POLAR CODES FOR DISTRIBUTED SOURCE CODING

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Saygun Önay
December, 2014

Polar Codes for Distributed Source Coding
By Saygun Önay
December, 2014

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Prof. Dr. Erdal Arıkan(Advisor)

Prof. Dr. Tolga M. Duman

Asst. Prof. Dr. Ayşe Melda Yüksel Turgut

Prof. Dr. Orhan Arıkan

Prof. Dr. Uğur Güdükbay

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

POLAR CODES FOR DISTRIBUTED SOURCE CODING

Saygun Önay
Ph.D. in Electrical and Electronics Engineering
Advisor: Prof. Dr. Erdal Arıkan
December, 2014

Polar codes were invented by Arıkan as the first "capacity achieving" codes for binary-input discrete memoryless symmetric channels with low encoding and decoding complexity. The "polarization phenomenon", which is the underlying principle of polar codes, can be applied to different source and channel coding problems both in single-user and multi-user settings. In this work, polar coding methods for multi-user distributed source coding problems are investigated. First, a restricted version of lossless distributed source coding problem, which is also referred to as the Slepian-Wolf problem, is considered. The restriction is on the distribution of correlated sources. It is shown that if the sources are "binary symmetric" then single-user polar codes can be used to achieve full capacity region without time sharing. Then, a method for two-user polar coding is considered which is used to solve the Slepian-Wolf problem with arbitrary source distributions. This method is also extended to cover multiple-access channel problem which is the dual of Slepian-Wolf problem.

Next, two lossy source coding problems in distributed settings are investigated. The first problem is the distributed lossy source coding which is the lossy version of the Slepian-Wolf problem. Although the capacity region of this problem is not known in general, there is a good inner bound called the Berger-Tung inner bound. A polar coding method that can achieve the whole dominant face of the Berger-Tung region is devised. The second problem considered is the multiple description coding problem. The capacity region for this problem is also not known in general. El Gamal-Cover inner bound is the best known bound for this problem. A polar coding method that can achieve any point on the dominant face of El Gamal-Cover region is devised.

Keywords: Polar codes, distributed coding, source coding, Slepian-Wolf, channel coding, multiple-access channel, lossy source coding, multiple descriptions, multi-user polar codes, successive cancellation decoding, list decoding.

ÖZET

DAĞITIK KAYNAK KODLAMA İÇİN KUTUPSAL KODLAR

Saygun Önay
Elektrik ve Elektronik Mühendisliği, Doktora
Tez Danışmanı: Prof. Dr. Erdal Arıkan
Aralık, 2014

Kutupsal kodlar, ikili-girişli ayrık hafızasız kanalların kapasitesine ulaşan ve düşük kodlama ve kod çözme karmaşıklığına sahip ilk kodlar olarak Arıkan tarafından keşfedilmişlerdir. Kutupsal kodların temel prensibi olan "kutuplaşma hadisesi", hem tekli hem de çoklu kullanıcılı kurgularda farklı kanal ve kaynak problemlerine uygulanabilmektedir. Bu çalışmada, çoklu kullanıcılı dağıtık kaynak kodlama problemleri için kutupsal kodlama metodları incelenmektedir. İlk olarak, aynı zamanda Slepian-Wolf problemi olarak da bilinen dağıtık kaynak kodlama probleminin kısıtlı bir hali ele alınmaktadır. Bu kısıt, ilişkili kaynakların dağılımı üzerinedir. Kaynakların "ikili simetrik" olduğu durumda tek kullanıcılı kutupsal kodlar kullanılarak tüm kapasite alanına zaman paylaşımsız erişilebildiği gösterilmektedir. Daha sonra, genel kaynak dağılımlarına sahip Slepian-Wolf problemini çözmek için kullanılan ikili kullanıcılı kutupsal kodlama metodu ele alınmaktadır. Ayrıca bu metod, Slepian-Wolf probleminin eşleniği olan çoklu erişimli kanal problemini de içerecek şekilde genişletilmektedir.

Daha sonra, dağıtık kurgulara sahip iki farklı kayıplı kaynak kodlama problemi incelenmektedir. Ele alınan ilk problem, Slepian-Wolf probleminin kayıplı hali olan kayıplı dağıtık kaynak kodlama problemidir. Bu problemin kapasite bölgesi genel olarak bilinmese de Berger-Tung iç sınırı olarak bilinen iyi bir iç sınır bulunmaktadır. Berger-Tung bölgesinin tümüne erişen bir kutupsal kodlama yöntemi tasarlanmaktadır. Ele alınan ikinci problem, çoklu tanım kodlaması problemidir. Bu problemin de genel kapasite alanı bilinmemektedir. El Gamal-Cover iç sınırı, bu problem için bilinen en iyi sınırdır. El Gamal-Cover kapasite sınırındaki herhangi bir noktaya erişebilen bir kutupsal kodlama yöntemi geliştirilmektedir.

Anahtar sözcükler: Kutupsal kodlar, dağıtık kodlama, kaynak kodlama, Slepian-Wolf, kanal kodlama, çoklu erişimli kanal, kayıplı kaynak kodlama, çoklu tanımlar, çoklu kullanıcılı kutupsal kodlar, sıralı elemeli kod çözme, liste kod çözme.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

C. E. Shannon laid the foundations of information theory in his 1948 seminal paper [1], where he presented the source coding and noisy channel coding theorems. He defined a mathematical framework in which these problems can be studied systematically. He gave the fundamental limits of both source coding and channel coding. He also proved that source and channel coding problems may be treated separately without any loss in system performance.

The basic model introduced by Shannon consists of a source, source encoding/decoding functions, channel encoding/decoding functions and a channel. The aim is to reconstruct the source at the sink perfectly with as few as possible channel transmissions. Source encoding makes it possible to most efficiently describe the source by throwing away the redundancy in it. Then, if the transmission medium is "noisy", transmitter adds redundancy to combat its effects and receiver decodes the received vectors into message symbols. If the message symbols are detected correctly, the source decoder decodes them into message vectors.

The source is modeled with a random process $\{Z_n\}$. Shannon showed that a source encoder can compress a source at most down to its entropy $H(Z)$ without a loss in fidelity. Thus, there exists an encoder that may represent a source with R bits per symbol if and only if $R > H(Z)$. After the source compression,

Figure 1.1: Basic communication setting.

the message needs to be transmitted over a noisy channel. The input and output symbols to channel are $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, respectively. The channel is denoted as $W : \mathcal{X} \to \mathcal{Y}$ and modeled with a conditional probability $P_{Y|X}$ which is called the channel transition probability. The channel encoder's job is to add redundancy to the source vector in such a way that it is recoverable at the channel decoder with *arbitrarily* high probability while keeping the the number of channel uses as few as possible. Shannon showed that these seemingly conflicting requirements may be satisfied as long as the size of transmission blocks (K user bits or N channel uses) is large and rate of transmission (K/N) is below the *channel capacity* which is a fundamental property of a given channel. Then, the receiver estimates the sent message from the channel output and source decoder reconstructs the source. It was also proven that the probability of decoding error goes to zero exponentially with the block length N. Thus long block lengths are needed to make the error probability low which means that encoding and decoding complexities need to be low for practical codes.

Shannon's proof depended on the *random coding* argument which considered the average performance of ensembles of codes. Thus he proved the existence of capacity-achieving codes without explicitly constructing any such code. His concept of information and viewpoint of a communication system changed the telecommunications field forever. The existence of information theory and coding theory fields is attributed to him.

Since then, construction of capacity-achieving codes with low encoding and decoding complexities has been the focus of coding theory. For sixty years there has been an enormous amount of research in the area. Many different coding methods have been developed which may be broadly categorized under two titles: algebraic coding and probabilistic coding. Hamming [2], Golay [3], Reed-Muller [4] [5], BCH [6] and Reed-Solomon [7] codes may be counted as the most famous examples of algebraic codes developed. Convolutional codes [8], LDPC codes [9]

and Turbo codes [10] are the most important examles of probabilistic codes. In the last two decades, with the discovery of Turbo Codes and "re-discovery" of LDPC codes, it has been possible to practically come very close to Shannon's capacity bounds in certain situations. However, because of the way those codes are constructed and decoded, it is not possible, except in very special cases, to theoretically prove that they can achieve the capacity bounds asymptotically. Interested reader is referred to the excellent survey by Costello and Forney [11].

Polar coding method [12], introduced by Arıkan, is the first provably capacity-achieving coding method with low encoding and decoding complexity for the class of binary input discrete memoryless channels (B-DMC). In addition to being used for constructing capacity achieving channel codes, the polarization concept is a promising new theoretical advancement that may find applications in other areas of information theory.

1.1 Polar Codes

Polar codes were introduced in [12] as the first provably capacity achieving channel codes for symmetric binary-input discrete memoryless channels (B-DMC) with low encoding and decoding complexity. Later, the polarization concept extended to non-binary alphabets, source coding scenarios and distributed settings. The time complexity of both encoder and decoder is $O(N \log N)$, where N is the block length. The primitive ideas for polar coding first appeared in Arıkan's earlier paper [13] on channel cutoff rate improvement.

The idea of polar coding can be summarized as generating N *extremal* channels from N independent uses of the same *base* channel $W : \mathcal{X} \to \mathcal{Y}$. By extremal we mean that the channels are either perfect or completely noisy. This is achieved by applying a transformation to the input of N independent copies of channel W and employing *successive cancellation* (SC) decoding in a special order at the receiver. The successive cancellation decoder at step i not only observes the base channel outputs but also the $i-1$ previously decoded bits. These *coordinate channels*

experienced by successive cancellation decoding are either worse or better than the original channel W. The interesting result proved by Arıkan is that these channels *polarize* to either a perfect or a completely useless channel with the ratio of perfect channels to block length N approaching to the capacity of the original channel W as $N \to \infty$. Then, how to use this method to construct a capacity achieving channel code becomes obvious: just send information from those inputs corresponding to perfect channels and fix the other ones and reveal to both encoder and decoder.

Polarization transform is a linear operation identified with an $N \times N$ matrix G_N, where $N = 2^n$. The construction has a recursive structure and starts with size-2 base matrix $G_2 = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. A higher order polar transform is obtained as $G_N = G_2^{\otimes n}$, where "$\otimes n$" denotes the n-th Kronecker power. Polar channel coding method summarized above dictates sending information from some of the input bits and fixing others to known values. This corresponds to selecting those rows of G_N corresponding to information bit indexes to compose the generator matrix of the code. Polar codes share a lot of structure with Reed-Muller (RM) codes which was surveyed by Arıkan in a later paper [14]. The generator matrix of RM codes is also selected from the rows of G_N. The difference is in the selection rule. The selection rule for RM codes maximizes the minimum distance whereas the selection rule for polar codes is dependent on the underlying channel and minimizes the decoding error under successive cancellation decoding.

Shortly after polar codes were introduced a number of work has been published on its performance and extension of its areas of applicability. In [15] authors improved the bound on the rate of polarization to $O(2^{-N^\beta})$ for $\beta < 1/2$, from $O(N^{-\frac{1}{4}})$ bound proved in [12]. Korada made most of the early contribution to the applicability of polar codes in his thesis [16]. He tackled lossless and lossy source coding problems. It was also shown in that thesis that polar codes could be used for some multi-terminal scenarios like Wyner-Ziv coding, Slepian-Wolf coding, degraded broadcast channel and multiple-access channel. However, all of those problems were considered under constrained assumptions like *symmetric* distributions and accessing corner points of capacity regions. Later, other researchers considered the same problems in their more generality.

Another major contribution to the theory of polarization from a single researcher came from Şaşoğlu. Şaşoğlu et al. [17] extended polar codes to non-binary alphabets. Polar code for multiple-access channels were first considered by Şaşoğlu et al [18]. However, their *joint* polarization approach was not able to reach any point on the dominant face of the capacity region. Şaşoğlu [19] proved an entropy inequality which made polarization proofs much simpler and direct. He made use of that result to prove in [20] that Arıkan's recursive transform also polarizes ergodic Markov processes of finite order. He assembled all of these results in his thesis [21].

Arıkan [22] reassessed source coding with polar codes from a direct source polarization approach. Systematic polar codes were introduced by Arıkan [23]. Polar code construction was considered in [24] and low complexity approximations were suggested. In [25], low complexity and efficient list decoding algorithms for SC decoding were proposed. In addition, authors proposed augmenting polar codes with a CRC to be used in list decoding. This approach increased the performance considerably and generated a polar coding scheme with the best known performance to date. Polar codes for broadcast channels were considered in [26]. Polar coding for asymmetric distributions without alphabet extension were considered in [27]. A new polar coding method for multi-terminal settings were introduced in Slepian-Wolf coding context by Arıkan [28]. The so called *monotone chain rule* approach could reach any point on the dominant face of Slepian-Wolf achievable rate region. In [29], authors applied this method to multiple-access channel, built list decoders based on [25] and presented simulation results. Multiple description problem using polar codes was considered in [30] and [31]. Early hardware architectures for successive cancellation decoding of polar codes were presented in [32] and [33]. Recently, hardware architectures for successive cancellation *list* decoders has also emerged [34], [35], [36], [37].

1.2 Contribution of this Thesis

This thesis mostly concentrates on polar coding methods for distributed source coding settings. We first present a technique in which single-user polar codes may be efficiently used to achieve any point on the dominant face of the achievable region of Slepian-Wolf coding of sources with *special* distributions. Then, we review in detail the monotone chain rule technique introduced by Arıkan for *general* Slepian-Wolf problem. We give detailed proofs of the method as well as explicit formulas and algorithms for decoder construction. We also include results on multiple-access channel (MAC) which is considered as the *dual* of Slepian-Wolf (SW) problem. Then we turn our attention to *lossy* source coding schemes. We show that the known bounds for distributed lossy source coding and multiple-description coding can be achieved with polar coding methods.

In the seminal paper by Slepian and Wolf [38], bounds on compression rate pairs of the noiseless coding of two correlated information sources were proved. The two correlated information sources (X, Y) are obtained by repeated independent drawings from a discrete bivariate distribution $P_{XY}(x, y)$ where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. The setting comprises of two *separate* encoders for X and Y sources and a joint decoder. The encoders compress the sources and the decoder's job is to reconstruct sources perfectly. This particular setting is the basic distributed lossless source coding setup and has since been synonymously referred to as the Slepian-Wolf coding problem. The details of the problem are presented in Section 2.1. With the discovery of capacity achieving channel codes and using the known dualities of source coding and channel coding there has been extensive amount of work in applying channel codes in distributed source coding setup of which we give a brief survey in Section 2.1. Most of the works mentioned in that section assume binary symmetric sources (BSS). That is, the correlated sources X and Y have uniform marginals. It is a restricted version of SW problem and also what we assume in Chapter 3. Furthermore, some of the works surveyed in Section 2.1 solve *asymmetric* SW coding problem, i.e. the corner points of the SW region are targeted. The common argument is that the other points on the dominant face may achieved with *time-sharing* [39]. However, in practice direct

achievement of a rate-point without time-sharing is more desirable. In a previous work by Korada [16] polar codes for BSS and asymmetric setting was considered. In Chapter 3, we show polar coding can be used to achieve any point on the SW region for BSS sources without time-sharing.

In Chapter 4, we show how the *general* Slepian-Wolf problem may be solved using polar codes. By "general" we mean that any discrete source distribution is allowed and any point on the dominant face of the rate region may be targeted directly. The method was introduced by Arıkan in [28]. We present an extended treatment of the method which consists of two sources with prime-sized alphabets and a side-information with an arbitrary alphabet. This treatment forms the basis of our other methods in later sections for distributed settings. The method describes a two-user joint successive cancellation (SC) decoder. This decoder is obviously more complicated than its normal single-user counterpart which is used in "special" SW problem investigated in Chapter 3. But in exchange of this increase in complexity it becomes possible to solve general SW problem. In Section 4.2, we give explicit formulas and algorithms for implementing joint decoder. We show how joint SC *list* (SCL) decoder may be implemented as an extension to single-user list decoder that was introduced in [40]. We also present experimental results on the performance of our joint SCL decoder. Then, we move on to multiple-access channel problem which is considered as the dual problem. We show that polar coding may be used to achieve the whole capacity region of MAC and not only the *symmetric* capacity region. Then we present experimental results on the performance.

In Chapter 5, we consider lossy source coding problems in distributed settings. The first problem we consider in Section 5.1 is the distributed lossy source coding which is the lossy version of SW problem. The setting is the same as the SW problem. The only difference is that the source reconstructions are subject to distortion constraints. Although the capacity region of this problem is not known in general, there is a good inner bound called the Berger-Tung (BT) inner bound. We devise the polar coding method for that problem and show that it can achieve the whole dominant face of the BT region. Then we present simulation results. The second problem we consider in Chapter 5 is the multiple description coding

(MDC) problem. There is single source in MDC setting. Two different representations of the source is generated by two encoders. There are three different decoders in the setting. Decoder 1, 2 and 0 has access to representation 1, 2 and both, respectively. Each reconstruction has a different distortion constraint. The capacity region for this problem is also not known in general. However, there is a good inner bound called the El Gamal-Cover (EGC) inner bound. We construct the polar coding method for this problem and prove that it can achieve any point on the dominant face of EGC region.

1.3 Notation

In this work we follow the notation of [12] and [22]. Random variables are denoted by upper case letters like X and its realization is denoted by lower case letter x. x_1^N or x^N denotes a row-vector $(x_1, \ldots, x_N)$ of length N. x_i^j denotes the subvector $(x_i, \ldots, x_j)$ of x^N when $i \leq j$. If $i > j$, then x_i^j is a null vector. We use $x_{i,o}^j$ and $x_{i,e}^j$ to denote subvectors consisting of only odd and even indices, respectively. Alternatively, lower-case bold characters ($\mathbf{x}$) also denote row vectors. Matrices are denoted with upper-case characters such as G. We use $[N]$ to denote set $\{1, 2, \ldots, N\}$. For any set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality. Let $\mathcal{A} \subseteq [N]$ be an index set, then $x_{\mathcal{A}}$ denotes the row-vector formed by those elements of x^N with indices in set $\mathcal{A}$ in ascending order of indices, i.e. $x_{\mathcal{A}}$ denotes $x_{i_1}, \ldots, x_{i_{|\mathcal{A}|}}$ where $\{i_k \in \mathcal{A} : i_k < i_{k+1}\}$. Similarly, $(G)_{\mathcal{A}}$ denotes the sub-matrix formed by those rows of G with indices in set $\mathcal{A}$ in ascending order of indices.

We also use $P_e(X|Y)$ to denote the average error probability in optimally decoding $X \in \mathcal{X}$ given $Y \in \mathcal{Y}$. That is,

$$P_e(X|Y) \triangleq \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{XY}(x, y) \cdot \mathbb{1}_{\left\{x' \in \mathcal{X} : P_{X|Y}(x'|y) \geq P_{X|Y}(x|y)\right\}}.$$

Chapter 2

Background

In this chapter we give some background information supplementing the main topics of this thesis. In the first section we present lossless distributed source coding problem also known as Slepian-Wolf problem [38]. We give a literature survey on the practical implementation methods using channel codes. In the second section, we present a review of polarization and polar codes for both channel and source coding problems.

2.1 Distributed Source Coding

The well-known paper by D. Slepian and J. Wolf [38] generalizes certain well known results on the noiseless coding of a single discrete information source to the case of two correlated information sources. The two correlated information sources (X, Y) are obtained by repeated independent drawings from a discrete bivariate distribution $P_{XY}(x, y)$ where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. The paper analyses all possible cases depending upon the information available to encoders and decoders. But by far the most interesting case presents itself when the encoder of each source is constrained to operate without knowledge of the other source, while the decoder has available both encoded message streams as in Figure 2.1. This particular

setting has since been known as the Slepian-Wolf (SW) coding problem and used interchangeably with distributed source coding problem, although there are other source coding problems in distributed settings like distributed lossy source coding or multiple-description problem which we will describe in later chapters.

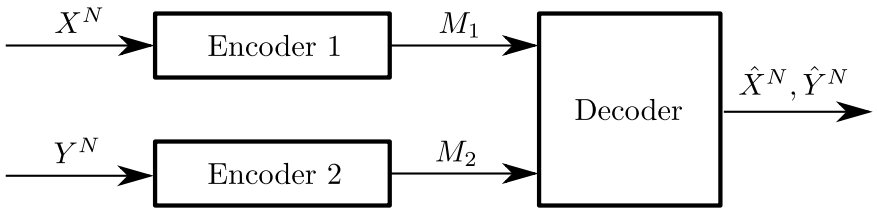


Figure 2.1: Correlated coding of two sources.

It is well known from the results of source coding that the rate of a source must be greater than its entropy, $R_1 \geq H(X)$. The same result generalizes to joint coding of correlated random variables X and Y easily: the jointly encoded data rate must be greater than the joint entropy, namely $R > H(X,Y)$. This is because a pair of random variables X, Y can be regarded as a single random variable Z taking $|\mathcal{X}| \cdot |\mathcal{Y}|$ values. The entropy of this variable is $H(X,Y)$. The interesting case occurs when sources are encoded separately and decoded jointly. One might expect that the lower bound $H(X,Y)$ may not be reached due to the fact that encoders not sharing information. However, the result of SW paper proves that there is no asymptotic loss in performance due to separate encoding, i.e. rate lower bound $H(X,Y)$ is still achievable. This is the central and the most surprising result presented in the paper.

Definition 1. *A* $((2^{NR_1}, 2^{NR_2}), N)$ distributed source code *for the joint source* (X,Y) *consists of two sets of integers* $\mathcal{M}_1 = \{1, 2, \ldots, 2^{NR_1}\}$ *and* $\mathcal{M}_2 = \{1, 2, \ldots, 2^{NR_2}\}$*,two* encoding functions*,*

$$f_1 : \quad \mathcal{X}^N \to \mathcal{M}_1 \tag{2.1}$$

and

$$f_2 : \quad \mathcal{Y}^N \to \mathcal{M}_2, \tag{2.2}$$

and a decoding function*,*

$$g : \quad \mathcal{M}_1 \times \mathcal{M}_2 \to \mathcal{X}^N \times \mathcal{Y}^N. \tag{2.3}$$

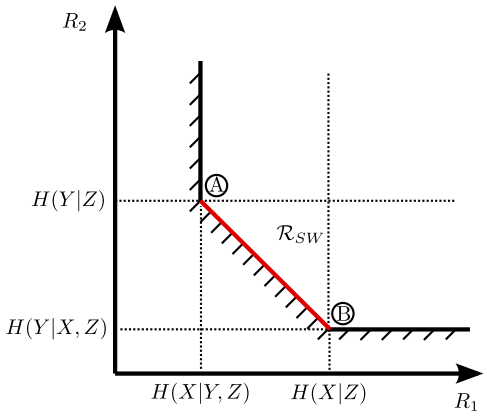


Figure 2.2: Admissible rate region.

Here $M_1 = f_1(X^N)$ is the index corresponding to X^N, $M_2 = f_2(Y^N)$ is the index corresponding to Y^N and (R_1, R_2) is the rate pair of the code.

Definition 2. *The* probability of error *for a distributed source code is defined as*

$$P_e^{(N)} = \Pr\{g(f_1(X^N), f_2(Y^N)) \neq (X^N, Y^N)\}. \tag{2.4}$$

Definition 3. *A rate pair* (R_1, R_2) *is said to be* achievable *for a distributed source if there exists a sequence of* $((2^{NR_1}, 2^{NR_2}), N)$ *distributed source codes with* $P_e^{(N)} \to 0$. *The* achievable rate region *is the closure of the set of achievable rates.*

Theorem 1 (Slepian-Wolf)**.** *For the distributed source coding problem for the source* (X, Y)*, the achievable rate region is given by*

$$\begin{aligned} R_1 &\geq H(X|Y), \\ R_2 &\geq H(Y|X), \\ R_1 + R_2 &\geq H(X, Y). \end{aligned} \tag{2.5}$$

The result of Slepian-Wolf paper can be presented as a two dimensional rate region $\mathcal{R}_{SW}$ for the two encoded message streams as shown in Figure 2.2. It is seen that both R_1 can go below $H(X)$ and R_2 can go below $H(Y)$, while their total $R_1 + R_2$ must stay above $H(X, Y)$. The line segment between points A and B in Figure 2.2 is referred to as the *dominant face* of the SW rate region. It is enough for a coding scheme to show that it can reach one of the corner points

A or B on this graph, namely $R_1 = H(X)$ and $R_2 = H(Y|X)$ or $R_2 = H(Y)$ and $R_1 = H(X|Y)$. At these points, one source (say Y) is compressed at its entropy rate and can therefore be reconstructed at the decoder independently of the information received from the other source X. The source Y is called the *side information* (SI) (available at the decoder only). X is compressed at a smaller rate than its entropy. More precisely, X is compressed at the conditional entropy $H(X|Y)$ and can therefore be reconstructed only if Y is available at the decoder. The sources X and Y play different roles in this scheme, and therefore the scheme is usually referred to as *asymmetric* SW coding.

2.1.1 Constructive Approaches to Slepian-Wolf Coding

The proof of the SW theorem depends on random coding argument and is non-constructive. In 1974, Wyner [39] suggested using a binary linear channel code for construction of SW codes and showed the optimality of this construction. He proved that if a linear block code achieves the capacity of the BSC that models the correlation between the two sources, then this capacity achieving code can be turned into SW bound achieving source code. His method is called the *syndrome approach*. The method assumes asymmetric setting, i.e. SI Y is available at the decoder and the problem is reduced to compressing X to $H(X|Y)$ at the encoder.

In syndrome approach a binary (N, K) code $\mathcal{C}$ is constructed with size $(N - K, N)$ parity check matrix H. The well-known properties of such a code and syndrome decoding are summarized in the following. The code contains all N-vectors $\mathbf{x}$ such that $\mathbf{xH^T} = \mathbf{0}$. The code partitions the space of N-vectors (2^N vectors) into $2^{(N-K)}$ *cosets* of 2^K words. Each coset is indexed by the $(N - K)$-vector syndrome $\mathbf{s}$. All sequences in a coset share the same syndrome $\mathcal{C}_s = \{\mathbf{x} : \mathbf{s} = \mathbf{xH^T}\}$. In addition, because of the linearity of the code, a coset results from the translation of the code by any representative of the coset: $\forall \mathbf{v} \in \mathcal{C}_\mathbf{s}$, $\mathcal{C}_s = \mathcal{C} \oplus \mathbf{v}$. The minimum Hamming weight representative of the coset is called the coset leader. It is used for maximum-likelihood decoding of the code, which

is formulated as follows:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{C}}{\operatorname{argmin}} \quad \mathrm{d}_H(\mathbf{x}, \mathbf{y}),$$

where $\mathrm{d}_H(\cdot,\cdot)$ is the Hamming distance function. This decoding procedure is called the syndrome decoding. The decoder first calculates the syndrome $\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathbf{T}}$ of the received word $\mathbf{y}$. Since $\mathbf{x} \in \mathcal{C}$ the syndrome of $\mathbf{y}$ equals to the syndrome of error $\mathbf{e}$ where $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}$: $\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathbf{T}} = (\mathbf{x} \oplus \mathbf{e})\mathbf{H}^{\mathbf{T}} = \mathbf{e}\mathbf{H}^{\mathbf{T}}$. The function $f(\mathbf{s})$ computes the coset leader for syndrome $\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathbf{T}}$. This coset leader is the ML estimate of the error pattern $\mathbf{e}$. Then the ML estimate of $\mathbf{x}$ is given by $\hat{\mathbf{x}} = \mathbf{y} \oplus f(\mathbf{y}\mathbf{H}^{\mathbf{T}})$.

Such a code is used in the asymmetric SW problem as follows. The encoding operation is defined as sending only the syndrome corresponding to input n-vector $\mathbf{x}$: $\mathbf{s} = \mathbf{x}\mathbf{H}^{\mathbf{T}}$. The N-vector $\mathbf{x}$ is mapped into its corresponding $(N-K)$-vector syndrome $\mathbf{s}$. Therefore, a compression ratio of N:$(N-K)$ is achieved. The decoder, given the correlation between sources X and Y, received coset index $\mathbf{s}$ and the SI $\mathbf{y}$, searches for the sequence that is closest to $\mathbf{y}$ in $\mathbf{s}$-coset $\mathcal{C}_s$:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{C}_\mathbf{s}}{\operatorname{argmin}} \quad \mathrm{d}_H(\mathbf{x}, \mathbf{y}).$$

It is important to note that the minimization may be performed over a coset whose coset leader is not the all-zero vector (syndrome is not zero). Therefore, the classical ML channel decoder has to be adapted in order to be able to enumerate all vectors in a given coset $\mathcal{C}_s$. Because of the linearity of the code, by adding the syndrome of $\mathbf{y}$ onto $\mathbf{s}$ we get $\mathbf{s} \oplus \mathbf{y}\mathbf{H}^{\mathbf{T}} = (\mathbf{x} \oplus \mathbf{y})\mathbf{H}^{\mathbf{T}} = \mathbf{e}\mathbf{H}^{\mathbf{T}}$. Therefore, the ML estimate of the error pattern $\mathbf{e}$ in this case is $f(\mathbf{s} \oplus \mathbf{y}\mathbf{H}^{\mathbf{T}})$

Another way to look at the syndrome decoding principle is as follows. As mentioned above, because of the linearity of the code, a coset $\mathcal{C}_s$ can be formed from the translation of code $\mathcal{C}$ by any representative of the coset. For different representatives only the order of words are shuffled. Therefore, we can get a codeword $\mathbf{x}' \in \mathcal{C}$ from $\mathbf{x} \in \mathcal{C}_\mathbf{s}$ by adding any representative $\mathbf{a}$ of the coset $\mathcal{C}_s$: $\mathbf{x}' = \mathbf{x} \oplus \mathbf{a}$. Since, $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}$, by adding a representative $\mathbf{a}$ to both sides we

get $\mathbf{y} \oplus \mathbf{a} = \mathbf{x} \oplus \mathbf{a} \oplus \mathbf{e}$. Setting $\mathbf{y}' = \mathbf{y} \oplus \mathbf{a}$ and $\mathbf{x}' = \mathbf{x} \oplus \mathbf{a}$, we get $\mathbf{y}' = \mathbf{x}' \oplus \mathbf{e}$. Hence, we can do the decoding on $\mathcal{C}$ instead of $\mathcal{C}_s$, which is the normal channel decoding operation, to find an estimate of $\hat{\mathbf{x}}'$ of $\mathbf{x}'$. And, by adding $\mathbf{a}$ onto estimate $\hat{\mathbf{x}}'$ we get the estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$.

Another approach on constructing SW codes from capacity-achieving linear codes is called the *parity approach*. Although the syndrome approach is optimal, it may be difficult to construct rate-adaptive codes by puncturing the syndrome. The parity approach is originally proposed to get rate adjustable codes easily via puncturing. In parity approach, parity bits are sent instead of syndrome.

Let $\tilde{\mathcal{C}}$ be an $(N, 2N-K)$ systematic binary linear code, defined by its $N \times 2N-K$ generator matrix $\mathrm{G} = [\ \mathrm{I}\ \mathrm{P}\ ] : \tilde{\mathcal{C}} = \{[\ \mathbf{x}\ \mathbf{x_p}\ ] = \mathbf{x}\mathrm{G}\}$. The compression is achieved by only sending the parity bits $\mathbf{x_p}$. The systematic bits $\mathbf{x}$ are not transmitted. This gives a compression ratio of $N : (N-K)$. The correlation between the source X and SI Y is modeled as a virtual channel in this approach, too. The pair $[\ \mathbf{y}\ \mathbf{x_p}\ ]$ is regarded as the noisy version of $[\ \mathbf{x}\ \mathbf{x_p}\ ]$. Therefore, the total channel is a parallel combination of a BSC and a perfect channel. The decoder corrects the *virtual* channel noise and estimates $\mathbf{x}$ given the parity bits $\mathbf{x_p}$ and the SI $\mathbf{y}$ which is regarded as the noisy version of the original sequence $\mathbf{x}$. Therefore, the usual ML decoder must be adapted to take into account that some bits (parity bits) of the received sequence are perfectly known.

2.1.2 Practical Slepian-Wolf Codes Based on Channel Codes

Development of good channel codes spurred interest in using them in constructing practical SW codes which started with the work of Pradhan *et al.* in [41]. Then, a number of researchers developed different methods which we briefly survey in this section. Practical Slepian-Wolf (SW) coding schemes can be divided into two main categories: asymmetric and nonasymmetric. Asymmetric SW coding refers to the case where one source, for example Y, is transmitted at its entropy

rate $H(Y)$ and is used as side information (SI) to decode the second source X, which compressed at rate $H(X|Y)$. Nonasymmetric SW coding refers to the case where both sources are compressed at a rate lower than their respective entropy rates. Both syndrome and parity approaches are used to construct asymmetric and nonasymmetric schemes. In both of the approaches, Y is treated as a noisy version of X, i.e. the correlation between source X and SI Y is modeled as a "virtual" channel. If a linear block code achieves the capacity of the binary symmetric channel that models the correlation between the two sources, then this capacity-achieving channel code can be turned into a SW-achieving source code. Both the LDPC and Turbo codes are used in this way to construct practical SW codes [42].

2.1.2.1 Asymmetric SW Coding

The first practical approach to syndrome decoding appeared in a scheme called DISCUS [41]. For convolutional codes, Viterbi decoding on a modified trellis is proposed. The method takes advantage of the linearity of the code. For systematic convolutional codes a representative of the coset is the concatenation of K-length all zero vector and $(N-K)$-vector syndrome $\mathbf{s}$: [$\mathbf{0}$ $\mathbf{s}$]. This representative is then added to all the codewords labelling the edges of the trellis. And Viterbi decoding is done on this modified trellis. The novelty in this paper is to apply the syndrome principle to modify the normal trellis decoder. This is accomplished by using a systematic code.

Another approach to syndrome decoding of convolutional codes is proposed in [43]. In this approach the translation by a coset representative is performed outside the decoder. The encoder calculates the syndrome which is referred as syndrome forming (SF). In the decoder, first, a representative is computed from the received syndrome $\mathbf{s}$ (this step is called inverse syndrome forming - ISF) and added to SI $\mathbf{y}$. Since there are many representatives, ISF operation is not unique. However, it is particularly easy to perform ISF operation when the code is systematic. This is the same as in DISCUS [41] method, where the representative used is [$\mathbf{0}$ $\mathbf{s}$]. Therefore, systematic codes are assumed in this paper. Then the

decoding is performed in usual manner using the original trellis. As a last step the representative is added onto the output of the decoder to get the estimate of the original stream $\mathbf{x}$. The advantage of this method is that it does not modify the decoder. This method is also applied to Turbo codes. They show a method to get the ISF of a parallel or serially concatenated Turbo code from ISFs of its constituent codes.

In [44] a SW scheme based on convolutional and turbo codes that can be used for any code (not only systematic) is proposed. This scheme also uses the syndrome approach. In this scheme the decoder is based on a syndrome trellis rather than the usual trellis based on the generator matrix of the code. The concept of syndrome trellis was first introduced for binary linear block codes [45] and then extended to convolutional codes [46]. In this scheme again the syndrome trellis is modified by the received syndrome $\mathbf{s}$. The syndrome trellis construction in this paper is new and different than the one in [46]. The states of the trellis are marked differently than the conventional way in [46] which is the partial syndrome value at that particular stage. The method in [44] gives a simpler construction of the trellis in the sense that there is no need to expand the parity check polynomial matrix into a matrix of an equivalent block code of large dimension. Each stage k of the trellis is one of the two possible trellises corresponding to $s_k = 0$ and $s_k = 1$. One of the advantages of this construction is that it is possible to perform optimal decoding even if the syndrome is punctured. The trellis stage corresponding to punctured syndrome bit consists of the union of the two possible trellis stages for $s_k = 0$ and $s_k = 1$. This way both possibilities are taken into account optimally and the complexity grows only linearly with the punctured positions.

For LDPC codes, belief propagation decoder can be modified to take into account the syndrome [47]. Here, the syndrome bits are added to the graph such that each bit is connected to the parity check equation to which it is related. This modification to the LDPC decoder is very natural and minimal. Only the update rule at the check node is modified to take into account the value of the syndrome bit.

The parity approach is also used in constructing SW codes using turbo codes [48][49]. In [48] a conventional turbo encoder/decoder pair is used. The systematic output of the encoders are not used and the parity outputs are punctured to get the desired rates. The encoders considered in the paper are rate $(N-1)/N$, but actually method is applicable to any encoder rate. The method described here is the direct application of parity approach to turbo decoding. In a conventional turbo encoder used for DSC, convolutional encoders are used and their rate K/N is less than 1, i.e. $K < N$. The required rate for source encoding is achieved by heavily puncturing the encoder outputs as in [48]. In [49], authors take an alternative way to construct constituent encoders. They use two identical finite state machine (FSM) encoders. These encoders are custom designed using Latin squares. Their rate is greater than 1, i.e. $K > N$. They are used instead of convolutional encoders in a parallel turbo encoder scheme with an interleaver in between. There is no need for puncturing the output in this setting. The decoder employs the turbo principle in a conventional way. Only the constituent decoders are custom designed for these FSM encoders and they perform trellis decoding using BCJR algorithm like in a conventional turbo decoder.

2.1.2.2 Nonasymmetric SW Coding

The methods proposed in the aforementioned works construct asymmetric SW coding scheme which refers to source Y being encoded at its entropy rate $H(Y)$ and perfectly recovered at the decoder as a side-information (SI) and source X encoded below its entropy rate at $H(X|Y)$ to be decoded with the help of SI Y. Nonasymmetric SW schemes are also possible where the rate of each encoder may vary while the total sum rate is kept constant. In this setting, the rates of encoded streams may be varied to reach any point on the dominant face of SW region. An asymmetric SW scheme can be turned into a nonasymmetric one using *time sharing* [39]. All points of the segment between A and B of the SW rate bound are achievable by time sharing. A fraction λ of samples (λn samples) is coded at the vertex point A, i.e. at rates $(H(Y), H(X|Y))$, and a fraction $(1-\lambda)$ of samples is coded at rates $(H(X), H(Y|X))$ corresponding to the corner point

B of the SW rate region. This leads to the rates $R_X = \lambda H(X) + (1-\lambda)H(X|Y)$ and $R_Y = (1-\lambda)H(Y) + \lambda H(Y|X)$.

In [50] two independent Turbo encoders are used to construct nonasymmetric coding scheme using *parity approach*. Here, instead of treating one of the sources as SI and assuming it is compressed at its entropy rate and losslessly recovered at the decoder, both of the sources are encoded using turbo encoders independently. In this scheme source X is input into the encoder directly while source Y is interleaved before encoding. Half of the systematic bits of each encoder is sent and the parity bits are punctured to get the desired rate. At the decoder, two separate Turbo decoders perform conventional Turbo decoding with the addition of an extra extrinsic information shared between the two Turbo decoders.

The *parity approach* can be modified to generate nonasymmetric schemes without using *time sharing* [51]. This approach can be described as follows. n-vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ are partitioned into sub-sequences $\mathbf{x}^h = (x_1, \ldots, x_l)$, $\mathbf{y}^h = (y_{l+1}, \ldots, y_n)$, $\mathbf{x}^s = (x_{l+1}, \ldots, x_n)$, and $\mathbf{y}^s = (y_1, \ldots, y_l)$. Sequences $\mathbf{x}^h$ and $\mathbf{y}^h$ are compressed by independent source encoders at their corresponding entropy rate $H(X)$ and $H(Y)$. The sequences $\mathbf{x}^s$ and $\mathbf{y}^s$ are encoded by independent systematic channel encoders, producing parity sequences $\mathbf{c}^x = (c_1^x, \ldots, c_a^x)$ and $\mathbf{c}^y = (c_1^y, \ldots, c_b^y)$, respectively. In the decoder, sub-sequences $\mathbf{x}^h$ and $\mathbf{y}^h$ can be easily recovered by the source decoders. In order to recover subsequence $\mathbf{x}^s$ from $\mathbf{c}^x$ and $\mathbf{y}^h$ channel decoding of $\mathbf{c}^x$ using $\mathbf{y}^h$ as side information is performed much as the same in normal parity approach. Similarly, $\mathbf{y}^s$ is recovered from $\mathbf{c}^y$ and $\mathbf{x}^h$ as SI. Because of the correlation between sources X and Y, $\mathbf{y}^h$ is interpreted as a corrupted version of $\mathbf{x}^s$. On the other hand parity bits $\mathbf{c}^x$ are considered to be sent through a noiseless channel. Letting $a \geq (n-l)H(X|Y)$ and $b \geq lH(Y|X)$, the following compression rate pair is achieved:

$$\begin{aligned} R_X &\geq \frac{l}{n}H(X) + \frac{a}{n} \quad \geq \frac{l}{n}H(X) + \left(1 - \frac{l}{n}\right) H(X|Y), \\ R_Y &\geq \frac{n-l}{n}H(Y) + \frac{b}{n} \geq \left(1 - \frac{l}{n}\right) H(Y) + \frac{l}{n}H(Y|X). \end{aligned}$$

Any point on the dominant face of Slepian-Wolf region is achieved by varying the

ratio l/n between 0 and 1. The above mentioned approach is applied to LDPC codes for constructing a practical DSC scheme in [52]. In this paper, for the purpose of practicality and efficiency, authors also proposed to design and use a single type of LDPC encoder for both of the sources X and Y, assuming they are both uniformly distributed. The paper also presents methods to construct suitable degree distribution pairs for LDPC decoders to be used in this scheme. In [53], the methods in [52] are extended to three correlated sources and a scheme to handle rate-adaptation by adding more parity bits.

Nonasymmetric SW coding schemes using the *syndrome approach* were also proposed recently. A syndrome approach was first proposed in [54], based on the partitioning of a single systematic linear channel code $\mathcal{C}$. The main code $\mathcal{C}$ is partitioned into $\mathcal{C}^1$ and $\mathcal{C}^2$ with generator matrices G_1 and G_2. The sources are assumed to be uniform. The generator matrices G_1 and G_2 of the two subcodes are formed by extracting m_1 and m_2 lines, respectively, where $m_1 + m_2 = k$, from the matrix G of the code $\mathcal{C}$. The code construction has been extended in [55] to the case where the sources X and Y are binary but nonuniformly distributed. The method presented in [54] is further developed in [56] [57], for more than two sources scenario using systematic codes. Also, methods for using this scheme for systematic IRA and Turbo codes are proposed and performance simulations are presented.

2.2 Polarization and Polar Codes

In this section, we give a review of polarization and polar codes for single user channel and source coding. The treatment here is based entirely on the works of Arıkan [12], Korada [16] and Şaşoğlu [21].

We consider a pair of correlated discrete random variables (X, Y) with $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. $\mathcal{X}$ can be any discrete alphabet of prime size. In the following we assume $\mathcal{X} = \{0, 1, \ldots, q-1\}$, where q is a prime number. The alphabet $\mathcal{Y}$ is an arbitrary discrete alphabet. (X, Y) is assumed to be distributed according to

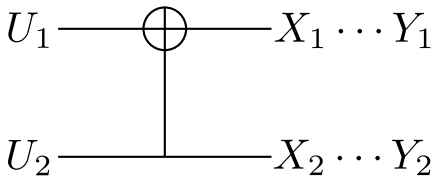


Figure 2.3: First step of polar transformation.

