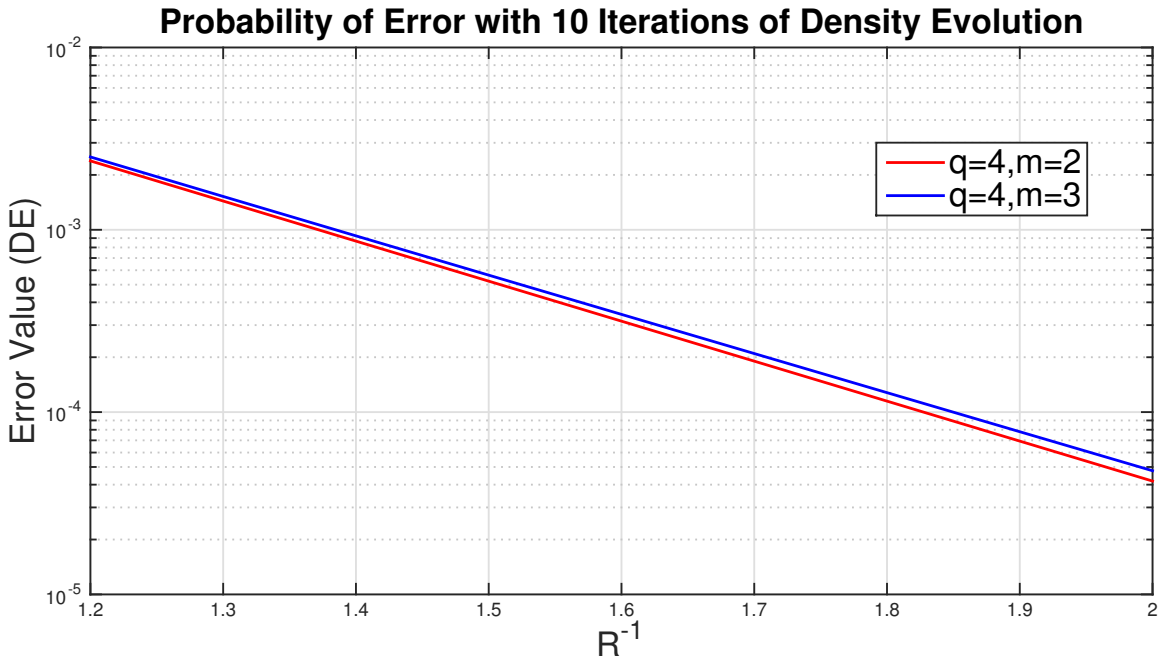


Figure 5.7: Performance of LT codes on QPEC for $\varepsilon = 0.1$ using a modified Ideal Soliton distribution.

Chapter 6

Partial Erasure Relay Channels

Relay channels were introduced by van der Meulen in [18]. In a relay channel, information is sent to a receiver through a channel both in the usual way (see Figure 2.1) and through a relay node. Information from the relay node can be combined with information sent directly to assist with decoding at the receiver (for example, decode-and-forward, compress-and-forward, or amplify and forward [55]). Relay channels may be used to model a variety of applications in wireless networks [56, 57, 58, 59]. In this section, we examine the case of a relay channel in which one of the links is a partial erasure channel. This is the first time partial erasure channels have been considered in this setting.

6.1 Background: The Erasure Relay Channel

We now explain the simple relay network of [19] consisting of a sender, a receiver, and a relay node that helps communicate packets from the sender to the receiver.

Definition 16. [60] The relay channel with one relay node is described by random variables X^0 (the channel input), X^1 (the message sent by the relay), Y^0 (the sender-receiver channel output), Y^1 (the sender-relay channel output), and Y^2 (the relay-receiver channel output) and a conditional probability density function $p(y^0, y^1, y^2|x^0, x^1)$.

A model of the relay channel is shown in Figure 6.1.

Definition 17. ([19]) The relay channel is said to be *(physically) degraded* if

$$p(y^0, y^1, y^2|x^0, x^1) = p(y^1|x^0, x^1)p(y^0, y^2|y^1, x^1)$$

(equivalently, $X^0 \to (X^1, Y^1) \to (Y^0, Y^2)$ is a Markov Chain).

The definition of a degraded relay channel tells us that on such a channel, the probability of receiving y^0, y^2 at the receiver depends only on the probabilities of symbols being sent and received by the relay. In particular, on a degraded erasure relay channel, any symbol received intact by the receiver from the source is also received intact at the relay [61].

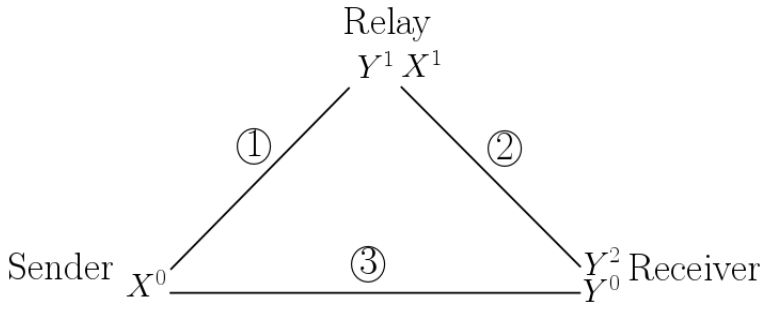


Figure 6.1: The three terminal erasure relay channel. At each link ①, ②, ③, information is sent through a channel.

The capacity of several types of relay channels as well as an information theoretic Min-Cut, Max-Flow cut-set bound for the capacity of a general relay channel were given in [19]. The standard three-terminal erasure relay network is referred to as the QEC-QEC-QEC relay channel, and achievability of the cut-set bound for the physically degraded QEC-QEC-QEC relay channel, was shown in [60]. An extension to the achievability of the cut-set bound for a channel with multiple relays in the erasure relay setting is given in [61]. However, the problem of determining the capacity of a relay channel remains open

in general. It was shown that the cut-set bound is not met for Gaussian relay networks [62]. The cut-set bound of [19] is given next.