P_{XY} which is an arbitrary discrete distribution. X is considered to be the *source* and Y is the *side information*. We use $H(X|Y)$ to denote conditional entropy which is given by

$$H(X|Y) = -\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} P_{XY}(x,y)\ \log\ P_{X|Y}(x|y). \tag{2.6}$$

The value of entropy is in $[0,1]$ [1]. If $H(X|Y) = 0$ then it means that X is deterministic given the observation Y.

Polarization is a transformation that takes N independent copies of (X,Y) and generates new N pairs of RVs. While all of the conditional entropy terms of original pairs are the same and equal to $H(X|Y)$, the conditional entropy terms of the transformed pairs are all different and close to either 0 or 1. As the size of the transform increases, more and more percentage of the entropy terms gets close to *extremal* values and the percentage of the intermediate ones decay to zero. Thus, the entropy terms of the transformed pairs *polarize*. The bigger transforms are obtained from smaller transforms by recursive construction. At each step of recursion, two identical size-$N/2$ transforms are combined to generate a size-N transform. Therefore, the block size of polar transform is always a power of 2, i.e. $N = 2^n$, $n \in \mathbb{Z}^+$. The recursive nature of the construction makes low complexity encoders and decoders possible.

2.2.1 Polarization

The first step of transformation is depicted in Figure 2.3 which is the basis of polarization. Here, $N = 2$ and we have two identical copies of the source pair denoted by (X_1, Y_1) and (X_2, Y_2). The mapping generates two new variables

$$U_1 = X_1 + X_2 \qquad \text{and} \qquad U_2 = X_2, \tag{2.7}$$

where '+' denotes modulo-q addition. Note that the following is true for the conditional entropy terms

$$2H(X|Y) = H(X^2|Y^2) = H(U_1|Y^2) + H(U_2|Y^2U_1), \tag{2.8}$$

due to the chain rule of entropy. The newly generated RV pairs are thus (U_1, Y^2) and (U_2, Y^2U_1). It is easy to see that

$$H(U_2|Y^2, U_1) \leq H(X|Y) \leq H(U_1|Y^2). \tag{2.9}$$

This first step shows the essence of polar transformation. The first inequality in (2.9) means that observing Y^2U_1 gives a more reliable estimate of U_2 (X_2) than observing Y_2 alone. But, observing Y^2 alone gives a less reliable estimate of U_1 ($X_1 + X_2$). Thus, two new entropy terms are created, one of which is closer to 0 than the original and the other closer to 1. However, it seems that there may be a problem which is the possibility of equality in (2.9). As we will show later, the inequalities are strict as long as the entropies are not one of the *extremal* values of 0 or 1.

Continuing with the same idea we reach at the two step transformation shown in Figure 2.4. Here, we combine two independent first step transformations in Figure 2.3. First define

$$\tilde{Y}_1 \triangleq Y_1^2 \quad \text{and} \quad \tilde{Y}_2 \triangleq Y_3^4. \tag{2.10}$$

[1]Logarithms are to the base q.

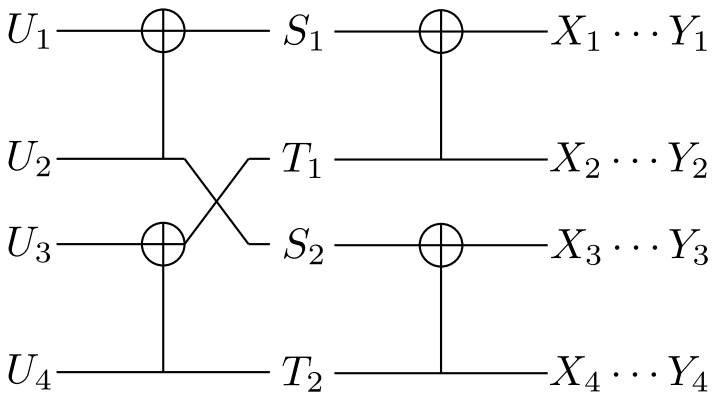


Figure 2.4: Two steps of polar transformation.

Then, note that $(S_1, \tilde{Y}_1)$ and $(S_2, \tilde{Y}_2)$ are i.i.d. We combine them to yield pairs $(U_1, \tilde{Y}^2)$ and $(U_2, \tilde{Y}^2 U_1)$, where $U_1 = S_1 + S_2$ and $U_2 = S_2$. Thus, we have the same relation of entropies as follows

$$\begin{aligned} H(S_2|\tilde{Y}^2, S_1 + S_2) \le H(S_1|\tilde{Y}_1) \le H(S_1 + S_2|\tilde{Y}^2), \\ H(U_2|\tilde{Y}^2, U_1) \le H(S_1|\tilde{Y}_1) \le H(U_1|\tilde{Y}^2), \\ H(U_2|Y^4, U_1) \le H(S_1|\tilde{Y}_1) \le H(U_1|Y^4). \end{aligned} \tag{2.11}$$

Similarly, if we define

$$\bar{Y}_1 \triangleq Y_1^2 S_1 \quad \text{and} \quad \bar{Y}_2 \triangleq Y_3^4 S_2, \tag{2.12}$$

we see that $(T_1, \bar{Y}_1)$ and $(T_2, \bar{Y}_2)$ are i.i.d. We combine them to yield pairs $(U_3, \bar{Y}^2)$ and $(U_4, \bar{Y}^2 U_3)$, where $U_3 = T_1 + T_2$ and $U_4 = T_2$. Thus, we have the same relation of entropies as follows

$$\begin{aligned} H(T_2|\bar{Y}^2, T_1 + T_2) \le H(T_1|\bar{Y}_1) \le H(T_1 + T_2|\bar{Y}^2), \\ H(U_4|\bar{Y}^2, U_3) \le H(T_1|\bar{Y}_1) \le H(U_3|\bar{Y}^2), \\ H(U_4|Y^4, U^3) \le H(T_1|\bar{Y}_1) \le H(U_3|Y^4, U^2). \end{aligned} \tag{2.13}$$

Note that the resulting entropy terms are again give the chain rule expansion on

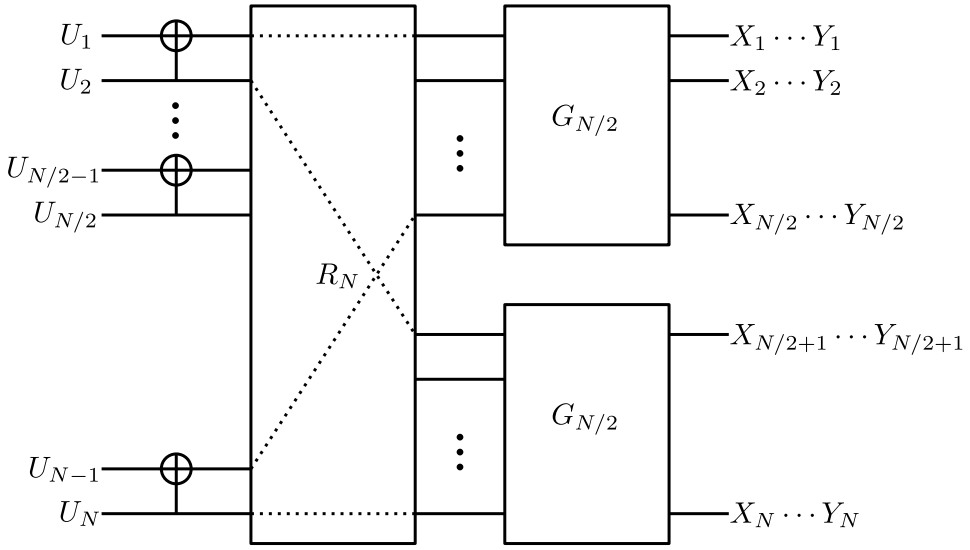


Figure 2.5: Polar transformation of size N (G_N).

the total entropy:

$$\begin{aligned} H(X^4|Y^4) &= 4H(X|Y) = H(U^4|Y^4) \\ &= H(U_1|Y^4) + H(U_2|Y^4, U_1) + H(U_3|Y^4, U^2) + H(U_4|Y^4, U^3). \end{aligned}$$

However, these newly created entropy terms are closer to 0 or 1 than the entropy of the original variable pair. At the first step, generated terms $H(S_1|\tilde{Y}_1)$ and $H(T_1|\bar{Y}_1)$ were somewhat polarized as indicated with equation (2.9). Now, with the second application of the transform to two i.i.d. copies enhances the polarization as indicated with equations (2.11) and (2.13).

The general form of Arıkan's transformation is obtained by recursive application of the the transform in (2.9) to the newly created variables at each step. There are a number of different ways this recursive nature of the transform may be depicted, one of which is shown in Figure 2.5. The polar transform of size N is denoted by G_N and as it can be seen from the figure, it is recursively constructed by two half-size transforms $G_{N/2}$. R_N is called the *reverse shuffle* operator. It shuffles the places of the variables so that natural ordering of U^N from top to down is the correct decoding order. It places the i.i.d. variables from the outputs of two identical transforms ($G_{N/2}$) next to each other so that the basic transform

can be applied again at left hand side of the figure.

The linear transform can be written as a matrix of size $N \times N$. Then, we may write the relation between row vectors U^N and X^N as

$$U^N = X^N G_N. \tag{2.14}$$

Let

$$F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{2.15}$$

and R_N denote the reverse-shuffle operation matrix. Then, by direct observation, size-N polar transform in Figure 2.5 can be algebraically written as

$$G_N = (I_{N/2} \otimes F) R_N (I_2 \otimes G_{N/2}), \tag{2.16}$$

where "$\otimes$" denotes the *Kronecker* product of matrices. Some algebraic manipulations ([12]) yield the following most used representation:

$$G_N = B_N F^{\otimes n}, \tag{2.17}$$

where $B_N = R_N(I_2 \otimes B_{N/2})$ and "$^{\otimes n}$" denotes the n-th Kronecker power of a matrix. B_N is a symmetric permutation matrix referred to as the *bit-reversal* matrix. Since it is symmetric and permutation we have $B_N = B_N^{-1}$. Sometimes just $F^{\otimes n}$ may be used as polarizing transformation. B_N is a permutation and thus just reordering of indices. Another useful fact is that $G_N = G_N^{-1}$.

The main polarization result states that the conditional entropy terms of the transformed variables polarize to either 0 or 1. The following theorem makes this precise.

Theorem 2 ([19]). *For all $\epsilon > 0$,*

$$\lim_{N\to\infty} \frac{1}{N} \left|\left\{i \in [N] :\ H(U_i|Y^N, U^{i-1}) > 1 - \epsilon\right\}\right| = H(X|Y),$$
$$\lim_{N\to\infty} \frac{1}{N} \left|\left\{i \in [N] :\ H(U_i|Y^N, U^{i-1}) < \epsilon\right\}\right| = 1 - H(X|Y).$$

To prove this, one may take the indirect approach as in Arıkan's original work [12] or the direct approach of Şaşoğlu [21]. Here, we follow Şaşoğlu's method. First, we need the following key lemma which states that the single step transformation (2.9) always enhances polarization except at the boundary cases.

Lemma 1 ([21]). *Let $X_1, X_2 \in \mathcal{X}$ and $Y_1, Y_2 \in \mathcal{Y}$ be random variables with the following joint probability density*

$$P_{X_1Y_1X_2Y_2}(x_1, y_1, x_2, y_2) = P_{X_1Y_1}(x_1, y_1)P_{X_2Y_2}(x_2, y_2). \tag{2.18}$$

If $H(X_1|Y_1), H(X_2|Y_2) \in (\delta, 1-\delta)$ for some $\delta > 0$, then there exists and $\epsilon(\delta) > 0$ such that

$$H(X_1 + X_2|Y_1, Y_2) - \max\{H(X_1|Y_1), H(X_2|Y_2)\} \geq \epsilon(\delta). \tag{2.19}$$

Definition 4. *For i.i.d. (X_1, Y_1) and (X_2, Y_2) with $H \triangleq H(X_1|Y_1)$, we define*

$$\begin{aligned} H^0 &\triangleq H(X_1 + X_2|Y^2), \\ H^1 &\triangleq H(X_2|Y^2, X_1 + X_2). \end{aligned} \tag{2.20}$$

Polar transformation of length $N = 2^n$ transforms N i.i.d. copies of (X_1, Y_1) with an average conditional entropy to N different pairs of the form $(U_i, Y^N U^{i-1})$ whose conditional entropy terms are closer to *extremal* values of 0 or 1. The N different entropy terms can be obtained by applying the above definition repeatedly. At each step of transformation the number of entropy terms doubles. The nature of transform G_N results in the following identities

$$\begin{aligned} H(U_1|Y^N) &= H^{0\cdots000} \\ H(U_2|Y^N, U^1) &= H^{0\cdots001} \\ H(U_3|Y^N, U^2) &= H^{0\cdots010} \\ &\vdots \\ H(U_{N-1}|Y^N, U^{N-2}) &= H^{1\cdots10} \\ H(U_N|Y^N, U^{N-1}) &= H^{1\cdots11}. \end{aligned} \tag{2.21}$$

For block length of N the binary vector b as the superscript of H is of length n. Each of the N different conditional entropy terms are enumerated with the binary vector b.

We define two related random processes. We define i.i.d. process $B_1, B_2, \ldots$ where B_i is distributed uniformly over $\{0, 1\}$. Then, we define a $[0, 1]$-valued random process $H_0, H_1, \ldots$ recursively as

$$\begin{aligned} H_0 &= H(X|Y), \\ H_1 &= H_{n-1}^{B_n}, \qquad n = 1, 2, \ldots \end{aligned} \tag{2.22}$$

Because of the relations in (2.21) and the fact that B_i is uniformly distributed the following is true for all n and any $\mathcal{I} \subseteq [0, 1]$:

$$\Pr[H_n \in \mathcal{I}] = \frac{1}{N} \left| \left\{ i : \ H(U_i|Y^N, U^{i-1}) \in \mathcal{I} \right\} \right|. \tag{2.23}$$

Thus, Theorem 2 is implied with the following theorem.

Theorem 3 ([21])**.** *H_n converges almost surely to a $\{0, 1\}$-valued random variable H_∞ with $\Pr[H_\infty = 1] = H(X|Y)$.*

Theorem 3 shows that as the block size increases to infinity the conditional entropies of the transformed pairs approach to either 0 or 1. And combined with (2.23), it implies Theorem 2.

If the entropy is 0 then the variable can be estimated given the observation with certainty. On the other hand if the entropy is 1, then it is not possible to reliably estimate the value of the random variable in any condition. The above reasoning suggests the use of polarization in source coding with side information as follows. Let's fix $\epsilon > 0$ and define the set

$$\mathcal{A} \triangleq \left\{ i \in [N] : \ H(U_i|Y^N, U^{i-1}) \leq \epsilon \right\}. \tag{2.24}$$

From Theorem 2, the size of $\mathcal{A}$ must be greater than $(1 - H(X|Y) - \delta)$ for some $\delta > 0$. Then, the encoder observing a realization x^N calculates $u^N = x^N G^N$

and transmits $u_{\mathcal{A}^c}$ to the decoder, because those are precisely the variables which cannot be estimated reliably using the observations. The remaining, given the observations, can be estimated at the decoder. The decoder, observing $u_{\mathcal{A}^c}$ and side information y^N, generates the estimation $\hat{u}^N$ bit-by-bit *successively* as

$$\hat{u}_i = \begin{cases} u_i, & \text{if } i \in \mathcal{A}^c, \\ 0, & \text{if } i \in \mathcal{A} \text{ and } L(y^N, \hat{u}^{i-1}) > 1, \\ 1, & \text{otherwise.} \end{cases} \tag{2.25}$$

The likelihoods are given as

$$L(y^N, u^{i-1}) = \frac{\Pr\left[U_i = 0 | Y^N = y^n, U^{i-1} = u^{i-1}\right]}{\Pr\left[U_i = 1 | Y^N = y^n, U^{i-1} = u^{i-1}\right]}. \tag{2.26}$$

Let's denote the error probability of optimally decoding i^{th} bit given the observations as $P_e(U_i|Y^N, U^{i-1})$. Obviously $H(U_i|Y^N, U^{i-1}) \to 0$ implies $P_e(U_i|Y^N, U^{i-1}) \to 0$. We need to make ϵ small to keep error probability small. However, for small block lengths if we keep ϵ too small the size of $\mathcal{A}$ may shrink too much away from the ideal value of $(1 - H(X|Y))$. But, as the block length goes to infinity both the error probability decay to zero and the size of $\mathcal{A}$ goes to $(1 - H(X|Y))$. The question is how fast this decay occurs. The above results just show that the polarization happens, but they do not give any indication of how fast it occurs. We discuss this next.

2.2.2 Polarization Rate and Probability of Error

The choice of set $\mathcal{A}$ in (2.24) can be modified to include block length dependent ϵ to yield codes with vanishing error probability. Define block length dependent set $\mathcal{A}_\beta$ as

$$\mathcal{A}_\beta \triangleq \left\{ i \in [N] : \; P_e(U_i|Y^N, U^{i-1}) \leq 2^{-N^\beta} \right\}, \tag{2.27}$$

for some positive $\beta < 1/2$. Following theorem shows that we may consider $\mathcal{A}_\beta$ instead of $\mathcal{A}$ for construction of capacity achieving polar codes.

Theorem 4. *For all* $0 < \beta < 1/2$ *and* $\epsilon > 0$*, there exists* $N_0 = N_0(\beta, \epsilon)$ *such that*

$$|A_\beta| > (1 - H(X|Y) - \epsilon)N \tag{2.28}$$

for all $N \geq N_0$.

Theorem 4 shows that at large block lengths those bit error probabilities $P_e(U_i|Y^N, U^{i-1})$ that go to zero, go to zero exponentially fast in the square root of the block length.

Error terms $P_e(U_i|Y^N, U^{i-1})$ need to be calculated to prove the theorem. The calculation of error terms $P_e(U_i|Y^N, U^{i-1})$ become analytically intractable as the block size increases. Therefore, the error analysis is performed by finding good bounds on error. For this purpose the Bhattacharyya parameter $Z(X|Y)$ is defined as [21]

$$Z(X|Y) \triangleq \frac{1}{q-1} \sum_{\substack{x,x': \\ x \neq x'}} \sum_{y} \sqrt{P_{XY}(x,y) P_{XY}(x',y)}. \tag{2.29}$$

It is well-known that the Bhattacharyya parameter upper bounds the error probability:

Proposition 1 ([21])**.**

$$P_e(X|Y) \leq (q-1) Z(X|Y). \tag{2.30}$$

Since $Z(X|Y)$ bounds the error probability $P_e(X|Y)$, it is expected that $Z(X|Y)$ is close to 0 whenever $H(X|Y)$ is close to 0 and it is close to 1 whenever $H(X|Y)$ is close to 1. It is made precise by the following proposition.

Proposition 2 ([21])**.**

$$\begin{aligned} Z(X|Y)^2 &\leq H(X|Y), \\ H(X|Y) &\leq \log(1 + (q-1) Z(X|Y)). \end{aligned}$$

By the above proposition we have $Z(X|Y) \geq 1-\delta \Rightarrow H(X|Y) \geq 1-2\delta$ and $Z(X|Y) \leq \delta \Rightarrow H(X|Y) \leq \log[1+(q-1)\delta] \leq \kappa\delta$ where $\kappa = (q-1)/\ln q$. Therefore, $Z(X|Y)$ is also considered as a measure of reliability. The Bhattacharyya parameters of the newly created variables after one step transformation also polarize just like the entropies. We have the following relation

$$Z(U_2|Y^2, U_1) \leq Z(X|Y) \leq Z(U_1|Y^2). \tag{2.31}$$

As it is the case for entropies, these inequalities are strict as long as $Z(X|Y)$ is not one of the *extremal* values of 0 or 1. The following lemma gives the bounds on these.

Lemma 2 ([21])**.**

$$\begin{aligned} Z(U_1|Y^2) &\leq (q^2-q+1)Z(X|Y), \\ Z(U_2|Y^2, U_1) &\leq (q-1)Z(X_1|Y_1)^2. \end{aligned}$$

To prove Theorem 4, we need to define a random process that tracks the behavior of Bhattacharyya parameters under recursive polarization construction. For that, we make the following definition in parallel with Definition 4.

Definition 5. *For i.i.d.* (X_1, Y_1) *and* (X_2, Y_2) *with* $Z \triangleq Z(X_1|Y_1)$*, we define*

$$\begin{aligned} Z^0 &\triangleq Z(X_1 + X_2|Y^2), \\ Z^1 &\triangleq Z(X_2|Y^2, X_1 + X_2). \end{aligned} \tag{2.32}$$

With the above definition, Bhattacharyya parameters under recursive polarization construction satisfy

$$\begin{aligned} Z(U_1|Y^N) &= Z^{0\cdots000} \\ Z(U_2|Y^N, U^1) &= Z^{0\cdots001} \\ Z(U_3|Y^N, U^2) &= Z^{0\cdots010} \\ &\vdots \\ Z(U_{N-1}|Y^N, U^{N-2}) &= Z^{1\cdots10} \\ Z(U_N|Y^N, U^{N-1}) &= Z^{1\cdots11}. \end{aligned} \tag{2.33}$$

Just like the entropy process in (2.22), we define two related random processes. We define i.i.d. process $B_1, B_2, \ldots$ where B_i is distributed uniformly over $\{0, 1\}$. Then, we define a $[0, 1]$-valued random process $Z_0, Z_1, \ldots$ recursively as

$$\begin{aligned} Z_0 &= Z(X|Y), \\ Z_1 &= Z_{n-1}^{B_n}, \qquad n = 1, 2, \ldots \end{aligned} \tag{2.34}$$

By proposition 1, $Z(U_i|Y^N, U^{i-1})$ upper bounds average symbol error probability and thus, we could also have defined set $\mathcal{A}_\beta$ as

$$\mathcal{A}_\beta \triangleq \left\{ i \in [N] : \ Z(U_i|Y^N, U^{i-1}) \leq 2^{-N^\beta} \right\}. \tag{2.35}$$

Then, Theorem 4 is proved as a corollary to Lemma 2 and the following lemma.

Lemma 3 ([21])**.** *Let $B_1, B_2, \ldots$ be an i.i.d. binary process where B_i is uniformly distributed over $\{0, 1\}$. Also let $Z_0, Z_1, \ldots$ be a $[0, 1]$-valued process where Z_0 is constant and*

$$\begin{aligned} Z_{n+1} &\leq K Z_n^2 \qquad \text{if } B_{n+1} = 1, \\ Z_{n+1} &\leq K Z_n \qquad \text{if } B_{n+1} = 0, \end{aligned}$$

for some $K > 0$. Suppose that $\{Z_n\}$ converges a.s. to a $\{0, 1\}$-valued random

variable Z_∞ *with* $\Pr[Z_\infty = 0] = z$. *Then, for any* $\beta < 1/2$,

$$\lim_{n\to\infty} \Pr[Z_n \leq 2^{-2^{n\beta}}] = z.$$

Random process $\{Z_n\}$ converges a.s. to a $\{0,1\}$-valued random variable Z_∞ with $\Pr[Z_\infty = 0] = (1 - H(X|Y))$ by Proposition 2 and Theorem 3. Thus, with Lemma 2 it satisfies the conditions of Lemma 3. Then, since by Proposition 1 $P_e(U_i|Y^N, U^{i-1})$ is bounded by $Z(U_i|Y^N, U^{i-1})$, Theorem 4 is implied by Lemma 3. This result shows that we may impose exponentially small bound on probability of decoding error of symbols of *information set* $\mathcal{A}$ and still reach the bound $\frac{1}{N}|\mathcal{A}| \to (1 - H(X|Y))$ as $N \to \infty$.

By the above results, it obvious how source coding with side information can be done using polarization. Encoding and decoding operations were given before in this section. However, how to perform channel coding is not so obvious. It requires some more treatment to prove that channel capacity may be reached by polar coding. This will be discussed next.

2.2.3 Channel Coding

Polar codes were first introduced in Arıkan's original work [12] as binary channel codes that achieve the capacity of *symmetric* channels. Since then there has been numerous work expanding the area of applicability of polarization. Previous section presents some of the results of those work. There is no constraint on the distribution of X and joint distribution of (X, Y) in previous results. Therefore, we may construct polar channel codes that achieve the capacity of any discrete channel, not only symmetric. The theoretical analysis of average probability of error depends on *randomized maps* and was introduced in [27] as an extension to Korada's *randomized rounding* method in [16] which was in lossy source coding context.

Let (X, Y) be a pair of correlated random variables with properties defined

as in previous sections. We may consider X as input to a channel described by conditional probability $P_{Y|X}$ and Y as the channel's output. We consider a block of $N = 2^n$ i.i.d. channel uses resulting in (X^N, Y^N). In addition, let $U^N = X^N G_N$ as always. Note that the following are true for the joint distributions of the random variables:

$$P_{X^N Y^N}(x^N, y^N) = \prod_{i=1}^{N} P_{XY}(x_i, y_i),$$
$$P_{U^N Y^N}(u^N, y^N) = P_{X^N Y^N}(u^N G_N, y^N).$$

Also, for polar coding purposes we decompose the joint distribution as

$$P_{U^N Y^N}(u^N, y^N) = P_{Y^N}(y^N) \prod_{i=1}^{N} P_{U_i|Y^N, U^{i-1}}(u_i|y^N, u^{i-1}). \tag{2.36}$$

Similarly the following is true for P_{U^N}:

$$P_{U^N}(u^N) = P_{X^N}(u^N G_N),$$
$$P_{U^N}(u^N) = \prod_{i=1}^{N} P_{U_i|U^{i-1}}(u_i|u^{i-1}). \tag{2.37}$$

We define the following *polarization sets*:

$$\mathcal{H}_X \triangleq \left\{ i \in [N] : Z(U_i|U^{i-1}) \geq 1 - \delta_N \right\},$$
$$\mathcal{L}_{X|Y} \triangleq \left\{ i \in [N] : Z(U_i|Y^N, U^{i-1}) \leq \delta_N \right\},$$

where $\delta_N = 2^{-N^\beta}$ for some positive $\beta < 1/2$. Then, we define the *information* ($\mathcal{I}$) and *frozen* ($\mathcal{F}$) sets as

$$\mathcal{I} \triangleq \mathcal{H}_X \cap \mathcal{L}_{X|Y}, \quad \mathcal{F} \triangleq [N] \setminus \mathcal{I}. \tag{2.38}$$

Proposition 3. *For all $0 < \beta < 1/2$, there exists $N_0 = N_0(\beta, \epsilon)$ such that*

$$|\mathcal{I}| > (H(X) - H(X|Y) - \epsilon)N, \tag{2.39}$$

for all $N > N_0$.

First, let's elaborate on the definitions made. Arıkan's polar transform is often considered as a transform that *distills* the randomness in a block of i.i.d. random variables. By that we mean that it takes N i.i.d. variables X^N with the same *mediocre* distribution and creates N different variables U^N with almost *extremal* distributions, i.e. either almost uniform or almost deterministic distributions. $\mathcal{H}_X$ represents the set of transformed variables with almost uniform distribution and thus called the *high entropy set*. The entropy of each variable in this set is close to 1. The remaining part of the variables comprise the *low entropy set* and have almost deterministic distributions and thus have entropies close to 0. Therefore, we have $|\mathcal{H}_X| \sim H(X)N$. Similarly, $\mathcal{L}_{X|Y}$ represents the set of transformed variables with almost deterministic distributions under the *side information* Y^N. The variables in this set are almost deterministic given the previous variables and the side information. A subset of $\mathcal{L}_{X|Y}$ is $\mathcal{L}_X \triangleq \{i \in [N] : Z(U_i|U^{i-1}) \leq \delta_N\}$ which is the "null set" if original variables have *uniform* distribution. Thus $\mathcal{L}_X$ represents the non-uniformity in distribution of X. The information set $\mathcal{I}$ is precisely those variables that have almost uniform distribution without side information Y^N and almost deterministic distribution with the side information, and thus can be estimated at the decoder.

To prove Proposition 3 we first make the following definition.

Definition 6. *Let* $P_{Y_1|X}$ *and* $P_{Y_2|X}$ *denote two DMCs with same input alphabet and possibly different output alphabets. We say* $P_{Y_2|X}$ *is degraded with respect to* $P_{Y_1|X}$*, denoted as* $P_{Y_2|X} \preceq P_{Y_1|X}$*, if there exists a distribution* $P_{Y_2|Y_1}$ *such that*

$$P_{Y_2|X}(y_2|x) = \sum_{y_1} P_{Y_1|X}(y_1|x)\ P_{Y_2|Y_1}(y_2|y_1). \tag{2.40}$$

Lemma 4 (Degradation [16])**.** *Let* $P_{Y_1|X}$ *and* $P_{Y_2|X}$ *denote two DMCs with* $P_{Y_2|X} \preceq P_{Y_1|X}$*. Then,*

$$Z(X|Y_1) \leq Z(X|Y_2). \tag{2.41}$$

Proof. See Appendix A.2. □

Proof of Proposition 3. From Theorem 4 we know that $|\mathcal{L}_{X|Y}| > (1 - H(X|Y) - \epsilon_1)N$. Define $\mathcal{L}_X \triangleq \{i \in [N] : Z(U_i|U^{i-1}) \leq \delta_N\}$. By following similar steps to proof of Theorem 4, we may prove $|\mathcal{L}_X| > (1 - H(X) - \epsilon_2)N$. Note that we can write $\mathcal{I} = \mathcal{L}_{X|Y} \setminus \{\mathcal{L}_X \cup \{[N] \setminus (\mathcal{L}_X \cup \mathcal{H}_X)\}\}$. $\Delta \triangleq \{[N] \setminus (\mathcal{L}_X \cup \mathcal{H}_X)\}$ is the set of partially polarized indices and its fraction goes to zero as $N \to \infty$ by polarization, i.e. $\frac{|\Delta|}{N} < \epsilon_3$. Since obviously channel $P_{U_i|U^{i-1}}$ is degraded with respect to $P_{U_i|Y^N,U^{i-1}}$, we have $Z(U_i|Y^N, U^{i-1}) \leq Z(U_i|U^{i-1})$. Thus, we have $\mathcal{L}_X \subseteq \mathcal{L}_{X|Y}$. From that the claim follows. □

2.2.3.1 Encoding

The encoder first constructs u^N symbol by symbol and then calculates $x^N = u^N G_N$ to be supplied to the channel. The subset of indices of u^N identified by set $\mathcal{I}$ are the message symbols intended for the receiver. They are determined uniformly. The remaining non-message indices are computed according to a set of maps that are *shared* between the encoder and decoder. These maps will be identified with λ_i and defined for $i \in \mathcal{I}^c$. We use $\lambda_{\mathcal{I}^c}$ to denote the set of maps shared between the encoder and the decoder.

We will define two different versions of these maps. The first one will be *maximum a posteriori* based deterministic rules. The second one will be random maps. In the analysis, random maps will be used for the sake of analytic tractability. The analysis of error probability will be done as an average over all possible maps.

We define deterministic maps $\bar{\lambda}_i : \mathcal{X}^{i-1} \to \mathcal{X}$ as

$$\bar{\lambda}_i(u^{i-1}) \triangleq \arg\max_{u' \in \mathcal{X}} \left\{ P_{U_i|U^{i-1}}(u'|u^{i-1}) \right\}. \tag{2.42}$$

We also define class of random maps $\Lambda_i : \mathcal{X}^{i-1} \to \mathcal{X}$ as

$$\Lambda_i(u^{i-1}) \triangleq q, \quad \text{w.p. } P_{U_i|U^{i-1}}(q|u^{i-1}). \tag{2.43}$$

Maps λ_i are the realizations of random maps Λ_i. Each realization of set of maps

$\lambda_{\mathcal{I}^c}$ results in different encoding and decoding protocols. The distribution over the choice of maps is induced with the above equation (2.43). The encoder uses the input symbols $u_{\mathcal{I}}$ and identical shared maps λ_i to construct the length-N vector u^N *successively* as

$$u_i = \begin{cases} u_i, & \text{if } i \in \mathcal{I}, \\ \lambda_i(u^{i-1}), & \text{otherwise.} \end{cases} \tag{2.44}$$

Then, $x^N = u^N G_N$ is supplied to the channel.

2.2.3.2 Decoding

Decoder decodes the sequence $\hat{u}^N$ symbol by symbol using the observations y^N. We define the following decoding functions:

$$\zeta_i(y^N, u^{i-1}) \triangleq \arg\max_{u' \in \mathcal{X}} \left\{ P_{U_i|Y^N U^{i-1}}(u'|y^N, u^{i-1}) \right\}. \tag{2.45}$$

The decoder uses the identical shared maps λ_i to reconstruct the estimate $\hat{u}^N$ *successively* as

$$\hat{u}_i = \begin{cases} \zeta_i(y^N, \hat{u}^{i-1}), & \text{if } i \in \mathcal{I}, \\ \lambda_i(\hat{u}^{i-1}), & \text{otherwise.} \end{cases} \tag{2.46}$$

Instead of λ_i, the decoder could also use $\bar{\lambda}_i$ when doing deterministic operation. As stated before the encoder and decoder are using the same shared maps for non-message indices. A realization of set of random maps has a probability of occurrence induced by probabilities $P_{U_i|U^{i-1}}$ as given in (2.43). Each realization results in different encoding / decoding protocols. We use randomized map concept to bound the expected average error probability by taking expectation over all possible set of maps, thus showing that there exists at least one good set of maps.

For different shared maps $\lambda_{\mathcal{I}^c}$, the results of encoding operation may be different for the same input $u_{\mathcal{I}}$. For encoder at step $i \in \mathcal{I}$ of the process the inputs are

inserted which are assumed to be uniformly distributed. Thus, for a realization of set of maps $\lambda_{\mathcal{I}^c}$, a particular x^N occurs with a certain probability induced by input distribution and maps. We define the resulting average (over $u_{\mathcal{I}}$) probability of error of above encoding and decoding operations as $P_e[\lambda_{\mathcal{I}^c}]$. In the following we show that for set $\mathcal{I}$ defined in (2.38) and encoding and decoding methods defined in 2.2.3.1 and 2.2.3.2, there exists a set of maps $\lambda_{\mathcal{I}^c}$ such that $P_e[\lambda_{\mathcal{I}^c}] \leq O(2^{-N^\beta})$, for $0 < \beta < 1/2$. We do that by determining the *expected* average probability of error over the ensembles of codes generated by different encoding maps $\lambda_{\mathcal{I}^c}$. The distribution over the choices of maps is given in (2.43). That is, we take expectation of $P_e[\Lambda_{\mathcal{I}^c}]$ which is a random quantity. Then we show that *expected* average probability of error decay to zero as $O(2^{-N^\beta})$. This implies that for at least one choice of $\lambda_{\mathcal{I}^c}$ the average probability of error decays to zero as $O(2^{-N^\beta})$.

2.2.3.3 Total Variation Bound

To analyze the average error probability P_e via the probabilistic method we define the following probability measure

$$Q(u^N) = \prod_{i=1}^{N} Q(u_i|u^{i-1}), \tag{2.47}$$

where conditional probabilities are defined as

$$Q(u_i|u^{i-1}) \triangleq \begin{cases} \frac{1}{q}, & \text{if } i \in \mathcal{I}, \\ P_{U_i|U^{i-1}}(u_i|u^{i-1}), & \text{otherwise.} \end{cases} \tag{2.48}$$

The probability measure Q defined in (2.47) is a perturbation of P_{U^N} in (2.37). The difference between P and Q is due to those indices in message set $\mathcal{I}$. The following lemma provides a bound on the total variation distance between P and Q. The lemma shows that by inserting uniformly distributed message bits in the proper indices at the encoder does not perturb the statistics too much.

Lemma 5. *Let probability measures P and Q be defined as* (2.37) *and* (2.47)*, respectively. For sufficiently large N and $0 < \beta < 1/2$, the total variation distance*

between P and Q is bounded as

$$\sum_{u^N \in \mathcal{X}^N} \left| P_{U^N}(u^N) - Q(u^N) \right| \leq 2^{-N^\beta}. \tag{2.49}$$

Proof. See Appendix A.3. □

2.2.3.4 Average Error Probability

The encoding and decoding rules were established in Sections 2.2.3.1 and 2.2.3.2, respectively. Consider the sequence u^N formed at the encoder and observation y^N received by the decoder. The decoder makes an SC decoding error on the i-th symbol for the following tuples:

$$\mathcal{T}^i \triangleq \big\{ (u^N, y^N) : \exists u' \in \mathcal{X} \text{ s.t. } u' \neq u_i, \\ P_{U_i|Y^N U^{i-1}}(u_i|y^N, u^{i-1}) \leq P_{U_i|Y^N U^{i-1}}(u'|y^N, u^{i-1}) \big\}. \tag{2.50}$$

The set $\mathcal{T}^i$ represents those tuples causing error at the decoder in the case u_i is inconsistent with respect to observations and the decoding rule. The complete set of tuples causing errors is

$$\mathcal{T} \triangleq \bigcup_{i \in \mathcal{I}} \mathcal{T}^i. \tag{2.51}$$

Assuming randomized maps shared between encoder and decoder, the average error probability is a random quantity given as follows

$$P_e[\Lambda_{\mathcal{I}^c}] = \sum_{(u^N, y^N) \in \mathcal{T}} \left[P_{Y^N|U^N}(y^N|u^N) \cdot \frac{1}{q^{|\mathcal{I}|}} \prod_{i \in \mathcal{I}^c} \mathbb{1}_{\{\Lambda_i(u^{i-1}) = u_i\}} \right] \tag{2.52}$$

The expected average block error probability is calculated by averaging over the randomness in the encoder and decoder

$$\bar{P}_e \triangleq \mathbb{E}_{\{\Lambda_{\mathcal{I}^c}\}} \left[P_e[\Lambda_{\mathcal{I}^c}] \right]. \tag{2.53}$$

The following lemma bounds the expected average block error probability.

Lemma 6. *Consider the polarization based channel code described in Sections 2.2.3.1 and 2.2.3.2. Let the information set $\mathcal{I}$ be selected as in Proposition 3. Then for $0 < \beta < 1/2$ and sufficiently large N,*

$$\mathbb{E}_{\{\Lambda_{\mathcal{I}^c}\}}\left[P_e[\Lambda_{\mathcal{I}^c}]\right] < 2^{-N^\beta}.$$

Proof. First, note that the expectation of average probability of error is written as

$$\mathbb{E}_{\{\Lambda_{\mathcal{I}^c}\}}\left[P_e[\Lambda_{\mathcal{I}^c}]\right] = \sum_{(u^N,y^N)\in\mathcal{T}} \left[P_{Y^N|U^N}(y^N|u^N) \cdot \frac{1}{q^{|\mathcal{I}|}} \prod_{i\in\mathcal{I}^c} \mathbb{P}\left\{\Lambda_i(u^{i-1}) = u_i\right\} \right].$$

From the definition of random mappings Λ_i it follows that

$$\mathbb{P}\left\{\Lambda_i(u^{i-1}) = u_i\right\} = P_{U_i|U^{i-1}}(u_i|u^{i-1}).$$

Then, we may substitute the definition for $Q(u^N)$ in (2.47) into the expression of expected average probability of error to get

$$\mathbb{E}_{\{\Lambda_{\mathcal{I}^c}\}}\left[P_e[\Lambda_{\mathcal{I}^c}]\right] = \sum_{(u^N,y^N)\in\mathcal{T}} P_{Y^N|U^N}(y^N|u^N)\; Q(u^N).$$

Then we split the error into two main parts, one due to the polar decoding function and the other due to the total variation distance between probability measures.

$$\begin{aligned}
\mathbb{E}_{\{\Lambda_{\mathcal{I}^c}\}}\left[P_e[\Lambda_{\mathcal{I}^c}]\right] &= \sum_{(u^N,y^N)\in\mathcal{T}} P_{Y^N|U^N}(y^N|u^N)\;\left[Q(u^N) - P(u^N) + P(u^N)\right], \\
&\leq \sum_{(u^N,y^N)\in\mathcal{T}} P_{U^NY^N}(u^N,y^N) + \sum_{u^N}\left|Q(u^N) - P(u^N)\right|.
\end{aligned}$$

The second part of the error which is due to total variation distance is upper bounded as $O(2^{-N^\beta})$ by Lemma 5. Thus, it remains to upper bound the error term due to polar decoding. Remember that $\mathcal{T} \triangleq \cup_{i\in\mathcal{I}}\mathcal{T}^i$. We may upper bound

each error symbol by symbol. Define error probability for symbol $i \in \mathcal{I}$ as

$$\varepsilon^i \triangleq \sum_{(u^N, y^N) \in \mathcal{T}^i} P_{U^N Y^N}(u^N, y^N).$$

But this is the average probability of error for symbol i, i.e. $\varepsilon^i = P_e(U_i|Y^N, U^{i-1})$. Probability of error is upper bounded by the Bhattacharyya parameter by Proposition 1. By union bound, total average probability of error is $\varepsilon \leq \sum_i \varepsilon^i$. Then we have

$$\begin{aligned} \varepsilon &\leq \sum_{i \in \mathcal{I}} (q-1) Z(U_i|Y^N, U^{i-1}), \\ &\leq (q-1) N \delta_N. \end{aligned}$$

This completes the proof that the expected average probability of error is upper bounded as $O(2^{-N^\beta})$. □

Since the expected value over the random maps of average probability of error decays to zero, there must be at least one deterministic set of maps for which $P_e \to 0$.

2.2.3.5 Symmetric Channels and Uniform Distributions

It is well known that the capacity of a symmetric channel is achieved with uniform distribution at its input. For uniform distributions, some of the concepts in previous sections simplify. The random maps defined in (2.43) always results in uniform distribution: $\Lambda_i(u^{i-1}) = a$, w.p. $1/q$, $\forall a \in \mathcal{X}$. Thus instead of sharing set of maps $\lambda_{\mathcal{I}^c}$ between encoder and decoder, we may generate a vector for $\mathcal{I}^c$ uniformly at random and share that. Also, each realization of a set of maps $\lambda_{\mathcal{I}^c}$ have the same probability, which means that the expected average error probability $\bar{P}_e$ and average error probability for a realization $P_e[\lambda_{\mathcal{I}^c}]$ are the same. Thus, as proven in [12] the value of those symbols in $\mathcal{I}^c$ don't matter in the sense that each selection results in the same average error probability. We can choose any fixed vector for $\mathcal{I}^c$ and share it between encoder and decoder.

Chapter 3

Distributed Coding of Uniform Sources

In this chapter we present a simple method for performing Slepian-Wolf (SW) coding using polar codes, based on [58]. [58] defines a general framework in which a good *single user* channel code is used to obtain a good SW code that can achieve any point on the dominant face of SW region. However, there is one important limitation that the marginal distributions of the source variables must be *uniform*. In exchange of this limitation the encoding/decoding operations can be done using single user channel encoders and decoders. The method requires syndrome calculation and channel decoder to perform *coset decoding*. By coset decoding we mean that the channel decoder needs to be able to decode at an arbitrary coset of the code. But a normal channel decoder can only decode in a single coset (which is generally the zero syndrome one). Both coset decoding and syndrome calculation are not trivial operations for an arbitrary good channel code. For example, for turbo codes these operations are very hard. In this chapter we show that polar codes fit nicely into this method in the sense that normal channel encoders and decoders may be used and thus efficient low complexity implementations can be achieved. The general SW coding (not limited to uniform source marginals) using polar codes requires more complex decoders as presented in Chapter 4. The contents of this chapter are based on our work in [59].

3.1 Description of the Method

A method for constructing nonasymmetric SW scheme from a single channel code restricted to the case of *uniformly* distributed sources using syndrome approach was proposed in [58]. We show how this method may be applied to construct SW coding using polar codes. We assume two correlated sources X and Y to be binary RVs with uniform marginals. The correlation model between sources X and Y is given as $Y = X \oplus E$, where $E \sim \text{Bernoulli}(\epsilon)$. Thus, $H(X|Y) = H(Y|X) = H(E) = \mathcal{H}(\epsilon)$, where $\mathcal{H}(\epsilon) = -\epsilon \cdot \log \epsilon - (1-\epsilon) \cdot \log(1-\epsilon)$. Here, Y can also be viewed as a version of X passed through a *virtual* BSC with cross–over probability ϵ.

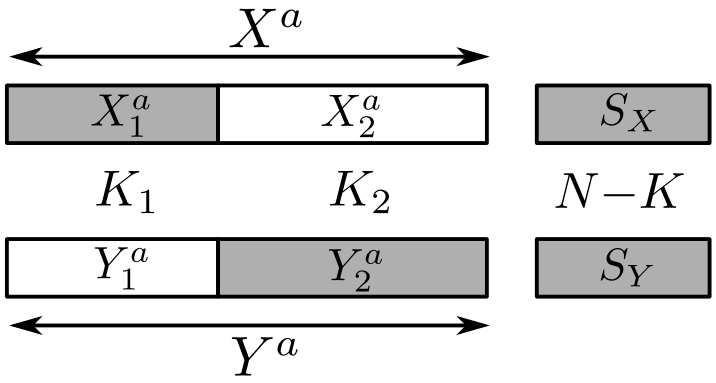


Figure 3.1: Encoding for nonasymmetric SW.

The method of [58] can be summarized as follows. Consider two i.i.d. distributed and correlated N–vectors $\mathbf{x} = [\mathbf{x}^a\ \mathbf{x}^b]$ and $\mathbf{y} = [\mathbf{y}^a\ \mathbf{y}^b]$ sampled from source RV (X, Y). $\mathbf{x}^a$ represents the first K bits and $\mathbf{x}^b$ represents the last $N-K$ bits of vector $\mathbf{x}$ (the same applies to $\mathbf{y}$). Also, let $\mathbf{x}^a = [\mathbf{x}_1^a\ \mathbf{x}_2^a]$ and $\mathbf{y}^a = [\mathbf{y}_1^a\ \mathbf{y}_2^a]$. $\mathbf{x}_1^a$ represents the first K_1 bits and $\mathbf{x}_2^a$ represents the last K_2 bits of $\mathbf{x}^a$ (the same applies to $\mathbf{y}^a$), where $K_1 + K_2 = K$. Let G be a $K \times N$ generator matrix and H a $(N-K) \times N$ parity check matrix of some block code. Assume that H has the form $[H_a\ H_b]$ where H_a is an $(N-K) \times K$ matrix and H_b is an $(N-K) \times (N-K)$ non–singular matrix. Notice that the systematic version of a code is a special case with $H_b = I_{N-K}$. The syndromes of $\mathbf{x}$ and $\mathbf{y}$ are calculated as $\mathbf{s_x} = \mathbf{x}H^T = \mathbf{x}^a H_a^T \oplus \mathbf{x}^b H_b^T$ and $\mathbf{s_y} = \mathbf{y}H^T = \mathbf{y}^a H_a^T \oplus \mathbf{y}^b H_b^T$, respectively. Then, X–encoder sends $(\mathbf{x}_1^a, \mathbf{s_x})$ and Y–encoder sends $(\mathbf{y}_2^a, \mathbf{s_y})$. The partitioning of variables may be visualized as in Figure 3.1. The total number of bits sent

by both encoders is $2N - K$ yielding a sum rate $R = 2 - K/N$. By choosing $K/N = 2 - H(X,Y) = 1 - H(E)$, this scheme results in a code operating on the dominant face of the SW region. Then by varying K_1 and K_2 subject to $K_1 + K_2 = K$, one can operate at any point on the dominant face.

The decoding of the above scheme, which is depicted in Figure 3.2, is done as follows. Let $\mathbf{e} = \mathbf{x} \oplus \mathbf{y}$ be the error vector. Then, $\mathbf{s_e} = \mathbf{e}H^T = (\mathbf{x} \oplus \mathbf{y})H^T = \mathbf{s_x} \oplus \mathbf{s_y}$. The method assumes that there is a syndrome decoder for the given code which is supplied with all–zeros vector as input and $\mathbf{s_e}$ as the coset index. The estimate $\hat{\mathbf{e}}$ is obtained as the output. With this estimated error pattern, $\mathbf{x}_2^a$ and $\mathbf{y}_1^a$ can be recovered using $\mathbf{y}_2^a$ and $\mathbf{x}_1^a$, respectively, as shown in right half of Figure 3.2. Finally, $\mathbf{x}^b$ and $\mathbf{y}^b$ are obtained as

$$\mathbf{x}^b = (\mathbf{s_x} \oplus \mathbf{x}^a H_a^T)(H_b^T)^{-1}, \tag{3.1}$$

$$\mathbf{y}^b = (\mathbf{s_y} \oplus \mathbf{y}^a H_a^T)(H_b^T)^{-1}. \tag{3.2}$$

Note that, although it is not shown explicitly in Figure 3.2, likelihood calculation of the the all–zeros vector input to the decoder is done using the *assumed* cross–over probability ϵ of the *virtual* BSC between sources X and Y. Thus, the LLRs input to the decoder are $L = \log \frac{1-\epsilon}{\epsilon}$.

A polar code is identified by a parameter set $(N, K, \mathcal{A}, \mathbf{u}_{\mathcal{A}^c})$, where $N = 2^n$ is the block length, K is the code dimension, $\mathcal{A}$ is the information index set of size K and $\mathbf{u}_{\mathcal{A}^c}$ is the frozen bits vector of size $N - K$. The frozen bits $\mathbf{u}_{\mathcal{A}^c}$ identify a coset of the linear block code and can be used as syndrome of the polar code [12]. An advantage of polar codes for this scheme is that the required syndrome decoding is readily available in SC polar decoder. The SC decoder can decode as easily for a given $\mathbf{u}_{\mathcal{A}^c}$ as it can for zero syndrome. However, this standard form of polar codes cannot be used in this method. Because, the second part of parity check matrix (H_b) of a normal polar code is not invertible, thus the second part of decoding given by (3.1) cannot be performed. But, the systematic version of polar codes [23] can be used. Systematic polar encoding operation does the mapping $(\mathbf{x}_{\mathcal{B}}, \mathbf{u}_{\mathcal{A}^c}) \to (\mathbf{x}_{\mathcal{B}^c}, \mathbf{u}_{\mathcal{A}})$, where $\mathbf{x}_{\mathcal{B}}$ is the K-bit systematic vector and $\mathbf{u}_{\mathcal{A}^c}$ is the $(N - K)$-bit frozen bits vector.

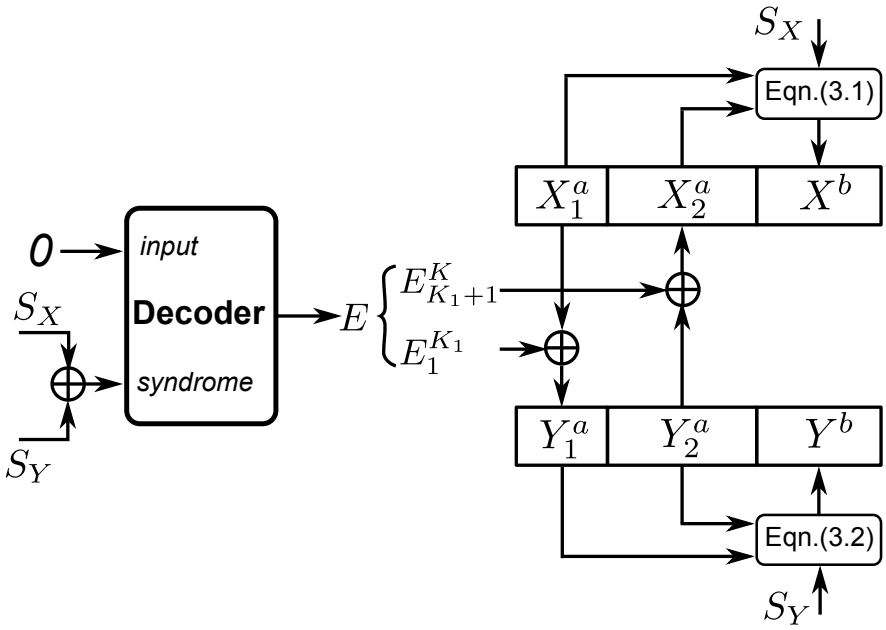


Figure 3.2: Decoding for nonasymmetric SW.

Now returning back to the nonasymmetric SW method of [58] described above, we set $\mathbf{x}^a = \mathbf{x}_{\mathcal{B}}$, $\mathbf{x}^b = \mathbf{x}_{\mathcal{B}^c}$ and $\mathbf{s}_{\mathbf{x}} = \mathbf{u}_{\mathcal{A}^c}$. This way we fulfill the requirements of the method such that when $\mathbf{x}^a$ is decoded using the estimated error vector $\hat{\mathbf{e}}$, the rest, $\mathbf{x}^b$, can be recovered from $\mathbf{x}^a$ and $\mathbf{s}_{\mathbf{x}}$. Given $\mathbf{x}^a = \mathbf{x}_{\mathcal{B}}$ and $\mathbf{s}_{\mathbf{x}} = \mathbf{u}_{\mathcal{A}^c}$, computing $\mathbf{x}^b = \mathbf{x}_{\mathcal{B}^c}$ is nothing but a systematic polar encoding operation.

We also use CRC to improve the short block length performance of SCL decoder, which was originally proposed by the authors of [25] in channel coding context. There are two ways to incorporate a CRC into the above method. The first one is to calculate the L_{crc}–bit CRCs of N–bit source blocks and transmit them separately. With this modification, X–encoder sends $(\mathbf{x}_1^a, \mathbf{s}_{\mathbf{x}}, \mathbf{c}_{\mathbf{x}})$ and Y–encoder sends $(\mathbf{y}_2^a, \mathbf{s}_{\mathbf{y}}, \mathbf{c}_{\mathbf{y}})$, where $\mathbf{c}_{\mathbf{x}}$ and $\mathbf{c}_{\mathbf{y}}$ are CRCs of $\mathbf{x}$ and $\mathbf{y}$, respectively. Since CRC operation is linear, the CRC of error vector $\mathbf{e} = \mathbf{x} \oplus \mathbf{y}$ is $\mathbf{c}_{\mathbf{x}} \oplus \mathbf{c}_{\mathbf{y}}$. Thus, the SCL syndrome decoder can use this information when estimating the error vector. To match the required sum rate R, the channel code is adjusted so that $K = N(2 - R) + 2L_{crc}$.

The second way is to complete $N' = N - L_{crc}$ length information blocks to N with L_{crc} bits of CRC. In this method, the CRCs are inside the N-bit $\mathbf{x}$ and $\mathbf{y}$

vectors. Thus, the LLR calculation for polar decoder is done differently:

$$L_i = \begin{cases} \log \frac{1-\epsilon}{\epsilon}, & i \in \{1, \ldots, N - L_{crc}\} \\ 0, & i \in \{N - L_{crc} + 1, \ldots, N\}, \end{cases} \tag{3.3}$$

Note that in (3.3), while the statistics of the first $(N - L_{crc})$ bits (correlated source bits) are known (Ber(ϵ)) and used for decoding, the statistics of the CRC bits are assumed to be uniform. To match the required sum rate R, the channel code is adjusted so that $K = N(2 - R) + RL_{crc}$. The two different methods of adding the CRC does not make any difference performance wise.

3.2 Complexity of the Method

Source encoding is essentially a syndrome calculation. It is done using a SC polar encoder which is of complexity $O(N \log N)$ [12]. Source decoding is done in two stages. First, the estimate of error vector is calculated. This is the critical step of decoding where errors are introduced. Here we use a SC *list* decoder with a list size L. Hence, the complexity is $O(L \cdot N \log N)$ [25]. The second part of decoding involves calculation of $\mathbf{x}^b(\mathbf{y}^b)$ from $\mathbf{x}^a(\mathbf{y}^a)$ and $\mathbf{s}_x(\mathbf{s}_y)$ using (3.1) (3.2). However, in practice matrix inversion and multiplication are not used. This calculation is effectively a systematic polar encoding operation and efficiently performed using a SC polar decoder. Thus, its complexity is $O(N \log N)$. Therefore, the total complexity of the source decoder is dominated by the first step which is of complexity $O(L \cdot N \log N)$.

3.3 Simulations

In this section, we present simulation results on performance of the source coding method discussed. The correlation model between sources X and Y is given as $Y = X \oplus Z$, where $Z \sim$ Ber(ϵ). In all of the plots, the rates of codes are kept

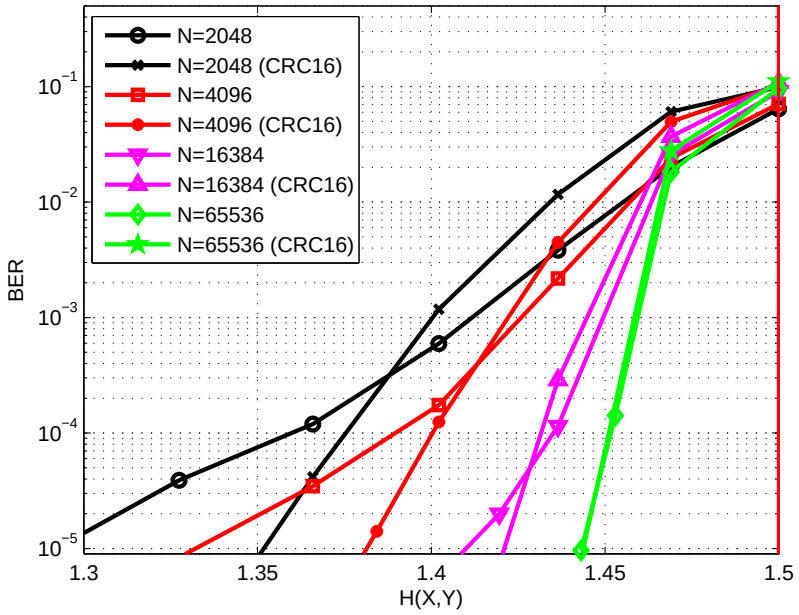


Figure 3.3: BER plot for rate allocation $R_X = 0.5$, $R_Y = 1$ (asymmetric).

at a defined constant value while ϵ is varied to achieve different $H(X|Y)$ points. The plotted BER corresponds to the averaged value over X and Y sources. The polar decoder used is the SCL decoder of [25]. To improve the performance, a 16-bit CRC (CCITT) is added. The list decoder selects the output from the final list with the aid of CRC. Note that, for source coding, CRC is appended to the "codeword" vector $\mathbf{x}$ as opposed to channel coding case where it is appended to "information" vector $\mathbf{u}_{\mathcal{A}}$. The list size is set to 32 for all cases. The code construction is done via the method proposed in [24] and optimized to $p = 0.09$ for $R = 0.5$.

The performance for $(R_X, R_Y) = (0.5, 1)$, which corresponds to asymmetric rate allocation, is presented in Figure 3.3. Figure 3.4 shows the BER plot with rates allocated such that it results in symmetric setting: $R_X = 0.75$ and $R_Y = 0.75$. It can be observed from the figure that the performance is slightly inferior to the asymmetric case given in Figure 3.3. This is expected, since as opposed to asymmetric case where no error is made for the source Y, in nonasymmetric cases estimation of Y is also prone to errors, furthermore these errors propagate

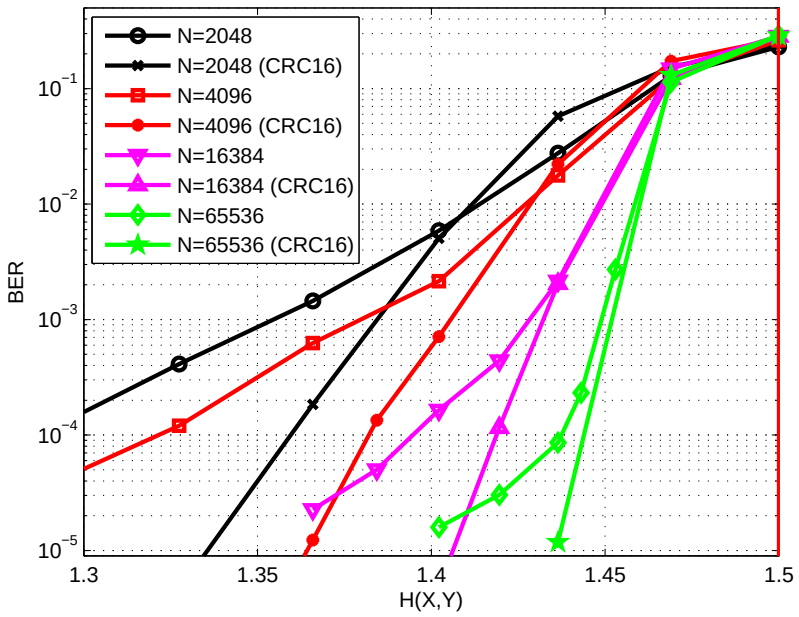


Figure 3.4: BER plot for rate allocation $R_X = 0.75$, $R_Y = 0.75$ (symmetric).

to the recovery of X. Simulation result of a nonasymmetric operating point is given in Figure 3.5. The rate allocation is such that $R_X = 0.875$ and $R_Y = 0.625$. It can be observed from the results that the performance is the same for all nonasymmetric points.

Table 3.1: Nonasymmetrical SW performance for $R = 1.5$ ($H(X,Y)$ values for a BER of 10^{-5}).

$(\mathbf{R_X}, \mathbf{R_Y})$ \ **N**	2048	4096	16384	65536
$(0.500, 1.000)$	1.361	1.388	1.424	1.444
$(0.625, 0.875)$	1.321	1.349	1.402	1.435
$(0.750, 0.750)$	1.321	1.349	1.402	1.435

Results for three different rate allocations are given in Table 3.1. A BER of 10^{-5} is considered to be lossless when determining the rate points. Figure 3.6 shows the performance of the method on SW rate region for $N = 65536$.

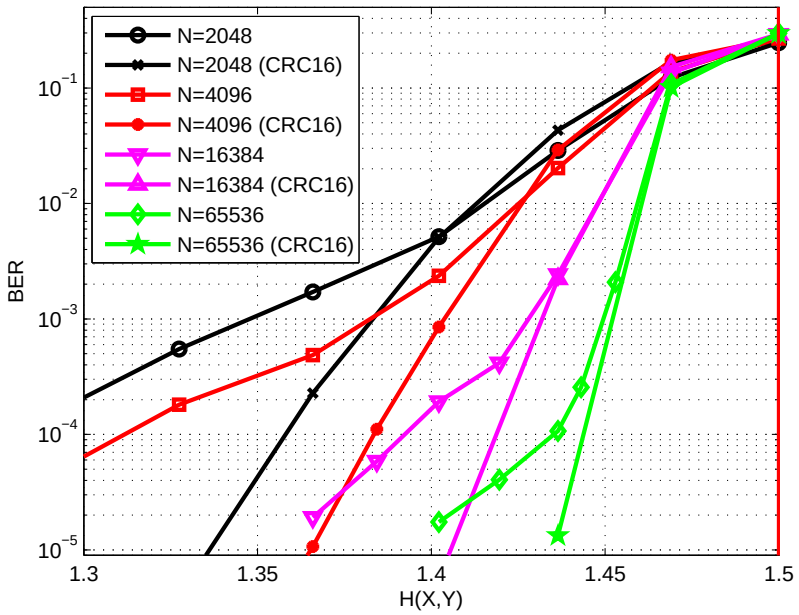


Figure 3.5: BER plot for rate allocation $R_X = 0.875$, $R_Y = 0.625$ (a nonasymmetric point).

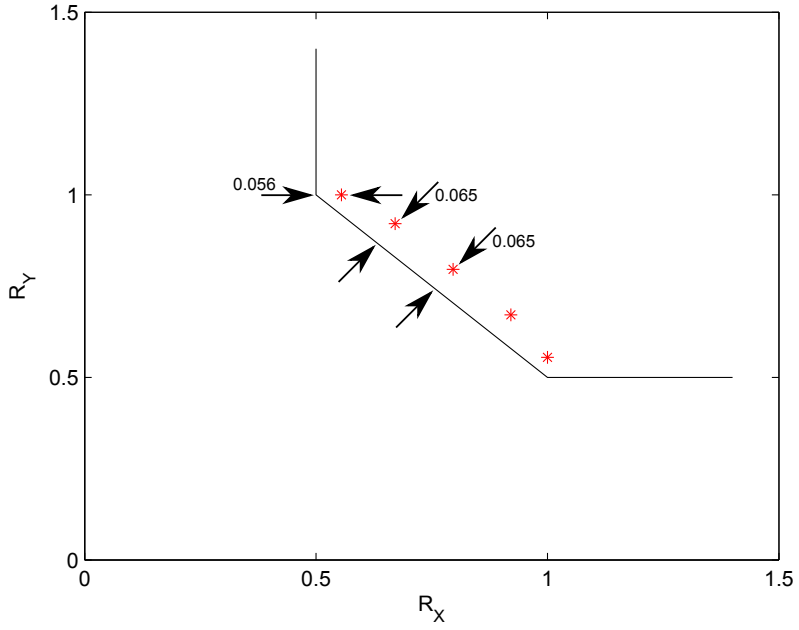


Figure 3.6: Nonasymmetric method for $N = 65536$ together with the SW bound.

Chapter 4

Distributed Lossless Coding

In this chapter, we present an extension of single-user polar codes to two-user settings. The approach pursued here is based on *monotone chain rule* expansion of entropy (or mutual information), which we will explain in detail later, and was introduced in [28] by Arıkan. Before [28], a slightly different method for two-user and multi-user generalization of polar codes were presented in [18] and [60], respectively, in multiple-access channel (MAC) context. The basis of those works rest on *joint polarization* approach that produces "extremal" channels which are also MACs. This is in contrast to the approach in [28] where "extremal" channels are single-user. The authors in [18] using *joint polarization* approach reached an interesting result stating that there are five types of "extremal" channels. However, in that work, it has also been shown that while polar coding can achieve a certain rate point on the dominant face of MAC capacity region, it cannot achieve an arbitrary rate point.

In Section 4.1, we explore in detail, polarization for distributed setting based on *monotone chain rule* approach introduced in [28]. We extend the treatment with an addition of *side-information* variable and using prime sized alphabets. Then, in Section 4.2 we show how a Slepian-Wolf (SW) polar code may be generated that achieves full dominant face of SW rate region. Note that, the approach used here works for arbitrary discrete source distributions, not only sources with

uniform marginals like the approaches in [16] or Chapter 3. In addition, we explicitly write recursive formulas and give detailed successive-cancellation (SC) *list* decoder implementation as a generalization of single-user list decoding introduced in [25]. We also present simulation results giving the performance of list decoder. Then, in Section 4.3 we show how to perform MAC (dual problem of SW coding) polar coding using the results of Section 4.1. In independent and contemporaneous works [61] and [62], authors pursue in MAC context, necessarily the same approach introduced in [28] for polar code construction like we do in Section 4.3. However, they remain restricted to uniform rate-region (uniform input distributions). We show that full MAC rate-region may be achieved for arbitrary input distributions. In addition, we give performance simulation results of successive cancellation *list* decoding.

4.1 Polarization for Distributed Setting

In this section we present the generalization of Section 2.2 to multi-user setting. For notational convenience we study two user setting however the treatment can easily be generalized to more than two users. This section is based on the work of Arıkan in [28]. We will denote user 1 and 2 with variables X and Y, respectively. We will denote the *side information* with variable Z. The user variables are from prime sized alphabets: $X, Y \in \mathcal{X} = \{0, 1, \ldots, q-1\}$, where q is prime. $Z \in \mathcal{Z}$ may be of any discrete alphabet. The variables are drawn from an arbitrary joint distribution P_{XYZ}.

The possible compression rates R_1 and R_2 for users X and Y that can be achieved with reliable lossless reconstruction under side information Z form a two dimensional region. We denote the points in this region with *rate vector* $\bar{R} \triangleq (R_1, R_2)$. This rate region is defined by

$$\mathcal{R} = \{\bar{R} : R_1 \geq H(X|Z,Y),\ R_2 \geq H(Y|Z,X),\ R_1 + R_2 \geq H(X,Y|Z)\}. \quad (4.1)$$

The subset of $\mathcal{R}$ consisting of points for which the sum-rate holds with equality

is referred to as the *dominant face* of the rate region:

$$\mathcal{J} = \{\bar{R} : R_1 \geq H(X|Z,Y),\ R_2 \geq H(Y|Z,X),\ R_1 + R_2 = H(X,Y|Z)\}. \quad (4.2)$$

The bounds of this region does not change even when the encoding of correlated source (X,Y) is done separately and without the knowledge of side information Z. The decoding must be done jointly with side information. This result is due to Slepian and Wolf in their seminal work [38]. Therefore, region $\mathcal{R}$ is also called Slepian-Wolf (SW) rate region.

4.1.1 Paths and Rates

Consider the i.i.d. block of random variables (X^N, Y^N, Z^N) with $N = 2^n$ for some $n \geq 1$. Let, U^N and V^N denote the polar transforms of X^N and Y^N, respectively:

$$U^N = X^N G_N, \qquad V^N = Y^N G_N. \quad (4.3)$$

The joint distribution of (X^N, Y^N, Z^N) through polar transformation induces a joint distribution on (U^N, V^N, Z^N). Since, G_N is a one-to-one mapping, we can write the total entropy as follows

$$H(X^N, Y^N|Z^N) = NH(X,Y|Z) = H(U^N, V^N|Z^N). \quad (4.4)$$

Recall that for single user polarization the total entropy term is expanded in the order of increasing indices of the user vector U^N: $H(U_i|Y^N, U^{i-1})$. This expansion reflects the decoding order of symbols with the *observations* (both Y^N and U^{i-1}) obtained so far. At each decoding step a single symbol is decoded. The polarization result shows that these conditional entropy terms polarize. They approach to 0 or 1 as $N \to \infty$. Thus, we use the ones with entropy close to 0 to convey information.

Similarly, in multi-user setting we consider expansions of entropy that preserve the order of indices of both user vector U^N and V^N. However, since there are

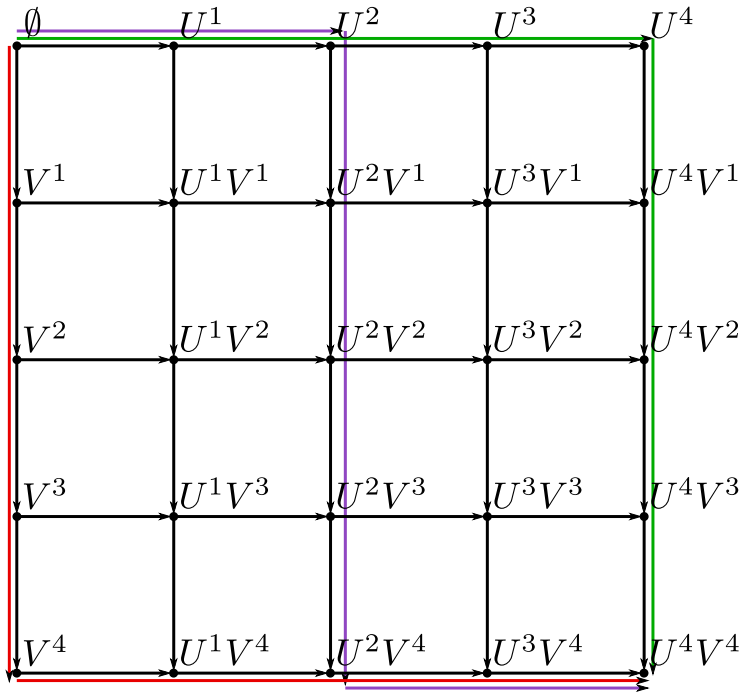


Figure 4.1: Monotone chain rule expansions.

two users, there is a freedom in the choice of expansion order. To capture the expansion order a new vector of length $2N$ is defined: $S^{2N} \triangleq \pi_N(U^N, V^N)$. $\pi_N(\cdot,\cdot)$ is from a special class of permutations where it takes 2 length-N vectors and permutes the elements with special consideration such that the relative order of indices of both vectors do not change. That is U_i comes before U_{i+1} and V_j comes before V_{j+1} in the permuted vector S^{2N}. For example, for $N = 4$ the following is a valid permutation

$$S^8 = (U_1, V_1, V_2, U_2, V_3, U_3, U_4, V_4). \tag{4.5}$$

But $S^8 = (U_1, V_1, V_2, U_3, V_3, U_2, U_4, V_4)$ is not a valid permutation. The allowed permutations may be visualized with a directed *path* on a two dimensional "chain rule diagram" as mentioned in [28] and shown in Figure 4.1. The path on diagram may also be represented with a *path* string b^{2N}, $b_k \in \{0,1\}, \forall k \in [2N]$. If $b_k = 0$ then S_k is the *next* variable from U^N vector. Similarly, if $b_k = 1$ then S_k is the *next* variable from V^N vector. Thus, b^{2N} has exactly N zeros and N ones. The expansion in (4.5) is represented by $b^8 = 01101001$.

The special type of expansions described above is referred to as *monotone* expansions. In the rest of the discussion we always use that kind of expansions. Then, using the S^{2N} vector defined above, *monotone* expansion of total entropy in (4.4) can be written as

$$H(U^N, V^N|Z^N) = \sum_{k=1}^{2N} H(S_k|Z^N, S^{k-1}). \tag{4.6}$$

Depending on the choice of path vector b^{2N}, the above expression represents all possible $\binom{2N}{N}$ different particular monotone expansions. The total entropy is decomposed into *incremental* entropy terms of the form $H(S_k|Z^N, S^{k-1})$. Those incremental entropy terms are visualized by edges on chain rule diagram and thus the variables S_k are called the *edge variables*.

We define the following two rates for two users:

$$R_1 = \frac{1}{N} \sum_{\substack{k=1: \\ b_k=0}}^{2N} H(S_k|Z^N, S^{k-1}), \qquad R_2 = \frac{1}{N} \sum_{\substack{k=1: \\ b_k=1}}^{2N} H(S_k|Z^N, S^{k-1}).$$

It is easy to see that R_1 attains its minimum $\frac{1}{N}H(U^N|Z^N, V^N) = H(X|Z, Y)$ with path $b^{2N} = (1^N 0^N)$ (red path in Figure 4.1). Similarly, R_2 attains its minimum $\frac{1}{N}H(V^N|Z^N, U^N) = H(Y|Z, X)$ with path $b^{2N} = (0^N 1^N)$ (green path in Figure 4.1). In any case the sum rate is constant at $R_{sum} = R_1 + R_2 = \frac{1}{N}H(U^N, V^N|Z^N)$. The following are true for any path:

$$R_1 \geq H(X|Z, Y), \qquad R_2 \geq H(Y|Z, X), \qquad R_1 + R_2 = H(X, Y|Z).$$

Thus, the defined rates lie on the dominant face of the rate region and span its two end points.

However, it is not clear that if there exists paths that achieve any point on the dominant face with arbitrary precision as $N \to \infty$. In the following we show that it is indeed the case. To that end, we first define a *distance* between two paths as follows.

Definition 7. *Let b^{2N} and $\tilde{b}^{2N}$ be two paths with rate pairs (R_1, R_2) and $(\tilde{R}_1, \tilde{R}_2)$, respectively. The distance between b^{2N} and $\tilde{b}^{2N}$ is defined as*

$$d(b^{2N}, \tilde{b}^{2N}) \triangleq |R_1 - \tilde{R}_1|.$$

Note that, since $R_1 + R_2 = \tilde{R}_1 + \tilde{R}_2 = H(X, Y|Z)$, the distance is also equal to $|R_2 - \tilde{R}_2|$. Then, we make a definition of neighborhood between two paths as follows.

Definition 8. *Let two paths b^{2N} and $\tilde{b}^{2N}$ be defined as neighbors if $\tilde{b}^{2N}$ can be obtained from b^{2N} by transposing b_i with b_j for some $i < j$ such that*

(i). $b_i \neq b_j$,

(ii). the substring $b_{i+1} \dots b_{j-1}$ is either all 0s or all 1s.

The following proposition limits the distance for neighboring paths.

Proposition 4 ([28])**.** *For neighbor paths b^{2N} and $\tilde{b}^{2N}$, the following is true:*

$$d(b^{2N}, \tilde{b}^{2N}) \leq \frac{1}{N}.$$

We define a class of paths as $\mathcal{V}_{2N} \triangleq \{0^i 1^N 0^{N-i} : 0 \leq i \leq N\}$ similar to magenta path in Figure 4.1. Then, the following theorem shows that we may attain any point on the dominant face of rate region with arbitrary precision using paths from this class as $N \to \infty$.

Theorem 5 ([28])**.** *Let (R_x, R_y) be any given rate point on the dominant face of rate region $\mathcal{R}$. For any given $\epsilon > 0$, there exists a N and a path b^{2N} from class $\mathcal{V}_{2N}$ with rate pair (R_1, R_2) satisfying*

$$|R_1 - R_x| \leq \epsilon, \qquad |R_2 - R_y| \leq \epsilon.$$

The theorem may be proven easily using Proposition 4. Let, $1/N < \epsilon$. For paths $1^N 0^N (i = 0)$ and $0^N 1^N (i = N)$ in class $\mathcal{V}_{2N}$, R_1 attains $H(X|Z, Y)$ and

$H(X|Z)$, respectively. Thus, it spans the two end points of values possible for R_x. For any $0 < i < N-1$ the distance between paths $0^i1^N0^{N-i}$ and $0^{i+1}1^N0^{N-i-1}$ is less than $1/N$ by Proposition 4. Thus there is an i such that $|R_1 - R_x| < \epsilon$. Also, since $R_1 + R_2 = R_x + R_y$, $|R_2 - R_y| < \epsilon$ is also true for that i.

There may be other paths that achieve the desired rate on the dominant face. The class $\mathcal{V}_{2N}$ gives a simple rule to obtain paths that achieve any point on the dominant face as $N \to \infty$.

4.1.2 Path Scaling and Polarization

In the previous section we have seen that the total entropy term $H(U^N, V^N|Z^N)$ is decomposed into $2N$ *incremental* entropy terms of the form $H(S_k|Z^N, S^{k-1})$. An incremental term is the entropy of single variable $S_k \in \mathcal{X}$. Thus, it is $[0,1]$-valued. However, we have not said anything about polarization of those incremental entropy terms, yet. In this section we define the conditions under which they polarize.

Similar to single user case we define the following information sets:

$$\mathcal{A}_1(\beta) \triangleq \left\{k \in [2N] :\ b_k = 0,\ Z(S_k|Z^N, S^{k-1}) \leq 2^{-N^\beta}\right\},$$
$$\mathcal{A}_2(\beta) \triangleq \left\{k \in [2N] :\ b_k = 1,\ Z(S_k|Z^N, S^{k-1}) \leq 2^{-N^\beta}\right\}.$$

Definition 9 (Path Scaling)**.** *For any path b^{2N} representing a monotone chain rule for (U^NV^N) and any integer $L = 2^l$, let Lb^{2N} denote*

$$\underbrace{b_1 \cdots b_1}_{L}\underbrace{b_2 \cdots b_2}_{L} \cdots \underbrace{b_{2N} \cdots b_{2N}}_{L}, \tag{4.7}$$

which represents a monotone chain rule for $(U^{LN}V^{LN})$. This scaling operation preserves the "shape" of the path.

Proposition 5 ([28])**.** *Let (R_1, R_2) be the rate pair for a fixed path b^{2N}. Then for any $l \geq 1$, (R_1, R_2) is also the rate pair for scaled path 2^lb^{2N}.*

Theorem 6 ([28]). *Fix $N_0 = 2^{n_0}$ and b^{2N_0} for some $n_0 \geq 1$. Let (R_1, R_2) be the rate pair for b^{2N_0}. Let $N = 2^l N_0$ for $l \geq 1$ and S^{2N} be edge variables for $2^l b^{2N_0}$. Then for any given $0 < \beta < 1/2$ we have*

$$\lim_{l\to\infty} \frac{1}{2N} \left| \left\{ k \in [2N] : \ 2^{-N^\beta} < Z(S_k|Z^N, S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$
$$\lim_{l\to\infty} \frac{1}{N} \left|\mathcal{A}_1(\beta)\right| = 1 - R_1, \qquad \lim_{l\to\infty} \frac{1}{N} \left|\mathcal{A}_2(\beta)\right| = 1 - R_2.$$

To prove this theorem it is enough to realize the following simple fact. Fix a block length $N_0 = 2^{n_0}$ and path b^{2N_0} for $(U^{N_0}V^{N_0})$. Then consider the scaled path $2b^{2N_0}$ for $(U^{2N_0}V^{2N_0})$. Let S^{2N_0} and T^{4N_0} be edge variables for b^{2N_0} and $2b^{2N_0}$, respectively. Let $\tilde{S}^{2N_0}$ be and independent copy of S^{2N_0}. Then, polar transformations (4.3) under this one step path scaling result in the following identities:

$$T_{2k-1} = S_k + \tilde{S}_k, \qquad T_{2k} = \tilde{S}_k, \tag{4.8}$$

for all $k \in [N_0]$. This is one step polar transformation. As we increase the size of the block length through *path scaling* by $L = 2^l$ times, we will be generating polarized variables from L independent copies of each of the base variables. There are $2N_0$ different variable pairs $(S_k, Z^{N_0}S^{k-1})$ and corresponding entropies $H(S_k|Z^{N_0}S^{k-1})$ at "base block" of length N_0. The entropy terms $H(S_k|Z^{N_0}S^{k-1})$ for an arbitrary "base path" b^{2N_0} are not necessarily polarized. As we scale the path by L, we will achieve a block length of $N = LN_0$ and "scaled path" b^{2N}. The rate pair for the scaled path is the same as the base path. However, for each entropy term in base block there are L entropy terms in scaled block which are polarized.

Let the edge variables of the base path b^{2N_0} be denoted with S^{2N_0} as before and the edge variables of l-step scaled path b^{2N} be denoted with T^{2N}. Let's focus on a single index k in the base block and make the following variable substitution:

$$\bar{S} = S_k, \qquad \bar{Y} = Z^{N_0}S^{k-1}. \tag{4.9}$$

Let $\bar{S}^L$ and $\bar{Y}^L$ be L i.i.d. replica of $\bar{S}$ and $\bar{Y}$, respectively. Let the transform

of $\bar{S}^L$ be denoted with $\bar{T}^L = \bar{S}^L G_L$. Note that Lemma 1 and Theorem 3 apply. Thus we have

$$\lim_{l\to\infty} \frac{1}{L} \left|\left\{i \in [L] :\ H(\bar{T}_i|\bar{Y}^L, \bar{T}^{i-1}) \geq 1-\epsilon\right\}\right| = H(\bar{S}|\bar{Y}),$$
$$\lim_{l\to\infty} \frac{1}{L} \left|\left\{i \in [L] :\ H(\bar{T}_i|\bar{Y}^L, \bar{T}^{i-1}) \leq \epsilon\right\}\right| = 1 - H(\bar{S}|\bar{Y}).$$

Note that the l-step scaling operation extends the basic step transform in (4.8) and we have the following relations

$$\bar{T}_i = T_{L(k-1)+i},$$
$$\bar{Y}^L = Z^N T^{L(k-1)},$$

for all $i \in [L]$. The above is true for each $k \in [2N_0]$. We can write the same result in original variables as follows

$$\lim_{l\to\infty} \frac{1}{L} \left|\left\{i \in [L] :\ H(T_{L(k-1)+i}|Z^N, T^{L(k-1)+i-1}) \geq 1-\epsilon\right\}\right| = H(S_k|Z^{N_0}, S^{k-1}),$$
$$\lim_{l\to\infty} \frac{1}{L} \left|\left\{i \in [L] :\ H(T_{L(k-1)+i}|Z^N, T^{L(k-1)+i-1}) \leq \epsilon\right\}\right| = 1 - H(S_k|Z^{N_0}, S^{k-1}).$$

for all $k \in [2N_0]$. Thus the polarization occurs through path scaling. An arbitrary path b^{2N_0} at base block selects an arbitrary decoding order (in the monotone class). The entropy terms are not necessarily polarized. But, after the path "shape" is fixed and it is scaled, each variable may be substituted as in (4.9) and the setting reduces to single user case. All of the discussions in Section 2.2.2 apply. Using definition of Bhattacharyya parameter in (2.29) we have $Z(T_k|Z^N, T^{k-1})$ bound symbol error probabilities. By Theorem 4 we can say that the polarization occurs and its rate and the probability of error are order $O(2^{-N^\beta})$.

Now we may employ the above polarization concept in distributed source coding setup as follows. If the entropy is 0 then the variable can be estimated given the observations with certainty. On the other hand if the entropy is 1, then it is not possible to reliably estimate the value of the random variable in any condition.

Let's fix a path b^{2N}, $\epsilon > 0$ and define the sets

$$\begin{aligned} \mathcal{A}_1 &\triangleq \left\{k \in [2N] :\ b_k = 0,\ Z(S_k|Z^N, S^{k-1}) \leq \epsilon\right\}, \\ \mathcal{A}_2 &\triangleq \left\{k \in [2N] :\ b_k = 1,\ Z(S_k|Z^N, S^{k-1}) \leq \epsilon\right\}. \end{aligned} \tag{4.10}$$

From Theorem 6, the size of $\mathcal{A}_1$ must be greater than $(1-R_1-\delta)$ and the size of $\mathcal{A}_2$ must be greater than $(1-R_2-\delta)$ for some $\delta > 0$, where rate pair (R_1, R_2) is path dependent and $R_1 \geq H(X|Z,Y)$, $R_2 \geq H(Y|Z,X)$ and $R_1 + R_2 = H(X,Y|Z)$.

Definition 10. *In the following discussions, we frequently need to refer to corresponding indices of U^N, V^N and their permuted form $S^{2N} = \pi_N(U^N, V^N)$. The permutation is characterized by path string b^{2N}. Thus, all indices are considered in the context of an assumed path b^{2N}. We define indices i and j of U^N and V^N corresponding to index k of S^{2N}, respectively as*

$$i = \sum_{l=1}^{k} \mathbb{1}_{\{b_l=0\}}, \qquad j = \sum_{l=1}^{k} \mathbb{1}_{\{b_l=1\}}. \tag{4.11}$$

Also, let $\pi_N^{(k)}$ denote the permutation for the first k elements as $S^k = \pi_N^{(k)}(U^i V^j)$. Note that with this definition we have $0 \leq i, j \leq N$ and $i + j = k$, for $1 \leq k \leq N$. Then we have

$$S_k = \begin{cases} U_i, & \text{if } b_k = 0, \\ V_j, & \text{if } b_k = 1. \end{cases}$$

Using these definitions we define the following information sets:

$$\begin{aligned} \tilde{\mathcal{A}}_1 &\triangleq \left\{i \in [N] :\ b_k = 0,\ Z(S_k|Z^N, S^{k-1}) \leq \epsilon\right\}, \\ \tilde{\mathcal{A}}_2 &\triangleq \left\{j \in [N] :\ b_k = 1,\ Z(S_k|Z^N, S^{k-1}) \leq \epsilon\right\}. \end{aligned} \tag{4.12}$$

$\tilde{\mathcal{A}}_1$ gives the indices of U^N corresponding to the indices of S^{2N} in set $\mathcal{A}_1$. Similarly, $\tilde{\mathcal{A}}_2$ gives the indices of V^N corresponding to the indices of S^{2N} in set $\mathcal{A}_2$. Note that, for set $\tilde{\mathcal{A}}_1$ we have $Z(S_k|Z^N, S^{k-1}) = Z(U_i|Z^N, U^{i-1}V^j)$ and for set $\tilde{\mathcal{A}}_2$ we have $Z(S_k|Z^N, S^{k-1}) = Z(V_j|Z^N, U^i V^{j-1})$.