Theorem ([19]). *For a general relay channel,*

$$C \leq \max_{p(x^0,x^1)} \min \left\{ I(X^0, X^1; Y^0, Y^2), I(X^0; Y^0, Y^1, Y^2 \mid X^1) \right\}.$$

If Y^0 is a degraded form of Y^1, then

$$C \leq \max_{p(x^0,x^1)} \min \left\{ I(X^0, X^1; Y^0, Y^2), I(X^0; Y^1 \mid X^1) \right\}.$$

This can be interpreted as saying that the capacity on a relay channel is bounded above by the minimum of (i) the information passed from the source to the relay or receiver and (ii) the information passed from the source or relay to the receiver. Each of these corresponds to a *cut* in the network, as shown in Figure 6.2. The former is sometimes referred to as the *cooperative broadcast bound* and the latter is sometimes referred to as the *cooperative multiple access bound* [55].

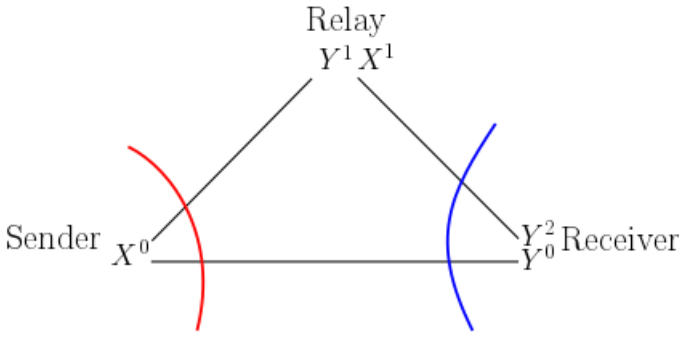


Figure 6.2: Cuts in the relay channel network from (i) the source to the relay or receiver and (ii) the source or relay to the receiver.

At the relay, several different schemes may be employed to assist with decoding. As in [60], we will assume that the message sent by the relay in the i^{th} time step is a function

of the messages received by the relay in the first $i-1$ time steps. For a given time step, it is then reasonable to assume that the messages sent by the sender and the relay are independent. To consider partial erasures in a relay setting, we can let any combination of the sender-receiver, sender-relay, and relay-receiver links be partial erasure channels.

6.2 Cut-set Bound for the MEC-QEC-QEC Relay Channel

We will consider a degraded relay channel in which the source-relay link is a MEC, and the other links are QECs. We will refer to this as the MEC-QEC-QEC relay channel. We derive the cut-set bound for this relay channel setting as follows.

Theorem 8. *The capacity region of the degraded erasure relay channel where the sender-to-receiver and the relay-to-receiver links are QECs over* $\mathrm{GF}(p^j)$ *with erasure probabilities* ε *and* ε_2*, respectively and the sender-to-relay is a MEC over* $\mathrm{GF}(p^j)$ *with erasure probabilities* $\gamma_1, \ldots, \gamma_j$ *and there is no interference at the receiver is upper bounded by*

$$R \leq \max_{\alpha \in [0,1]} \min \left\{ \frac{1}{j} \sum_{i=1}^{j} (1-\gamma_i), (1-\varepsilon) + \alpha(1-\varepsilon_2) \right\}.$$

For the case of the the constrained MEC with $\gamma_1 = \cdots = \gamma_j = \gamma$*, we have*

$$R \leq \max_{\alpha \in [0,1]} \min \left\{ (1-\gamma), (1-\varepsilon) + \alpha(1-\varepsilon_2) \right\}.$$

Let α^* be the value of α maximizing the bound. Note that as in [60], α^* can be found

explicitly. Specifically,

$$\alpha^* = \begin{cases} 0 & \text{if } \frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i) \leq 1-\varepsilon \\ \frac{\varepsilon - 1 + \frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i)}{1-\varepsilon_2} & \text{if } (1-\varepsilon) < \frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i) < (1-\varepsilon) + (1-\varepsilon_2) \cdot \\ 1 & \text{if } (1-\varepsilon) + (1-\varepsilon_2) \leq \frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i) \end{cases}$$

When $\alpha = \alpha^*$ as above, $\frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i)$ will be the minimum term in the bound exactly when $(1-\varepsilon) + (1-\varepsilon_2) \geq \frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i)$.

Proof. Recall that the MEC may be decomposed into independent p-ary erasure channels. The cut-set bound of [19] informs us that the capacity region of a degraded relay channel may be bounded above by

$$R \leq \max_{p(x^0,x^1)} \min \left\{ I(X^0; Y^1 \mid X^1), I(X^0, X^1; Y^0, Y^2) \right\}.$$

As in [60], this may be rewritten as

$$R \leq \max_{p(x^0,x^1)} \min \left\{ I(X^0; Y^1) - I(X^1; Y^1), I(X^0; Y^0) + I(X^1, Y^2 \mid X^0) \right\}.$$

Both terms in the bound are maximized when X^1 and X^0 are independent. In this case, the first term becomes $I(X^0; Y^1)$. Recall, for the MEC, we have $I(X^0; Y^1) \leq \frac{1}{j}\sum_{i=1}^{j}(1-\gamma_i)$. Now consider the second term. We assume no interference at the receiver, so we may view the relay-receiver and sender-receiver links as channels with independent erasures. As in [60], when X^0 and X^1 are independent and $H(X^1) = \alpha H(X^0)$ for some $0 \leq \alpha \leq 1$, we have

$$I(X^0; Y^0) + I(X^1, Y^2 \mid X^0) = (1-\varepsilon)H(X^0) + (1-\varepsilon_2)\alpha H(X^0).$$

□

6.3 Achievability

We now show how the cut-set bound of Theorem 8 may be achieved using MDS codes and multilevel coding. For relay channels with simple erasure channel links, it is known that MDS codes achieve capacity [60]. We next show how we can meet the cut-set bound in the MEC-QEC-QEC relay setting by using MDS codes and multilevel coding. This then proves that the cut-set bound defines the capacity region of this channel.

6.3.1 On the Constrained MEC

Consider the degraded MEC-QEC-QEC relay channel where the sender-to-receiver and the relay-to-receiver links are QECs with erasure probabilities ε and ε_2, respectively and the sender-to-relay is a constrained $q = p^j$-ary MEC with p-ary symbol erasure probability γ.