The calculation of sets $\mathcal{A}_1$ and $\mathcal{A}_2$ may be done using the successive cancellation decoder, described in the next section, in large number of Monte-Carlo

simulations and integrating the results. Or, one may use a construction method adapted from the single user case described in [63].

Encoding

The setting consists of two separate encoders each observing one of either x^N or y^N. The decoder receives compressed sequences from both encoders and generates estimates $\hat{x}^N$ and $\hat{y}^N$ with the help of side information z^N.

Encoder 1 observing a realization x^N calculates $u^N = x^N G_N$ and transmits $u_{\tilde{\mathcal{A}}_1^c}$ to the decoder. Encoder 2 observing a realization y^N calculates $v^N = y^N G_N$ and transmits $v_{\tilde{\mathcal{A}}_2^c}$ to the decoder. Because those are precisely the variables which cannot be estimated reliably using the observations. The remaining, given the observations, can be estimated at the decoder.

Decoding

The decoder, receiving $(u_{\tilde{\mathcal{A}}_1^c}, v_{\tilde{\mathcal{A}}_2^c})$ first assembles them into sub-vector $s_{\mathcal{A}^c}$, where $\mathcal{A} \triangleq \mathcal{A}_1 \cup \mathcal{A}_2$. Then using side information z^N, generates the estimation $\hat{s}^{2N}$ bit-by-bit *successively* as

$$\hat{s}_k = \begin{cases} s_k & \text{if } k \in \mathcal{A}^c, \\ 0 & \text{if } k \in \mathcal{A} \text{ and } L(z^N, \hat{s}^{k-1}) > 1, \\ 1 & \text{otherwise.} \end{cases} \tag{4.13}$$

The likelihoods are given as

$$L(z^N, s^{k-1}) = \frac{\Pr\left[S_k = 0 | Z^N = z^n, S^{k-1} = s^{k-1}\right]}{\Pr\left[S_k = 1 | Z^N = z^n, S^{k-1} = s^{k-1}\right]}. \tag{4.14}$$

4.2 Slepian-Wolf Coding

In the previous section we have seen that polarization occurs under path scaling and how we can use it to construct distributed source coding. Encoders just employ standard polar transforms separately and send subsets of calculated vectors. The decoder must calculate likelihood ratios in (4.14). Likelihoods depends on calculations of probabilities of the form $\Pr\left[S_k = s_k | Z^N = z^n, S^{k-1} = s^{k-1}\right]$. Similar to single user case we need to find recursive formulas for calculation of this probability to be able to achieve low complexity decoder implementation.

4.2.1 Recursive Formulas

First, for an i.i.d. block of variables (X^N, Y^N, Z^N), transformed vectors (U^N, V^N) as defined in (4.3) and permuted vector $S^{2N} = \pi_N(U^N, V^N)$, we make the following definitions.

$$P_N(s_k | z^N, s^{k-1}) \triangleq \Pr\left[S_k = s_k | Z^N = z^N, S^{k-1} = s^{k-1}\right], \tag{4.15}$$

$$P_N^{(i,j)}(u_i, v_j | z^N, u^{i-1}, v^{j-1}) \triangleq \\ \Pr\left[U_i = u_i, V_j = v_j | Z^N = z^N, U^{i-1} = u^{i-1}, V^{j-1} = v^{j-1}\right]. \tag{4.16}$$

We make use of the Definition 10 when we talk about vectors U^N, V^N, S^{2N} and their corresponding indices i, j, k under assumed path b^{2N}. Then, we have the following identity:

$$P_N(s_k | z^N, s^{k-1}) = \begin{cases} \sum_{v_{j+1}} P_N^{(i,j+1)}(u_i, v_{j+1} | z^N, u^{i-1}, v^j) & \text{if } b_k = 0 \\ \sum_{u_{i+1}} P_N^{(i+1,j)}(u_{i+1}, v_j | z^N, u^i, v^{j-1}) & \text{if } b_k = 1. \end{cases} \tag{4.17}$$

Now we will show that we can recursively calculate probabilities

$P_N^{(i,j)}(u_i, v_j|z^N, u^{i-1}, v^{j-1})$. Let's first define the following four scaling constants.

$$
\begin{aligned}
&C_1 \triangleq 1, \\
&C_2 \triangleq \Pr\left[U_{2i-1} = u_{2i-1}|Z^{2N} = z^{2N}, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right], \\
&C_3 \triangleq \Pr\left[V_{2j-1} = v_{2j-1}|Z^{2N} = z^{2N}, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right], \\
&C_4 \triangleq \Pr\left[U_{2i-1} = u_{2i-1}, V_{2j-1} = v_{2j-1}|Z^{2N} = z^{2N}, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right].
\end{aligned} \tag{4.18}
$$

The constants are calculated as follows.

$$
\begin{aligned}
C_2 = \sum_{u_{2i}, v_{2j-1}, v_{2j}} & P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}) \\
& P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}).
\end{aligned} \tag{4.19}
$$

$$
\begin{aligned}
C_3 = \sum_{u_{2i-1}, u_{2i}, v_{2j}} & P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}) \\
& P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}).
\end{aligned} \tag{4.20}
$$

$$
\begin{aligned}
C_4 = \sum_{u_{2i}, v_{2j}} & P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}) \\
& P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}).
\end{aligned} \tag{4.21}
$$

We have the following four recursive formulas depending on the indices of u^{2N} and v^{2N} being odd or even:

$$
\begin{aligned}
P_{2N}^{(2i-1,2j-1)}&(u_{2i-1}, v_{2j-1}|z^{2N}, u^{2i-2}, v^{2j-2}) = (1/C_1) \\
&\cdot \sum_{u_{2i}, v_{2j}} P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}) \\
&\cdot P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}),
\end{aligned} \tag{4.22}
$$

$$P_{2N}^{(2i,2j-1)}(u_{2i}, v_{2j-1}|z^{2N}, u^{2i-1}, v^{2j-2}) = (1/C_2)$$
$$\cdot \sum_{v_{2j}} P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2})$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}), \quad (4.23)$$

$$P_{2N}^{(2i-1,2j)}(u_{2i-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-1}) = (1/C_3)$$
$$\cdot \sum_{u_{2i}} P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2})$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}), \quad (4.24)$$

$$P_{2N}^{(2i,2j)}(u_{2i}, v_{2j}|z^{2N}, u^{2i-1}, v^{2j-1}) = (1/C_4)$$
$$\cdot P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2})$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}). \quad (4.25)$$

The proofs are given in Appendix B.1.

Note that the scaling constants are not actually needed for decoder implementation. We could also calculate slightly different types of probabilities: the *total* probabilities. Then, the calculated probabilities would be of the following form:

$$P_{2N}^{(2i-1,2j-1)}(u_{2i-1}, v_{2j-1}, u^{2i-2}, v^{2j-2}|z^{2N}) =$$
$$\sum_{u_{2i}, v_{2j}} P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}|z_1^N)$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}|z_{N+1}^{2N}), \quad (4.26)$$

$$P_{2N}^{(2i,2j-1)}(u_{2i}, v_{2j-1}, u^{2i-1}, v^{2j-2}|z^{2N}) =$$
$$\sum_{v_{2j}} P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}|z_1^N)$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}|z_{N+1}^{2N}), \quad (4.27)$$

$$P_{2N}^{(2i-1,2j)}(u_{2i-1}, v_{2j}, u^{2i-2}, v^{2j-1}|z^{2N}) =$$
$$\sum_{u_{2i}} P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}|z_1^N)$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}|z_{N+1}^{2N}), \quad (4.28)$$

$$P_{2N}^{(2i,2j)}(u_{2i}, v_{2j}, u^{2i-1}, v^{2j-1}|z^{2N}) =$$
$$P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}|z_1^N)$$
$$\cdot P_N^{(i,j)}(u_{2i}, v_{2j}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}|z_{N+1}^{2N}). \quad (4.29)$$

In SC decoder, either type of formulas may be used. However, in SC *list* decoder the latter type must be used. This is because the list decoder makes comparisons of alternate decoding paths and the comparisons are meaningful over the total probabilities.

4.2.2 Decoder Implementation

In this section, we give an explicit recursive implementation of the SC decoder for two-user monotone chain rule based polar codes. The algorithms described in this section are generalizations of those by Tal and Vardy [40] to two-user setting. We keep the general structure and the names of the algorithms intact. We modify the inside of the algorithms to implement two-user monotone chain rule based successive cancellation decoder. Similarly, we modify the data structures to accommodate two-user probabilities and bit decisions. We first give an efficient implementation of SC decoder and then go on to describing the list decoder.

For algorithmic purposes we use the notation in [40]. The code block length in consideration is given by $N = 2^n$. Unlike previous treatment, here vector indices start at zero. User 1, 2 and side information vectors are represented by u_0^{N-1}, v_0^{N-1} and z_0^{N-1}, respectively. Also we use the notation $P_n^{(i,j)}(u_i, v_j|z_0^{N-1}, u_0^{i-1}, v_0^{j-1})$ instead of $P_N^{(i+1,j+1)}(u_i, v_j|z_0^{N-1}, u_0^{i-1}, v_0^{j-1})$ defined in (4.16). We give the high-level algorithm for SC decoding in Algorithm 1. In each step k of the algorithm we

calculate four probabilities of the form $P_n^{(i,j)}(a, b|z_0^{N-1}, u_0^{i-1}, v_0^{j-1})$ for $a, b \in \{0, 1\}$. The exact form of the probability that needs to be calculated depends on the path. Then, using the calculated probabilities we make a decision for either user 1 or user 2, depending on the path.

Algorithm 1: High-level description of the two-user SC source decoder

input : Received vectors $u_{\tilde{\mathcal{A}}_1^c}$, $v_{\tilde{\mathcal{A}}_2^c}$, side information z^N and decoding path b^{2N}
output: Decoded user bits $(\hat{u}^N, \hat{v}^N)$

```
// Initialization
i ← −1, j ← −1
// Main Loop
for k = 0, ..., 2N − 1 do
    if b_k = 0 then
        // Horizontal step
        i ← i + 1
        calculate P_n^(i,j+1)(a, b|z_0^{N−1}, û_0^{i−1}, v̂_0^j)
        set P_n^u[i][a] ← Σ_b P_n^(i,j+1)(a, b|z_0^{N−1}, û_0^{i−1}, v̂_0^j)
        if i ∈ Ã_1^c then
            set û_i to the received value u_i
        else
            if P_n^u[i][1] > P_n^u[i][0] then
                set û_i ← 1
            else
                set û_i ← 0
    else
        // Vertical step
        j ← j + 1
        calculate P_n^(i+1,j)(a, b|z_0^{N−1}, û_0^i, v̂_0^{j−1})
        set P_n^v[j][b] ← Σ_a P_n^(i+1,j)(a, b|z_0^{N−1}, û_0^i, v̂_0^{j−1})
        if j ∈ Ã_2^c then
            set v̂_j to the received value v_j
        else
            if P_n^v[j][1] > P_n^v[j][0] then
                set v̂_j ← 1
            else
                set v̂_j ← 0
return (û^N, v̂^N)
```

We follow the footsteps of [40] and write the equations (4.22) through (4.25) in slightly different form. First we define the following variables. $0 \leq \lambda \leq n$ denotes the *layer* for which there are $2^{n-\lambda}$ independent *blocks* of size $\Lambda = 2^\lambda$. We denote the phases of users 1 and 2 with φ_0 and φ_1, respectively. We combine the *phases*

of users 1 and 2 into a two-tuple $\varphi = (\varphi_0, \varphi_1)$. We also use the notation φ_p for $p = 0, 1$ to address the individual elements of the two-tuple. As before, we use variables i and j to index u_0^{N-1} and v_0^{N-1}, respectively. Thus, for layer n we have $\varphi_0 = i$ and $\varphi_1 = j$. Note that for layer λ we have $0 \leq \varphi_0, \varphi_1 < \Lambda$. We write the following four recursive formulas using the new definitions.

$$P_\lambda^{(2\psi_0,2\psi_1)}(u_{2\psi_0}, v_{2\psi_1}|z_0^{\Lambda-1}, u^{2\psi_0-1}, v^{2\psi_1-1}) = (1/C_1)\cdot$$
$$\sum_{\substack{u_{2\psi_0+1} \\ v_{2\psi_1+1}}} P_{\lambda-1}^{(\psi)}(u_{2\psi_0} + u_{2\psi_0+1}, v_{2\psi_1} + v_{2\psi_1+1}|z_0^{\Lambda/2-1}, u_{0,e}^{2\psi_0-1} + u_{0,o}^{2\psi_0-1}, v_{0,e}^{2\psi_1-1} + v_{0,o}^{2\psi_1-1})$$
$$\cdot P_{\lambda-1}^{(\psi)}(u_{2\psi_0+1}, v_{2\psi_1+1}|z_{\Lambda/2}^{\Lambda-1}, u_{0,o}^{2\psi_0-1}, v_{0,o}^{2\psi_1-1}). \quad (4.30)$$

$$P_\lambda^{(2\psi_0+1,2\psi_1)}(u_{2\psi_0+1}, v_{2\psi_1}|z_0^{\Lambda-1}, u^{2\psi_0}, v^{2\psi_1-1}) = (1/C_2)\cdot$$
$$\sum_{v_{2\psi_1+1}} P_{\lambda-1}^{(\psi)}(u_{2\psi_0} + u_{2\psi_0+1}, v_{2\psi_1} + v_{2\psi_1+1}|z_0^{\Lambda/2-1}, u_{0,e}^{2\psi_0-1} + u_{0,o}^{2\psi_0-1}, v_{0,e}^{2\psi_1-1} + v_{0,o}^{2\psi_1-1})$$
$$\cdot P_{\lambda-1}^{(\psi)}(u_{2\psi_0+1}, v_{2\psi_1+1}|z_{\Lambda/2}^{\Lambda-1}, u_{0,o}^{2\psi_0-1}, v_{0,o}^{2\psi_1-1}). \quad (4.31)$$

$$P_\lambda^{(2\psi_0,2\psi_1+1)}(u_{2\psi_0}, v_{2\psi_1+1}|z_0^{\Lambda-1}, u^{2\psi_0-1}, v^{2\psi_1}) = (1/C_3)\cdot$$
$$\sum_{u_{2\psi_0+1}} P_{\lambda-1}^{(\psi)}(u_{2\psi_0} + u_{2\psi_0+1}, v_{2\psi_1} + v_{2\psi_1+1}|z_0^{\Lambda/2-1}, u_{0,e}^{2\psi_0-1} + u_{0,o}^{2\psi_0-1}, v_{0,e}^{2\psi_1-1} + v_{0,o}^{2\psi_1-1})$$
$$\cdot P_{\lambda-1}^{(\psi)}(u_{2\psi_0+1}, v_{2\psi_1+1}|z_{\Lambda/2}^{\Lambda-1}, u_{0,o}^{2\psi_0-1}, v_{0,o}^{2\psi_1-1}). \quad (4.32)$$

$$P_{\lambda}^{(2\psi_0+1,2\psi_1+1)}(u_{2\psi_0+1}, v_{2\psi_1+1}|z_0^{\Lambda-1}, u^{2\psi_0}, v^{2\psi_1}) = (1/C_4)\cdot$$
$$P_{\lambda-1}^{(\psi)}(u_{2\psi_0} + u_{2\psi_0+1}, v_{2\psi_1} + v_{2\psi_1+1}|z_0^{\Lambda/2-1}, u_{0,e}^{2\psi_0-1} + u_{0,o}^{2\psi_0-1}, v_{0,e}^{2\psi_1-1} + v_{0,o}^{2\psi_1-1})$$
$$\cdot P_{\lambda-1}^{(\psi)}(u_{2\psi_0+1}, v_{2\psi_1+1}|z_{\Lambda/2}^{\Lambda-1}, u_{0,o}^{2\psi_0-1}, v_{0,o}^{2\psi_1-1}). \quad (4.33)$$

Scaling constants C_m, $m = 1, 2, 3, 4$ are calculated as in (4.18). They are calculated using the same probabilities $P_{\lambda-1}^{(\psi)}$. To calculate probabilities for level λ we need to evaluate $P_{\lambda-1}^{(\psi)}$ with observation $(z_0^{\Lambda/2-1}, u_{0,e}^{2\psi_0-1}+u_{0,o}^{2\psi_0-1}, v_{0,e}^{2\psi_1-1}+v_{0,o}^{2\psi_1-1})$ and $(z_{\Lambda/2}^{\Lambda-1}, u_{0,o}^{2\psi_0-1}, v_{0,o}^{2\psi_1-1})$. Therefore, a *branch* number $0 \leq \beta < 2^{n-\lambda}$ is defined to keep track of this. At top layer $\lambda = n$ there is a single branch $\beta = 0$. For each layer $0 \leq \lambda \leq n$, consider probability $P_{\lambda}^{(\varphi)}$ corresponding to branch β with observation $(z_0^{\Lambda-1}, u_0^{\varphi_0-1}, v_0^{\varphi_1-1})$. Denote $\psi = \lfloor \varphi/2 \rfloor$. $P_{\lambda}^{(\varphi)}$ is calculated from probability $P_{\lambda-1}^{(\psi)}$ at layer $\lambda-1$ evaluated with observations $(z_0^{\Lambda/2-1}, u_{0,e}^{2\psi_0-1}+u_{0,o}^{2\psi_0-1}, v_{0,e}^{2\psi_1-1}+v_{0,o}^{2\psi_1-1})$ and $(z_{\Lambda/2}^{\Lambda-1}, u_{0,o}^{2\psi_0-1}, v_{0,o}^{2\psi_1-1})$. They are assigned branch numbers 2β and $2\beta+1$, respectively. The calculations descend down the layers recursively in this fashion.

For each layer λ we define the *probabilities array* structure denoted as P_{λ} indexed by $0 \leq k < 2N$. For a given layer λ, index k will correspond to $0 \leq \varphi_0, \varphi_1 < \Lambda$ and $0 \leq \beta < 2^{n-\lambda}$ as

$$k = \langle \varphi, \beta \rangle_{\lambda} = \varphi_0 + \varphi_1 + 2 \cdot 2^{\lambda} \cdot \beta. \quad (4.34)$$

If for layer λ, we denote the observation corresponding to branch β as $(z_0^{\Lambda-1}, u_0^{\varphi_0-1}, v_0^{\varphi_1-1})$, then probability array will hold for all values of $(a, b) \in \{0, 1\} \times \{0, 1\}$ that

$$P_{\lambda}[\langle \varphi, \beta \rangle][a, b] = P_{\lambda}^{(\varphi)}(a, b|z_0^{\Lambda-1}, u_0^{\varphi_0-1}, v_0^{\varphi_1-1}). \quad (4.35)$$

Also we need to keep track of the decided bit values. We use *bit array* structure B_{λ} indexed by $0 \leq k < 2N$ for each layer λ. Denote the user bits of $P_{\lambda}^{(\varphi)}$ at branch

β, layer λ and phase φ with $\hat{u}(\lambda, \varphi_0, \beta)$ and $\hat{v}(\lambda, \varphi_1, \beta)$. Then we have

$$B_\lambda[\langle \varphi, \beta \rangle][0] = \hat{u}(\lambda, \varphi_0, \beta) \tag{4.36}$$

$$B_\lambda[\langle \varphi, \beta \rangle][1] = \hat{v}(\lambda, \varphi_1, \beta). \tag{4.37}$$

Algorithm 2: Two-user SC decoder main loop

input : Received vectors $u_{\tilde{\mathcal{A}}_1^c}$, $v_{\tilde{\mathcal{A}}_2^c}$, side information z_0^{N-1} and decoding path b_0^{2N-1}
output: Decoded codewords matrix $\hat{\mathbf{c}}$

```
// Initialization
```
Generate $s_{\mathcal{A}^c}$ from $(u_{\tilde{\mathcal{A}}_1^c}, v_{\tilde{\mathcal{A}}_2^c})$
for $\beta = 0, 1, \ldots, N-1$ **do**
    $P_0[\beta][0,0] \leftarrow P_{XY|Z}(0,0|z_\beta)$, $P_0[\beta][0,1] \leftarrow P_{XY|Z}(0,1|z_\beta)$
    $P_0[\beta][1,0] \leftarrow P_{XY|Z}(1,0|z_\beta)$, $P_0[\beta][1,1] \leftarrow P_{XY|Z}(1,1|z_\beta)$
$\varphi' \leftarrow (-1, -1)$
```
// Main Loop
```
for $k = 0, \ldots, 2N-1$ **do**
    $p \leftarrow \mathbf{b}[k]$
    $\varphi'_p \leftarrow \varphi'_p + 1$
    $\varphi \leftarrow \varphi'$
    if $\varphi_{\bar{p}} = -1$ **then**
        $\varphi_{\bar{p}} \leftarrow 0$
    if $k = 0$ **or** $\varphi_p > 0$ **then**
        `recursivelyCalcP`(n, φ, p)
    `calcΓ`(φ', p)
    if $k \in \mathcal{A}^c$ **then**
        $C_\lambda[\beta][\varphi_p \mod 2, p] \leftarrow s_k$
    else
        if $\Gamma[1] > \Gamma[0]$ **then**
            $C_\lambda[\beta][\varphi_p \mod 2, p] \leftarrow 1$
        else
            $C_\lambda[\beta][\varphi_p \mod 2, p] \leftarrow 0$
    if $\varphi_p \bmod 2 = 1$ **then**
        `recursivelyUpdateC`(n, φ, p)
return $\hat{\mathbf{c}} \leftarrow (C_0[\beta][0,0], C_0[\beta][0,1])|_{\beta=0}^{N-1}$

The space complexity is $O(N \log N)$ with data structures stated as above. However, as noted in [40], we do not need to keep all of the values at each level and the space complexity can be reduced to $O(N)$. For probabilities array $P_\lambda^{(\varphi)}$ we need to keep only one location per branch. The phase information is not needed. Thus, we replace $P_\lambda^{(\varphi)}[\langle \varphi, \beta \rangle]$ with $P_\lambda^{(\varphi)}[\beta]$. The four probabilities for

Algorithm 3: calcΓ (φ', p)

```
if φ'_p̄ = −1 then
    for u' = 0, 1 do
        if p = 0 then
            Γ[u'] ← Σ_{u''} P_n[0][u', u'']
        else
            Γ[u'] ← Σ_{u''} P_n[0][u'', u']
else
    for u' = 0, 1 do
        if p = 0 then
            Γ[u'] ← (1/D_1) P_n[0] [u', C_n[0][φ'_p̄ mod 2, p̄]]
        else
            Γ[u'] ← (1/D_2) P_n[0] [C_n[0][φ'_p̄ mod 2, p̄], u']
```

Algorithm 4: recursivelyCalcP(λ, φ, p)

```
// Stopping condition
if λ = 0 then return
ψ ← ⌊φ/2⌋
// Recurse if necessary
if φ_p mod 2 = 0 then recursivelyCalcP(λ − 1, ψ, p)
// Perform calculation
for β = 0, 1, ..., 2^{n−λ} − 1 do
    if φ_0 mod 2 = 0 then // Even u index
        if φ_1 mod 2 = 0 then // Even v index
            // Apply equation 4.30
            for (u', v') ∈ {0, 1} × {0, 1} do
                P_λ[β][u', v'] ← (1/C_1) Σ_{u'',v''} P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β + 1][u'', v'']
        else // Odd v index
            // Apply equation 4.31
            v' ← C_λ[β][0, 1]
            for (u', v'') ∈ {0, 1} × {0, 1} do
                P_λ[β][u', v''] ← (1/C_3) Σ_{u''} P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β + 1][u'', v'']
    else // Odd u index
        u' ← C_λ[β][0, 0]
        if φ_1 mod 2 = 0 then // Even v index
            // Apply equation 4.32
            for (u'', v') ∈ {0, 1} × {0, 1} do
                P_λ[β][u'', v'] ← (1/C_2) Σ_{v''} P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β + 1][u'', v'']
        else // Odd v index
            // Apply equation 4.33
            v' ← C_λ[β][0, 1]
            for (u'', v'') ∈ {0, 1} × {0, 1} do
                P_λ[β][u'', v''] ← (1/C_4) P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β + 1][u'', v'']
```

Algorithm 5: recursivelyUpdateC(λ, φ, p)

```
   // Stopping condition
1  if λ = 0 then return
2  ψ ← ⌊φ/2⌋
3  for β = 0, 1, ..., 2^{n−λ} − 1 do
4  |  C_{λ−1}[2β][ψ_p mod 2, p] ← C_λ[β][0, p] ⊕ C_λ[β][1, p]
5  |  C_{λ−1}[2β + 1][ψ_p mod 2, p] ← C_λ[β][1, p]
6  if ψ_p mod 2 = 1 then
7  |  recursivelyUpdateC(λ − 1, ψ, p)
```

each value of the user bits $(a, b) \in \{0, 1\} \times \{0, 1\}$ are stored as $P_\lambda[\beta][a, b]$. Thus at each layer λ, probabilities array stores $4 \cdot 2^{n-\lambda}$ floating values. The total space requirement becomes $4 \cdot (N - 1)$. Similarly, for bit array structure we need to keep two locations per branch as noted in [40]. Here phase information cannot be completely thrown away but only the parity of the phase is needed. We need to keep two separate locations corresponding to users. Denoting the user with variable $p \in \{0, 1\}$, we replace $B_\lambda[\langle \varphi, \beta \rangle]$ with $C_\lambda[\beta][\varphi_p \mod 2, p]$. Thus, for user 1 ($p = 0$) and 2 ($p = 1$) we keep two locations for bit values corresponding to even and odd parity of that user's phase.

Algorithms 2 through 5 completely describe the two-user monotone chain rule based SC polar decoder. Algorithm 2 presents the main loop. The algorithm accepts outputs of encoders $u_{\tilde{\mathcal{A}}_1^c}$, $v_{\tilde{\mathcal{A}}_2^c}$ and side information z_0^{N-1} as data inputs. In addition, decoding path b_0^{2N-1} is supplied as a *configuration* input. First, layer 0 of probabilities array is initialized with the probabilities from the conditional distribution $P_{XY|Z}$ under observation of side-information vector z_0^{N-1}. Main loop runs over $2N$ steps on the decoding path. At each step either a bit from u_0^{N-1} or v_0^{N-1} is decoded. The algorithm makes the necessary probability calculation to decide the next user bit. The probabilities calculated by "recursivelyCalcP" function recursively are of the form $P_n^{(\varphi)}(a, b|z_0^{N-1}, u_0^{\varphi_0-1}, v_0^{\varphi_1-1})$ which are functions of two variables. However, we need to calculate a second probability from these which is function of single variable which is the variable we are interested in at this step. That calculation is done in "calcΓ" function in Algorithm 3. After probability calculation, a decision is made on the user bit value. If that bit value is received from encoders the value is inserted here. Otherwise, a decision is made

based on the calculated probability. For each user after odd indexed bit value is calculated, the decisions are propagated recursively down to the lower layers by "recursivelyUpdateC" function.

For example, let $b_0 = 0$, then at the first step of algorithm we actually need $\Pr[U_0 = a|Z_0^{N-1} = z_0^{N-1}]$. To calculate that probability, the algorithm first calculates $P_n^{(0,0)}(a, b|z_0^{N-1})$ and then sums over b. As another example, at step 11 of the algorithm assume that $b_{10} = 1$ and we need to calculate $\Pr[V_3 = b|Z_0^{N-1} = z_0^{N-1}, U_0^6 = \hat{u}_0^6, V_0^2 = \hat{v}_0^2]$. The algorithm first calculates $P_n^{(6,3)}(a, b|z_0^{N-1}, \hat{u}_0^5, \hat{v}_0^2)$ and then gets the result as

$$\Pr[V_3 = b|Z_0^{N-1} = z_0^{N-1}, U_0^6 = \hat{u}_0^6, V_0^2 = \hat{v}_0^2] = \frac{P_n^{(6,3)}(\hat{u}_6, b|z_0^{N-1}, \hat{u}_0^5, \hat{v}_0^2)}{\sum_{b'} P_n^{(6,3)}(\hat{u}_6, b'|z_0^{N-1}, \hat{u}_0^5, \hat{v}_0^2)}. \tag{4.38}$$

"recursivelyCalcP" function is an extension of the one in [40]. Note that it has four different recursive equations corresponding to indices of two users being odd or even. The normalization constants in the formulas are given as

$$\begin{aligned}
C_1 &= 1, \\
C_2 &= \sum_{u'',v',v''} P_{\lambda-1}[2\beta][u' \oplus u'', v' \oplus v''] \cdot P_{\lambda-1}[2\beta+1][u'', v''], \\
C_3 &= \sum_{u',u'',v''} P_{\lambda-1}[2\beta][u' \oplus u'', v' \oplus v''] \cdot P_{\lambda-1}[2\beta+1][u'', v''], \\
C_4 &= \sum_{u'',v''} P_{\lambda-1}[2\beta][u' \oplus u'', v' \oplus v''] \cdot P_{\lambda-1}[2\beta+1][u'', v''].
\end{aligned}$$

The normalization constants in function "calcΓ" are given as

$$\begin{aligned}
D_1 &= \sum_{u'} P_n[0]\left[u', C_n[0][\varphi'_{\bar{p}} \bmod 2, \bar{p}]\right], \\
D_2 &= \sum_{u'} P_n[0]\left[C_n[0][\varphi'_{\bar{p}} \bmod 2, \bar{p}], u'\right].
\end{aligned}$$

"recursivelyUpdateC" function updates the bits array recursively as new bit decisions are made at layer n.

4.2.2.1 List Decoder

In this section, we give explicit algorithms for two-user SC list (SCL) decoder based on monotone chain rules. The algorithms given here are also generalizations of list decoder algorithms by Tal and Vardy [40] to two-user setting. We keep the general structure and the names of the algorithms intact. We modify the inside of the algorithms to implement two-user monotone chain rule based SC decoder. We already explained the two-user generalization of SC decoder in detail. In this section, two-user SC decoder is modified to explore L parallel decoding paths. We only make the necessary modifications to the SC decoder to achieve an efficient list decoder which was explained in detail in [40]. Thus, we do not go into details of describing how to implement efficient list version of a SC decoder of polar codes.

The list decoder accepts list size L in addition to the parameters accepted by two-user SC decoder described before. Increasing the list size both increases the effectiveness and complexity of the decoder. SCL decoder keeps a list of decoding paths to increase the effectiveness of the decoder. At each step of the SC decoder for an unfrozen bit, a decision is made for either $\hat{u}_{\varphi_0}$ or $\hat{v}_{\varphi_1}$. List decoder attempts to split the decoding path to inspect both options so that it does not discard the less likely path at that step of the algorithm. However, this means that the number of paths to be inspected grows exponentially. Therefore, a limit to the number of decoding paths the decoder can hold is set as the list size L. The decoder prunes the paths at each step according to their probability and keeps the list size less than or equal to L.

Algorithm 6: Two–user SCL decoder

input : Received vectors $u_{\tilde{\mathcal{A}}_1^c}$, $v_{\tilde{\mathcal{A}}_2^c}$, side information z_0^{N-1}, decoding path b_0^{2N-1} and list size L

output: Decoded codewords matrix $\hat{\mathbf{c}}$

`// Initialization`

$l \leftarrow$ `assignInitialPath()`

$P_0 \leftarrow$ `getArrayPointerP(0,`l`)`

for $\beta = 0, 1, \ldots, N-1$ **do**

  $P_0[\beta][0,0] \leftarrow P_{XY|Z}(0,0|z_\beta)$, $P_0[\beta][0,1] \leftarrow P_{XY|Z}(0,1|z_\beta)$

  $P_0[\beta][1,0] \leftarrow P_{XY|Z}(1,0|z_\beta)$, $P_0[\beta][1,1] \leftarrow P_{XY|Z}(1,1|z_\beta)$

$\varphi' \leftarrow [-1, -1]$

`// Main Loop`

for $k = 0, 1, \ldots, 2N-1$ **do**

  $p \leftarrow \mathbf{b}[k]$

  $\varphi'_p \leftarrow \varphi'_p + 1$

  $\varphi \leftarrow \varphi'$

  if $\varphi_{\bar{p}} = -1$ **then**

    $\varphi_{\bar{p}} \leftarrow 0$

  if $k = 0$ **or** $\varphi_p > 0$ **then**

    `recursivelyCalcP(`n, φ, p`)`

    `calcΓ(`φ', p`)`

  if φ_p *is frozen index* **then**

    `continuePathsFrozenBit(`φ, p`)`

  else

    `continuePathsUnfrozenBit(`φ, p`)`

  if $\varphi_p \bmod 2 = 1$ **then**

    `recursivelyUpdateC(`n, φ, p`)`

`// Find the best codeword`

$l \leftarrow$ `findBestCodeWord()`

$C_0 \leftarrow$ `getArrayPointerC(`$0, l$`)`

return $\hat{\mathbf{c}} \leftarrow (C_0[\beta][0,0], C_0[\beta][0,1])|_{\beta=0}^{N-1}$

Algorithm 7: calcΓ (φ', p)

```
for l = 0, 1, ..., L − 1 do
    if pathIndexInactive(l) then
        continue
    P_n ← getArrayPointerP(n, l)
    C_n ← getArrayPointerC(n, l)
    if φ'_p̄ = −1 then
        for u' = 0, 1 do
            if p = 0 then
                Γ_l[u'] ← Σ_{u''} P_n[0][u', u'']
            else
                Γ_l[u'] ← Σ_{u''} P_n[0][u'', u']
    else
        for u' = 0, 1 do
            if p = 0 then
                Γ_l[u'] ← P_n[0][u', C_n[0][φ'_p̄ mod 2, p̄]]
            else
                Γ_l[u'] ← P_n[0][C_n[0][φ'_p̄ mod 2, p̄], u']
```

Algorithm 8: findBestCodeWord()

```
// Find the best codeword
l' ← 0, P' ← 0
for l = 0, 1, ..., L − 1 do
    if pathIndexInactive(l) then
        continue
    C_n ← getArrayPointerC(n, l)
    if P' < Γ_l[C_n[0][1, p]] then
        l' ← l, P' ← Γ_l[C_n[0][1, p]]
return l'
```

Algorithm 9: continuePathsFrozenBit(φ, p)

```
for l = 0, 1, ..., L − 1 do
    if pathIndexInactive(l) then continue

    C_n ← getArrayPointerC(n, l)
    C_n[0][φ_p mod 2, p] ← frozen value of user p index φ_p
```

Algorithm 10: continuePathsUnfrozenBit(φ, p)

```
forksArray ← new (float,bit,index) – triplets of size 2L
i ← 0
// populate forksArray
for l = 0, 1, ..., L − 1 do
    if pathIndexInactive(l) then continue
    P_n ← getArrayPointerP(n, l)
    forksArray[2i] ← (Γ_l[0], 0, l)
    forksArray[2i + 1] ← (Γ_l[1], 1, l)
    i ← i + 1
// Pivot forksArray
ρ ← min(2i, L)
rearrange forksArray such that ∀α < ρ and ∀β ≥ ρ we have
forksArray[α][0] ≥ forksArray[β][0]
// Pick the best rho forks
contForks ← new(boolean, boolean) – pairs of size L
for r = 0, 1, ..., ρ − 1 do
    l ← forksArray[r][2]
    b ← forksArray[r][1]
    contForks[l][b] ← true
// Kill-off non-continuing paths
for l = 0, 1, ..., L − 1 do
    if pathIndexInactive(l) then continue
    if contForks [l][0]=false and contForks [l][1]=false then
        killPath(l)
// Continue relevant paths and duplicate if necessary
for l = 0, 1, ..., L − 1 do
    if contForks [l][0]=false and contForks [l][1]=false then    // both forks are bad
        continue
    C_n ← getArrayPointerC(n, l)
    if contForks [l][0]=true and contForks [l][1]=true then   // both forks are good
        C_n[0][φ_p mod 2, p] ← 0
        l' ← clonePath(l)
        C_n ← getArrayPointerC(n, l')
        C_n[0][φ_p mod 2, p] ← 1
    else                                                  // exactly one fork is good
        if contForks [l][0]=true then
            C_n[0][φ_p mod 2, p] ← 0
        else
            C_n[0][φ_p mod 2, p] ← 1
```

Algorithm 11: recursivelyCalcP(λ, φ, p)

```
   // Stopping condition
1  if λ = 0 then return
2  ψ ← ⌊φ/2⌋
   // Recurse if necessary
3  if φ_p mod 2 = 0 then recursivelyCalcP(λ − 1, ψ, p)
   // Perform calculation
4  for l = 0, 1, ..., L − 1 do
5      if pathIndexInactive(l) then continue
6
7      P_λ ← getArrayPointerP(λ,l)
8      P_{λ−1} ← getArrayPointerP(λ − 1,l)
9      C_λ ← getArrayPointerC(λ,l)
10     for β = 0, 1, ..., 2^{n−λ} − 1 do
11         if φ_0 mod 2 = 0 then // Even u index
12             if φ_1 mod 2 = 0 then // Even v index
13                 for (u', v') ∈ {0,1} × {0,1} do
14                     P_λ[β][u', v'] ← Σ_{u'',v''} P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β+1][u'', v'']
15             else // Odd v index
16                 v' ← C_λ[β][0,1]
17                 for (u', v'') ∈ {0,1} × {0,1} do
18                     P_λ[β][u', v''] ← Σ_{u''} P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β+1][u'', v'']
19         else // Odd u index
20             u' ← C_λ[β][0,0]
21             if φ_1 mod 2 = 0 then // Even v index
22                 for (u'', v') ∈ {0,1} × {0,1} do
23                     P_λ[β][u'', v'] ← Σ_{v''} P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β+1][u'', v'']
24             else // Odd v index
25                 v' ← C_λ[β][0,1]
26                 for (u'', v'') ∈ {0,1} × {0,1} do
27                     P_λ[β][u'', v''] ← P_{λ−1}[2β][u' ⊕ u'', v' ⊕ v''] · P_{λ−1}[2β+1][u'', v'']
```

Algorithm 12: recursivelyUpdateC(λ, φ, p)

```
1  for l = 0, 1, ..., L − 1 do
2      if pathIndexInactive(l) then continue
3
4      C_λ ← getArrayPointerC(λ,l)
5      C_{λ−1} ← getArrayPointerC(λ − 1,l)
6      ψ ← ⌊φ/2⌋
7      for β = 0, 1, ..., 2^{n−λ} − 1 do
8          C_{λ−1}[2β][ψ_p mod 2, p] ← C_λ[β][0,p] ⊕ C_λ[β][1,p]
9          C_{λ−1}[2β+1][ψ_p mod 2, p] ← C_λ[β][1,p]
10     if ψ_p mod 2 = 1 then
11         recursivelyUpdateC(λ − 1, ψ, p, l)
```

Algorithm 13: getArrayPointerP(λ, l)

```
1  s ← pathIndexToArrayIndex[λ][l]
   // Copy array if necessary
2  if arrayReferenceCount[λ][s] > 1 then
3      s′ ← pop(inactiveArrayIndices)
4      P_λ ← arrayPointerP[λ][s], P′_λ ← arrayPointerP[λ][s′]
5      C_λ ← arrayPointerC[λ][s], C′_λ ← arrayPointerC[λ][s′]
6      for β = 0, 1, ..., 2^(n−λ) do
7          for (i, j) ∈ {0, 1} × {0, 1} do
8              P′_λ[β][i, j] ← P_λ[β][i, j]
9              C′_λ[β][i, j] ← C_λ[β][i, j]
10     arrayReferenceCount[λ][s] ← arrayReferenceCount[λ][s] − 1
11     arrayReferenceCount[λ][s′] ← 1
12     pathIndexToArrayIndex[λ][l] ← s′
13     s ← s′
   // Return array pointer
14 return arrayPointerP[λ][s]
```

Algorithm 14: getArrayPointerC(λ, l)

```
1  s ← pathIndexToArrayIndex[λ][l]
   // Copy array if necessary
2  if arrayReferenceCount[λ][s] > 1 then
3      s′ ← pop(inactiveArrayIndices)
4      P_λ ← arrayPointerP[λ][s], P′_λ ← arrayPointerP[λ][s′]
5      C_λ ← arrayPointerC[λ][s], C′_λ ← arrayPointerC[λ][s′]
6      for β = 0, 1, ..., 2^(n−λ) do
7          for (i, j) ∈ {0, 1} × {0, 1} do
8              P′_λ[β][i, j] ← P_λ[β][i, j]
9              C′_λ[β][i, j] ← C_λ[β][i, j]
10     arrayReferenceCount[λ][s] ← arrayReferenceCount[λ][s] − 1
11     arrayReferenceCount[λ][s′] ← 1
12     pathIndexToArrayIndex[λ][l] ← s′
13     s ← s′
   // Return array pointer
14 return arrayPointerC[λ][s]
```

Algorithm 15: killPath(l)

```
  // First, free arrays referenced by this path
1 for λ = 0, 1, ..., n do
2     s ← pathIndexToArrayIndex[λ][l]
3     arrayReferenceCount[λ][s] ← arrayReferenceCount[λ][s] − 1
4     if arrayReferenceCount[λ][s] = 0 then
5         push (inactiveArrayIndices, s)
  // Then, kill path
6 activePath[l] ← false
7 push (inactivePathIndices, l)
```

Algorithm 16: clonePath(l)

```
  // First, get a free path and activate it
1 l′ ← pop(inactivePathIndices)
2 activePath[l′] ← true
  // Then, just copy references not the actual data (lazy copy)
3 for λ = 0, 1, ..., n do
4     s ← pathIndexToArrayIndex[λ][l]
5     pathIndexToArrayIndex[λ][l′] ← s
6     arrayReferenceCount[λ][s] ← arrayReferenceCount[λ][s] + 1
```

4.2.3 Simulations

In this section, we present simulation results showing the performance of SCL decoder used in Slepian–Wolf distributed source coding problem. In this simulation, the probability distribution of source is given by

$$p_{XY} = \begin{bmatrix} 0.1286 & 0.0175 \\ 0.0175 & 0.8364 \end{bmatrix}.$$

This distribution results in the following entropies:

$$H(X) = H(Y) = 0.6, \qquad H(X|Y) = H(Y|X) = 0.2.$$

Thus the SW achievable rate region is

$$\{(R_x, R_y) : \ R_x \geq 0.2, \ R_y \geq 0.2, \ R_x + R_y \geq 0.8\}.$$

The capacity region is shown in Figure 4.2 along with simulation results. We adjust the rates of the codes on straight lines yielding operating rate pairs $(R_x^A, R_y^A) = \rho_A \cdot (0.4, 0.4)$, $(R_x^B, R_y^B) = \rho_B \cdot (0.5, 0.3)$ and $(R_x^C, R_y^C) = \rho_C \cdot (0.6, 0.2)$ with $\rho_A, \rho_B, \rho_C \geq 1$ for code classes A, B and C, respectively. The markings show the points where the block error rate (BLER) falls down to 10^{-4}. The list size L used in these simulations is 32.

Figures 4.3 thru 4.5 show the detailed performance results for code classes C, B and A, respectively. We see how the performance increases as we allow more sum rate.

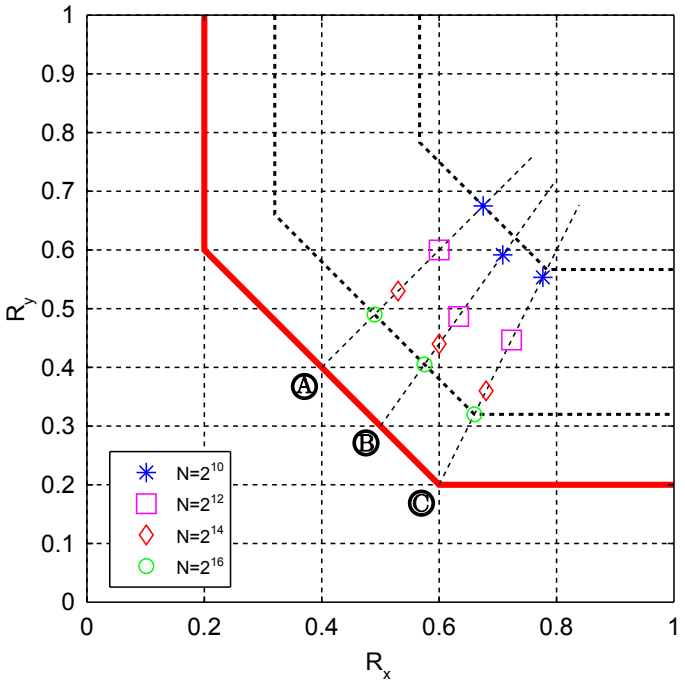


Figure 4.2: 10^{-4} BLER marked on SW region for n=10,12,14,16 (L=32).

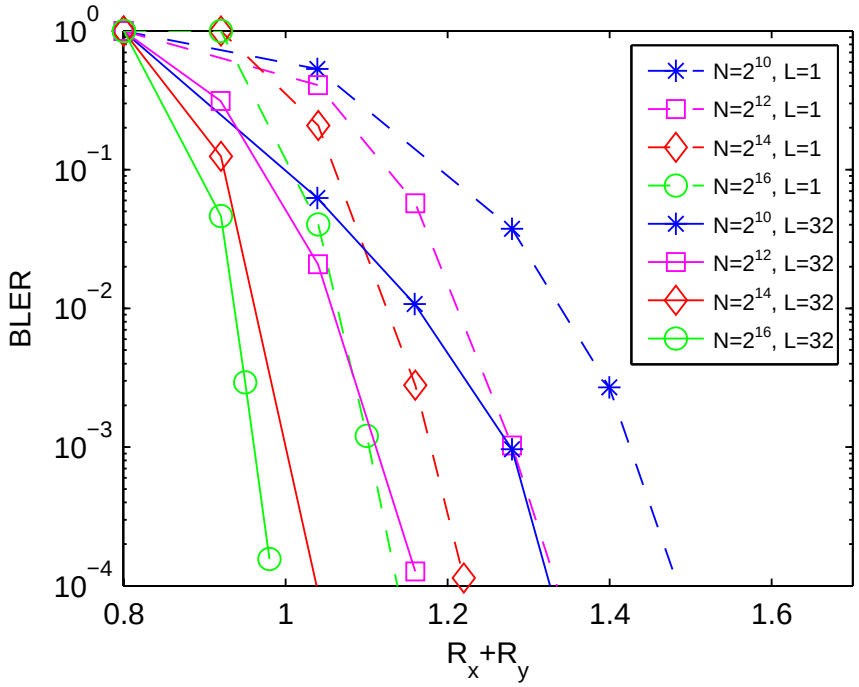


Figure 4.3: Rate point C $(0.6, 0.2)$.

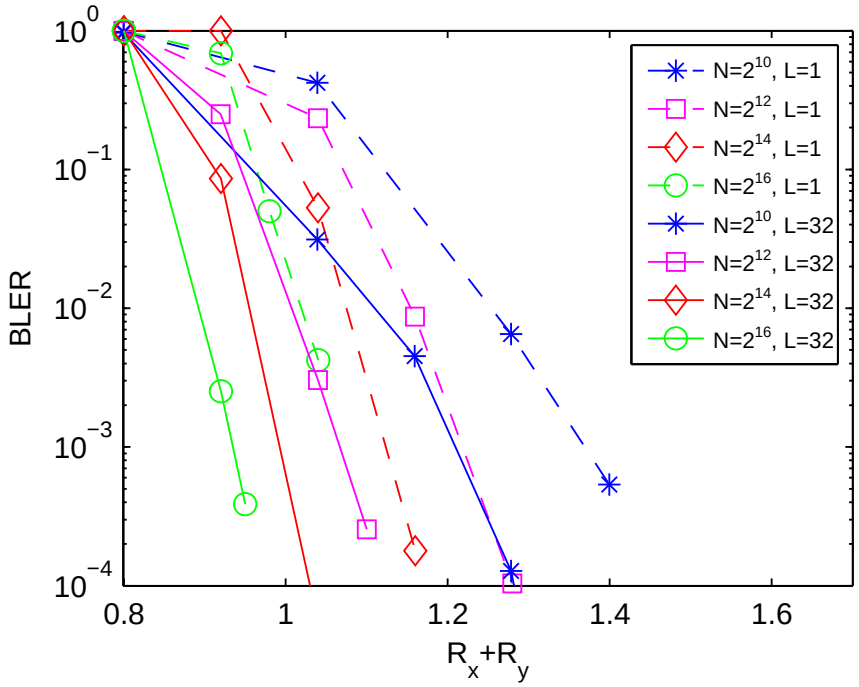


Figure 4.4: Rate point B $(0.5, 0.3)$.

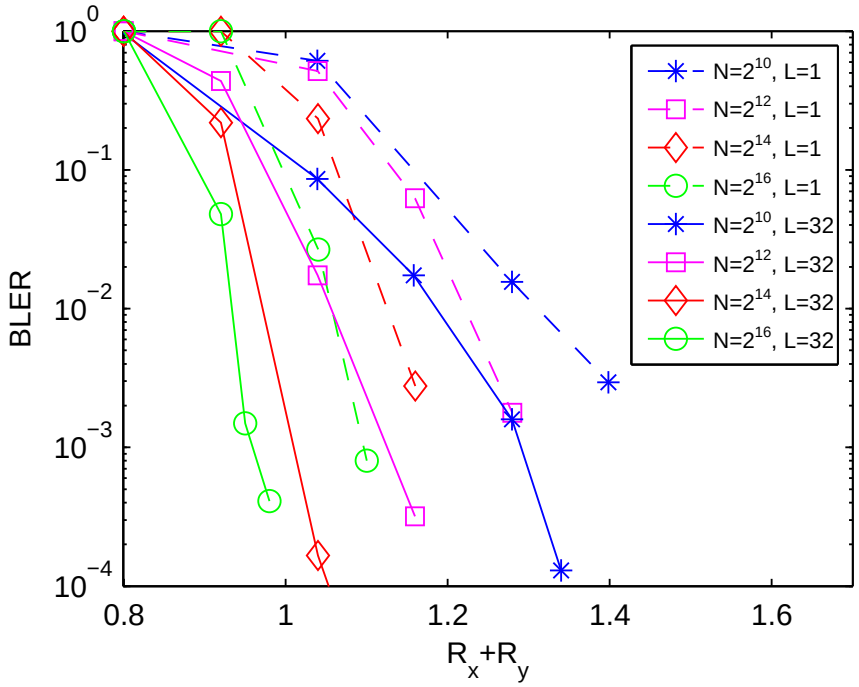


Figure 4.5: Rate point A $(0.4, 0.4)$.

4.3 Multiple-Access Channel

A two-user multiple-access channel (MAC) communication setup is depicted in Figure 4.6. The setup consists of two independent users trying to communicate to a common receiver. A *discrete memoryless* MAC consists of three alphabets $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$, and a probability transition matrix $p(z|x,y)$.

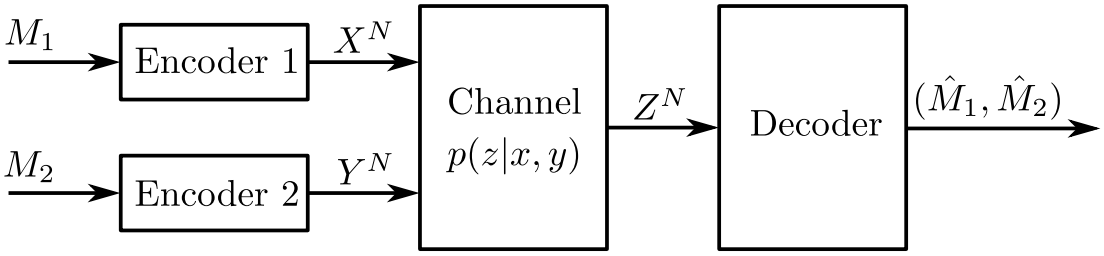


Figure 4.6: Multiple-access channel communication setup.

Definition 11. *A* $((2^{NR_1}, 2^{NR_2}), N)$ *code for MAC consists of two sets of integers* $\mathcal{M}_1 = \{1, 2, \ldots, 2^{NR_1}\}$ *and* $\mathcal{M}_2 = \{1, 2, \ldots, 2^{NR_2}\}$*, called the* message sets*, two* encoding functions*,*

$$f_1 : \quad \mathcal{M}_1 \to \mathcal{X}^N \tag{4.39}$$

and

$$f_2 : \quad \mathcal{M}_2 \to \mathcal{Y}^N, \tag{4.40}$$

and a decoding function*,*

$$g : \quad \mathcal{Z}^N \to \mathcal{M}_1 \times \mathcal{M}_2. \tag{4.41}$$

Sender 1 chooses a message index M_1 *uniformly* from $\mathcal{M}_1$ and sender 2 chooses a message index M_2 *uniformly* from $\mathcal{M}_2$. They both send their messages over the channel. Assuming that the distribution of messages over the product set $\mathcal{M}_1 \times \mathcal{M}_2$ is uniform, we define the *average probability of error* for the $((2^{NR_1}, 2^{NR_2}), N)$ code as

$$P_e^{(N)} = \frac{1}{2^{N(R_1+R_2)}} \sum_{(m_1,m_2)\in\mathcal{M}_1\times\mathcal{M}_2} \Pr\{g(Z^N) \neq (m_1, m_2) | (m_1, m_2) \text{ sent}\}. \tag{4.42}$$

Definition 12. *A rate pair* (R_1, R_2) *is said to be* achievable *for the multiple-access channel if there exists a sequence of* $((2^{NR_1}, 2^{NR_2}), N)$ *codes with* $P_e^{(N)} \to 0$.

Definition 13. *The* capacity region $\mathcal{R}_{MAC}$ *of the multiple-access channel is the closure of the set of achievable* (R_1, R_2) *rate pairs.*

Theorem 7 (Multiple-access channel capacity)**.** *The capacity of a multiple-access channel* $(\mathcal{X} \times \mathcal{Y}, p(z|x,y), \mathcal{Z})$ *is the closure of the convex hull of all* (R_1, R_2) *satisfying*

$$R_1 < I(X; Z|Y), \tag{4.43}$$

$$R_2 < I(Y; Z|X), \tag{4.44}$$

$$R_1 + R_2 < I(X, Y; Z) \tag{4.45}$$

for some product distribution $p_X(x)p_Y(y)$ *on* $\mathcal{X} \times \mathcal{Y}$.

4.3.1 Polar Coding

Let (X, Y, Z) be a triple of correlated random variables with properties defined as in Section 4.1. Let $X, Y \in \mathcal{X} = \{0, 1, \ldots, q-1\}$, where q is prime. Let $Z \in \mathcal{Z}$ where $\mathcal{Z}$ is an arbitrary discrete alphabet. We may consider (X, Y) as input to a multiple-access channel described by conditional probability $P_{Z|XY}$ and Z as the channel's output. Note that, in MAC setting we have a special case distribution $P_{XY}(x, y) = P_X(x)P_Y(y)$. We consider a block of $N = 2^n$ i.i.d. channel uses resulting in (X^N, Y^N, Z^N). In addition, let $U^N = X^N G_N$ and $V^N = Y^N G_N$ as always. Note that the following are true for the joint distributions of the random variables:

$$P_{X^N Y^N Z^N}(x^N, y^N, z^N) = \prod_{i=1}^{N} P_{XYZ}(x_i, y_i, z_i),$$

$$P_{U^N V^N Z^N}(u^N, v^N, z^N) = P_{X^N Y^N Z^N}(u^N G_N, v^N G_N, z^N).$$

As in section 4.1, we use S^{2N} and b^{2N} to denote *monotone permutation* on user

vectors (U^N, V^N) and corresponding path vector, respectively. For polar coding purposes we decompose the joint distribution as

$$P_{U^N V^N Z^N}(u^N, v^N, z^N) = P_{Z^N}(z^N) \prod_{k=1}^{2N} P_{S_k|Z^N, S^{k-1}}(s_k|z^N, s^{k-1}). \tag{4.46}$$

Then, *monotone* expansion of total mutual information is given as

$$NI(X, Y; Z) = I(U^N, V^N; Z^N) = \sum_{k=1}^{2N} I(Z^N; S_k|S^{k-1}). \tag{4.47}$$

The channel rates R_1 and R_2 for a given b^{2N} and S^{2N} are defined as

$$R_1 = \frac{1}{N} \sum_{\substack{k=1: \\ b_k=0}}^{2N} I(Z^N; S_k|S^{k-1}), \tag{4.48}$$

$$R_2 = \frac{1}{N} \sum_{\substack{k=1: \\ b_k=1}}^{2N} I(Z^N; S_k|S^{k-1}). \tag{4.49}$$

For any path on $U^N V^N$ the rate pair satisfies

$$R_1 \leq \frac{1}{N} I(Z^N; U^N|V^N) = I(Z; X|Y), \tag{4.50}$$

$$R_2 \leq \frac{1}{N} I(Z^N; V^N|U^N) = I(Z; Y|X), \tag{4.51}$$

$$R_1 + R_2 = \frac{1}{N} I(Z^N; U^N, V^N) = I(Z; X, Y). \tag{4.52}$$

The first inequality is satisfied with equality for $b^{2N} = 1^N 0^N$ and the second inequality is satisfied with equality for $b^{2N} = 0^N 1^N$.

The rate pairs (R_1, R_2) span the dominant face of the MAC region spanning its two end points. They also form a dense subset of the dominant face from the results in section 4.1.

Theorem 8. *Fix a path b^{2N_0} for $U^{N_0} V^{N_0}$. Let $\bar{R} = (R_1, R_2)$ be the associated rate vector. Let, S^{2N} be the edge variables for scaled path $2^l b^{2N_0}$, where $N = 2^l N_0$.*

Then, for all $\epsilon > 0$,

$$\lim_{l\to\infty} \frac{1}{2N} \left| \left\{ k \in [2N] : 2^{-N^\beta} < I(Z^N; S_k|S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l\to\infty} \frac{|\tilde{\mathcal{I}}_m(\beta)|}{N} = R_m, \qquad m = 1, 2,$$

where $\tilde{\mathcal{I}}_m(\beta) = \{k \in [2N] : b_k = m-1, I(Z^N; S_k|S^{k-1}) \geq 1 - 2^{-N^\beta}\}$, *for* $m \in \{1, 2\}$.

Proof. Note that for MAC setting Section 4.1 and polarization theorem 6 applies. From Theorem 6, we have the following fact:

$$\lim_{l\to\infty} \frac{1}{2N} \left| \left\{ k \in [2N] : 2^{-N^\beta} < H(S_k|S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l\to\infty} \frac{|\tilde{\mathcal{L}}_m(\beta)|}{N} = 1 - R'_m, \qquad m = 1, 2,$$

where $\tilde{\mathcal{L}}_m(\beta) = \{k \in [2N] : b_k = m-1, H(S_k|S^{k-1}) \leq 2^{-N^\beta}\}$ and $R'_m = \frac{1}{N} \sum_{k:b_k=m-1}^{2N} H(S_k|S^{k-1})$, for $m \in \{1, 2\}$. Note that, since X and Y are independent (without observation Z) the following is true for R'_m:

$$R'_1 = H(X), \qquad R'_2 = H(Y).$$

Sets $\tilde{\mathcal{L}}_1(\beta)$ and $\tilde{\mathcal{L}}_2(\beta)$ are not dependent on particular path.

We also have the following fact from Theorem 6:

$$\lim_{l\to\infty} \frac{1}{2N} \left| \left\{ k \in [2N] : 2^{-N^\beta} < H(S_k|Z^N, S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l\to\infty} \frac{|\tilde{\mathcal{H}}_m(\beta)|}{N} = R''_m, \qquad m = 1, 2,$$

where $\tilde{\mathcal{H}}_m(\beta) = \{k \in [2N] : b_k = m-1, H(S_k|Z^N, S^{k-1}) > 1 - 2^{-N^\beta}\}$ and $R''_m = \frac{1}{N} \sum_{k:b_k=m-1}^{2N} H(S_k|Z^N, S^{k-1})$, for $m \in \{1, 2\}$. Also, the following is true

for R''_m:

$$H(X|Z,Y) \leq R''_1 \leq H(X|Z), \qquad H(Y|Z,X) \leq R''_2 \leq H(Y|Z).$$

The lower and upper bounds of first and second expressions, respectively, are satisfied with path $b^{2N} = 1^N 0^N$. Similarly, the upper and lower bounds of first and second expressions, respectively, are satisfied with path $b^{2N} = 0^N 1^N$.

First define two index sets as follows:

$$\tilde{\mathcal{K}}_m \triangleq \{k \in [2N] : \ b_k = m - 1\}, \quad m = 1, 2. \tag{4.53}$$

We define complements of sets for user m with respect to the corresponding index set $\tilde{\mathcal{K}}_m$, i.e. $\tilde{\mathcal{F}}_m^c \triangleq \tilde{\mathcal{K}}_m \setminus \tilde{\mathcal{F}}_m$. Since $H(S_k|Z^N, S^{k-1}) \leq H(S_k|S^{k-1})$, we have $\tilde{\mathcal{L}}_m \cap \tilde{\mathcal{H}}_m = \emptyset$ and $\tilde{\mathcal{H}}_m \subseteq \tilde{\mathcal{L}}_m^c$. Let $\tilde{\mathcal{F}}_m = \tilde{\mathcal{L}}_m \cup \tilde{\mathcal{H}}_m$. Then, we may write $\tilde{\mathcal{I}}'_m = \tilde{\mathcal{F}}_m^c = (\tilde{\mathcal{L}}_m \cup \tilde{\mathcal{H}}_m)^c = \tilde{\mathcal{L}}_m^c \setminus \tilde{\mathcal{H}}_m$. $\tilde{\mathcal{I}}'_m$ has some extra partially polarized indices compared to $\tilde{\mathcal{I}}_m$. By polarization, the ratio of the size of the set of partially polarized indices to N go to zero. Thus, the result follows by observing $\frac{|\tilde{\mathcal{I}}'_m|}{N} \to \frac{|\tilde{\mathcal{I}}_m|}{N}$ as $N \to \infty$. □

In the following sections we define the polarization sets and give the encoding and decoding protocols for polar codes for two-user MAC. Then, we prove that for those encoding and decoding rules, the average error probability goes to zero as block size goes to infinity.

4.3.1.1 Polarization Sets

In the following discussion, we will refer to three interrelated index variables k, i and j, repeatedly, all in the context of an assumed path b^{2N}. k will mark the index of edge vector S^{2N}. i and j will mark the corresponding index of U^N and V^N, respectively. We will make use of the Definition 10 here.

For the purpose of polar coding, the total probability is also expanded as

follows:

$$P_{U^N V^N Z^N}(u^N, v^N, z^N) = P_{S^{2N} Z^N}(\pi_N u^N, v^N, z^N),$$
$$P_{S^{2N} Z^N}(s^{2N}, z^N) = P_{Z^N}(z^N) \prod_{k=1}^{2N} P_{S_k|Z^N S^{k-1}}(s_k|z^N, s^{k-1}). \tag{4.54}$$

Similarly the following is true for $P_{S^{2N}}$ and $P_{U^N V^N}$:

$$P_{S^{2N}}(s^{2N}) = \prod_{k=1}^{2N} P_{S_k|S^{k-1}}(s_k|s^{k-1}), \tag{4.55}$$
$$P_{U^N V^N}(u^N, v^N) = P_{S^{2N}}(\pi_N(u^N, v^N)). \tag{4.56}$$

Let $\delta_N = 2^{-N^\beta}$ for $0 < \beta < \frac{1}{2}$. We use Bhattacharyya parameters $Z(\cdot|\cdot)$ instead of entropies $H(\cdot|\cdot)$ when defining polarization sets. Because, $Z(\cdot|\cdot)$ bounds average probability of error by Proposition 1 and $Z(\cdot|\cdot)$ and $H(\cdot|\cdot)$ polarize together by Proposition 2. First, we define the following general "path dependent" polarization sets:

$$\tilde{\mathcal{L}} \triangleq \left\{k \in [2N] : Z(S_k|S^{k-1}) \leq \delta_N\right\}, \tag{4.57}$$
$$\tilde{\mathcal{H}} \triangleq \left\{k \in [2N] : Z(S_k|Z^N, S^{k-1}) \geq 1 - \delta_N\right\}. \tag{4.58}$$

Then, we define the following related sets for users 1 and 2:

$$\tilde{\mathcal{L}}_1 \triangleq \left\{k \in [2N] : b_k = 0, k \in \tilde{\mathcal{L}}\right\}, \quad \tilde{\mathcal{L}}_2 \triangleq \left\{k \in [2N] : b_k = 1, k \in \tilde{\mathcal{L}}\right\}, \tag{4.59}$$
$$\tilde{\mathcal{H}}_1 \triangleq \left\{k \in [2N] : b_k = 0, k \in \tilde{\mathcal{H}}\right\}, \quad \tilde{\mathcal{H}}_2 \triangleq \left\{k \in [2N] : b_k = 1, k \in \tilde{\mathcal{H}}\right\}. \tag{4.60}$$

We define the following *frozen* and *information* sets

$$\tilde{\mathcal{F}} \triangleq \tilde{\mathcal{L}} \cup \tilde{\mathcal{H}}, \qquad \tilde{\mathcal{I}} \triangleq [2N] \setminus \tilde{\mathcal{F}}, \tag{4.61}$$
$$\tilde{\mathcal{F}}_1 \triangleq \tilde{\mathcal{L}}_1 \cup \tilde{\mathcal{H}}_1, \qquad \tilde{\mathcal{I}}_1 \triangleq \tilde{\mathcal{K}}_1 \setminus \tilde{\mathcal{F}}_1, \tag{4.62}$$
$$\tilde{\mathcal{F}}_2 \triangleq \tilde{\mathcal{L}}_2 \cup \tilde{\mathcal{H}}_2, \qquad \tilde{\mathcal{I}}_2 \triangleq \tilde{\mathcal{K}}_2 \setminus \tilde{\mathcal{F}}_2. \tag{4.63}$$

and

$$\mathcal{F}_1 \triangleq \{i \in [N] : k \in \tilde{\mathcal{F}}_1\}, \qquad \mathcal{I}_1 \triangleq [N] \setminus \mathcal{F}_1, \tag{4.64}$$

$$\mathcal{F}_2 \triangleq \{j \in [N] : k \in \tilde{\mathcal{F}}_2\}, \qquad \mathcal{I}_2 \triangleq [N] \setminus \mathcal{F}_2, \tag{4.65}$$

where $\tilde{\mathcal{K}}_1$ and $\tilde{\mathcal{K}}_2$ are as defined in (4.53).

4.3.1.2 Encoding

Encoder 1 and 2 first construct u^N and v^N, respectively, symbol by symbol and then calculate $x^N = u^N G_N$, $y^N = v^N G_N$ to be supplied to the channel. The subset of indices of u^N, v^N identified by sets $\mathcal{I}_1$, $\mathcal{I}_2$, respectively, are the message symbols intended for the receiver. They are determined uniformly. The remaining non-message indices are computed according to a set of maps that are *shared* between the encoders and decoder. These maps will be identified with $(\lambda_i^{(1)}, \lambda_j^{(2)})$ and defined for $i \in \mathcal{F}_1$, $j \in \mathcal{F}_2$. We use $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ to denote the set of maps shared between the encoders and the decoder.

We will define two different versions of these maps. The first one will be *maximum a posteriori* based deterministic rules. The second one will be random maps. In the analysis, random maps will be used for the sake of analytic tractability. The analysis of error probability will be done as an average over all possible maps.

We define deterministic maps $\bar{\lambda}_i^{(1)} : \mathcal{X}^{i-1} \to \mathcal{X}$ and $\bar{\lambda}_j^{(2)} : \mathcal{X}^{j-1} \to \mathcal{X}$ as

$$\begin{aligned} \bar{\lambda}_i^{(1)}(u^{i-1}) &\triangleq \arg\max_{u' \in \mathcal{X}} \left\{ P_{U_i|U^{i-1}}(u'|u^{i-1}) \right\}, \\ \bar{\lambda}_j^{(2)}(v^{j-1}) &\triangleq \arg\max_{v' \in \mathcal{X}} \left\{ P_{V_j|V^{j-1}}(v'|v^{j-1}) \right\}. \end{aligned} \tag{4.66}$$

We define class of random maps $\Lambda_i^{(1)} : \mathcal{X}^{i-1} \to \mathcal{X}$ and $\Lambda_j^{(2)} : \mathcal{X}^{j-1} \to \mathcal{X}$ as

$$\begin{aligned} \Lambda_i^{(1)}(u^{i-1}) &\triangleq a, \quad \text{w.p. } P_{U_i|U^{i-1}}(a|u^{i-1}), \\ \Lambda_j^{(2)}(v^{j-1}) &\triangleq a, \quad \text{w.p. } P_{V_j|V^{j-1}}(a|v^{j-1}), \end{aligned} \tag{4.67}$$

where $a \in \mathcal{X}$. Maps $(\lambda_i^{(1)}, \lambda_j^{(2)})$ are the realizations of random maps $(\Lambda_i^{(1)}, \Lambda_j^{(2)})$. Each realization of set of maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ results in different encoding and decoding protocols. The distribution over the choice of maps is induced with the above equation (4.67).

The encoder 1 (2) uses the input symbols $u_{\mathcal{I}_1}$ $(v_{\mathcal{I}_2})$ and identical shared maps $\lambda_i^{(1)}$ $(\lambda_j^{(2)})$ to construct the length-N vector u^N (v^N) *successively* as

$$u_i = \begin{cases} u_i, & \text{if } i \in \mathcal{I}_1, \\ \lambda_i^{(1)}(u^{i-1}), & \text{otherwise.} \end{cases} \tag{4.68}$$

$$v_j = \begin{cases} v_j, & \text{if } j \in \mathcal{I}_2, \\ \lambda_j^{(2)}(v^{j-1}), & \text{otherwise.} \end{cases} \tag{4.69}$$

Encoder 1 calculates $x^N = u^N G_N$ and applies it to the channel. Similarly, encoder 2 calculates $y^N = v^N G_N$ and applies it to the channel.

4.3.1.3 Decoding

Decoder decodes the sequence $\hat{s}^{2N} = \pi_N(\hat{u}^N, \hat{v}^N)$ symbol by symbol using the observations z^N. We define the following decoding functions:

$$\zeta_k(z^N, s^{k-1}) \triangleq \arg\max_{s' \in \mathcal{X}} \left\{ P_{S_k|Z^N S^{k-1}}(s'|z^N, s^{k-1}) \right\}. \tag{4.70}$$

The decoder uses the identical shared maps $\lambda_i^{(1)}$ and $\lambda_j^{(2)}$ to reconstruct the estimate $\hat{s}^{2N}$ *successively* as

$$\hat{s}_k = \begin{cases} \lambda_i^{(1)}(\hat{u}^{i-1}), & \text{if } k \in \tilde{\mathcal{F}}_1, \\ \lambda_j^{(2)}(\hat{v}^{j-1}), & \text{if } k \in \tilde{\mathcal{F}}_2, \\ \zeta_k(z^N, \hat{s}^{k-1}), & \text{otherwise.} \end{cases} \tag{4.71}$$

Instead of $\lambda_i^{(m)}$, $m = 1, 2$, the decoder could also use $\bar{\lambda}_i^{(m)}$ when doing deterministic operation. As stated before the encoder and decoder are using the same shared maps for non-message indices. A realization of class of random maps has a probability of occurrence induced by probabilities $P_{U_i|U^{i-1}}$ and $P_{V_j|V^{j-1}}$ as given in (4.67). Each realization results in different encoding/decoding protocol. We use randomized map concept to bound the average error probability by averaging over all possible maps, thus showing that there exists at least one good map.

Note that the encoding operations are almost the same as single-user channel coding: information bits are inserted into indices in $\mathcal{I}_1$ ($\mathcal{I}_2$), the remaining bits are determined by shared set of maps and the resulting sequence u^N (v^N) is passed through polar transformation to obtain x^N (y^N). The difference is that the size of the *information* sets $\mathcal{I}_1$ ($\mathcal{I}_2$) may be adjusted using different paths b^{2N} while keeping their sum at constant. That way all rate allocation pairs (R_1, R_2) on the dominant face of the MAC capacity region may be reached. The decoder is very different compared to single-user channel polar coding where a single-user polar SC decoder was used. Here, two-user polar successive cancellation decoding is used at the decoder.

As in the single-user case, for analysis purposes, the encoding functions are random. The results of encoding operations may be different for the same inputs $(u_{\mathcal{I}_1}, v_{\mathcal{I}_2})$. For encoder 1 (2) at step $i \in \mathcal{I}_1$ ($j \in \mathcal{I}_2$) of the process the inputs are inserted which are assumed to be uniformly distributed. Thus, for a realization of set of maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$, a particular (x^N, y^N) occurs with a certain probability induced by input distributions and maps. We define the resulting average (over $u_{\mathcal{I}_1}$, $v_{\mathcal{I}_2}$) probability of error of above encoding and decoding operations as $P_e[\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)}]$. In the following we show that for sets $\mathcal{I}_1$, $\mathcal{I}_2$ defined in 4.3.1.1 and encoding and decoding methods defined in 4.3.1.2 and 4.3.1.3, there exists maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ such that $P_e[\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)}] \leq O(2^{-N^\beta})$, for $0 < \beta < 1/2$. We do that by determining the *expected* average probability of error over the ensembles of codes generated by different encoding maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$. The distribution over the choices of maps is given in (4.67). That is, we take expectation of $P_e[\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}]$ which is a random quantity. Then we show that *expected* average probability of error decay to zero as $O(2^{-N^\beta})$. This implies that for at least one choice of $(\lambda_{\mathcal{F}_1}^{(1)},$

$\lambda_{\mathcal{F}_2}^{(2)}$) the average probability of error decays to zero as $O(2^{-N^\beta})$. The following theorem makes this precise.

Theorem 9. *Let $\mathcal{I}_1$, $\mathcal{I}_2$ be sets as defined in 4.3.1.1 and encoding and decoding methods be as defined in 4.3.1.2 and 4.3.1.3. Then the expectation of average probability of error $P_e[\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}]$ over the maps $(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})$ satisfy $\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}\}}\left[P_e[\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}]\right] \leq 2^{-N^\beta}$ for any $(R_1, R_2) \in \mathcal{R}_{MAC}$ and $0 < \beta < 1/2$. Consequently, there exists deterministic maps that satisfy the above relations.*

The following sections give necessary steps for proving the theorem. We first prove a total variation bound on two probability measures. Then, we use that result to bound the *expected* average probability of error of the code.

4.3.1.4 Total Variation Bound

Assume a given path b^{2N}. Then the edge variables vector S^{2N} is a monotone permutation π_N (identified by b^{2N}) on $U^N V^N$, i.e. $S^{2N} = \pi_N(U^N V^N)$. To analyze the average error probability P_e via the probabilistic method we define the following probability measure.

$$Q_{S^{2N}}(s^{2N}) \triangleq \prod_{k=1}^{2N} Q_{S_k|S^{k-1}}(s_k|s^{k-1}), \tag{4.72}$$

where conditional probability measures are defined as

$$Q_{S_k|S^{k-1}}(s_k|s^{k-1}) \triangleq \begin{cases} P_{U_i|U^{i-1}}(u_i|u^{i-1}), & \text{if } k \in \tilde{\mathcal{F}}_1, \\ P_{V_j|V^{j-1}}(v_j|v^{j-1}), & \text{if } k \in \tilde{\mathcal{F}}_2, \\ \frac{1}{q}, & \text{otherwise.} \end{cases} \tag{4.73}$$

The probability measure Q defined in (4.72) is a perturbation of $P_{S^{2N}}$ in (4.55). The following lemma provides a bound on the total variation distance between P and Q.

Lemma 7 (Total Variation Bound)**.** *Let probability measures P and Q be defined as in* (4.55) *and* (4.72)*, respectively. For $0 < \beta < 1/2$ and sufficiently large N,*

the total variation distance between P and Q is bounded as

$$\sum_{s^{2N}} \left| P_{S^{2N}}(s^{2N}) - Q_{S^{2N}}(s^{2N}) \right| \leq 2^{-N^{\beta}}. \tag{4.74}$$

Proof. See Appendix B.2. □

4.3.1.5 Average Error Probability

The encoding and decoding rules were established in Sections 4.3.1.2 and 4.3.1.3, respectively. Consider the sequences u^N and v^N formed at the encoders and observation z^N received by the decoder. The decoder makes an SC decoding error on the k-th symbol for the following tuples:

$$\mathcal{T}^k \triangleq \big\{ (s^{2N}, z^N) : \exists s' \in \mathcal{X} \text{ s.t. } s' \neq s_k, \\ P_{S_k|Z^N S^{k-1}}(s_k|z^N, s^{k-1}) \leq P_{S_k|Z^N S^{k-1}}(s'|z^N, s^{k-1}) \big\}. \tag{4.75}$$

The set $\mathcal{T}^k$ represents those tuples causing an error at the decoder in the case s_k is inconsistent with respect to observations and the decoding rule. The complete set of tuples causing an error is

$$\mathcal{T} \triangleq \bigcup_{k \in \tilde{\mathcal{I}}} \mathcal{T}^k. \tag{4.76}$$

Assuming randomized maps shared between encoder and decoder, the average error probability is a random quantity given as

$$P_e[\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}] = \sum_{(s^{2N}, z^N) \in \mathcal{T}} \Bigg[P_{Z^N|S^{2N}}(z^N|s^{2N}) \cdot \frac{1}{q^{|\mathcal{I}|}} \\ \cdot \prod_{i \in \mathcal{F}_1} \mathbb{1}_{\left\{\Lambda^{(1)}_i(u^{i-1}) = u_i\right\}} \prod_{j \in \mathcal{F}_2} \mathbb{1}_{\left\{\Lambda^{(2)}_j(v^{j-1}) = v_j\right\}} \Bigg]. \tag{4.77}$$

The expected average block error probability is calculated by averaging over the

randomness in the encoders and decoder

$$\bar{P}_e \triangleq \mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}\}}\left[P_e[\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}]\right]. \tag{4.78}$$

The following lemma bounds the expected average block error probability.

Lemma 8. *Consider the polarization based channel code described in Sections 4.3.1.2 and 4.3.1.3. Let the information set $\mathcal{I}$ and frozen set $\mathcal{F}$ be selected as in* (4.61)*. Then for $0 < \beta < 1/2$ and sufficiently large N,*

$$\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}\}}\left[P_e[\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}]\right] \leq 2^{-N^\beta}.$$

Proof. Note that the expectation of average probability of error is written as

$$\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}\}}\left[P_e[\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}]\right] = \sum_{(s^{2N},z^N)\in\mathcal{T}}\Bigg[P_{Z^N|S^{2N}}(z^N|s^{2N})\cdot\frac{1}{q^{|\mathcal{I}|}} \cdot\prod_{i\in\mathcal{F}_1}\mathbb{P}\left\{\Lambda_i^{(1)}(u^{i-1})=u_i\right\}\prod_{j\in\mathcal{F}_2}\mathbb{P}\left\{\Lambda_j^{(2)}(v^{j-1})=v_j\right\}\Bigg].$$

From the definition of random mappings $\Lambda_m^{(i)}$ it follows that

$$\mathbb{P}\left\{\Lambda_i^{(1)}(u^{i-1})=u_i\right\} = P_{U_i|U^{i-1}}(u_i|u^{i-1}),$$
$$\mathbb{P}\left\{\Lambda_j^{(2)}(v^{j-1})=v_j\right\} = P_{V_j|V^{j-1}}(v_j|v^{j-1}).$$

Then, we may substitute the definition for $Q_{S^{2N}}(s^{2N})$ in (4.72) into the expression of expected average probability of error to get

$$\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}\}}\left[P_e[\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}]\right] = \sum_{(s^{2N},z^N)\in\mathcal{T}} P_{Z^N|S^{2N}}(z^N|s^{2N})\; Q_{S^{2N}}(s^{2N}).$$

Then we split the error into two main parts, one due to the polar decoding function and the other due to the total variation distance between probability

measures.

$$\begin{aligned}
&\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}\}}\left[P_e[\Lambda_{\mathcal{F}_1}^{(1)},\Lambda_{\mathcal{F}_2}^{(2)}]\right] \\
&\qquad = \sum_{(s^{2N},z^N)\in\mathcal{T}} P_{Z^N|S^{2N}}(z^N|s^{2N})\ \left[Q_{S^{2N}}(s^{2N}) - P_{S^{2N}}(s^{2N}) + P_{S^{2N}}(s^{2N})\right], \\
&\qquad \leq \sum_{(s^{2N},z^N)\in\mathcal{T}} P_{S^{2N}Z^N}(s^{2N},z^N) + \sum_{s^{2N}} \left|Q_{S^{2N}}(s^{2N}) - P_{S^{2N}}(s^{2N})\right|.
\end{aligned}$$

The second part of the error which is due to total variation distance is upper bounded as $O(2^{-N^\beta})$ by Lemma 7. Thus, it remains to upper bound the error term due to polar decoding. Remember that $\mathcal{T} \triangleq \cup_{k\in\mathcal{I}}\mathcal{T}^k$. We may upper bound each error symbol by symbol. Define error probability for symbol $k \in \tilde{\mathcal{I}}$ as

$$\varepsilon^k \triangleq \sum_{(s^{2N},z^N)\in\mathcal{T}^k} P_{S^{2N}Z^N}(s^{2N},z^N).$$

But this is the average probability of error for symbol k, i.e. $\varepsilon^k = P_e(S_k|Z^N,S^{k-1})$. Probability of error is upper bounded by the Bhattacharyya parameter by Proposition 1. By union bound, total average probability of error is $\varepsilon \leq \sum_k \varepsilon^k$. Then we have

$$\begin{aligned}
\varepsilon &\leq \sum_{k\in\tilde{\mathcal{I}}}(q-1)Z(S_k|Z^N,S^{k-1}), \\
&\leq (q-1)2N\delta_N.
\end{aligned}$$

This completes the proof that the expected average probability of error is upper bounded as $O(2^{-N^\beta})$. □

Since by Lemma 8 the expected value over the random maps of average probability of error decays to zero, there must be at least one deterministic class of maps for which $P_e \to 0$.

4.3.1.6 Uniform Distributions

Similar to single user case, different polarization regions, encoding and decoding tasks simplify if input distributions are uniform. The random mapping functions defined in (4.67) always results in uniform distribution: $\Lambda_i^{(1)}(u^{i-1}) = \Lambda_j^{(2)}(v^{j-1}) = a, \quad \text{w.p. } 1/q, \forall a \in \mathcal{X}$. Thus instead of sharing set of maps $(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})$ between encoders and decoder we may generate a vector for $\tilde{\mathcal{F}}$ uniformly at random and share that. Also, each realization of a set of maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ have the same probability, which means that the expected average error probability $\bar{P}_e$ and average error probability for a realization $P_e[\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)}]$ are the same. Thus, similar to single user case the value of those symbols in $\tilde{\mathcal{F}}$ don't matter in the sense that each selection results in the same average error probability. We can choose any fixed vector for $\tilde{\mathcal{F}}$ and share it between encoders and decoder.

4.3.2 Simulations

In this section, we present simulation results showing the performance of MAC SCL decoder. We present the performance of the decoder on a well known MAC channel: binary erasure MAC (BE-MAC). The capacity region of this channel is maximized with a single distribution, namely the uniform distribution in both its inputs.

Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ denote the inputs of the channel corresponding to user 1 and 2, respectively. And let $\mathcal{X} = \mathcal{Y} = \{0, 1\}$. The channel output is given as $Z = X + Y$ which is of a ternary alphabet, $Z \in \mathcal{Z} = \{0, 1, 2\}$. The capacity region of this channel is well known [64, Sec. 15.3] and shown in Figure 4.7. The figure shows results for three different code classes labelled by A, B and C which target three different rate pairs $(0.75, 0.75)$, $(0.625, 0.875)$ and $(0.5, 1)$ on the dominant face of the MAC region, respectively. Figure 4.7 shows the summary result of the simulations. We adjust the rates of the codes on straight lines yielding operating rate pairs $(R_u^A, R_v^A) = \rho_A \cdot (0.75, 0.75)$, $(R_u^B, R_v^B) = \rho_B \cdot (0.625, 0.875)$ and $(R_u^C, R_v^C) = \rho_C \cdot (0.5, 1)$ with $0 \leq \rho_A, \rho_B, \rho_C \leq 1$ for code classes A, B and C, respectively. The markings show the points where the block error rate (BLER) falls down to 10^{-4}. The list size L used in these simulations is 32. Code class B performs the best (closest to boundary). Code class A comes the second and code class C is the worst.

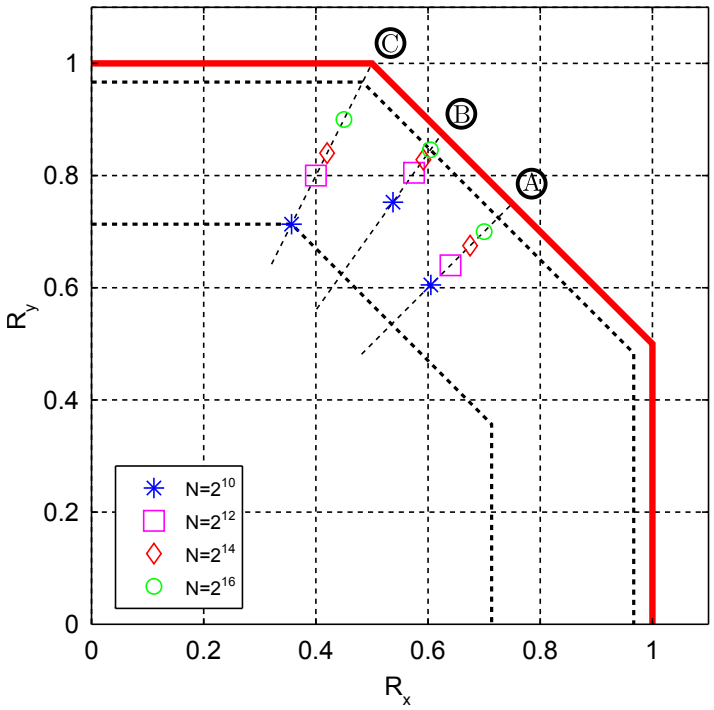


Figure 4.7: 10^{-4} BLER marked on MAC region for n=10,12,14,16 (L=32).