Assume there are k q-ary information symbols to be sent. Recall that the MEC may be decomposed into j independent p-ary erasure channels. Each q-ary symbol to be sent by the sender can be expanded into j p-ary symbols. The sender will encode the resulting kj p-ary symbols using multilevel coding as explained in Chapter 4 and transmit the multilevel encoded sequence across both the MEC and the QEC. Moreover, we will require that the component codes each be MDS. On the constrained MEC we can use the same component code for each component. Let $[I_{k\times k}|A_{k\times m}]$ be the generator matrix for the component MDS code over GF(p). Thus there are nj p-ary symbols which are then converted to n q-ary symbols which the sender transmits to the receiver over a q-ary erasure channel and also transmits to the relay over the MEC.

The erasure probability of the sender-receiver connection is ε, thus on average for

large n, the receiver obtains $n(1-\varepsilon)$ q-ary symbols intact by the law of large numbers. Moreover, when a symbol is erased on this channel, it causes an erasure at the p-ary level in each of the j component codes (due to the structure of this multilevel coding scheme). Thus, there are $n(1-\varepsilon)$ p-ary symbols intact in each of the component codes.

At the relay, for each of the j component codes, there are $n(1-\gamma)$ intact p-ary symbols on average for large n, by the law of large numbers. Here, in contrast to the QEC, the positions need not be the same for each of them (i.e. different locations may be erased for different components) because the channels for the components are independent. Recall that for an $[n,k,d]_q$ MDS code, a codeword may be recovered from any k symbols within the word. If $[I_{k\times k}|A_{k\times m}]$ is MDS over GF(p), the relay will be able to recover all symbols in each of the component codes as long as $k \leq n(1-\gamma)$. Then, the relay encodes the k original q-ary information symbols also using multilevel coding (so that both the symbols sent over the sender-receiver and relay-receiver links use multilevel coding). In this case, each component code has generator matrix $[B_{k\times \ell}]$ over GF(p) where $[I_{k\times k}|A_{k\times m}|B_{k\times \ell}]$ is also MDS. The k original q-ary information symbols are encoded to give a length ℓ q-ary codeword that is transmitted over a QEC with erasure probability ε_2 to the receiver.

On average, for large n, the receiver receives $\ell(1-\varepsilon_2)$ intact q-ary symbols from the relay, which means that there are on average $\ell(1-\varepsilon)$ p-ary symbols intact on each component code. So long as $k \leq n(1-\varepsilon)+\ell(1-\varepsilon_2)$, each component code can recover all symbols if $[I_{k\times k}|A_{k\times m}|B_{k\times \ell}]$ is also MDS.

6.3.2 On the Unconstrained MEC

Now we consider the same relay setting with the unconstrained MEC with component erasure probabilities $\gamma_1,\ldots,\gamma_k$. On the unconstrained MEC, j potentially different component codes $C_1,\ldots,C_j$, each of block length n, are used. Code C_i has rate $\dfrac{k_i}{n}$ for $i=1,\ldots,j$, and $\sum_{i=1}^{j} k_i = kj$. Assume each component code C_i is chosen to be an MDS code with gen-

erator matrix $[I_{k_i \times k_i} | A_{k_i \times m_i}]$ over GF(p), where $k_i + m_i = n$ and $\sum_{i=1}^{j} m_i = j(n-k) = jm$. Again, there are nj p-ary symbols which are then converted to n q-ary symbols which the sender transmits to the receiver over a q-ary erasure channel and also transmits to the relay over the MEC.

The erasure probability of the sender-receiver connection is ε, thus on average, the receiver obtains $n(1-\varepsilon)$ q-ary symbols intact. Moreover, when a symbol is erased on this channel, it causes an erasure at the p-ary level in each of the j component codes. Thus, there are $n(1-\varepsilon)$ p-ary symbols intact in each of the component codes.

At the relay, for component code C_i, there are $n(1-\gamma_i)$ intact p-ary symbols. As long as $k_i \leq n(1-\gamma_i)$ for every i, the relay will be able to recover all symbols in each of the component codes, and thus, recover all information symbols. Note that this implies that $kj \leq \sum_{i=1}^{j}(1-\gamma_i)n$.

Then, the relay encodes the k_i original q-ary information symbols, again using an MDS code with length ℓ and generator matrix $[B_{k_i \times \ell}]$ over GF(p) where $[I_{k_i \times k_i} | A_{k_i \times m_i} | B_{k_i \times \ell}]$ is also MDS. The q-ary codeword is transmitted over a QEC with erasure probability ε_2 to the receiver.

On average, the receiver receives $\ell(1-\varepsilon_2)$ intact q-ary symbols from the relay, which means that there are on average $\ell(1-\varepsilon)$ p-ary symbols intact on each component code. So long as $k_i \leq n(1-\varepsilon) + \ell(1-\varepsilon_2)$ for each i, each component code can recover all symbols so long as the code generated by $[I_{k_i \times k_i} | A_{k_i \times m_i} | B_{k_i \times \ell}]$ is also MDS.

Let $M_i = [I_{k_i \times k_i} | A_{k_i \times m_i} | B_{k_i \times \ell}]$ for $i = 1, 2, \ldots, j$. Assuming that the p-ary symbols are assigned to component codes in order (i.e. the first k_1 symbols are encoded by C_1, the next k_2 symbols are encoded by C_2, and so on), the receiver decodes using the MDS code with generator matrix M_i on the ith component. The overall representation for this

coding scheme may be given by the $jk \times (jk + jm + j\ell)$ generator matrix

$$\begin{pmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & M_j \end{pmatrix}$$

for the p-ary expansion that has MDS codes on each component.

Chapter 7

Hypergraph Codes

The design of codes for distributed storage has recently become a popular area of study [4, 63, 64]. Increasing amounts of data must be stored and accessed. When storing large amounts of data, it is inconvenient or infeasible to frequently access all the data stored in the system. Consequently, there is interest in codes that can be decoded locally by accessing small amounts of data [22, 23, 65]. Other properties of interest include the storage overhead and repair bandwidth costs [5], latency [66, 67], and privacy of codes [68] for distributed storage.

As we saw in Chapter 2, codes may be defined using graphs. Codes may also be defined on *hypergraphs*, a generalization of graphs in which edges are formed by subsets of vertices. Codes from regular hypergraphs with expansion-like properties were introduced in [20] as a family of asymptotically good binary codes. The existence of hypergraph codes meeting the Gilbert-Varshamov bound and an improved decoding algorithm for codes on hypergraphs was found in [21]. More recently, codes from expander graphs were considered for locality in[22]. In this section, we introduce a construction of a biregular hypergraph and examine the locality, availability, and erasure-correcting capabilities of the resulting hypergraph codes.