4.3.2.1 Construction

The polar codes are designed by Monte-Carlo simulations using the SC MAC decoder. Our decoder outputs soft likelihood ratios for both u^N and v^N which are averaged over large number of simulations and used as reliability values. The code design is specific to the underlying MAC and involves finding a path b^{2N} and information index sets $\mathcal{A}_u$ and $\mathcal{A}_v$ for a desired target rate pair (R_x, R_y). Although there may be many possible paths satisfying the required rate pair we restricted ourselves to a class of paths of the form $0^i 1^N 0^{N-i}$ for $0 \leq i \leq N$. These paths produce rate pairs that span the entire dominant face of the MAC region.

The following three figures shows the code construction simulation results for code class C, B and A, in that order. The sorted reliability values of the coordinate channels of users U and V are plotted. The red vertical lines mark the $(0.75, 0.75)$ rate point for reference. The green vertical lines mark the target rate point for that simulation.

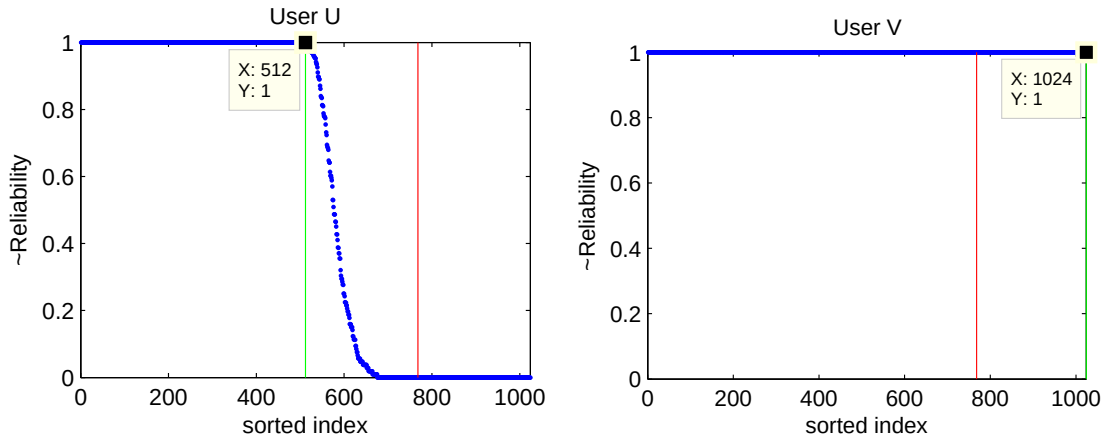


Figure 4.8: $N = 2^{10}$, Path $= 0^N 1^N$, Rate C $= (0.5, 1.0)$.

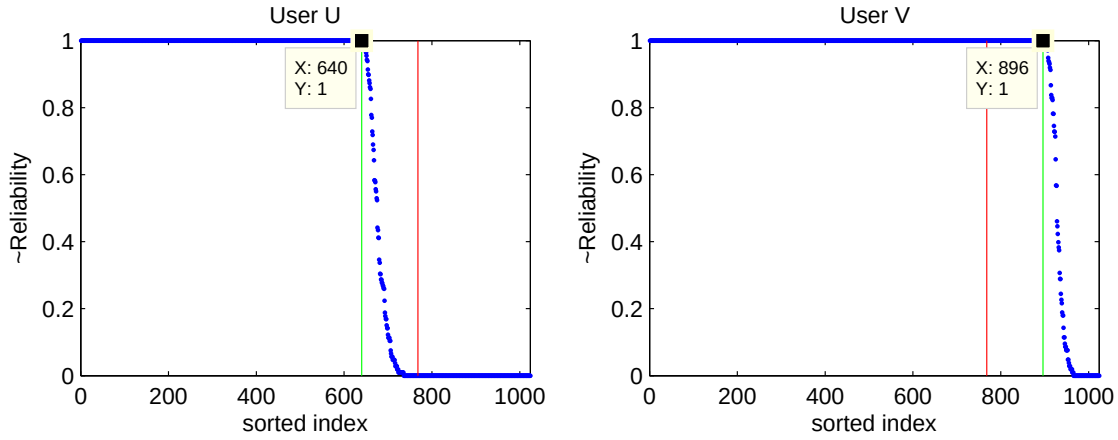


Figure 4.9: $N = 2^{10}$, Path $= 0^{N/2} 1^N 0^{N/2}$, Rate B $= (0.625, 0.875)$.

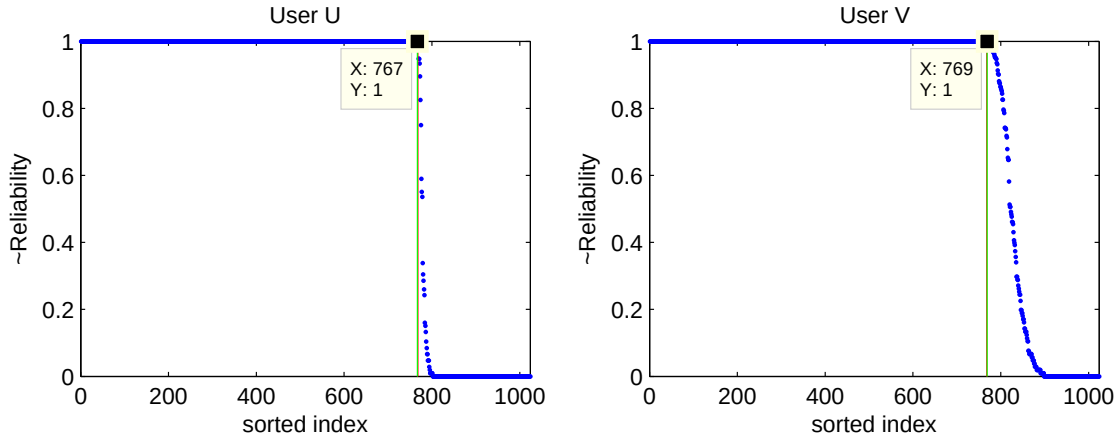


Figure 4.10: $N = 2^{10}$, Path $= 0^{17N/64} 1^N 0^{47N/64}$, Rate A $= (0.75, 0.75)$.

4.3.2.2 Performance Simulations

Here we present detailed simulation results. The following three figures compare BLER performances of four block sizes ($n = 10, 12, 14, 16$) and two list sizes ($L = 1, 32$) for three rate points A, B and C.

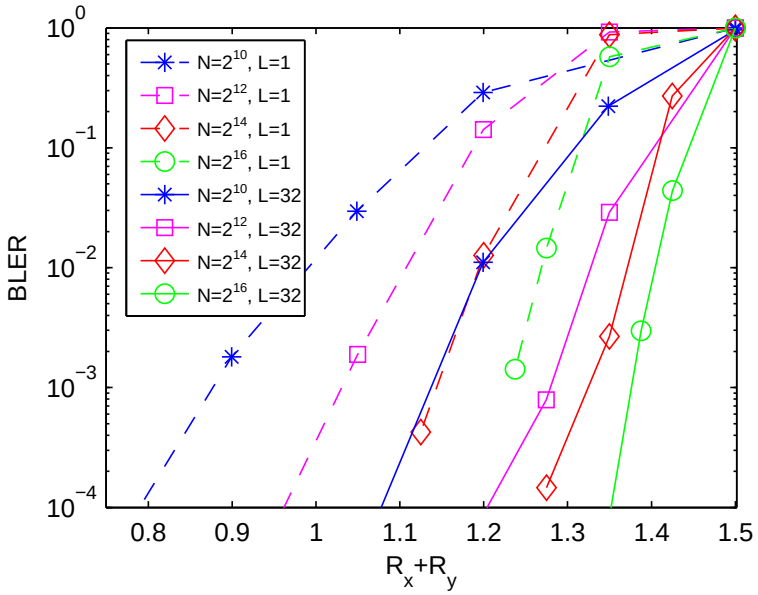


Figure 4.11: Rate point C $(0.5, 1.0)$.

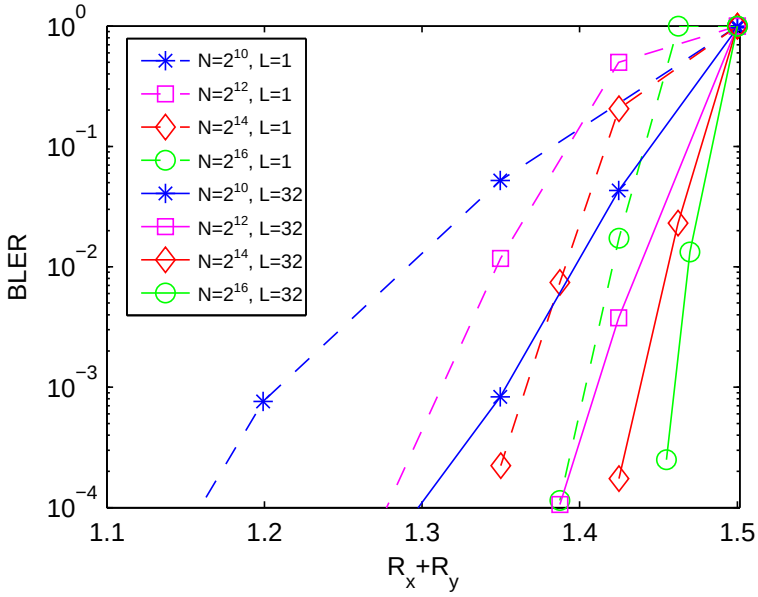


Figure 4.12: Rate point B $(0.625, 0.875)$.

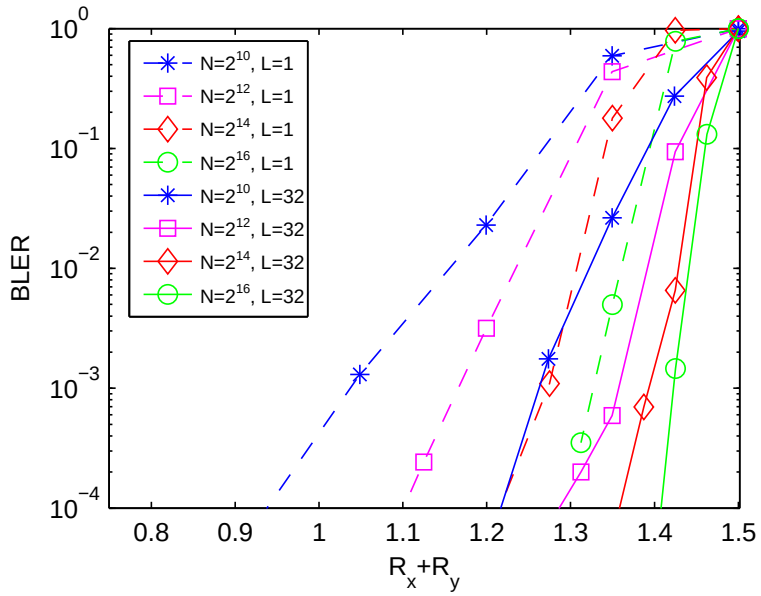


Figure 4.13: Rate point A $(0.75, 0.75)$.

4.3.2.3 Performance Simulations with CRC

In this section, we present performance results with CRC. The CRC is appended to the user bits for aid in best path selection at the end of list decoding just like in the single-user case as presented in [40]. However, in two-user MAC there is more freedom in the position to insert CRC. It may be appended to the information bits of user 1, user 2 or both. In all of the figures list size is fixed at 32. We used simulations to compare performances of different CRC options. We found that CRC of size 16 embedded into either user 1 or user 2 gave the best results most of the time. Therefore, we used that option in the following figures. The "CRC(16,0)" annotations in the figures means that a CRC of length 16 is inserted into user 1 data and no CRC is inserted into user 2. Note that, inserting CRC causes user rate to decrease a little. This is taken into account in the rate calculation.

In the following three figures, we compare the performances with and without CRC for rate points C, B and A, in order. In all cases list size is 32. The dotted lines are without CRC and the solid lines are with CRC. We can see clearly

how much CRC improves the performance for short block lengths (2^{10} and 2^{12}). However, as the block size increases the effect of CRC on performance decreases. This behavior is analogous to single-user case.

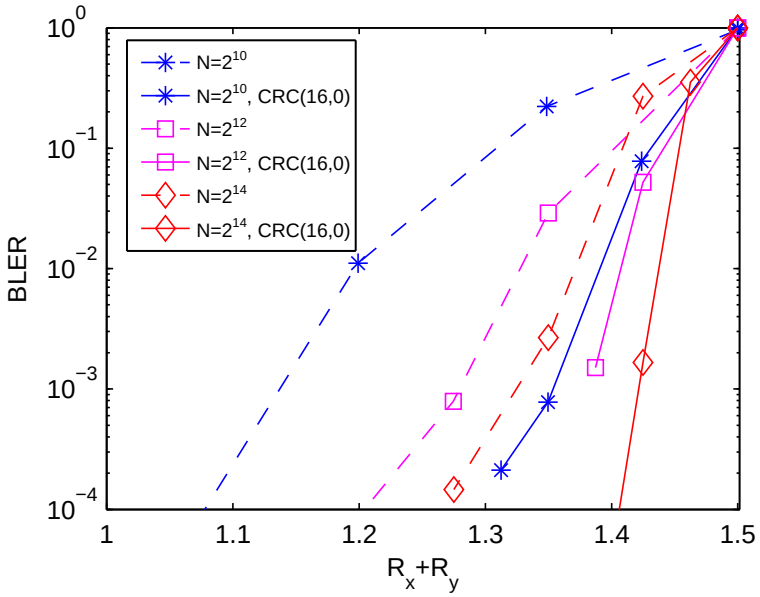


Figure 4.14: Comparison of CRC performance for rate point C $(0.5, 1.0)$.

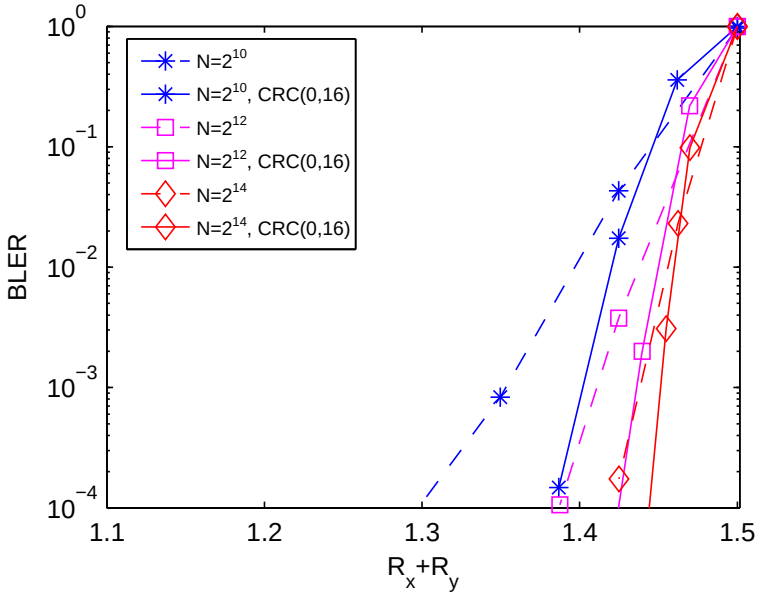


Figure 4.15: Comparison of CRC performance for rate point B $(0.625, 0.875)$.

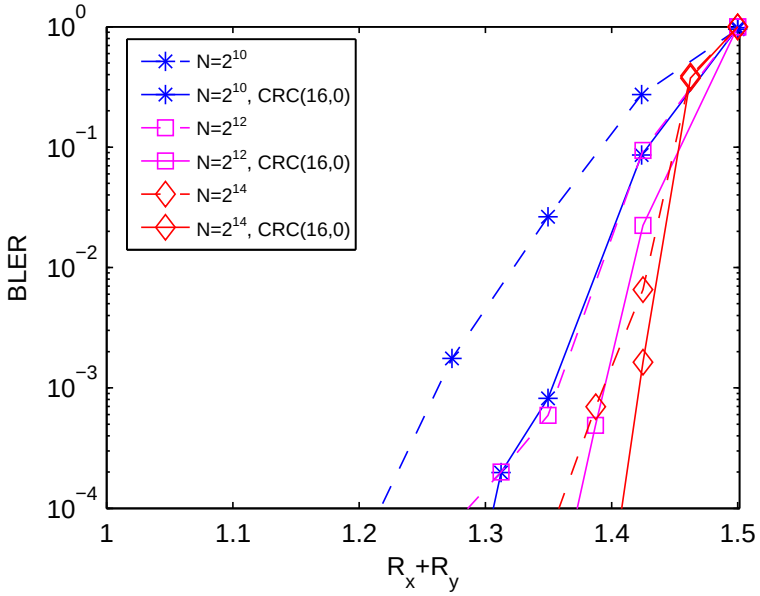


Figure 4.16: Comparison of CRC performance for rate point A $(0.75, 0.75)$.

Chapter 5

Distributed Lossy Coding

In this chapter, we consider two different distributed source coding setups where reconstructions of sources are subject to distortion constraints. We prove that the bounds of the known achievable rate-regions of both setups may be attained by polar coding (PC) methods.

The first of these setups is the distributed lossy source coding (DLSC) setup which we consider in Section 5.1. This setup consists of two correlated sources, their two *separate* encoders and a *joint* decoder. The reconstructions of two sources at the decoder are subject to different distortion constraints. The achievable rate-region of this problem is not known in general but there is a good inner bound called the Berger-Tung (BT) inner bound. The only work we are aware of that mentions DLSC problem using polar codes is given in [65] which is independent and contemporaneous to ours. The authors very briefly claim that PC for DLSC setup can be done using "nested polar codes" [66]. The polar coding method for DLSC problem described in our work is based on *monotone chain rule* approach introduced in [28] and also analyzed in detail in Section 4.1. We show that using our method, any point on the dominant face of BT region may be achieved for arbitrary source distributions. In our method, two single user successive-cancellation (SC) polar decoders are needed for encoding and a single two-user joint SC polar decoder based on monotone chain rules is needed for

decoding.

The second setup is the multiple description coding (MDC) setup which we consider in Section 5.2. In this setup, there is a *single* source and *two encoders* generate two different representations of the same source. Then, *three decoders* that has access to representation 1, representation 2 and both, respectively, generate three different reconstructions subject to three different distortion constraints. The achievable rate-region for this problem is not known in general but there is an inner bound called El-Gamal Cover (EGC) inner bound. In the following we mention other work on polar coding for MDC setup briefly. All of these work consider achieving the EGC inner bound. Polar coding for MDC problem was considered in [30] and two different methods were proposed. The first one is based on joint polarization approach that was introduced in [18]. Using this method a certain point on the dominant face of EGC region may be achieved. However, this method has an important drawback such that the achieved rate-pair is determined by the coding scheme rather than being a design choice. The second method is based on rate-splitting approach described in [67] and achieves any point on the dominant face. However, the method is expensive in the sense that it uses three successive encoding steps, each of which is a SC polar decoder. Recently PC for MDC problem was also considered in [31]. The method considered in that work is only for a special case of uniform and binary sources. Another recent work that mentions MDC problem using polar codes is given in [65]. The authors very briefly claim that PC for MDC setup can be done using "nested polar codes"; however, they do not go into the specifics. The polar coding method for MDC problem described in our work is based on *monotone chain rule* approach introduced in [28] and also analyzed in detail in Section 4.1. We show that using our method any point on the dominant face of EGC region may be achieved for arbitrary source distributions. In our method, a single two-user joint polar decoder based on monotone chain rules is needed for encoding.

5.1 Distributed Lossy Source Coding

General lossy source coding setup is depicted in Figure 5.1. (X, Y) is discrete memoryless source (DMS) and $\mathrm{d}_x(x, \hat{x})$, $\mathrm{d}_y(y, \hat{y})$ are two bounded distortion measures. Encoder 1 wants to compress source X to be reconstructed with a maximum distortion D_x and similarly encoder 2 wants to compress source Y to be reconstructed with a maximum distortion D_y at the joint decoder. This problem obviously includes Slepian-Wolf and lossy source coding with side information (Wyner-Ziv) problems as its special cases. However, unlike these special cases, the rate-distortion region of this problem is not known in general.

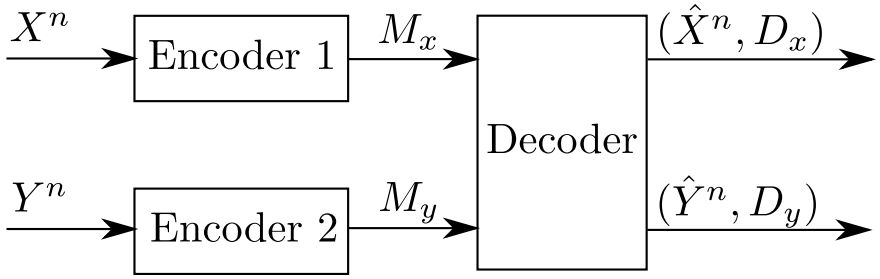


Figure 5.1: Distributed lossy compression setup.

There are Berger-Tung inner and outer bounds for the rate region, none of which is tight in general. However, there are well known specializations where either inner or outer bound is tight. Before, giving the bound theorems let's define the setting formally.

Definition 14. *An (n, R_x, R_y) source code in this setup is defined as*

- *Encoding function for X:* $f_X : \mathcal{X}^n \mapsto \{1, \ldots, 2^{nR_x}\}$.
- *Encoding function for Y:* $f_Y : \mathcal{Y}^n \mapsto \{1, \ldots, 2^{nR_y}\}$.
- *Decoding function :* $g : \{1, \ldots, 2^{nR_x}\} \times \{1, \ldots, 2^{nR_y}\} \mapsto \mathcal{X}^n \times \mathcal{Y}^n$.

Let $\mathrm{d_x} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ denote the distortion function with maximum value $\mathrm{d_{max}}$. The distortion function extends to vectors as $\mathrm{d_x}(x^N, \hat{x}^N) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{d_x}(x_i, \hat{x}_i)$. The distortion function $\mathrm{d_y} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ for Y is defined similarly.

Definition 15. *A tuple (R_x, R_y, D_x, D_y) is said to be achievable if there exists a sequence of (n, R_x, R_y) codes such that*

$$\limsup_{n\to\infty} \mathbb{E}[d_x(X^n, \hat{X}^n)] \leq D_x$$
$$\limsup_{n\to\infty} \mathbb{E}[d_y(Y^n, \hat{Y}^n)] \leq D_y$$

The rate–distortion region $\mathcal{R}(D_x, D_y)$ for distributed lossy source coding is the closure of the set of all rate pairs (R_x, R_y) such that (R_x, R_y, D_x, D_y) is achievable.

The rate-distortion region for this problem is not known in general. However, there is an inner bound called Berger-Tung inner bound [68], which is tight in some special cases. In this work we focus on the Berger-Tung inner bound.

Theorem 10 (Berger-Tung Inner Bound)**.** *Let (X, Y) be a 2-DMS with joint density $p(x, y)$ and $d_x(x, \hat{x})$ and $d_y(y, \hat{y})$ be two distortion measures. A rate pair (R_x, R_y) is achievable with distortion pair (D_x, D_y) for distributed lossy source coding if*

$$R_x \geq I(X; \bar{X}|\bar{Y}), \tag{5.1}$$
$$R_y \geq I(Y; \bar{Y}|\bar{X}), \tag{5.2}$$
$$R_x + R_y \geq I(X, Y; \bar{X}, \bar{Y}) \tag{5.3}$$

for some conditional pmf $p(\bar{x}|x)p(\bar{y}|y)$ with $|\bar{X}| \leq |X| + 4$, $|\bar{Y}| \leq |Y| + 4$, and functions $\hat{x}(\bar{x}, \bar{y})$ and $\hat{y}(\bar{x}, \bar{y})$ such that $\mathbb{E}[d_x(X, \hat{X})] \leq D_x$, $\mathbb{E}[d_y(Y, \hat{Y})] \leq D_y$.

Following are some special cases where the bound is tight.

- Suppose d_y is a Hamming distortion measure and $D_y = 0$. In this case conditions are minimized with $\bar{Y} = Y$. Then the Berger-Tung inner bound

is tight and reduces to the set of rate pairs (R_x, R_x) such that

$$\begin{aligned} R_x &\geq I(X;\bar{X}|Y), \\ R_y &\geq H(Y|\bar{X}), \\ R_x + R_y &\geq I(X;\bar{X}|Y) + H(Y) = I(X;\bar{X}) + H(Y|\bar{X}) \end{aligned}$$

for some conditional pmf $p(\bar{x}|x)$ and function $\hat{x}(\bar{x}, y)$ that satisfy the constraint $\mathrm{E}[\mathrm{d}_x(X, \hat{X})] \leq D_x$.

- It reduces to the Wyner-Ziv rate-distortion function when there is no rate limit on describing Y, i.e., $R_y \geq H(Y)$. In this case, the only active constraint is $R_x \geq I(X;\bar{X}|Y)$.
- It reduces to the Slepian-Wolf region when $D_x = 0$ in addition to D_y (set $\bar{X} = X$).

The Berger-Tung (BT) region can be defined as

$$\mathcal{R}_{BT} \triangleq \{(R_1, R_2) : R_1 \geq I(X;\bar{X}|\bar{Y}), R_2 \geq I(Y;\bar{Y}|\bar{X}), R_1 + R_2 \geq I(X,Y;\bar{X},\bar{Y})\} \tag{5.4}$$

where the random variables have the joint density $p(\bar{x}|x)p(\bar{y}|y)p(x,y)$. The *dominant face* of the BT region is defined as follows

$$\mathcal{J} \triangleq \{(R_1, R_2) \in \mathcal{R}_{BT} : R_1 + R_2 = I(X,Y;\bar{X},\bar{Y})\}. \tag{5.5}$$

Because of the special distribution, the two corner points of the dominant face are given as $(I(X;\bar{X}), I(Y;\bar{Y}|\bar{X}))$ and $(I(X;\bar{X}|\bar{Y}), I(Y;\bar{Y}))$. The first point can be shown as follows

$$\begin{aligned} I(X,Y;\bar{X},\bar{Y}) &= I(X,Y;\bar{X}) + I(X,Y;\bar{Y}|\bar{X}) && (5.6)\\ &= H(\bar{X}) - H(\bar{X}|X,Y) + H(\bar{Y}|\bar{X}) - H(\bar{Y}|X,Y,\bar{X}) && (5.7)\\ &\overset{(a)}{=} H(\bar{X}) - H(\bar{X}|X) + H(\bar{Y}|\bar{X}) - H(\bar{Y}|Y,\bar{X}) && (5.8)\\ &= I(X;\bar{X}) + I(Y;\bar{Y}|\bar{X}). && (5.9) \end{aligned}$$

(a) is due to the $\bar{X}-X-Y-\bar{Y}$ Markov chain dependency of the random variables which also gives the special form of the conditional distribution $p(\bar{x},\bar{y}|x,y) = p(\bar{x}|x)p(\bar{y}|y)$. The other corner point can be shown similarly.

We can also write the corner points as $(I(X;\bar{X}), I(Y;\bar{Y}) - I(\bar{X};\bar{Y}))$ and $(I(X;\bar{X}) - I(\bar{X};\bar{Y}), I(Y;\bar{Y}))$. This can be shown as follows

$$\begin{aligned} I(X;\bar{X}|\bar{Y})) &= H(\bar{X}|\bar{Y}) - H(\bar{X}|X,\bar{Y}) && (5.10)\\ &\overset{(a)}{=} H(\bar{X}) - H(\bar{X}|X) - [H(\bar{X}) - H(\bar{X}|\bar{Y})] && (5.11)\\ &= I(X;\bar{X}) - I(\bar{X};\bar{Y}). && (5.12) \end{aligned}$$

(a) is due to the Markov chain dependency of the random variables. Thus, the sum rate can also be written as

$$I(X,Y;\bar{X},\bar{Y}) = I(X;\bar{X}) + I(Y;\bar{Y}) - I(\bar{X};\bar{Y}). \tag{5.13}$$

5.1.1 Polar Coding

Let source variables $(X,Y) \in \mathcal{X} \times \mathcal{Y}$ be from arbitrary discrete alphabets. The external variables are from prime sized alphabets: $\bar{X},\bar{Y} \in \bar{\mathcal{X}} = \{0,1,\ldots,q-1\}$, where q is prime. Given the source distribution $(X,Y) \sim P_{XY}$, let the conditional distribution $P_{\bar{X}\bar{Y}|XY} = P_{\bar{X}|X}P_{\bar{Y}|Y}$ give rise to the design distortions D_x^* and D_y^*, i.e.

$$\begin{aligned} D_x^* &= \mathbb{E}_{P_{\bar{X}\bar{Y}XY}}[\mathrm{d_x}(X,\hat{x}(\bar{X},\bar{Y}))], && (5.14)\\ D_y^* &= \mathbb{E}_{P_{\bar{X}\bar{Y}XY}}[\mathrm{d_y}(Y,\hat{y}(\bar{X},\bar{Y}))], && (5.15) \end{aligned}$$

where

$$P_{\bar{X}\bar{Y}XY}(\bar{x},\bar{y},x,y) = P_{XY}(x,y)P_{\bar{X}|X}(\bar{x}|x)P_{\bar{Y}|Y}(\bar{y}|y). \tag{5.16}$$

Consider the i.i.d. block of random variables $(\bar{X}^N,\bar{Y}^N,X^N,Y^N)$ with $N = 2^n$

for some $n \geq 1$. The joint distribution is given by

$$P_{\bar{X}^N \bar{Y}^N X^N Y^N}(\bar{x}^N, \bar{y}^N, x^N, y^N) = \prod_{i=1}^{N} P_{XY}(x_i, y_i) P_{\bar{X}|X}(\bar{x}_i|x_i) P_{\bar{Y}|Y}(\bar{y}_i|y_i). \quad (5.17)$$

Let, $\bar{X}^N$ and $\bar{Y}^N$ denote the polar transforms of N-vectors U^N and V^N, respectively, i.e.

$$U^N = \bar{X}^N G_N, \qquad V^N = \bar{Y}^N G_N. \quad (5.18)$$

Then we have

$$P_{U^N V^N X^N Y^N}(u^N, v^N, x^N, y^N) = P_{\bar{X}^N \bar{Y}^N X^N Y^N}(u^N G^N, v^N G_N, x^N, y^N). \quad (5.19)$$

Since, G_N is a one-to-one mapping, we can write the total mutual information as follows

$$I(X^N, Y^N; \bar{X}^N, \bar{Y}^N) = NI(X, Y; \bar{X}, \bar{Y}) = I(X^N, Y^N; U^N, V^N). \quad (5.20)$$

Let $S^{2N} = (S_1, \ldots, S_{2N})$ be a permutation on (U^N, V^N) such that relative order of elements of U^N and V^N are preserved. Then, *monotone* expansion of total mutual information is given as

$$I(X^N, Y^N; U^N, V^N) = \sum_{k=1}^{2N} I(X^N, Y^N; S_k|S^{k-1}). \quad (5.21)$$

Let b^{2N} be the *path string* s.t. $b_k \in \{0, 1\}$ which denotes the decoding path. The channel rates R_1 and R_2 for a given b^{2N} and S^{2N} are defined as

$$R_1 = \frac{1}{N} \sum_{\substack{k=1: \\ b_k=0}}^{2N} I(X^N, Y^N; S_k|S^{k-1}), \quad (5.22)$$

$$R_2 = \frac{1}{N} \sum_{\substack{k=1: \\ b_k=1}}^{2N} I(X^N, Y^N; S_k|S^{k-1}). \quad (5.23)$$

For any path on $U^N V^N$ the rate pair satisfies

$$R_1 \geq \frac{1}{N} I(X^N; U^N | V^N) \qquad = I(X; \bar{X}|\bar{Y}), \tag{5.24}$$

$$R_2 \geq \frac{1}{N} I(Y^N; V^N | U^N) \qquad = I(Y; \bar{Y}|\bar{X}), \tag{5.25}$$

$$R_1 + R_2 = \frac{1}{N} I(X^N, Y^N; U^N, V^N) = I(X, Y; \bar{X}, \bar{Y}). \tag{5.26}$$

The first inequality is satisfied with equality for $b^{2N} = 1^N 0^N$ and the second inequality is satisfied with equality for $b^{2N} = 0^N 1^N$.

The rate pairs (R_1, R_2) span the dominant face $\mathcal{J}$ of the Berger-Tung region spanning its two end points. They also form a dense subset of $\mathcal{J}$.

Theorem 11. *Fix a path b^{2N_0} for $U^{N_0} V^{N_0}$. Let $\bar{R} = (R_1, R_2)$ be the associated rate vector. Let, S^{2N} be the edge variables for scaled path $2^l b^{2N_0}$, where $N = 2^l N_0$. Then, for all $\epsilon > 0$,*

$$\lim_{l \to \infty} \frac{1}{2N} \left| \left\{ k \in [2N] : 2^{-N^\beta} < I(X^N, Y^N; S_k | S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l \to \infty} \frac{|\tilde{\mathcal{I}}_m(\beta)|}{N} = R_m, \qquad m = 1, 2,$$

where $\tilde{\mathcal{I}}_m(\beta) = \{k \in [2N] : b_k = m - 1, I(X^N, Y^N; S_k | S^{k-1}) \geq 1 - 2^{-N^\beta}\}$, for $m \in \{1, 2\}$.

Proof. Note that when we define $Z = XY$, Section 4.1 and polarization theorem 6 applies. From Theorem 6 we have the following fact:

$$\lim_{l \to \infty} \frac{1}{2N} \left| \left\{ k \in [2N] : 2^{-N^\beta} < H(S_k | S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l \to \infty} \frac{|\tilde{\mathcal{L}}_m(\beta)|}{N} = 1 - R'_m, \qquad m = 1, 2,$$

where $\tilde{\mathcal{L}}_m(\beta) = \{k \in [2N] : b_k = m - 1, H(S_k | S^{k-1}) \leq 2^{-N^\beta}\}$ and $R'_m = \frac{1}{N} \sum_{k: b_k = m-1}^{2N} H(S_k | S^{k-1})$, for $m \in \{1, 2\}$. Also, the following is true for R'_m:

$$H(\bar{X}|\bar{Y}) \leq R'_1 \leq H(\bar{X}), \qquad H(\bar{Y}|\bar{X}) \leq R'_2 \leq H(\bar{Y}).$$

The lower and upper bounds of first and second expressions, respectively, are satisfied with path $b^{2N} = 1^N 0^N$. Similarly, the upper and lower bounds of first and second expressions, respectively, are satisfied with path $b^{2N} = 0^N 1^N$.

We also have the following fact from Theorem 6:

$$\lim_{l\to\infty} \frac{1}{2N} \left| \left\{ k \in [2N] : 2^{-N^\beta} < H(S_k|Z^N, S^{k-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l\to\infty} \frac{|\tilde{\mathcal{H}}_m(\beta)|}{N} = R''_m, \qquad m = 1, 2,$$

where $\tilde{\mathcal{H}}_m(\beta) = \{k \in [2N] : b_k = m-1, H(S_k|Z^N, S^{k-1}) \geq 1 - 2^{-N^\beta}\}$ and $R''_m = \frac{1}{N} \sum_{k:b_k=m-1}^{2N} H(S_k|Z^N, S^{k-1})$, for $m \in \{1, 2\}$. Note that, because of the special total probability distribution of the problem R''_1 and R''_2 are constant and not path dependent. The following is true for R''_m:

$$R''_1 = H(\bar{X}|X), \qquad R''_2 = H(\bar{Y}|Y).$$

First define two index sets as follows:

$$\tilde{\mathcal{K}}_m \triangleq \{k \in [2N] : \ b_k = m-1\}, \quad m = 1, 2. \tag{5.27}$$

We define complements of sets for user m with respect to the corresponding index set $\tilde{\mathcal{K}}_m$, i.e. $\tilde{\mathcal{F}}_m^c \triangleq \tilde{\mathcal{K}}_m \setminus \tilde{\mathcal{F}}_m$. Since $H(S_k|Z^N, S^{k-1}) \leq H(S_k|S^{k-1})$, we have $\tilde{\mathcal{L}}_m \cap \tilde{\mathcal{H}}_m = \emptyset$ and $\tilde{\mathcal{H}}_m \subseteq \tilde{\mathcal{L}}_m^c$. We may write $\tilde{\mathcal{I}}'_m = (\tilde{\mathcal{L}}_m \cup \tilde{\mathcal{H}}_m)^c = \tilde{\mathcal{L}}_m^c \setminus \tilde{\mathcal{H}}_m$. $\tilde{\mathcal{I}}'_m$ has some extra partially polarized indices compared to $\tilde{\mathcal{I}}_m$. By polarization, the ratio of the size of the set of partially polarized indices to N go to zero. Thus, the result follows by observing $\frac{|\tilde{\mathcal{I}}'_m|}{N} \to \frac{|\tilde{\mathcal{I}}_m|}{N}$ as $N \to \infty$. □

5.1.1.1 Polarization Sets

In the following discussion, we will refer to three interrelated index variables k, i and j, repeatedly, all in the context of an assumed path b^{2N}. k will mark the index of edge vector S^{2N}. i and j will mark the corresponding index of U^N and

V^N, respectively. We will make use of Definition 10 here.

For the purpose of polar coding, the total probability is also expanded as follows:

$$P_{S^{2N}X^NY^N}(s^{2N}, x^N, y^N) = P_{X^NY^N}(x^N, y^N) \prod_{k=1}^{2N} P_{S_k|S^{k-1}X^NY^N}(s_k|s^{k-1}, x^N, y^N). \tag{5.28}$$

Let $\delta_N = 2^{-N^\beta}$ for $0 < \beta < \frac{1}{2}$. We use Bhattacharyya parameters $Z(\cdot|\cdot)$ instead of entropies $H(\cdot|\cdot)$ when defining polarization sets as we did in previous chapters. Because, $Z(\cdot|\cdot)$ bounds average probability of error by Proposition 1 and $Z(\cdot|\cdot)$ and $H(\cdot|\cdot)$ polarize together by Proposition 2. First, we define the following general "path dependent" polarization sets:

$$\tilde{\mathcal{L}} \triangleq \left\{k \in [2N] : Z(S_k|S^{k-1}) \leq \delta_N\right\}, \tag{5.29}$$

$$\tilde{\mathcal{H}} \triangleq \left\{k \in [2N] : Z(S_k|S^{k-1}, X^N, Y^N) \geq 1 - \delta_N\right\}. \tag{5.30}$$

Then, we define the following "low entropy" sets for user 1:

$$\tilde{\mathcal{L}}_1 \triangleq \left\{k \in [2N] : b_k = 0, k \in \tilde{\mathcal{L}}\right\}, \tag{5.31}$$

$$\tilde{\mathcal{L}}_{\bar{X}} \triangleq \left\{k \in [2N] : b_k = 0, Z(U_i|U^{i-1}) \leq \delta_N\right\}, \tag{5.32}$$

$$\tilde{\mathcal{L}}_{\bar{X}|\bar{Y}} \triangleq \left\{k \in [2N] : b_k = 0, Z(U_i|U^{i-1}, V^N) \leq \delta_N\right\}, \tag{5.33}$$

and user 2:

$$\tilde{\mathcal{L}}_2 \triangleq \left\{k \in [2N] : b_k = 1, k \in \tilde{\mathcal{L}}\right\}, \tag{5.34}$$

$$\tilde{\mathcal{L}}_{\bar{Y}} \triangleq \left\{k \in [2N] : b_k = 1, Z(V_j|V^{j-1}) \leq \delta_N\right\}, \tag{5.35}$$

$$\tilde{\mathcal{L}}_{\bar{Y}|\bar{X}} \triangleq \left\{k \in [2N] : b_k = 1, Z(V_j|V^{j-1}, U^N) \leq \delta_N\right\}. \tag{5.36}$$

The following relations hold for any path b^{2N}

$$\tilde{\mathcal{L}}_{\bar{X}} \subseteq \tilde{\mathcal{L}}_1 \subseteq \tilde{\mathcal{L}}_{\bar{X}|\bar{Y}}, \tag{5.37}$$

$$\tilde{\mathcal{L}}_{\bar{Y}} \subseteq \tilde{\mathcal{L}}_2 \subseteq \tilde{\mathcal{L}}_{\bar{Y}|\bar{X}}. \tag{5.38}$$

Note that $\tilde{\mathcal{L}}_{\bar{X}} = \tilde{\mathcal{L}}_1$ and $\tilde{\mathcal{L}}_2 = \tilde{\mathcal{L}}_{\bar{Y}|\bar{X}}$ for $b^{2N} = 0^N 1^N$. Similarly, $\tilde{\mathcal{L}}_{\bar{X}|\bar{Y}} = \tilde{\mathcal{L}}_1$ and $\tilde{\mathcal{L}}_2 = \tilde{\mathcal{L}}_{\bar{Y}}$ for $b^{2N} = 1^N 0^N$.

Then, we define low entropy sets for users 1 and 2 in terms of i and j indices:

$$\mathcal{L}_1 \triangleq \left\{ i \in [N] : k \in \tilde{\mathcal{L}}_1 \right\}, \qquad \mathcal{L}_2 \triangleq \left\{ j \in [N] : k \in \tilde{\mathcal{L}}_2 \right\}, \tag{5.39}$$

$$\mathcal{L}_{\bar{X}} \triangleq \left\{ i \in [N] : k \in \tilde{\mathcal{L}}_{\bar{X}} \right\}, \qquad \mathcal{L}_{\bar{Y}} \triangleq \left\{ j \in [N] : k \in \tilde{\mathcal{L}}_{\bar{Y}} \right\}, \tag{5.40}$$

$$\mathcal{L}_{\bar{X}|\bar{Y}} \triangleq \left\{ i \in [N] : k \in \tilde{\mathcal{L}}_{\bar{X}|\bar{Y}} \right\}, \quad \mathcal{L}_{\bar{Y}|\bar{X}} \triangleq \left\{ j \in [N] : k \in \tilde{\mathcal{L}}_{\bar{Y}|\bar{X}} \right\}. \tag{5.41}$$

Now we define the high entropy sets as

$$\tilde{\mathcal{H}}_1 \triangleq \left\{ k \in [2N] : b_k = 0, Z(S_k|S^{k-1}, X^N, Y^N) \geq 1 - \delta_N \right\}, \tag{5.42}$$

$$\tilde{\mathcal{H}}_{\bar{X}|X} \triangleq \left\{ k \in [2N] : b_k = 0, Z(U_i|U^{i-1}, X^N) \geq 1 - \delta_N \right\}, \tag{5.43}$$

and

$$\tilde{\mathcal{H}}_2 \triangleq \left\{ k \in [2N] : b_k = 1, Z(S_k|S^{k-1}, X^N, Y^N) \geq 1 - \delta_N \right\}, \tag{5.44}$$

$$\tilde{\mathcal{H}}_{\bar{Y}|Y} \triangleq \left\{ k \in [2N] : b_k = 1, Z(V_j|V^{j-1}, Y^N) \geq 1 - \delta_N \right\}. \tag{5.45}$$

Observe that the following are true for above sets

$$\tilde{\mathcal{H}}_{\bar{X}|X} = \tilde{\mathcal{H}}_1, \text{ and } \tilde{\mathcal{H}}_{\bar{Y}|Y} = \tilde{\mathcal{H}}_2, \tag{5.46}$$

for any path b^{2N}. Similar to above sets, we define the following sets which contain i and j indices:

$$\mathcal{H}_{\bar{X}|X} \triangleq \left\{ i \in [N] : Z(U_i|U^{i-1}, X^N) \geq 1 - \delta_N \right\}, \tag{5.47}$$

$$\mathcal{H}_{\bar{Y}|Y} \triangleq \left\{ j \in [N] : Z(U_i|U^{i-1}, X^N) \geq 1 - \delta_N \right\}. \tag{5.48}$$

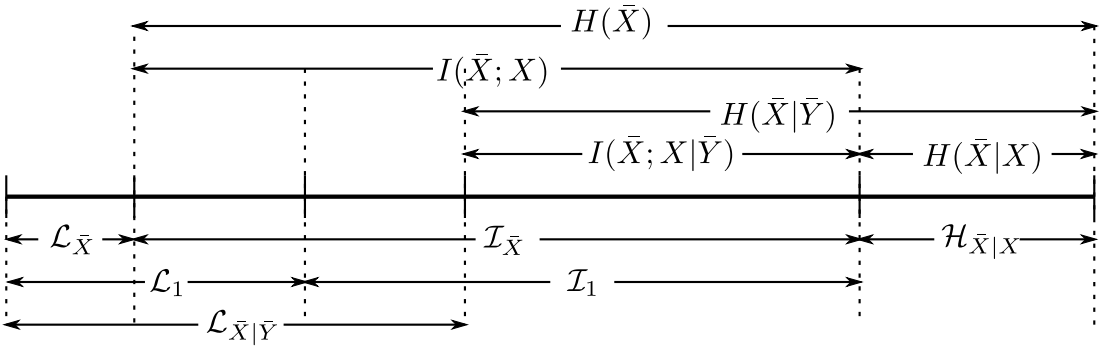


Figure 5.2: Polarization Sets for $\bar{X}$.

Definition 16 (Frozen and Information Sets)**.** *The following frozen sets are defined using the polarization sets defined above:*

$$\mathcal{F}_{\bar{X}} \triangleq \mathcal{L}_{\bar{X}} \cup \mathcal{H}_{\bar{X}|X}, \quad \mathcal{I}_{\bar{X}} \triangleq [N] \setminus \mathcal{F}_{\bar{X}}, \tag{5.49}$$

$$\mathcal{F}_1 \triangleq \mathcal{L}_1 \cup \mathcal{H}_{\bar{X}|X}, \quad \mathcal{I}_1 \triangleq [N] \setminus \mathcal{F}_1, \tag{5.50}$$

$$\mathcal{F}_{\bar{Y}} \triangleq \mathcal{L}_{\bar{Y}} \cup \mathcal{H}_{\bar{Y}|Y}, \quad \mathcal{I}_{\bar{Y}} \triangleq [N] \setminus \mathcal{F}_{\bar{Y}}, \tag{5.51}$$

$$\mathcal{F}_2 \triangleq \mathcal{L}_2 \cup \mathcal{H}_{\bar{Y}|Y}, \quad \mathcal{I}_2 \triangleq [N] \setminus \mathcal{F}_2, \tag{5.52}$$

and

$$\tilde{\mathcal{F}}_{\bar{X}} \triangleq \tilde{\mathcal{L}}_{\bar{X}} \cup \tilde{\mathcal{H}}_{\bar{X}|X}, \quad \tilde{\mathcal{I}}_{\bar{X}} \triangleq \tilde{\mathcal{K}}_1 \setminus \tilde{\mathcal{F}}_{\bar{X}}, \tag{5.53}$$

$$\tilde{\mathcal{F}}_1 \triangleq \tilde{\mathcal{L}}_1 \cup \tilde{\mathcal{H}}_{\bar{X}|X}, \quad \tilde{\mathcal{I}}_1 \triangleq \tilde{\mathcal{K}}_1 \setminus \tilde{\mathcal{F}}_1, \tag{5.54}$$

$$\tilde{\mathcal{F}}_{\bar{Y}} \triangleq \tilde{\mathcal{L}}_{\bar{Y}} \cup \tilde{\mathcal{H}}_{\bar{Y}|Y}, \quad \tilde{\mathcal{I}}_{\bar{Y}} \triangleq \tilde{\mathcal{K}}_2 \setminus \tilde{\mathcal{F}}_{\bar{Y}}, \tag{5.55}$$

$$\tilde{\mathcal{F}}_2 \triangleq \tilde{\mathcal{L}}_2 \cup \tilde{\mathcal{H}}_{\bar{Y}|Y}, \quad \tilde{\mathcal{I}}_2 \triangleq \tilde{\mathcal{K}}_2 \setminus \tilde{\mathcal{F}}_2, \tag{5.56}$$

where $\tilde{\mathcal{K}}_1$ and $\tilde{\mathcal{K}}_2$ are as defined in (5.27)*. Also let*

$$\tilde{\mathcal{F}} \triangleq \tilde{\mathcal{L}} \cup \tilde{\mathcal{H}}, \quad \mathcal{I} \triangleq [2N] \setminus \tilde{\mathcal{F}}. \tag{5.57}$$

Note that, $\tilde{\mathcal{F}} = \tilde{\mathcal{F}}_1 \cup \tilde{\mathcal{F}}_2$ and $\tilde{\mathcal{I}} = \tilde{\mathcal{I}}_1 \cup \tilde{\mathcal{I}}_2$.

5.1.1.2 Encoding

We define family of functions $\lambda_i^{(1)} : \bar{\mathcal{X}}^{i-1} \to \bar{\mathcal{X}}$, $\forall i \in \mathcal{F}_{\bar{X}}$ and $\lambda_j^{(2)} : \bar{\mathcal{X}}^{j-1} \to \bar{\mathcal{X}}$, $\forall j \in \mathcal{F}_{\bar{Y}}$. We assume that they are shared between the encoders and the decoder. We also define the corresponding random variables $\Lambda_i^{(1)}$ and $\Lambda_j^{(2)}$ such that

$$\begin{aligned} \Lambda_i^{(1)}(u^{i-1}) &\triangleq a, \quad \text{w.p. } P_{U_i|U^{i-1}}\left(a|u^{i-1}\right), \\ \Lambda_j^{(2)}(v^{j-1}) &\triangleq a, \quad \text{w.p. } P_{V_j|V^{j-1}}\left(a|v^{j-1}\right), \end{aligned} \tag{5.58}$$

where $a \in \bar{\mathcal{X}}$. Maps $(\lambda_i^{(1)}, \lambda_j^{(2)})$ are the realizations of random maps $(\Lambda_i^{(1)}, \Lambda_j^{(2)})$. Each realization of set of maps $(\lambda_{\mathcal{F}_{\bar{X}}}^{(1)}, \lambda_{\mathcal{F}_{\bar{Y}}}^{(2)})$ results in different encoding and decoding protocols. The distribution over the choice of maps is induced with the above equation (5.58). The set of maps $(\lambda_{\mathcal{F}_{\bar{X}}}^{(1)}, \lambda_{\mathcal{F}_{\bar{Y}}}^{(2)})$ are used to determine the bits in sets $\mathcal{F}_{\bar{X}}, \mathcal{F}_{\bar{Y}}$. The theoretical analysis of the distortions are made much easier using the randomized maps and calculating the average distortion over maps.

The bits in *information* sets $\mathcal{I}_{\bar{X}}$ and $\mathcal{I}_{\bar{Y}}$ are calculated either the deterministic or the random rules given below.

Deterministic rules:

$$\begin{aligned} \bar{\psi}_i^{(1)}(u^{i-1}, x^N) &\triangleq \arg\max_{u' \in \bar{\mathcal{X}}} \left\{ P_{U_i|U^{i-1}X^N}\left(u'|u^{i-1}, x^N\right) \right\}, \\ \bar{\psi}_j^{(2)}(v^{j-1}, y^N) &\triangleq \arg\max_{v' \in \bar{\mathcal{X}}} \left\{ P_{V_j|V^{j-1}Y^N}\left(v'|v^{j-1}, y^N\right) \right\}. \end{aligned} \tag{5.59}$$

Random rules:

$$\begin{aligned} \Psi_i^{(1)}(u^{i-1}, x^N) &\triangleq a, \quad \text{w.p. } P_{U_i|U^{i-1}X^N}\left(a|u^{i-1}, x^N\right), \\ \Psi_j^{(2)}(v^{j-1}, y^N) &\triangleq a, \quad \text{w.p. } P_{V_i|V^{j-1}Y^N}\left(a|v^{j-1}, y^N\right), \end{aligned} \tag{5.60}$$

where $a \in \bar{\mathcal{X}}$. Maps $(\psi_i^{(1)}, \psi_j^{(2)})$ are the realizations of random maps $(\Psi_i^{(1)}, \Psi_j^{(2)})$. In the analysis we use the random rules for tractability. This approach is called *randomized rounding* [16]. The encoding operations are given as follows.

Encoder 1 constructs the sequence u^N bit-by-bit successively,

$$u_i = \begin{cases} \lambda_i^{(1)}(u^{i-1}), & \text{if } i \in \mathcal{F}_{\bar{X}}, \\ \psi_i^{(1)}(u^{i-1}, x^N), & \text{otherwise.} \end{cases} \tag{5.61}$$

Encoder 2 constructs the sequence v^N bit-by-bit successively,

$$v_j = \begin{cases} \lambda_j^{(2)}(v^{j-1}), & \text{if } j \in \mathcal{F}_{\bar{Y}}, \\ \psi_j^{(2)}(v^{j-1}, y^N), & \text{otherwise.} \end{cases} \tag{5.62}$$

Then, encoder 1 transmits the compressed message $u_{\mathcal{I}_1}$ and encoder 2 transmits the compressed message $v_{\mathcal{I}_2}$. The randomness in the encoding process ensures that bits of u^N and v^N have the correct statistics as if drawn from the joint distribution of (U^N, V^N).

Remark 1. *Note that although in the analysis we use randomized rounding approach and thus make use of random rules $\Psi_i^{(1)}$ and $\Psi_j^{(2)}$ for calculating bits in $\mathcal{I}_{\bar{X}}$ and $\mathcal{I}_{\bar{Y}}$, in practice we use the deterministic rules. In either case, the probabilities $P(u_i|u^{i-1}, x^N)$ and $P(v_j|v^{j-1}, y^N)$ have to be calculated. These are calculated using SC decoding. Therefore, SC decoders are employed at the encoders just like single user rate-distortion coding. Thus, we refer to this operation as SC encoding.*

Remark 2. *If the distribution of $\bar{X}$ is uniform then we could determine the bits in $\mathcal{F}_{\bar{X}}$ beforehand uniformly from $\bar{\mathcal{X}}^{|\mathcal{F}_{\bar{X}}|}$ and the randomized maps $\Lambda_{\mathcal{F}_{\bar{X}}}^{(1)}$ are not required. In general case however, the set $\mathcal{F}_{\bar{X}}$ actually comprises of two distinct parts and we could use a simplified rule for $i \in \mathcal{F}_{\bar{X}}$:*

$$u_i = \begin{cases} \bar{u}_i, & \text{if } i \in \mathcal{H}_{\bar{X}|X}, \\ \arg\max_{u' \in \bar{\mathcal{X}}} P_{U_i|U^{i-1}}(u'|u^{i-1}), & \text{if } i \in \mathcal{L}_{\bar{X}}, \end{cases} \tag{5.63}$$

where $\bar{u}_i$ is determined beforehand uniformly from $\bar{\mathcal{X}}$. The same reasoning applies to $\bar{Y}$, too. However, since this rule makes the proof harder, we use the maps $\Lambda_{\mathcal{F}_{\bar{X}}}^{(1)}$ for simplicity. But, in simulations, the above presented rules are used.

5.1.1.3 Decoding

Joint decoding is performed at the decoder along the path b^{2N}. Decoding is performed in $2N$ steps. In step $k \in [2N]$, if $b_k = 0$ then a bit from u^N is decoded else a bit from v^N is decoded. We define the following decoding functions:

$$\zeta_k(s^{k-1}) \triangleq \arg\max_{s' \in \bar{\mathcal{X}}} \left\{ P_{S_k|S^{k-1}}(s'|s^{k-1}) \right\}, \tag{5.64}$$

for $k \in \tilde{\mathcal{I}}$.

First, the decoder assembles the received vectors $(u_{\mathcal{I}_1}, v_{\mathcal{I}_2})$ into $s_{\tilde{\mathcal{I}}}$. Then, the decoder uses the identical shared maps $\lambda_i^{(1)}$ and $\lambda_j^{(2)}$ to reconstruct the estimate $\hat{s}^{2N}$ *successively* as

$$\hat{s}_k = \begin{cases} \lambda_i^{(1)}(\hat{u}^{i-1}), & \text{if } k \in \tilde{\mathcal{F}}_{\bar{X}}, \\ \lambda_j^{(2)}(\hat{v}^{j-1}), & \text{if } k \in \tilde{\mathcal{F}}_{\bar{Y}}, \\ s_k, & \text{if } k \in \tilde{\mathcal{I}}, \\ \zeta_k(\hat{s}^{k-1}), & \text{otherwise.} \end{cases} \tag{5.65}$$

Then, $\hat{u}^N$ and $\hat{v}^N$ are extracted as $\hat{u}^N = \hat{s}_{\tilde{\mathcal{K}}_1}$ and $\hat{v}^N = \hat{s}_{\tilde{\mathcal{K}}_2}$. Finally, the estimations are generated as $\hat{x}^N = \hat{x}(\hat{u}^N G_N, \hat{v}^N G_N)$ and $\hat{y}^N = \hat{y}(\hat{u}^N G_N, \hat{v}^N G_N)$.

Note that the encoding operations are almost the same as single-user rate distortion coding. The difference is that only a subset $u_{\mathcal{I}_1}$ $(v_{\mathcal{I}_2})$ of $u_{\mathcal{I}_{\bar{X}}}$ $(v_{\mathcal{I}_{\bar{Y}}})$ (all bits generated by polar successive cancellation encoding operation) is sent to the decoder. The rest can be estimated with the combined knowledge of $(u_{\mathcal{I}_1}, v_{\mathcal{I}_2})$. Thus, the decoder is very different compared to single-user rate-distortion polar coding where a simple polar encoder was used. Here, two-user polar successive cancellation decoding is used at the decoder. Therefore, in addition to bounding the average distortions, we need to bound the average probability of decoding error, too. As in the single-user case, for analysis purposes, the encoding functions are random. The results of encoding operations may be different for the same inputs (x^N, y^N). For encoder 1, at step $i \in \mathcal{I}_{\bar{X}}$ of the process, $u_i = a$ with probability proportional to $P_{U_i|U^{i-1}X^N}(a|u^{i-1}, x^N)$. Similarly for encoder 2, at step $j \in \mathcal{I}_{\bar{Y}}$

of the process, $v_j = a$ with probability proportional to $P_{V_j|V^{j-1}Y^N}(a|v^{j-1}, y^N)$. Thus, for a given pair of maps $(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$, a particular (u^N, v^N) occurs with a certain probability induced by the distributions of $(\Psi^{(1)}_{\mathcal{I}_{\bar{X}}}, \Psi^{(2)}_{\mathcal{I}_{\bar{Y}}})$ and maps.

We define the resulting average (over x^N, y^N and randomness of the information bits induced by the distributions of $(\Psi^{(1)}_{\mathcal{I}_{\bar{X}}}, \Psi^{(2)}_{\mathcal{I}_{\bar{Y}}})$) distortions of above encoding and decoding operations as $D_x(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$ and $D_y(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$. In the following we show that for sets $\mathcal{F}_{\bar{X}}$, $\mathcal{F}_{\bar{Y}}$, $\mathcal{I}_1$, $\mathcal{I}_2$ defined in 5.1.1.1 and encoding and decoding methods defined in 5.1.1.2 and 5.1.1.3, there exists maps $(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$ such that $D_x(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}) \sim D_x^*$ and $D_y(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}) \sim D_y^*$, where D_x^*, D_y^* are the design distortions. We do that by determining the *expected* average distortions over the ensembles of codes generated by different encoding maps $(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$. The distribution over the choices of maps is given in (5.58). Then we show that *expected* average distortions are roughly D_x^* and D_y^*. This implies that for at least one choice of $(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$ the average distortions are close to D_x^* and D_y^*. The following theorem makes this precise.

Theorem 12. *Let $\mathcal{F}_{\bar{X}}$, $\mathcal{F}_{\bar{Y}}$, $\mathcal{I}_1$, $\mathcal{I}_2$ be sets as defined in 5.1.1.1 and encoding and decoding methods be as defined in 5.1.1.2 and 5.1.1.3. Then the expectations of average distortions $D_x(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$, $D_y(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$ over the maps $\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}$, $\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}$ satisfy $\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}\left[D_x(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})\right] = D_x^* + O(2^{-N^\beta})$ and $\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}\left[D_y(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})\right] = D_y^* + O(2^{-N^\beta})$ for any $(R_1, R_2) \in \mathcal{R}_{BT}$ and $0 < \beta < 1/2$. Consequently, there exist deterministic maps that satisfy the above relations.*

The following sections give necessary steps for proving the theorem. We first prove a total variation bound on two probability measures. Then, we use that result to bound the *expected* average distortions of the code. However, before showing the average distortions are bounded, we need to show that the decoding error is bounded, which is done in similar steps as the MAC case in Section 4.3.1.5.

5.1.1.4 Total Variation Bound

First, we define the following probability measure:

$$\begin{aligned} Q_{S^{2N}X^NY^N}(s^{2N},x^N,y^N) &\triangleq Q_{X^NY^N}(x^N,y^N)Q_{S^{2N}|X^NY^N}(s^{2N}|x^N,y^N) \\ &= Q_{X^NY^N}(x^N,y^N)\prod_{k=1}^{2N} Q_{S_k|S^{k-1}X^NY^N}(s_k|s^{k-1},x^N,y^N), \end{aligned} \tag{5.66}$$

where $Q_{X^NY^N}(x^N,y^N) = P_{X^NY^N}(x^N,y^N)$. Also we have the following relation:

$$Q_{U^NV^NX^NY^N}(u^N,v^N,x^N,y^N) \triangleq Q_{S^{2N}X^NY^N}(\pi_N(u^N,v^N),x^N,y^N). \tag{5.67}$$

The conditional probability measures are defined as

$$Q_{S_k|S^{k-1}X^NY^N}(s_k|s^{k-1},x^N,y^N) \triangleq \begin{cases} P_{U_i|U^{i-1}}(u_i|u^{i-1}), & b_k = 0 \text{ and } k \in \tilde{\mathcal{F}}_{\bar{X}}, \\ P_{U_i|U^{i-1}X^N}(u_i|u^{i-1},x^N), & b_k = 0 \text{ and } k \in \tilde{\mathcal{I}}_{\bar{X}}, \\ P_{V_j|V^{j-1}}(v_j|v^{j-1}), & b_k = 1 \text{ and } k \in \tilde{\mathcal{F}}_{\bar{Y}}, \\ P_{V_j|V^{j-1}Y^N}(v_j|v^{j-1},y^N), & b_k = 1 \text{ and } k \in \tilde{\mathcal{I}}_{\bar{Y}}, \end{cases} \tag{5.68}$$

Lemma 9 (Total Variation Bound)**.** *Let probability measures P and Q be defined as in* (5.28) *and* (5.66)*, respectively. For $0 < \beta < 1/2$ and sufficiently large N, the total variation distance between P and Q is bounded as*

$$\sum_{s^{2N},x^N,y^N} \left| P_{S^{2N}X^NY^N}(s^{2N},x^N,y^N) - Q_{S^{2N}X^NY^N}(s^{2N},x^N,y^N) \right| \leq 2^{-N^\beta}. \tag{5.69}$$

Proof. See Appendix C.1. □

5.1.1.5 Average Error Probability

The encoding and decoding rules were established in Sections 5.1.1.2 and 5.1.1.3, respectively. The two-user decoder defined in Section 5.1.1.3 is used to estimate unknown values. s_k for $k \in \{\tilde{\mathcal{F}}_{\bar{X}} \cup \tilde{\mathcal{F}}_{\bar{Y}}\}$ are decided using shared maps $\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}$, $\lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}$. s_k for $k \in \tilde{\mathcal{I}}$ are received from encoders. The remaining bits in $\{\tilde{\mathcal{I}}_{\bar{X}} \cup \tilde{\mathcal{I}}_{\bar{Y}}\} \setminus \tilde{\mathcal{I}}$ are estimated by the joint decoder. Note that, that set is also given by $\tilde{\mathcal{L}} \setminus \{\tilde{\mathcal{L}}_{\bar{X}} \cup \tilde{\mathcal{L}}_{\bar{Y}}\}$. Consider the sequences $s^{2N} = \pi_N(u^N, v^N)$ formed at the encoders under observations x^N and y^N. The decoder makes an SC decoding error on the k-th symbol for the following sequences:

$$\mathcal{T}^k \triangleq \big\{(s^{2N}, x^N, y^N) : \exists s' \in \bar{\mathcal{X}} \text{ s.t. } s' \neq s_k, \\ P_{S_k|S^{k-1}}(s_k|s^{k-1}) \leq P_{S_k|S^{k-1}}(s'|s^{k-1})\big\}. \quad (5.70)$$

The set $\mathcal{T}^k$ represents those tuples causing an error at the decoder in the case s_k is inconsistent with respect to the decoding rule. The complete set of tuples causing an error is

$$\mathcal{T} \triangleq \bigcup_{k \in \tilde{\mathcal{L}} \setminus \{\tilde{\mathcal{L}}_{\bar{X}} \cup \tilde{\mathcal{L}}_{\bar{Y}}\}} \mathcal{T}^k. \quad (5.71)$$

Assuming randomized maps shared between encoder and decoder, the average error probability is a random quantity given as follows

$$P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}] = \sum_{(s^{2N}, x^N, y^N) \in \mathcal{T}} P_{X^N, Y^N}(x^N, y^N) \\ \cdot \prod_{i \in \mathcal{I}_{\bar{X}}} P_{U_i|U^{i-1}X^N}(u_i|u^{i-1}, x^N) \prod_{j \in \mathcal{I}_{\bar{Y}}} P_{V_j|V^{j-1}Y^N}(v_j|v^{j-1}, y^N) \\ \cdot \prod_{i \in \mathcal{F}_{\bar{X}}} \mathbb{1}_{\left\{\Lambda^{(1)}_i(u^{i-1}) = u_i\right\}} \prod_{j \in \mathcal{F}_{\bar{Y}}} \mathbb{1}_{\left\{\Lambda^{(2)}_j(v^{j-1}) = v_j\right\}}. \quad (5.72)$$

The expected average block error probability is calculated by averaging over the randomness in the encoders

$$\bar{P}_e \triangleq \mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}} \left[P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}]\right]. \quad (5.73)$$

The following lemma bounds the expected average block error probability.

Lemma 10. *Consider the polarization based channel code described in Sections 5.1.1.2 and 5.1.1.3. Let the sets $\mathcal{F}_{\bar{X}}$, $\mathcal{F}_{\bar{Y}}$, $\mathcal{I}_1$, $\mathcal{I}_2$ be as defined in 5.1.1.1. Then for $0 < \beta < 1/2$ and sufficiently large N,*

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}} \left[P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}] \right] \leq 2^{-N^\beta}.$$

Proof. Note that the expectation of average probability of error is written as

$$\begin{aligned}\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}} \left[P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}] \right] = & \sum_{(s^{2N}, x^N, y^N) \in \mathcal{T}} P_{X^N, Y^N}(x^N, y^N) \\ & \cdot \prod_{i \in \mathcal{I}_{\bar{X}}} P_{U_i | U^{i-1} X^N}(u_i | u^{i-1}, x^N) \prod_{j \in \mathcal{I}_{\bar{Y}}} P_{V_j | V^{j-1} Y^N}(v_j | v^{j-1}, y^N) \\ & \cdot \prod_{i \in \mathcal{F}_{\bar{X}}} \mathbb{P}\left\{ \Lambda^{(1)}_i(u^{i-1}) = u_i \right\} \prod_{j \in \mathcal{F}_{\bar{Y}}} \mathbb{P}\left\{ \Lambda^{(2)}_j(v^{j-1}) = v_j \right\}.\end{aligned}$$

From the definition of random mappings it follows that

$$\begin{aligned}\mathbb{P}\left\{ \Lambda^{(1)}_i(u^{i-1}) = u_i \right\} &= P_{U_i | U^{i-1}}(u_i | u^{i-1}), \\ \mathbb{P}\left\{ \Lambda^{(2)}_j(v^{j-1}) = v_j \right\} &= P_{V_j | V^{j-1}}(v_j | v^{j-1}).\end{aligned}$$

Then, we may substitute the definition for $Q_{S^{2N} | X^N Y^N}(s^{2N} | x^N, y^N)$ in (5.66) into the expression of expected average probability of error to get

$$\begin{aligned}\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}} \left[P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}] \right] = & \sum_{(s^{2N}, x^N, y^N) \in \mathcal{T}} P_{X^N, Y^N}(x^N, y^N) \cdot \\ & Q_{S^{2N} | X^N Y^N}(s^{2N} | x^N, y^N).\end{aligned}$$

Then we split the error into two main parts, one due to the polar decoding function and the other due to the total variation distance between probability

measures.

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}\left[P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}]\right] = \sum_{(s^{2N},x^N,y^N)\in\mathcal{T}} P_{X^N,Y^N}(x^N,y^N)\cdot \left[Q(s^{2N}|x^N,y^N) - P(s^{2N}|x^N,y^N) + P(s^{2N}|x^N,y^N)\right].$$

Then we have

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}\left[P_e[\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}]\right] \leq \sum_{(s^{2N},x^N,y^N)\in\mathcal{T}} P_{S^{2N}X^N,Y^N}(s^{2N},x^N,y^N) + \sum_{s^{2N},x^N,y^N} \left|Q(s^{2N},x^N,y^N) - P(s^{2N},x^N,y^N)\right|.$$

The second part of the error which is due to total variation distance is upper bounded as $O(2^{-N^\beta})$ by Lemma 9. Thus, it remains to upper bound the error term due to polar decoding. Remember that $\mathcal{T} \triangleq \cup_{k\in\tilde{\mathcal{L}}\setminus\{\tilde{\mathcal{L}}_{\bar{X}}\cup\tilde{\mathcal{L}}_{\bar{Y}}\}}\mathcal{T}^k$. We may upper bound each error symbol by symbol. Define error probability for symbol $k \in \tilde{\mathcal{L}} \setminus \{\tilde{\mathcal{L}}_{\bar{X}} \cup \tilde{\mathcal{L}}_{\bar{Y}}\}$ as

$$\begin{aligned}\varepsilon^k &\triangleq \sum_{(s^{2N},x^N,y^N)\in\mathcal{T}^k} P_{S^{2N}X^NY^N}(s^{2N},x^N,y^N),\\ &= \sum_{s^k} P_{S^K}(s^k)\cdot \mathbb{1}_{\left\{s':P_{S_k|S^{k-1}}(s_k|s^{k-1})\leq P_{S_k|S^{k-1}}(s'|s^{k-1})\right\}}.\end{aligned}$$

But this is the average probability of decoding error for symbol k, i.e. $\varepsilon^k = P_e(S_k|S^{k-1})$. Probability of error is upper bounded by the Bhattacharyya parameter by Proposition 1. By union bound, the total average probability of error is $\varepsilon \leq \sum_k \varepsilon^k$. Then we have

$$\begin{aligned}\varepsilon &\leq \sum_{k\in\tilde{\mathcal{L}}\setminus\{\tilde{\mathcal{L}}_{\bar{X}}\cup\tilde{\mathcal{L}}_{\bar{Y}}\}} (q-1)Z(S_k|S^{k-1}),\\ &\leq (q-1)2N\delta_N.\end{aligned}$$

The second inequality is from the definition of polarization sets in Section 5.1.1.1. This completes the proof that the expected average probability of error is upper bounded as $O(2^{-N^\beta})$. □

Since by Lemma 10 the expected value over the random maps of average probability of error decays to zero, there must be at least one deterministic class of maps for which $P_e \to 0$.

5.1.1.6 Average Distortion

For a source sequence (x^N, y^N), random encoding maps $(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$ and encoding rule (5.61), (u^N, v^N) appears with probability

$$\left(\prod_{i\in\mathcal{I}_{\bar{X}}} P_{U_i|U^{i-1},X^N}(u_i|u^{i-1},x^N)\right)\left(\prod_{i\in\mathcal{F}_{\bar{X}}} \mathbb{1}_{\left\{\Lambda_i^{(1)}(u^{i-1})=u_i\right\}}\right)\cdot \left(\prod_{j\in\mathcal{I}_{\bar{Y}}} P_{V_j|V^{j-1},Y^N}(v_j|v^{j-1},y^N)\right)\left(\prod_{j\in\mathcal{F}_{\bar{Y}}} \mathbb{1}_{\left\{\Lambda_j^{(2)}(v^{j-1})=v_j\right\}}\right).$$

For random set of maps $(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})$, the average distortion of X is a random quantity given by

$$D_x(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}}, \Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}) = \sum_{\substack{u^N,v^N\\ x^N,y^N}} P_{X^NY^N}(x^N,y^N)\cdot \mathrm{d_x}(x^N,\hat{x}(u^NG_N,v^NG_N)) \left(\prod_{i\in\mathcal{I}_{\bar{X}}} P_{U_i|U^{i-1},X^N}(u_i|u^{i-1},x^N)\right)\left(\prod_{i\in\mathcal{F}_{\bar{X}}} \mathbb{1}_{\left\{\Lambda_i^{(1)}(u^{i-1})=u_i\right\}}\right)\cdot \left(\prod_{j\in\mathcal{I}_{\bar{Y}}} P_{V_j|V^{j-1},Y^N}(v_j|v^{j-1},y^N)\right)\left(\prod_{j\in\mathcal{F}_{\bar{Y}}} \mathbb{1}_{\left\{\Lambda_j^{(2)}(v^{j-1})=v_j\right\}}\right). \tag{5.74}$$

The expectation over maps is

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}[D_x(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})] = \sum_{\substack{u^N,v^N\\x^N,y^N}} P_{X^NY^N}(x^N,y^N)\cdot \mathrm{d_x}(x^N,\hat{x}(u^NG_N,v^NG_N))\cdot \left(\prod_{i\in\mathcal{I}_{\bar{X}}} P_{U_i|U^{i-1},X^N}(u_i|u^{i-1},x^N)\right)\left(\prod_{i\in\mathcal{F}_{\bar{X}}} P_{U_i|U^{i-1}}(u_i|u^{i-1})\right)\cdot \left(\prod_{j\in\mathcal{I}_{\bar{Y}}} P_{V_j|V^{j-1},Y^N}(v_j|v^{j-1},y^N)\right)\left(\prod_{j\in\mathcal{F}_{\bar{Y}}} P_{V_j|V^{j-1}}(v_j|v^{j-1})\right). \tag{5.75}$$

Using the probability distribution Q defined in (5.66) we can write the expectation as

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}\left[D_x(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})\right] = \mathbb{E}_Q\left[\mathrm{d_x}(X^N,\hat{x}(U^NG_N,V^NG_N))\right]. \tag{5.76}$$

Therefore, we get

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}\}}\left[D_x(\Lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\Lambda^{(2)}_{\mathcal{F}_{\bar{Y}}})\right] \le \mathbb{E}_P\left[\mathrm{d_x}(X^N,\hat{x}(U^NG_N,V^NG_N))\right] + \mathrm{d_{max}}||P_{S^{2N}X^NY^N} - Q_{S^{2N}X^NY^N}||. \tag{5.77}$$

Lemma 9 shows that the second term of the sum is $O(2^{-N^\beta})$. Therefore, there exist deterministic sets of maps $\lambda^{(1)}_{\mathcal{F}_{\bar{X}}}$ and $\lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}$ such that $D_x(\lambda^{(1)}_{\mathcal{F}_{\bar{X}}},\lambda^{(2)}_{\mathcal{F}_{\bar{Y}}}) = D^*_x + O(2^{-N^\beta})$. Similarly we can prove the same result for Y distortion.

5.1.2 Simulations

In this section, we implement the proposed polar coding scheme for distributed lossy compression and present simulation results for binary sources with Hamming

distortion measures. That is, $\mathrm{d}_x = \mathrm{d}_y = \mathrm{d}_H$ where

$$\mathrm{d}_H(x, \hat{x}) = \begin{cases} 0, & \text{if } x = \hat{x}, \\ 1, & \text{otherwise.} \end{cases} \tag{5.78}$$

The practical encoder implements the function in (5.63). The symbols for $\mathcal{H}_{\bar{X}|X}$ and $\mathcal{H}_{\bar{Y}|Y}$ are fixed to zero and ML encoding is used for the rest of the bits instead of randomized rounding. The practical decoder uses joint ML SC decoding for all of the bits except the known bits of sets $\mathcal{H}_{\bar{X}|X}$ and $\mathcal{H}_{\bar{Y}|Y}$.

Recall that the total probability distribution for Berger-Tung coding is of the form $P_{\bar{X}\bar{Y}XY}(\bar{x}, \bar{y}, x, y) = P_{\bar{X}|X}(\bar{x}|x)P_{\bar{Y}|Y}(\bar{y}|y)P_{XY}(x, y)$. $P_{XY}(x, y)$ is fixed for the given source. Each different selection of $P_{\bar{X}|X}(\bar{x}|x)$ and $P_{\bar{Y}|Y}(\bar{y}|y)$ distributions result in a different achievable region. However, that selection is not totally arbitrary; The resulting distribution must satisfy the distortion constraints:

$$\mathbb{E}[\mathrm{d}_x(X^n, \hat{X}^n)] = D_x, \quad \mathbb{E}[\mathrm{d}_y(Y^n, \hat{Y}^n)] = D_y. \tag{5.79}$$

5.1.2.1 Simulation 1

For this simulation we use the estimator functions $\hat{x}(\bar{x}, \bar{y}) = \bar{x}$ and $\hat{y}(\bar{x}, \bar{y}) = \bar{y}$. For the probability distributions used in this simulation it turns out that these estimators are optimal, i.e.

$$\arg\max_{x'} P_{X|\bar{X}\bar{Y}}(x'|\bar{x}, \bar{y}) = \bar{x}, \qquad \arg\max_{y'} P_{Y|\bar{X}\bar{Y}}(y'|\bar{x}, \bar{y}) = \bar{y}.$$

For our case with binary sources and Hamming distortion, above estimator results in the following equations:

$$P_{\bar{X}|X}(0|1) = \frac{D_x + P_X(0)(P_{\bar{X}|X}(0|0) - 1)}{1 - P_X(0)}, \tag{5.80}$$

$$P_{\bar{Y}|Y}(0|1) = \frac{D_y + P_Y(0)(P_{\bar{Y}|Y}(0|0) - 1)}{1 - P_Y(0)}. \tag{5.81}$$

Thus, $P_{\bar{X}|X}(0|0)$ is the only free variable in conditional distribution $P_{\bar{X}|X}$. The similar result is also valid for conditional distribution $P_{\bar{Y}|Y}$. In the simulations below, when selecting conditional distributions, we took this constraint into account and optimized the *sum-rate* ($I(\bar{X}, \bar{Y}; X, Y)$) over these two free variables.

For this simulation, the source distribution is selected as

$$P_{XY} = \begin{bmatrix} 0.50 & 0.15 \\ 0.05 & 0.30 \end{bmatrix}.$$

And the average distortion constraints are set to $D_x = 0.05$ and $D_y = 0.05$. The conditional distributions are selected with the average distortion constraints mentioned above and optimized to minimize the total sum rate. The graph of sum-rate versus $P_{\bar{X}|X}(0|0)$ and $P_{\bar{Y}|Y}(0|0)$ is given in Figure 5.3. For the simulations we select the parameters that minimize the sum rate which occurs at $P_{\bar{X}|X}(0|0) = 0.97$, $P_{\bar{Y}|Y}(0|0) = 0.96$. The corresponding conditional distributions are given in Table 5.1.

Table 5.1: Conditional probabilities $P_{\bar{X}|X}$ and $P_{\bar{Y}|Y}$.

$\mathbf{a}, \mathbf{b}$	$\mathbf{0}, \mathbf{0}$	$\mathbf{0}, \mathbf{1}$	$\mathbf{1}, \mathbf{0}$	$\mathbf{1}, \mathbf{1}$
$\mathbf{P_{\bar{X}|X}(a|b)}$	0.9700	0.0871	0.0300	0.9129
$\mathbf{P_{\bar{Y}|Y}(a|b)}$	0.9600	0.0622	0.0400	0.9378

The mutual information parameters calculated for this source distribution are given in Table 5.2. As it can be seen from the table, if we were to encode X alone we would need a rate of $I(X; \bar{X}) = 0.6481$ and similarly for Y alone we would need a rate of $I(Y; \bar{Y}) = 0.7064$. The total rate would be 1.3545. However, because of the correlation and joint decoding the sum rate of Berger-Tung region is $I(\bar{X}, \bar{Y}; X, Y) = 1.1821$ which is $I(\bar{X}; \bar{Y}) = 0.1723$ less. The corner points of the Berger-Tung region are given as $(0.6481, 0.5243)$ and $(0.4758, 0.7064)$. In the simulations, the distortions are averaged over 1000 blocks. The tables show the experimental rates required to obtain the target distortions approximately.

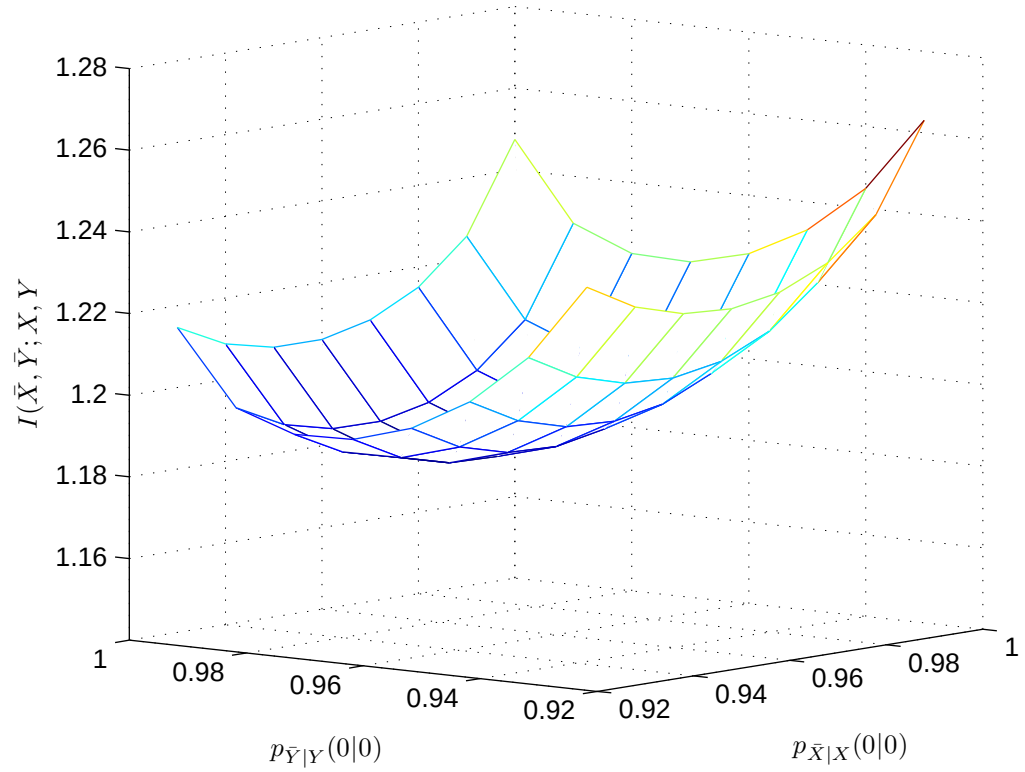


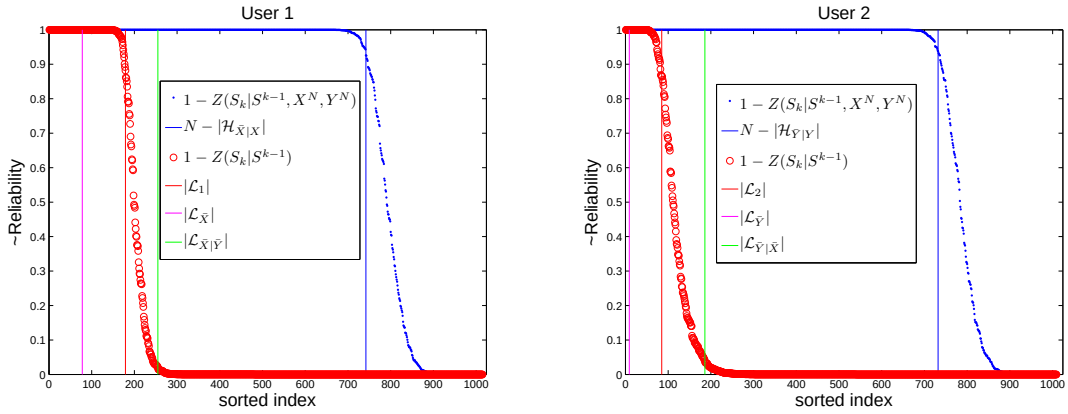
Figure 5.3: $I(\bar{X},\bar{Y};X,Y)$ vs. $P_{\bar{X}|X}(0|0)$ and $P_{\bar{Y}|Y}(0|0)$.

Table 5.2: Berger-Tung parameters.

$\mathbf{I}(\mathbf{X};\bar{\mathbf{X}})$	$\mathbf{I}(\mathbf{Y};\bar{\mathbf{Y}})$	$\mathbf{I}(\bar{\mathbf{X}};\bar{\mathbf{Y}})$	$\mathbf{I}(\bar{\mathbf{X}},\bar{\mathbf{Y}};\mathbf{X},\mathbf{Y})$
0.6481	0.7064	0.1723	1.1821

Construction

Code construction is done using two-user SC decoder in large number of Monte-Carlo simulations and averaging the results. The joint decoder runs in two different configurations, once for likelihoods calculated for known (X^N, Y^N) and once for (X^N, Y^N) unknown. The decoder runs along the given path (b^{2N}) and calculate reliability values of the bits. Figure 5.4 shows results for path $b^{2N} = 0^{\frac{3N}{4}}1^N0^{\frac{N}{4}}$ with $N = 2^{10}$. The rate allocation for this path is measured from the results of simulations as $(R_1, R_2) = (0.5514, 0.6307)$.



(a) User 1 (b) User 2

Figure 5.4: Sorted reliability values.

The output of the first run (known (X^N, Y^N)) gives approximately $1 - Z(S_k|S^{k-1}, X^N, Y^N)$. It is shown with dotted plot and blue color in both figures. Figures 5.4a and 5.4b show only those indices with $b_k = 0$ (user 1) and $b_k = 1$ (user 2), respectively. We identify the sets $\mathcal{H}_{\bar{X}|X}$ and $\mathcal{H}_{\bar{Y}|Y}$ using this simulation. The number of indices close to 1 in those plots are expected to be $N - |\mathcal{H}_{\bar{X}|X}|$ and $N - |\mathcal{H}_{\bar{Y}|Y}|$, which are marked with vertical blue solid lines.