Let $\mathcal{H}$ denote a hypergraph. $\mathcal{H}$ is said to be *t-uniform* if every edge contains exactly t vertices (so a graph is a 2-uniform hypergraph) and *t-partite* if its vertex set can be

partitioned into t sets $V_1, \ldots, V_t$ such that no edge contains more than one vertex from any part. $\mathcal{H}$ is Δ-regular if every vertex belongs to Δ edges. As in [20], we will use $\mathcal{H} = (V_1, V_2, \ldots, V_t; E)$ to denote a t-uniform, t-partite hypergraph with vertex sets $V_1, \ldots, V_t$ and edge set E. We will also assume that no two edges in E are identical.

The *locality* of a code measures how many code symbols must be used to recover an erased code symbol. While there are a variety of locality notions relevant to coding for distributed storage, we focus on (r, ℓ)-cooperative locality and (r, τ)-availability [22, 23].

Definition 18. A code C has (r, ℓ)-*cooperative locality* if for any $\mathbf{y} \in C$, any set of ℓ symbols in $\mathbf{y}$ are functions of at most r other symbols. Furthermore, C has (r, τ)-*availability* if any symbol in $\mathbf{y}$ can be recovered by using any of τ disjoint sets of symbols each of size at most r.

7.1 Biregular Hypergraph Code Construction and Properties

A construction of t-uniform t-partite Δ-regular hypergraphs is presented in [20] based on an underlying regular bipartite expander graph. We show how to obtain a t-uniform t-partite (Δ_1, Δ_2)-biregular hypergraph from a (c, d)-regular bipartite graph in a similar way. We provide bounds on the stopping set size, cooperative locality, rate, and minimum distance for the resulting hypergraph codes. The proofs of these results come from straightforward generalizations of a co-author's arguments for the regular hypergraph case [3]. However, they are included for completeness.

Definition 19. We say that a t-uniform t-partite hypergraph $\mathcal{H} = (V_1, \ldots, V_t; E)$ is (Δ_1, Δ_2)-*biregular* if the parts can be labeled such that each vertex in an odd (resp., even) index part is contained in Δ_1 (resp., Δ_2) edges.

Construction 1. *Let $G = V \cup W$ be a (c, d)-regular bipartite expander graph with $|V| \geq |W|$. For $t \in \mathbb{N}$, construct a t-uniform t-partite hypergraph $\mathcal{H}$ with parts $V_1, \ldots, V_t$ as follows. For odd (resp., even) i, let V_i be a copy of V (resp., W). Take $E(\mathcal{H})$ to be the set of edges corresponding to walks of length $t-1$ in G. That is $(v_1, \ldots, v_t)$ with $v_i \in V_i$ is in $E(\mathcal{H})$ if and only if $(v_1, \ldots, v_t)$ corresponds to a walk in G.*

Note that $\mathcal{H}$ is indeed t-uniform and t-partite, and has vertices of degree $\Delta_1 = c^{\lceil \frac{t}{2} \rceil} d^{\lfloor \frac{t}{2} \rfloor - 1}$ (resp., $\Delta_2 = c^{\lceil \frac{t}{2} \rceil - 1} d^{\lfloor \frac{t}{2} \rfloor}$) in odd (resp., even) index parts.

A *stopping set* in the Tanner graph of a generalized LDPC code is a subset S of the variable nodes such that each neighbor of S has at least $d_{min}(C)$ neighbors in S, where C is the subcode represented by the constraint vertices [69, 70]. The definition of a stopping set may be extended to biregular hypergraph codes. On erasure channels, stopping sets characterize iterative decoding failure, as when the symbols in a stopping set is erased, the decoder is unable to proceed.

Definition 20. Let $\mathcal{Z}$ be a code on a hypergraph $\mathcal{H} = (V_1, V_2, \ldots, V_t; E)$, with the edges representing the code symbols and the vertices representing the constraints of a subcode C_1 (resp., C_2) if the vertex is in an odd (resp., even) index part. Then a *stopping set* S is a subset of the edges of $\mathcal{H}$ such that every vertex contained in an element of S is contained in at least $d_{\min}(C_1)$ (resp., $d_{\min}(C_2)$) elements of S if the vertex is in an odd (resp., even) index part.

We now give a bound on the minimum stopping set size and (r, ℓ)-cooperative locality of codes resulting from Construction 1.

Theorem 9. *Let $\mathcal{H}$ be a t-uniform t-partite (Δ_1, Δ_2)-biregular hypergraph. If the vertices in an odd (resp., even) index part of $\mathcal{H}$ represent constraints of a subcode C_1 (resp., C_2) with block length*

Δ_1 *(resp.,* Δ_2*), then the size of the minimum stopping set,* $s_{\min}(\mathcal{H})$*, is bounded by*

$$s_{\min}(\mathcal{H}) \geq \left(d_{\min}(C_1)^{\lceil \frac{t}{2} \rceil} d_{\min}(C_2)^{\lfloor \frac{t}{2} \rfloor}\right)^{1/(t-1)}. \tag{7.1}$$

Proof. Let $\mathcal{H}$ be as above, and let S be a minimum stopping set for $\mathcal{H}$. As $\mathcal{H}$ is t-uniform and t-partite, each edge in S contains exactly one constraint node from each of the t parts of $\mathcal{H}$. Therefore each part of $\mathcal{H}$ has exactly $|S| = s_{\min}(\mathcal{H})$ incident edges belonging to S. Each constraint node contained in an edge in S in and odd (resp., even) index part must be contained in at least $d_{\min}(C_1)$ (resp., $d_{\min}(C_2)$) edges in S. By the pigeonhole principle, the number of vertices in any odd (resp., even) index part of $\mathcal{H}$ that are contained in some edge in S is bounded above by $\frac{s_{\min}(\mathcal{H})}{d_{\min}(C_1)}$ (resp., $\frac{s_{\min}(\mathcal{H})}{d_{\min}(C_2)}$).

There are $\lceil \frac{t}{2} \rceil$ odd index parts and $\lfloor \frac{t}{2} \rfloor$ even index parts of $\mathcal{H}$. There are therefore at most $\left(\frac{s_{\min}(\mathcal{H})}{d_{\min}(C_1)}\right)^{\lceil \frac{t}{2} \rceil} \left(\frac{s_{\min}(\mathcal{H})}{d_{\min}(C_2)}\right)^{\lfloor \frac{t}{2} \rfloor} = \frac{s_{\min}(\mathcal{H})^t}{d_{\min}(C_1)^{\lceil \frac{t}{2} \rceil} d_{\min}(C_2)^{\lfloor \frac{t}{2} \rfloor}}$ edges in S. Thus,

$$\frac{s_{\min}(\mathcal{H})^t}{d_{\min}(C_1)^{\lceil \frac{t}{2} \rceil} d_{\min}(C_2)^{\lfloor \frac{t}{2} \rfloor}} \geq s_{\min}(\mathcal{H})$$

and so

$$s_{\min}(\mathcal{H}) \geq \left(d_{\min}(C_1)^{\lceil \frac{t}{2} \rceil} d_{\min}(C_2)^{\lfloor \frac{t}{2} \rfloor}\right)^{1/(t-1)}.$$

□

Note that the guaranteed number of erasures that may be recovered at once is one less than the size of the smallest stopping set. Thus we have the following corollary.

Corollary 1. *If the subcodes* C_1 *(resp.,* C_2*) of the biregular hypergraph code* $\mathcal{Z}$ *have* r_1 *(resp.,* r_2*) locality then* $\mathcal{Z}$ *has* (r, ℓ)*-cooperative locality where*

$$r = r_1\lceil\frac{t}{2}\rceil\frac{s_{\min}(\mathcal{H})}{d_{\min}(C_1)} + r_2\lfloor\frac{t}{2}\rfloor\frac{s_{\min}(\mathcal{H})}{d_{\min}(C_2)}$$
$$s_{\min}(\mathcal{H}) - 1 \geq \ell \geq \left(d_{\min}(C_1)^{\lceil\frac{t}{2}\rceil} d_{\min}(C_2)^{\lfloor\frac{t}{2}\rfloor}\right)^{1/(t-1)} - 1.$$

Observe that $\mathcal{Z}$ has at least the (r, τ)-availability of its subcodes.

Sipser and Spielman showed that when the a code is defined using an *expander graph* (i.e. a graph in which small sets of vertices have large neighborhoods), it has improved minimum distance [30]. In [20], Bilu and Hoory define *ϵ-homogeneity*, an expansion-like property of hypergraphs. We extend the definition of ϵ-homogeneity to biregular hypergraphs and give an improved minimum stopping set bound for the corresponding codes.

Definition 21. Let $\mathcal{H} = (V_1, V_2, \ldots, V_t; E)$ be a t-uniform t-partite (Δ_1, Δ_2)-biregular hypergraph with n_1 vertices in each odd index part and n_2 vertices in each even index part. We say that $\mathcal{H}$ is *ϵ-homogeneous* if for every choice of $A_1, A_2, \ldots, A_t$, with $A_i \subseteq V_i$,

$$\frac{|E(A_1, A_2, \ldots A_t)|}{\Delta_1 n_1} \leq \prod_{i=1}^{t} \alpha_i + \epsilon\sqrt{\alpha_{\sigma(1)}\alpha_{\sigma(2)}},$$

where σ is a permutation on $[t]$ such that $\alpha_{\sigma(i)} \leq \alpha_{\sigma(i+1)}$ for each $i \in [t-1]$, and $|A_i| = \alpha_i n_1$ if i is odd and $|A_i| = \alpha_i n_2$ if i is even.

Theorem 10. *Let $\mathcal{H} = (V_1, \ldots, V_t; E)$ be a t-uniform t-partite (Δ_1, Δ_2)-regular ϵ-homogeneous hypergraph where there are n_1 (resp., n_2) vertices in each of the odd (resp., even) index parts. Let C_1 and C_2 be the subcodes of the odd and even index parts, respectively. Then $s_{\min}(\mathcal{H})$ is bounded below by*

$$\left(\frac{(n_1 d_{\min}(C_1))^{\lceil \frac{t}{2} \rceil}(n_2 d_{\min}(C_2))^{\lfloor \frac{t}{2} \rfloor}}{n_1\Delta_1}\left(1-\frac{\epsilon n_1\Delta_1}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}}\right)\right)^{\frac{1}{t-1}}.$$

For $\epsilon < \left(1-\frac{n_1\Delta_1}{n_1^{\lceil \frac{t}{2} \rceil} n_2^{\lfloor \frac{t}{2} \rfloor}}\right)\frac{\min_{i=1,2}\{n_i d_{\min}(C_i)\}}{n_1\Delta_1}$*, this improves the Theorem 9 bound.*

Proof. Let S be a minimum stopping set. By Theorem 9,

$$s_{\min}(\mathcal{H}) \geq \left(d_{\min}(C_1)^{\lceil \frac{t}{2} \rceil} d_{\min}(C_2)^{\lfloor \frac{t}{2} \rfloor}\right)^{1/(t-1)}.$$

Let $A_i \subseteq V_i$ be the set of vertices in V_i, for $i \in [t]$, contained in an edge in S. By ϵ-homogeneity,

$$s_{\min}(\mathcal{H}) \;=\; |S| \leq |E(A_1,\ldots,A_t)| \leq n_1\Delta_1\left(\prod_{i=1}^{t}\alpha_i + \epsilon\sqrt{\alpha_{\sigma(1)}\alpha_{\sigma(2)}}\right),$$

where $E(A_1,\ldots,A_t)$ is the set of edges which intersect all of the A_i's, $\alpha_i = \frac{|A_i|}{n_1}$ (resp., $\alpha_i = \frac{|A_i|}{n_2}$) if i is odd (resp., even), and σ is a permutation on $[t]$ such that $\alpha_{\sigma(1)} \leq \alpha_{\sigma(2)} \leq \cdots \leq \alpha_{\sigma(t)}$. Since $|A_i| \leq s_{\min}(\mathcal{H})/d_{\min}(C_1)$ (resp., $|A_i| \leq s_{\min}(\mathcal{H})/d_{\min}(C_2)$) if i is odd (resp., even), the above inequality simplifies to

$$\begin{aligned} s_{\min}(\mathcal{H}) &\leq n_1\Delta_1\left(\left(\frac{s_{\min}(\mathcal{H})}{n_1 d_{\min}(C_1)}\right)^{\lceil \frac{t}{2} \rceil}\left(\frac{s_{\min}(\mathcal{H})}{n_2 d_{\min}(C_2)}\right)^{\lfloor \frac{t}{2} \rfloor} + \epsilon\frac{s_{\min}(\mathcal{H})}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}}\right) \\ &= n_1\Delta_1\left(\frac{s_{\min}(\mathcal{H})^t}{(n_1 d_{\min}(C_1))^{\lceil \frac{t}{2} \rceil}(n_2 d_{\min}(C_2))^{\lfloor \frac{t}{2} \rfloor}} + \epsilon\frac{s_{\min}(\mathcal{H})}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}}\right) \end{aligned}$$

Therefore

$$\left(\frac{(n_1 d_{\min}(C_1))^{\lceil \frac{t}{2} \rceil}(n_2 d_{\min}(C_2))^{\lfloor \frac{t}{2} \rfloor}}{n_1\Delta_1}\left(1-\frac{\epsilon n_1\Delta_1}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}}\right)\right)^{\frac{1}{t-1}} \leq s_{\min}(\mathcal{H})$$

if

$$
\begin{aligned}
1 &< \frac{n_1^{\lceil \frac{t}{2} \rceil} n_2^{\lfloor \frac{t}{2} \rfloor}}{n_1 \Delta_1} \left(1 - \frac{\epsilon n_1 \Delta_1}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}} \right) \\
\frac{n_1 \Delta_1}{n_1^{\lceil \frac{t}{2} \rceil} n_2^{\lfloor \frac{t}{2} \rfloor}} &< 1 - \frac{\epsilon n_1 \Delta_1}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}} \\
\frac{\epsilon n_1 \Delta_1}{\min_{i=1,2}\{n_i d_{\min}(C_i)\}} &< 1 - \frac{n_1 \Delta_1}{n_1^{\lceil \frac{t}{2} \rceil} n_2^{\lfloor \frac{t}{2} \rfloor}} \\
\epsilon &< \left(1 - \frac{n_1 \Delta_1}{n_1^{\lceil \frac{t}{2} \rceil} n_2^{\lfloor \frac{t}{2} \rfloor}} \right) \frac{\min_{i=1,2}\{n_i d_{\min}(C_i)\}}{n_1 \Delta_1}.
\end{aligned}
$$

□

We now give bounds on the rate and minimum distance of a length $n_1\Delta_1$ code $\mathcal{Z}$ from an ϵ-homogeneous t-uniform t-partite (Δ_1, Δ_2)-regular hypergraph with n_1 (resp., n_2) vertices in each odd (resp., even) index part and $[\Delta_1, \Delta_1 R_1, \Delta_1 \delta_1]$ subcodes C_1 (resp., $[\Delta_2, \Delta_2 R_2, \Delta_2 \delta_2]$ subcodes C_2).

$$
\begin{aligned}
\text{rate}(\mathcal{Z}) &\geq R_1 \lceil \frac{t}{2} \rceil + R_2 \lfloor \frac{t}{2} \rfloor - (t-1) \\
d_{\min}(\mathcal{Z}) &\geq n_1 \Delta_1 \left((\delta_1^{\lceil \frac{t}{2} \rceil} \delta_2^{\lfloor \frac{t}{2} \rfloor})^{\frac{1}{t-1}} - c(\epsilon, \delta_1, \delta_2, t) \right)
\end{aligned}
$$

where $c(\epsilon, \delta_1, \delta_2, t) \to 0$ as $\epsilon \to 0$. The proofs follow those for the regular hypergraph bounds given in [20] and are included for completeness.

Proof. Let $\mathcal{Z}$ be a code on a (Δ_1, Δ_2)-biregular hypergraph with subcodes C_1 and C_2. Let R_1 (resp., R_2) be the rate and let δ_1 (resp., δ_2) be the relative distance of C_1 (resp., C_2). Since C_1 and C_2 are linear codes, codewords in C_1 (resp., C_2) must each satisfy $(1 - R_1)\Delta_1$ (resp., $(1 - R_2)\Delta_2$) linear equations.

A word in $\mathcal{Z}$ must then satisfy $(1-R_1)n_1\Delta_1\lceil\frac{t}{2}\rceil + (1-R_2)n_2\Delta_2\lfloor\frac{t}{2}\rfloor$ linear equations. Therefore

$$\begin{aligned}(1-\text{rate}(\mathcal{Z}))\Delta_1 n_1 \leq (1-R_1)n_1\Delta_1\lceil\frac{t}{2}\rceil + (1-R_2)n_2\Delta_2\lfloor\frac{t}{2}\rfloor \\ =(1-R_1)n_1\Delta_1\lceil\frac{t}{2}\rceil + (1-R_2)n_1\Delta_1\lfloor\frac{t}{2}\rfloor,\end{aligned}$$

and so

$$\text{rate}(\mathcal{Z}) \geq 1-(1-R_1)\lceil\frac{t}{2}\rceil - (1-R_2)\lfloor\frac{t}{2}\rfloor.$$

Let $\mathbf{x} \in \mathcal{Z}$ be a codeword of weight $d_{\min}(\mathcal{Z})$. Then $\mathbf{x}$ satisfies the constraints of each subcode. Therefore, either the weight of $\mathbf{x}$ is either 0 or $\geq \min_{i=1,2}\{d_{\min}(C_i)\}$. For each $i \in [t]$, let A_i be the set of vertices for which $\mathbf{x}$ is not $\mathbf{0}$ (i.e. the vertices that are in edges that were not assigned 0 in $\mathbf{x}$). Note that if $d_{\min}(C_1)|A_i| > d_{\min}(\mathcal{Z})$ for an odd i (resp., $d_{\min}(C_2)|A_i| > d_{\min}(\mathcal{Z})$ for an even i), then there are not enough nonzero symbols in $\mathbf{x}$ (i.e. edges assigned nonzero symbols) to satisfy the conditions of C_1 (resp., C_2). Therefore for odd (resp., even) i, we must have $|A_i| \leq \frac{d_{\min}(\mathcal{Z})}{d_{\min}(C_1)}$ (resp., $|A_i| \leq \frac{d_{\min}(\mathcal{Z})}{d_{\min}(C_2)}$). Each nonzero edge is incident to each A_i. Therefore $\{\text{nonzero edges}\} \subseteq E(A_1, A_2, \ldots, A_t)$, and by ϵ-homogeneity,

$$\begin{aligned}\frac{d_{\min}(\mathcal{Z})}{n_1\Delta_1} = \frac{|\{\text{nonzero edges}\}|}{n_1\Delta_1} &\leq \prod_{i=1}^{t}\alpha_i + \epsilon\sqrt{\alpha_{\sigma(1)}\alpha_{\sigma(2)}} \\ &\leq \left(\frac{d_{\min}(\mathcal{Z})}{d_{\min}(C_1)n_1}\right)^{\lceil\frac{t}{2}\rceil}\left(\frac{d_{\min}(\mathcal{Z})}{d_{\min}(C_2)n_2}\right)^{\lfloor\frac{t}{2}\rfloor} + \epsilon\frac{d_{\min}(\mathcal{Z})}{\min_{i=1,2}\{d_{\min}(C_i)n_i\}} \\ &= d_{\min}(\mathcal{Z})^t\left(\frac{1}{d_{\min}(C_1)n_1}\right)^{\lceil\frac{t}{2}\rceil}\left(\frac{1}{d_{\min}(C_2)n_2}\right)^{\lfloor\frac{t}{2}\rfloor} + \epsilon\frac{d_{\min}(\mathcal{Z})}{\min_{i=1,2}\{d_{\min}(C_i)n_i\}}.\end{aligned}$$

Thus

$$d_{\min}(\mathcal{Z}) \geq \left((d_{\min}(C_1)n_1)^{\lceil \frac{t}{2} \rceil} (d_{\min}(C_2)n_2)^{\lfloor \frac{t}{2} \rfloor} \left(\frac{1}{n_1\Delta_1} - \frac{\epsilon}{\min_{i=1,2}\{d_{\min}(C_i)n_i\}} \right) \right)^{\frac{1}{t-1}}.$$

□

The regular t-uniform t-partite hypergraphs constructed in [20] from regular expander graphs with second largest eigenvalue λ were shown to be $2(t-1)\lambda$-homogeneous. We conjecture that when Construction 1 starts with a (c, d)-regular bipartite expander graph, the resulting hypergraph will be ϵ-homogeneous, where ϵ depends on the second largest eigenvalue of the underlying expander graph.

Bibliography

[1] C. Mayer, K. Haymaker, and C. A. Kelley, "Channel decomposition for multilevel codes over multilevel and partial erasure channels," *Advances in Mathematics of Communications*, vol. 12, pp. 151–168, 2018. (document), 3, 1, 1, 2, 7

[2] C. Mayer and C. A. Kelley, "LT codes on partial erasure channels," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 804–808, June 2017. (document)

[3] A. Beemer, C. Mayer, and C. A. Kelley, "Erasure correction and locality of hypergraph codes," in *Coding Theory and Applications* (Á. I. Barbero, V. Skachek, and Ø. Ytrehus, eds.), (Cham), pp. 21–29, Springer International Publishing, 2017. (document), 7.1

[4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, pp. 4539–4551, Sept 2010. 1, 7

[5] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1243–1249, Sept 2009. 1, 7

[6] N. Silberstein and T. Etzion, "Optimal fractional repetition codes and fractional repetition batch codes," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2046–2050, June 2015. 1

[7] I. Tamo, M. Ye, and A. Barg, "Fractional decoding: Error correction from partial information," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 998–1002, June 2017. 1

[8] R. Cohen and Y. Cassuto, "Iterative decoding of LDPC codes over the q -ary partial erasure channel," *IEEE Transactions on Information Theory*, vol. 62, pp. 2658–2672, May 2016. 1, 2.4, 2.4.1, 2.4.1, 3, 3.1, 4, 5.4

[9] R. Cohen, N. Raviv, and Y. Cassuto, "LDPC codes over the q-ary multi-bit channel," *CoRR*, vol. abs/1706.09146, 2017. 1, 2.4, 2.4.2, 2.4.2, 3

[10] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, January 1962. 1, 2.3, 2.4.1

[11] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, pp. 533–547, Sep 1981. 1, 2.3, 2.3

[12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, pp. 1064–1070 vol.2, May 1993. 1, 2.3

[13] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, pp. 457–458, Mar 1997. 1

[14] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 619–637, Feb 2001. 1, 2.3, 5.4

[15] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423. 1, 2

[16] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Transactions on Information Theory*, vol. 23, pp. 371–377, May 1977. 1, 4.1, 4.2

[17] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 271–280, 2002. 1, 5, 5.1, 5.1, 5.1, 13, 5.1, 14, 15, 5.2, 5.2, 5.3, 5.4

[18] E. C. V. D. Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971. 1, 6

[19] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, pp. 572–584, September 1979. 1, 2.2.1, 6.1, 17, 6.1, 6.2

[20] Y. Bilu and S. Hoory, "On codes from hypergraphs," *European Journal of Combinatorics*, vol. 25, no. 3, pp. 339 – 354, 2004. 1, 7, 7.1, 7.1, 7.1

[21] A. Barg and G. Zemor, "Codes on hypergraphs," in *2008 IEEE International Symposium on Information Theory*, pp. 156–160, July 2008. 1, 7

[22] A. S. Rawat, A. Mazumdar, and S. Vishwanath, "Cooperative local repair in distributed storage," *CoRR*, vol. abs/1409.3900, 2014. 1, 7

[23] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Transactions on Information Theory*, vol. 62, pp. 4481–4493, Aug 2016. 1, 7

[24] R. W. Yeung, *Information theory and network coding*. Springer, 2011. 2.2.1, 2.2.1

[25] T. M. Cover and J. A. Thomas, *Elements of information theory*. Wiley-Interscience, 2006. 2.2.1

[26] T. Cover, "Broadcast channels," *IEEE Transactions on Information Theory*, vol. 18, pp. 2–14, Jan 1972. 2.2.1

[27] T. Cover and C. Leung, "An achievable rate region for the multiple-access channel with feedback," *IEEE Transactions on Information Theory*, vol. 27, pp. 292–298, May 1981. 2.2.1

[28] A. Carleial, "Interference channels," *IEEE Transactions on Information Theory*, vol. 24, pp. 60–70, January 1978. 2.2.1

[29] R. M. Roth, *Introduction to coding theory*. Cambridge Univ. Press, 2007. 2.2.2

[30] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inf. Theor.*, vol. 42, pp. 1710–1722, Sept. 2006. 2.3, 7.1

[31] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, pp. 3051–3073, July 2009. 2.3

[32] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information Theory*, vol. 47, pp. 585–598, Feb 2001. 2.3

[33] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in mlc nand flash memory: Measurement, characterization, and analysis," in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 521–526, March 2012. 2.4

[34] Z. Zhang, W. Xiao, N. Park, and D. J. Lilja, "Memory module-level testing and error behaviors for phase change memory," in *2012 IEEE 30th International Conference on Computer Design (ICCD)*, pp. 358–363, Sept 2012. 2.4

[35] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, Feb 2001. 2.4.1

[36] R. Gabrys, E. Yaakobi, L. Grupp, S. Swanson, and L. Dolecek, "Tackling intracell variability in tlc flash through tensor product codes," in *2012 IEEE International Symposium on Information Theory Proceedings*, pp. 1000–1004, July 2012. 3

[37] M. Moser and P. Chen, *A Student's Guide to Coding and Information Theory*. Cambridge University Press, 2012. 12, 3.2

[38] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," in *1998 Information Theory Workshop (Cat. No.98EX131)*, pp. 70–71, Jun 1998. 4

[39] G. D. Forney, "Concatenated codes," 1965. 4.1

[40] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: theoretical concepts and practical design rules," *IEEE Transactions on Information Theory*, vol. 45, pp. 1361–1391, Jul 1999. 4.2

[41] J. W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1528–1540, Oct 2002. 5

[42] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '98, (New York, NY, USA), pp. 56–67, ACM, 1998. 5

[43] D. MacKay, "Fountain codes," *IEE Proceedings - Communications*, vol. 152, pp. 1062–1068(6), December 2005. 5

[44] A. Shokrollahi, S. Lassen, and M. Luby, "Multi-stage code generator and decoder for communication systems," 2006. 5

[45] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, pp. 2551–2567, June 2006. 5, 5.4

[46] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Transactions on Information Theory*, vol. 52, pp. 2033–2051, May 2006. 5

[47] J. H. Sorensen, T. Koike-Akino, P. Orlik, J. Ostergaard, and P. Popovski, "Ripple design of LT codes for biawgn channels," *IEEE Transactions on Communications*, vol. 62, pp. 434–441, February 2014. 5, 5.1

[48] A. Khisti, "Tornado codes and Luby transform codes," 2003. 5.1

[49] J. H. S. rensen, P. Popovski, and J. Ostergaard, "Design and analysis of LT codes with decreasing ripple size," *IEEE Transactions on Communications*, vol. 60, pp. 3191–3197, November 2012. 5.1

[50] K. K. Yen, Y. C. Liao, and H. C. Chang, "Design of LT code degree distribution with profiled output ripple size," in *2015 IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 1–6, Oct 2015. 5.1

[51] Z. Zhiliang, L. Sha, Z. Jiawei, Z. Yuli, and Y. Hai, "Performance analysis of LT codes with different degree distribution," in *2012 Fifth International Workshop on Chaos-fractals Theories and Applications*, pp. 142–146, Oct 2012. 5.1

[52] H. Zhu, G. Zhang, and G. Li, "A novel degree distribution algorithm of LT codes," in *2008 11th IEEE International Conference on Communication Technology*, pp. 221–224, Nov 2008. 5.1

[53] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEU Int. J. Electron. Commun.*, vol. 54, pp. 389–398, Nov 2000. 5.4

[54] G. Joshi, J. Rhim, J. Sun, and D. Wang, "Fountain codes," 2010. 5.4

[55] E. G. A. A. and Y.-H. Kim, *Network information theory*. Cambridge University Press, 2014. 6, 6.1

[56] S. Girs, E. Uhlemann, and M. Björkman, "The effects of relay behavior and position in wireless industrial networks," in *2012 9th IEEE International Workshop on Factory Communication Systems*, pp. 183–190, May 2012. 6

[57] Y. Li, L. Liu, H. Li, J. Zhang, and Y. Yi, "Resource allocation for delay-sensitive traffic over lte-advanced relay networks," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 4291–4303, Aug 2015. 6

[58] K. B. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proceedings of the IEEE*, vol. 97, pp. 878–893, May 2009. 6

[59] R. U. Nabar, H. Bolcskei, and F. W. Kneubuhler, "Fading relay channels: performance limits and space-time signal design," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1099–1109, Aug 2004. 6

[60] R. Khalili and K. Salamatian, "On the achievability of cut-set bound for a class of erasure relay channels," in *International Workshop on Wireless Ad-Hoc Networks, 2004.*, pp. 275–280, May 2004. 16, 6.1, 6.1, 6.2, 6.3

[61] R. Khalili and K. Salamatian, "On the capacity of erasure relay channel: multi-relay case," in *IEEE Information Theory Workshop, 2005.*, pp. 5 pp.–, Aug 2005. 6.1, 6.1

[62] X. Wu and A. Özgür, "Cut-set bound is loose for Gaussian relay networks," in *2015 53th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1135–1142, Oct 2015. 6.1

[63] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, pp. 476–489, March 2011. 7

[64] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp. 1–10, May 2010. 7

[65] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Transactions on Information Theory*, vol. 58, pp. 6925–6934, Nov 2012. 7

[66] S. Chen, Y. Sun, U. C. Kozat, L. Huang, P. Sinha, G. Liang, X. Liu, and N. B. Shroff, "When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 1042–1050, April 2014. 7

[67] G. Joshi, Y. Liu, and E. Soljanin, "Coding for fast content download," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 326–333, Oct 2012. 7

[68] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2852–2856, June 2015. 7

[69] C. Kelley and D. Sridhara, "Eigenvalue bounds on the pseudocodeword weight of expander codes," vol. 1, 08 2007. 7.1

[70] N. Miladinovic and M. P. C. Fossorier, "Generalized LDPC codes and generalized stopping sets," *IEEE Transactions on Communications*, vol. 56, pp. 201–212, February 2008. 7.1