The output of the second run (unknown (X^N, Y^N)) gives approximately $1 - Z(S_k|S^{k-1})$. The output is plotted with small circles and red color in both figures. Figures 5.4a and 5.4b show only those indices with $b_k = 0$ (user 1) and $b_k = 1$ (user 2), respectively. We identify the sets $\mathcal{L}_1$ and $\mathcal{L}_2$ using this simulation. For Figure 5.4a, the number of indices close to 1 depends on the path (b^{2N}) chosen and gives the size of set $\mathcal{L}_1$ (marked with vertical red line). The size of $\mathcal{L}_1$ must be between $|\mathcal{L}_{\bar{X}}|$ $(b^{2N} = 0^N 1^N)$ and $|\mathcal{L}_{\bar{X}|\bar{Y}}|$ $(b^{2N} = 1^N 0^N)$ marked with vertical magenta and green lines, respectively. Similarly, for Figure 5.4b, the number of indices close to 1 depends on the path (b^{2N}) chosen and gives the size of set $\mathcal{L}_2$ (marked with vertical red line). The size of $\mathcal{L}_2$ must be between $|\mathcal{L}_{\bar{Y}}|$ $(b^{2N} = 1^N 0^N)$ and $|\mathcal{L}_{\bar{Y}|\bar{X}}|$ $(b^{2N} = 0^N 1^N)$ marked with vertical magenta and green lines, respectively.

As the result of the code construction simulations, we are able to calculate the sets given in previous sections which are necessary to perform encoding and decoding such as $\mathcal{L}_{\bar{X}}$, $\mathcal{H}_{\bar{X}|X}$, $\mathcal{L}_1$, $\mathcal{I}_1$, $\mathcal{L}_{\bar{Y}}$, $\mathcal{H}_{\bar{Y}|Y}$, $\mathcal{L}_2$, $\mathcal{I}_2$.

Results

Simulation results for path $b^{2N} = (0^{\frac{3N}{4}}1^N0^{\frac{N}{4}})$ are shown in Tables 5.3 and 5.4. Table 5.3 is for list size of 1 while Table 5.4 is for list size of 32. Both tables show two results for two different block lengths. The empirical rates $(\hat{R}_1, \hat{R}_2)$ as well as distortions $(\hat{D}_x, \hat{D}_y)$ are shown. During the simulations we increased both of the rates proportionally and recorded the values such that the distortion constraints are approximately satisfied.

Table 5.3: Experimental results for $b^{2N} = (0^{\frac{3N}{4}}1^N0^{\frac{N}{4}})$ and list size 1.

$\mathbf{N}$	$\hat{\mathbf{R}}_\mathbf{1}$	$\hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{R}}_\mathbf{1} + \hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{D}}_\mathbf{x}$	$\hat{\mathbf{D}}_\mathbf{y}$
$\mathbf{2^{10}}$	0.6943	0.7959	1.4902	0.0505	0.0464
$\mathbf{2^{12}}$	0.6785	0.7759	1.4543	0.0508	0.0483

In Table 5.3, we see that for $N = 2^{10}$, the total rate is 1.4902 instead of the theoretical limit 1.1821. The rate is approximately 1.26 times the theoretical limit. The rate expansion for $N = 2^{12}$ is approximately 1.23 which is lower as expected.

Table 5.4: Experimental results for $b^{2N} = (0^{\frac{3N}{4}}1^N0^{\frac{N}{4}})$ and list size 32.

$\mathbf{N}$	$\hat{\mathbf{R}}_\mathbf{1}$	$\hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{R}}_\mathbf{1} + \hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{D}}_\mathbf{x}$	$\hat{\mathbf{D}}_\mathbf{y}$
$\mathbf{2^{10}}$	0.6279	0.7207	1.3486	0.0552	0.0488
$\mathbf{2^{12}}$	0.6204	0.7097	1.3301	0.0497	0.0477

In Table 5.4, we see the results of same experiments only the decoders are list of 32. As we can see from the results, the performance increases considerably as expected. We see that for $N = 2^{10}$, the total rate is 1.3486 which is approximately 1.14 times the theoretical limit. The rate expansion for $N = 2^{12}$ is just 1.12, approximately.

5.1.2.2 Simulation 2

For this simulation the source distribution is selected as

$$P_{XY} = \begin{bmatrix} 0.7190 & 0.0050 \\ 0.0050 & 0.2710 \end{bmatrix}.$$

And we use the optimal estimator functions

$$\hat{x}(\bar{x},\bar{y}) = \arg\max_{x'} P_{X|\bar{X}\bar{Y}}(x'|\bar{x},\bar{y}), \qquad \hat{y}(\bar{x},\bar{y}) = \arg\max_{y'} P_{Y|\bar{X}\bar{Y}}(y'|\bar{x},\bar{y}).$$

The average distortion constraints are set to $D_x = 0.05$ and $D_y = 0.05$. The conditional distributions are selected as to satisfy the average distortion constraints mentioned above. If we use simple estimator functions $\hat{x}(\bar{x},\bar{y}) = \bar{x}$ and $\hat{y}(\bar{x},\bar{y}) = \bar{y}$

Table 5.5: Conditional probabilities $P_{\bar{X}|X}$ and $P_{\bar{Y}|Y}$.

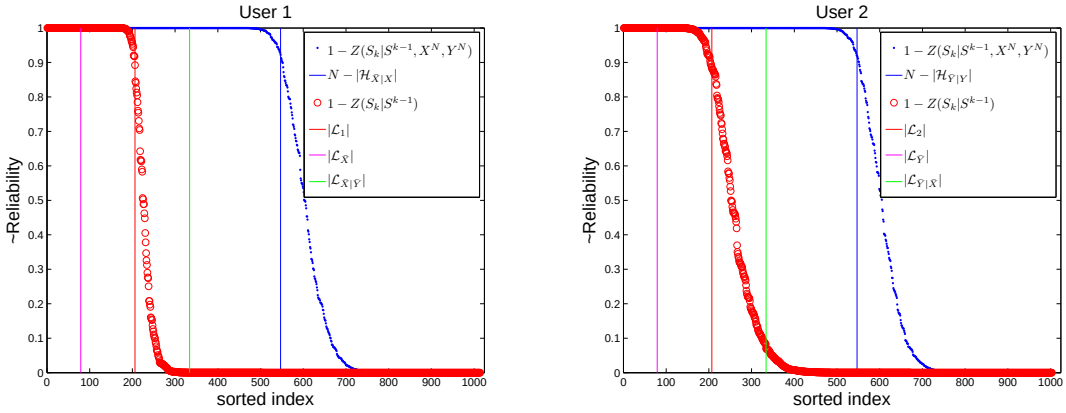
$\mathbf{a},\mathbf{b}$	$\mathbf{0},\mathbf{0}$	$\mathbf{0},\mathbf{1}$	$\mathbf{1},\mathbf{0}$	$\mathbf{1},\mathbf{1}$
$\mathbf{P_{\bar{X}|X}(a|b)}$	0.8880	0.0685	0.1120	0.9315
$\mathbf{P_{\bar{Y}|Y}(a|b)}$	0.8880	0.0685	0.1120	0.9315

in this setting the average distortions become $D_x = 0.1$ and $D_y = 0.1$, which are twice as bad compared to optimal estimators.

The mutual information parameters calculated for this source distribution are given in Table 5.6. As it can be seen from the table, if we were to encode X alone we would need a rate of $I(X;\bar{X}) = 0.4573$ and similarly for Y alone we would need a rate of $I(Y;\bar{Y}) = 0.4573$. The total rate would be 0.9146. However, because of the correlation and joint decoding the sum rate of Berger-Tung region is $I(\bar{X},\bar{Y};X,Y) = 0.666$ which is $I(\bar{X};\bar{Y}) = 0.2486$ less. The corner points of the Berger-Tung region are given as $(0.4573, 0.2087)$ and $(0.2087, 0.4573)$. In the simulations, the distortions are averaged over 1000 blocks. The tables show the experimental rates required to obtain the target distortions approximately.

Table 5.6: Berger-Tung parameters.

$\mathbf{I}(\mathbf{X};\bar{\mathbf{X}})$	$\mathbf{I}(\mathbf{Y};\bar{\mathbf{Y}})$	$\mathbf{I}(\bar{\mathbf{X}};\bar{\mathbf{Y}})$	$\mathbf{I}(\bar{\mathbf{X}},\bar{\mathbf{Y}};\mathbf{X},\mathbf{Y})$
0.4573	0.4573	0.2486	0.6660



(a) User 1 (b) User 2

Figure 5.5: Sorted reliability values.

Construction

Code construction is done using two-user SC decoder in large number of Monte-Carlo simulations and averaging the results. The joint decoder runs in two different configurations, once for likelihoods calculated for known (X^N, Y^N) and once for (X^N, Y^N) unknown. The decoder runs along the given path (b^{2N}) and calculate reliability values of the bits. Figure 5.5 shows results for path $b^{2N} = 0^{\frac{3N}{4}}1^N0^{\frac{N}{4}}$ with $N = 2^{10}$. The rate allocation for this path is measured from the results of simulations as $(R_1, R_2) = (0.333, 0.333)$. The interpretation of the figure is the same as given in previous simulation section.

Results

Simulation results for path $b^{2N} = (0^{\frac{3N}{4}}1^N0^{\frac{N}{4}})$ are shown in Tables 5.7 and 5.8. Table 5.7 is for list size of 1 while Table 5.8 is for list size of 32. Both tables show two results for two different block lengths. The empirical rates $(\hat{R}_1, \hat{R}_2)$ as well as

distortions $(\hat{D}_x, \hat{D}_y)$ are shown. During the simulations we increased both of the rates proportionally and recorded the values such that the distortion constraints are approximately satisfied.

Table 5.7: Experimental results for $b^{2N} = (0^{\frac{3N}{4}}1^N0^{\frac{N}{4}})$ and list size 1.

$\mathbf{N}$	$\hat{\mathbf{R}}_\mathbf{1}$	$\hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{R}}_\mathbf{1} + \hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{D}}_\mathbf{x}$	$\hat{\mathbf{D}}_\mathbf{y}$
$\mathbf{2^{10}}$	0.5128	0.5128	1.0256	0.0516	0.0517
$\mathbf{2^{12}}$	0.4829	0.4829	0.9658	0.0519	0.0520

In Table 5.7, we see that for $N = 2^{10}$, the total rate is 1.0256 instead of the theoretical limit 0.666. The rate is approximately 1.54 times the theoretical limit. The rate expansion for $N = 2^{12}$ is approximately 1.45 which is lower as expected.

Table 5.8: Experimental results for $b^{2N} = (0^{\frac{3N}{4}}1^N0^{\frac{N}{4}})$ and list size 32.

$\mathbf{N}$	$\hat{\mathbf{R}}_\mathbf{1}$	$\hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{R}}_\mathbf{1} + \hat{\mathbf{R}}_\mathbf{2}$	$\hat{\mathbf{D}}_\mathbf{x}$	$\hat{\mathbf{D}}_\mathbf{y}$
$\mathbf{2^{10}}$	0.4462	0.4462	0.8924	0.0499	0.0500
$\mathbf{2^{12}}$	0.4063	0.4063	0.8126	0.0502	0.0503

In Table 5.8, we see the results of same experiments only the decoders are list of 32. As we can see from the results, the performance increases considerably as expected. We see that for $N = 2^{10}$, the total rate is 0.8924 which is approximately 1.34 times the theoretical limit. The rate expansion for $N = 2^{12}$ is approximately just 1.22.

5.2 Multiple Description Coding

Multiple description coding (MDC) problem concerns with generating two descriptions of a source such that each description by itself can be used to reconstruct the source with some desired distortion and the two descriptions together can be used to reconstruct the source with a lower distortion. This problem is motivated by the need to efficiently communicate multimedia content over networks. Suppose that there are two paths to send a picture from the source to the viewer and the data may be lost over the paths. We may send the same description of the image over both of the paths to the viewer. However, such replication is inefficient and the viewer does not benefit from receiving more than one copy of the description. Multiple description coding provides a better way to achieve this path diversity. If a single description is received by the viewer, the image may be reconstructed with acceptable quality, and if both are received then the image may be reconstructed with a higher quality. Another application emerges when we want to communicate an image with different levels of quality to different viewers. Instead of sending different descriptions to each viewer we may use a special case of MDC called *successive refinement*. The idea is to send the common lowest quality description to all viewers and send successive refinements of it to different viewers with increasing quality expectations.

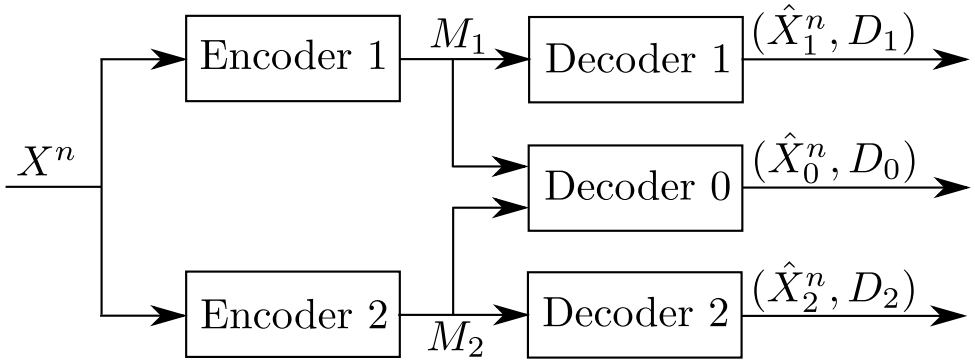


Figure 5.6: Multiple description coding setup.

Multiple description coding setup for a source X and three distortion measures $\mathrm{d}_j(x, \hat{x}_j)$ is depicted in Figure 5.6. Each encoder generates a description of X so that decoder 1 that receives only description M_1 can reconstruct X with distortion

D_1, decoder 2 that receives only description M_2 can reconstruct X with distortion D_2, and decoder 0 that receives both descriptions can reconstruct X with distortion D_0. The problem is to find the optimal trade-off between the description rate pair (R_1, R_2) and the distortion triple (D_0, D_1, D_2). Let $\mathrm{d}_j : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ denote the distortion function with maximum value less than $\mathrm{d}_{\max}$, for $j = 0, 1, 2$. The distortion function extends to vectors as $\mathrm{d}_j(x^N, \hat{x}^N) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{d}_j(x_i, \hat{x}_i)$.

A $(2^{nR_1}, 2^{nR_2}, n)$ multiple description code consists of two encoders, where encoder 1 assigns an index $m_1(x_n) \in [1 : 2^{nR_1})$ and encoder 2 assigns an index $m_2(x^n) \in [1 : 2^{nR_2})$ to each sequence $x^n \in \mathcal{X}^n$, and three decoders, where decoder 1 assigns an estimate $\hat{x}_1^n$ to each index m_1, decoder 2 assigns an estimate $\hat{x}_2^n$ to each index m_2, and decoder 0 assigns an estimate $\hat{x}_0^n$ to each index pair (m_1, m_2).

A rate-distortion quintuple $(R_1, R_2, D_0, D_1, D_2)$ is said to be achievable (and a rate pair (R_1, R_2) is said to be achievable for distortion triple (D_0, D_1, D_2)) if there exists a sequence of $(2^{nR_1}, 2^{nR_2}, n)$ codes with

$$\limsup_{n \to \infty} \mathrm{E}[\mathrm{d}_j(X^n, \hat{X}_j^n)] = D_j, \qquad j = 0, 1, 2. \tag{5.82}$$

The rate-distortion region $R(D_0, D_1, D_2)$ for multiple description coding is the closure of the set of rate pairs (R_1, R_2) such that $(R_1, R_2, D_0, D_1, D_2)$ is achievable. The rate-distortion region for multiple description coding is not known in general. The difficulty is that two good individual descriptions must be close to the source and so must be highly dependent. Thus the second description contributes little extra information beyond the first one. At the same time, to obtain a better reconstruction by combining two descriptions, they must be far apart and so must be highly independent. Two independent descriptions, however, cannot be individually good in general.

In this section, we will focus on an inner bound due to El Gamal and Cover [68]. We will use an alternate form of this bound given in [69], which is shown to be equivalent.

Theorem 13 (El Gamal-Cover (EGC) Inner Bound)**.** *Let X be a DMS, then*

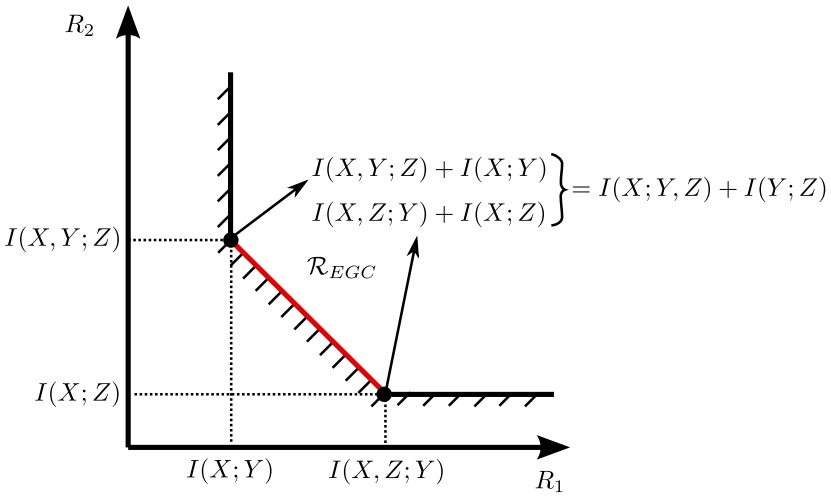


Figure 5.7: EGC rate region.

$(R_1, R_2, D_0, D_1, D_2)$ is achievable for multiple description coding if

$$R_1 \geq I(X;Y)$$
$$R_2 \geq I(X;Z)$$
$$R_1 + R_2 \geq I(X;Y,Z) + I(Y;Z)$$

for some pmf $p(x,y,z) = p(x)p(y,z|x)$ and deterministic mappings ϕ_j, $j = 0,1,2$, such that

$$D_0 \geq E[d_0(X, \phi_0(Y,Z))],$$
$$D_1 \geq E[d_1(X, \phi_1(Y))],$$
$$D_2 \geq E[d_2(X, \phi_2(Z))].$$

EGC region can be defined as

$$\mathcal{R}_{EGC} \triangleq \{(R_1,R_2) : R_1 \geq I(X;Y), R_2 \geq I(X;Z), \\ R_1 + R_2 \geq I(X;Y,Z) + I(Y;Z)\}. \quad (5.83)$$

The EGC region with its corner points are shown in Figure 5.7. The excess rate is represented with $I(Y;Z)$. If two descriptions were independent $I(Y;Z)$ would be zero. Then the total rate would be given as $I(X;Y) + I(X;Z) = I(X;Y,Z)$. The EGC inner bound is not tight in general. However, there are some special cases where it is tight. One particular case where the bound is tight is when there is no *excess rate*, that is, when rate pair (R_1, R_2) satisfies the condition $R_1 + R_2 = R(D_0)$ where $R(D_0)$ is the rate-distortion function of X with distortion measure d_0 evaluated at D_0.

Note the following equalities for the sum rate:

$$\begin{aligned} R_1 + R_2 &= I(X;Y) + I(Z;X,Y), \\ &= I(X;Z) + I(Y;X,Z), \\ &= I(X;Y) + I(X;Z) + I(Y;Z|X), \\ &= H(Y) + H(Z) - H(Y,Z|X). \end{aligned}$$

5.2.1 Polar Coding

Let source variable $X \in \mathcal{X}$ be from arbitrary discrete alphabet. Let external variables $Y \in \mathcal{Y}$ and $Z \in \mathcal{Z}$. We restrict the discussion to prime size alphabets $\mathcal{Y} = \mathcal{Z} = \{0, 1, \ldots, q-1\}$, where q is prime, for the purpose of polar coding. We assume $\mathcal{Y} = \mathcal{Z}$ to keep notation simple, but it is trivial to show that they may be of different size as long as their sizes are prime. Given the source distribution $X \sim P_X$, let the conditional distribution $P_{YZ|X}$ give rise to the design distortions D_1^*, D_2^* and D_0^*, i.e.

$$D_1^* = \mathbb{E}_{P_{XYZ}}[\mathrm{d}_1(X, \phi_1(Y))], \tag{5.84}$$

$$D_2^* = \mathbb{E}_{P_{XYZ}}[\mathrm{d}_2(X, \phi_2(Z))], \tag{5.85}$$

$$D_0^* = \mathbb{E}_{P_{XYZ}}[\mathrm{d}_0(X, \phi_0(Y,Z))], \tag{5.86}$$

where $P_{XYZ}(x,y,z) = P_X(x)P_{YZ|X}(y,z|x)$.

Consider the i.i.d. block of random variables (X^N, Y^N, Z^N) with $N = 2^n$ for some $n \geq 1$. The joint distribution is given by

$$P_{X^N Y^N Z^N}(x^N, y^N, z^N) = \prod_{i=1}^{N} P_X(x_i) P_{YZ|X}(y_i, z_i|x_i). \tag{5.87}$$

Let, U^N and V^N denote the polar transforms of N-vectors Y^N and Z^N, respectively, i.e.

$$U^N = Y^N G_N, \qquad V^N = X^N G_N. \tag{5.88}$$

Then we have

$$P_{X^N U^N V^N}(x^N, u^N, v^N) = P_{X^N Y^N Z^N}(x^N, u^N G^N, v^N G_N). \tag{5.89}$$

Since G_N is a one-to-one mapping, we can write the total rate for a block of length N as follows

$$R_1 + R_2 = \frac{1}{N}[H(Y^N) + H(Z^N) - H(Y^N, Z^N|X^N)] \tag{5.90}$$

$$= H(Y) + H(Z) - \frac{1}{N} H(U^N, V^N|X^N). \tag{5.91}$$

Let $S^{2N} = (S^1, \ldots, S^{2N})$ be a permutation π_N on (U^N, V^N) such that relative order of elements of U^N and V^N are preserved. Let b^{2N} be the *path string* s.t. $b_k \in \{0, 1\}$ which denotes the decoding path. Here, we make use of Section 4.1 and Definition 10.

Then, *monotone* expansion of total rate is given as

$$R_1 + R_2 = H(Y) + H(Z) - \frac{1}{N} \sum_{k=1}^{2N} H(S_k|X^N, S^{k-1}). \tag{5.92}$$

And individual rates are given as

$$R_1 = H(Y) - \frac{1}{N} \sum_{\substack{k=1: \\ b_k=0}}^{2N} H(S_k|X^N, S^{k-1}), \tag{5.93}$$

$$R_2 = H(Z) - \frac{1}{N} \sum_{\substack{k=1: \\ b_k=1}}^{2N} H(S_k|X^N, S^{k-1}). \tag{5.94}$$

Depending on the path the rate pairs span the entire dominant face of the EGC rate region. The first corner point $(I(X;Y), I(Z;X,Y))$ is achieved with $b^{2N} = (0^N 1^N)$. The second corner point $(I(Y;X,Z), I(X;Z))$ is achieved with $b^{2N} = (1^N 0^N)$.

For the purpose of polar coding, the total probabilities are also expanded as follows:

$$P_{X^N U^N V^N}(x^N, u^N, v^N) = P_{X^N S^{2N}}(x^N, \pi_N(u^N, v^N)) \tag{5.95}$$

$$P_{X^N S^{2N}}(x^N, s^{2N}) = P_{X^N}(x^N) \prod_{k=1}^{2N} P_{S_k|X^N S^{k-1}}(s_k|x^N, s^{k-1}). \tag{5.96}$$

5.2.1.1 Polarization Sets

In the following, we refer to three interrelated index variables k, i and j, repeatedly, all in the context of an assumed path b^{2N}. We make use of Definition 10 here. Let $\delta_N = 2^{-N^\beta}$ for $0 < \beta < \frac{1}{2}$. First, we define the following general path dependent polarization set:

$$\tilde{\mathcal{H}} \triangleq \left\{ k \in [2N] : Z(S_k|X^N, S^{k-1}) \geq 1 - \delta_N \right\}. \tag{5.97}$$

Then, we define the following low entropy set for Y:

$$\mathcal{L}_Y = \left\{ i \in [N] : Z(U_i|U^{i-1}) \leq \delta_N \right\}, \tag{5.98}$$

and for Z:

$$\mathcal{L}_Z \triangleq \left\{ j \in [N] : Z(V_j|V^{j-1}) \leq \delta_N \right\}. \tag{5.99}$$

Similar to above sets, we define the following sets that contain k indices:

$$\tilde{\mathcal{L}}_Y = \left\{ k \in [2N] : Z(U_i|U^{i-1}) \leq \delta_N \right\}, \tag{5.100}$$

and

$$\tilde{\mathcal{L}}_Z \triangleq \left\{ k \in [2N] : Z(V_j|V^{j-1}) \leq \delta_N \right\}. \tag{5.101}$$

Now we define the high entropy sets as

$$\mathcal{H}_{Y|X} \triangleq \left\{ i \in [N] : Z(U_i|X^N, U^{i-1}) \geq 1 - \delta_N \right\}, \tag{5.102}$$

$$\mathcal{H}_1 \triangleq \left\{ i \in [N] : b_k = 0, Z(S_k|X^N, S^{k-1}) \geq 1 - \delta_N \right\}, \tag{5.103}$$

and

$$\mathcal{H}_{Z|X} \triangleq \left\{ j \in [N] : Z(V_j|X^N, V^{j-1}) \geq 1 - \delta_N \right\}, \tag{5.104}$$

$$\mathcal{H}_2 \triangleq \left\{ j \in [N] : b_k = 1, Z(S_k|X^N, S^{k-1}) \geq 1 - \delta_N \right\}. \tag{5.105}$$

Observe that the following are true for above sets

$$\mathcal{H}_1 \subseteq \mathcal{H}_{Y|X}, \qquad \mathcal{H}_2 \subseteq \mathcal{H}_{Z|X} \tag{5.106}$$

for any path b^{2N}. Similar to above sets, we define the following sets which contain k indices:

$$\tilde{\mathcal{H}}_1 \triangleq \left\{ k \in [2N] : b_k = 0, Z(S_k|X^N, S^{k-1}) \geq 1 - \delta_N \right\}, \tag{5.107}$$

$$\tilde{\mathcal{H}}_2 \triangleq \left\{ k \in [2N] : b_k = 1, Z(S_k|X^N, S^{k-1}) \geq 1 - \delta_N \right\}. \tag{5.108}$$

First define two index sets as follows:

$$\tilde{\mathcal{K}}_m \triangleq \{k \in [2N] : \ b_k = m-1\}, \quad m = 1, 2. \tag{5.109}$$

Definition 17 (Frozen and Information Sets)**.** *The following frozen sets are defined using the polarization sets defined above:*

$$\mathcal{F}_Y \triangleq \mathcal{L}_Y \cup \mathcal{H}_{Y|X}, \quad \mathcal{I}_Y \triangleq [N] \setminus \mathcal{F}_Y \tag{5.110}$$

$$\mathcal{F}_1 \triangleq \mathcal{L}_Y \cup \mathcal{H}_1, \quad \mathcal{I}_1 \triangleq [N] \setminus \mathcal{F}_1 \tag{5.111}$$

$$\mathcal{F}_Z \triangleq \mathcal{L}_Z \cup \mathcal{H}_{Z|X}, \quad \mathcal{I}_Z \triangleq [N] \setminus \mathcal{F}_Z \tag{5.112}$$

$$\mathcal{F}_2 \triangleq \mathcal{L}_Z \cup \mathcal{H}_2, \quad \mathcal{I}_2 \triangleq [N] \setminus \mathcal{F}_2. \tag{5.113}$$

and

$$\tilde{\mathcal{F}} \triangleq \tilde{\mathcal{L}}_X \cup \tilde{\mathcal{L}}_Y \cup \tilde{\mathcal{H}}, \quad \tilde{\mathcal{I}} \triangleq [2N] \setminus \tilde{\mathcal{F}} \tag{5.114}$$

$$\tilde{\mathcal{F}}_Y \triangleq \tilde{\mathcal{L}}_Y \cup \tilde{\mathcal{H}}_{Y|X}, \quad \tilde{\mathcal{I}}_Y \triangleq \tilde{\mathcal{K}}_1 \setminus \tilde{\mathcal{F}}_Y \tag{5.115}$$

$$\tilde{\mathcal{F}}_1 \triangleq \tilde{\mathcal{L}}_Y \cup \tilde{\mathcal{H}}_1, \quad \tilde{\mathcal{I}}_1 \triangleq \tilde{\mathcal{K}}_1 \setminus \tilde{\mathcal{F}}_1 \tag{5.116}$$

$$\tilde{\mathcal{F}}_Z \triangleq \tilde{\mathcal{L}}_Z \cup \tilde{\mathcal{H}}_{Z|X}, \quad \tilde{\mathcal{I}}_Z \triangleq \tilde{\mathcal{K}}_2 \setminus \tilde{\mathcal{F}}_Z \tag{5.117}$$

$$\tilde{\mathcal{F}}_2 \triangleq \tilde{\mathcal{L}}_Z \cup \tilde{\mathcal{H}}_2, \quad \tilde{\mathcal{I}}_2 \triangleq \tilde{\mathcal{K}}_2 \setminus \tilde{\mathcal{F}}_2. \tag{5.118}$$

Proposition 6 (Polarization)**.** *Consider the information sets defined in Definition 17 for a fixed base path b^{2N_0} with rate pair (R_1, R_2). Fix a constant $\tau > 0$. Then there exists an $N'(\tau) = 2^l N_0$, $l = 1, 2, \ldots$, and the corresponding scaled path $b^{2N} = 2^l b^{2N_0}$, such that*

$$\frac{1}{N} |\mathcal{I}_1| \leq R_1 + \tau, \tag{5.119}$$

$$\frac{1}{N} |\mathcal{I}_2| \leq R_2 + \tau, \tag{5.120}$$

for all $N > N'$.

Proof. Note that from standard single user polar coding we have the following facts:

$$\lim_{l \to \infty} \frac{1}{N} \left| \left\{ i \in [N] : 2^{-N^\beta} < H(U_i | U^{i-1}) < 1 - 2^{-N^\beta} \right\} \right| = 0,$$

$$\lim_{l\to\infty}\frac{|\mathcal{L}_Y|}{N}=1-H(Y),$$

and

$$\lim_{l\to\infty}\frac{1}{N}\left|\left\{j\in[N]:2^{-N^\beta}<H(V_j|V^{j-1})<1-2^{-N^\beta}\right\}\right|=0,$$

$$\lim_{l\to\infty}\frac{|\mathcal{L}_Z|}{N}=1-H(Z).$$

For the MDC setting, Section 4.1 and polarization theorem 6 apply. From Theorem 6, we have the following fact:

$$\lim_{l\to\infty}\frac{1}{2N}\left|\left\{k\in[2N]:2^{-N^\beta}<H(S_k|X^N,S^{k-1})<1-2^{-N^\beta}\right\}\right|=0,$$

$$\lim_{l\to\infty}\frac{|\tilde{\mathcal{H}}_m|}{N}=R'_m,\qquad m=1,2,$$

where $\tilde{\mathcal{H}}_m=\{k\in[2N]:b_k=m-1,H(S_k|X^N,S^{k-1})\geq 1-2^{-N^\beta}\}$ and $R'_m=\frac{1}{N}\sum_{k:b_k=m-1}^{2N}H(S_k|Z^N,S^{k-1})$, for $m\in\{1,2\}$. Also, the following is true for R'_m:

$$H(Y|X,Z)\leq R'_1\leq H(Y|X),\qquad H(Z|X,Y)\leq R'_2\leq H(Z|X).$$

The lower and upper bounds of first and second expressions, respectively, are satisfied with path $b^{2N}=1^N0^N$. Similarly, the upper and lower bounds of first and second expressions, respectively, are satisfied with path $b^{2N}=0^N1^N$.

We define complements of sets for user m with respect to the corresponding index set $\tilde{\mathcal{K}}_m$, i.e. $\tilde{\mathcal{F}}_m^c\triangleq\tilde{\mathcal{K}}_m\setminus\tilde{\mathcal{F}}_m$. Since $H(S_k|X^N,S^{k-1})\leq H(S_k|S^{k-1})$, we have $\tilde{\mathcal{L}}_Y\cap\tilde{\mathcal{H}}_1=\emptyset$, $\tilde{\mathcal{L}}_Z\cap\tilde{\mathcal{H}}_2=\emptyset$ and $\tilde{\mathcal{H}}_1\subseteq\tilde{\mathcal{L}}_Y^c$, $\tilde{\mathcal{H}}_2\subseteq\tilde{\mathcal{L}}_Z^c$. And the result follows from observing $\tilde{\mathcal{I}}_1=(\tilde{\mathcal{L}}_Y\cup\tilde{\mathcal{H}}_1)^c=\tilde{\mathcal{L}}_Y^c\setminus\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{I}}_2=(\tilde{\mathcal{L}}_Z\cup\tilde{\mathcal{H}}_2)^c=\tilde{\mathcal{L}}_Z^c\setminus\tilde{\mathcal{H}}_2$. □

5.2.1.2 Encoding

We define family of functions $\lambda_i^{(1)}:\mathcal{Y}^{i-1}\to\mathcal{Y}$, $\forall i\in\mathcal{F}_1$ and $\lambda_j^{(2)}:\mathcal{Y}^{j-1}\to\mathcal{Y}$, $\forall j\in\mathcal{F}_2$. We assume that they are shared between the encoders and the decoder.

We also define the corresponding random variables $\Lambda_i^{(1)}$ and $\Lambda_j^{(2)}$ such that

$$\begin{aligned}\Lambda_i^{(1)}(u^{i-1}) &\triangleq a, \text{ w.p. } P_{U_i|U^{i-1}}\left(a|u^{i-1}\right), \\ \Lambda_j^{(2)}(v^{j-1}) &\triangleq a, \text{ w.p. } P_{V_j|V^{j-1}}\left(a|v^{j-1}\right),\end{aligned} \tag{5.121}$$

where $a \in \mathcal{Y}$. Maps $(\lambda_i^{(1)}, \lambda_j^{(2)})$ are the realizations of random maps $(\Lambda_i^{(1)}, \Lambda_j^{(2)})$. Each realization of set of maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ results in different encoding and decoding protocols. The distribution over the choice of maps is induced with the above equation (5.121). The set of maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(1)})$ are used to determine the bits in sets $\mathcal{F}_1$, $\mathcal{F}_2$. The theoretical analysis of the distortions are made much easier using the randomized maps and calculating the average distortion over maps.

The bits in *information* sets $\tilde{\mathcal{I}}$ are calculated either the deterministic or the random rules given below.

Deterministic rules:

$$\bar{\psi}_k(s^{k-1}, x^N) \triangleq \arg\max_{s' \in \mathcal{Y}} \left\{ P_{S_k|X^N S^{k-1}}\left(s'|x^N, s^{k-1}\right) \right\}. \tag{5.122}$$

Random rules:

$$\Psi_k(s^{k-1}, x^N) \triangleq a, \text{ w.p. } P_{S_k|X^N S^{k-1}}\left(a|x^N, s^{k-1}\right), \tag{5.123}$$

where $a \in \mathcal{Y}$. Maps ψ_k are the realizations of random maps Ψ_k. In the analysis we use the random rules for tractability. This approach is called *randomized rounding* [16]. The encoding operations are given as follows.

The encoders construct the sequence s^{2N} bit-by-bit *successively*,

$$s_k = \begin{cases} \lambda_i^{(1)}(u^{i-1}), & \text{if } k \in \tilde{\mathcal{F}}_1, \\ \lambda_j^{(2)}(v^{j-1}), & \text{if } k \in \tilde{\mathcal{F}}_2, \\ \psi_k(s^{k-1}, x^N), & \text{otherwise.} \end{cases} \tag{5.124}$$

Then, encoder 1 transmits the compressed message $u_{\mathcal{I}_1} = s_{\tilde{\mathcal{I}}_1}$ and encoder 2

transmits the compressed message $v_{\mathcal{I}_2} = s_{\tilde{\mathcal{I}}_2}$.

Remark 3. *Note that although in the analysis we use randomized rounding approach and thus make use of random rules Ψ_k for calculating bits in $\mathcal{I}_1$ and $\mathcal{I}_2$, in practice we use the deterministic rules. In either case, the probabilities $P(s_k|x^N, s^{k-1})$ have to be calculated. These are calculated using two-user joint SC decoding. Therefore, two-user joint SC decoders are employed at the encoders. Thus, we refer to this operation as SC encoding.*

Remark 4. *The set $\tilde{\mathcal{F}}$ actually comprises of two distinct parts and we could use a simplified rule for $k \in \tilde{\mathcal{F}}$:*

$$s_k = \begin{cases} \bar{s}_k, & \text{if } k \in \tilde{\mathcal{H}}, \\ \underset{s' \in \mathcal{Y}}{\arg\max}\ P_{S_k|S^{k-1}}\left(s'|s^{k-1}\right), & \text{if } k \in \tilde{\mathcal{L}}_Y \cup \tilde{\mathcal{L}}_Z, \end{cases} \tag{5.125}$$

where $\bar{s}_k$ is determined beforehand uniformly from $\mathcal{Y}$. However, since this rule makes the proof harder, we use the random maps ($\Lambda^{(1)}_{\mathcal{F}_1}$, $\Lambda^{(2)}_{\mathcal{F}_2}$) for simplicity.

5.2.1.3 Decoding

Three different decoding operations are performed at Decoders 1, 2 and 0. Decoder 1 has access to only $u_{\mathcal{I}_1}$, decoder 1 has access to only $v_{\mathcal{I}_2}$ and decoder 0 has access to both.

Decoder 1 first calculates:

$$\hat{u}_i = \begin{cases} \lambda^{(1)}_i(\hat{u}^{i-1}), & \text{if } i \in \mathcal{F}_1, \\ u_i, & \text{otherwise.} \end{cases} \tag{5.126}$$

Then, calculates the estimate as $\hat{y}^N = \hat{u}^N G_N$ and $\hat{x}_1^N = \phi_1(\hat{y}^N)$.

Decoder 2 first calculates:

$$\hat{v}_j = \begin{cases} \lambda^{(2)}_j(\hat{v}^{j-1}), & \text{if } j \in \mathcal{F}_2, \\ v_j, & \text{otherwise.} \end{cases} \tag{5.127}$$

Then, calculates the estimate as $\hat{z}^N = \hat{v}^N G_N$ and $\hat{x}_2^N = \phi_2(\hat{z}^N)$.

Decoder 0 first calculates:

$$\hat{s}_k = \begin{cases} \lambda_i^{(1)}(\hat{u}^{i-1}), & \text{if } k \in \tilde{\mathcal{F}}_1, \\ \lambda_j^{(2)}(\hat{v}^{j-1}), & \text{if } k \in \tilde{\mathcal{F}}_2, \\ s_k, & \text{otherwise.} \end{cases} \tag{5.128}$$

Then, the decoder extracts $\hat{u}^N = \hat{s}_{\tilde{\mathcal{K}}_1}$ and $\hat{v}^N = \hat{s}_{\tilde{\mathcal{K}}_2}$. Finally, it calculates the estimate as $\hat{y}^N = \hat{u}^N G_N$, $\hat{z}^N = \hat{v}^N G_N$ and $\hat{x}_0^N = \phi_0(\hat{y}^N, \hat{z}^N)$.

Note that although in Figure 5.6 two separate encoders are shown, the information available to both of the encoders is the same which is the source sequence x^N. They are not "separate" encoders in the sense that both can generate the other's output. Thus, we defined a single successive cancellation encoding operation in Section 5.2.1.2. Encoders 1 and 2 get differentiated at the last step where they transmit different subsets of the sequence s^{2N} formed by the single SC encoding operation. Even if encoders 1 and 2 are implemented separately they would generate the same s^{2N} for the same input x^N. The encoders use a two-user successive-cancellation polar decoder to calculate s^{2N} given the source x^N and path b^{2N}. The particular choice of b^{2N} results in a specific rate allocation pair (R_1, R_2) corresponding to the sizes of sets $\mathcal{I}_1$ and $\mathcal{I}_2$. For analysis purposes, we assume random encoding functions. The results of encoding operation may be different for the same input x^N. For encoder, at step $k \in \tilde{\mathcal{I}}$ of the process, $s_k = a$ with probability proportional to $P_{S_k|X^N S^{k-1}}(a|x^N, s^{k-1})$. Thus, for a given pair of maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$, a particular s^{2N} occurs with a certain probability induced by the distributions of $\Psi_{\tilde{\mathcal{I}}}$ and maps.

We define the resulting average (over x^N and randomness of the "information" bits induced by the distribution of $\Psi_{\tilde{\mathcal{I}}}$) distortions of above encoding and decoding operations as $D_1(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$, $D_2(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ and $D_0(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$. In the following we show that for sets $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{I}_1$, $\mathcal{I}_2$ defined in 5.2.1.1 and encoding and decoding methods defined in 5.2.1.2 and 5.2.1.3, there exist maps $(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)})$ such that $D_1(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)}) \sim D_1^*$, $D_2(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)}) \sim D_2^*$ and $D_0(\lambda_{\mathcal{F}_1}^{(1)}, \lambda_{\mathcal{F}_2}^{(2)}) \sim D_0^*$, where D_1^*, D_2^*

and D_0^* are the design distortions. We do that by determining the *expected* average distortions over the ensembles of codes generated by different encoding maps ($\lambda_{\mathcal{F}_1}^{(1)}$, $\lambda_{\mathcal{F}_2}^{(2)}$). The distribution over the choices of maps is given in (5.121). Then we show that the *expected* average distortions are roughly D_1^*, D_2^* and D_0^*. This implies that for at least one choice of ($\lambda_{\mathcal{F}_1}^{(1)}$, $\lambda_{\mathcal{F}_2}^{(2)}$) the average distortions are close to D_1^*, D_2^* and D_0^*. The following theorem makes this precise.

Theorem 14. *Let $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{I}_1$, $\mathcal{I}_2$ be sets as defined in 5.2.1.1 and encoding and decoding methods be as defined in 5.2.1.2 and 5.2.1.3. Then the expectations of average distortions $D_1(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})$, $D_2(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})$, $D_0(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})$ over the maps $\Lambda_{\mathcal{F}_1}^{(1)}$, $\Lambda_{\mathcal{F}_2}^{(2)}$ satisfy $\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}\}}\left[D_1(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})\right] = D_1^* + O(2^{-N^\beta})$, $\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}\}}\left[D_2(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})\right] = D_2^* + O(2^{-N^\beta})$ and $\mathbb{E}_{\{\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)}\}}\left[D_0(\Lambda_{\mathcal{F}_1}^{(1)}, \Lambda_{\mathcal{F}_2}^{(2)})\right] = D_0^* + O(2^{-N^\beta})$ for any $(R_1, R_2) \in \mathcal{R}_{EGC}$ and $\beta < 1/2$. Consequently, there exist deterministic maps that satisfy the above relations.*

The following sections give necessary steps for proving the theorem. We first prove a total variation bound on two probability measures. Then, we use that result to bound the *expected* average distortions of the code.

5.2.1.4 Total Variation Bound

Define the following probability measure.

$$Q_{X^N S^{2N}}(x^N, s^{2N}) \triangleq Q_{X^N}(x^N) \prod_{k=1}^{2N} Q_{S_k|X^N S^{k-1}}(s_k|x^N, s^{k-1}), \tag{5.129}$$

where $Q_{X^N}(x^N) = P_{X^N}(x^N)$. The conditional probability measures are defined as

$$Q_{S_k|X^N S^{k-1}}(s_k|x^N, s^{k-1}) \triangleq \begin{cases} P_{U_i|U^{i-1}}(u_i|u^{i-1}), & k \in \tilde{\mathcal{F}}_1, \\ P_{V_j|V^{j-1}}(v_j|v^{j-1}), & k \in \tilde{\mathcal{F}}_2, \\ P_{S_k|X^N S^{k-1}}(s_k|x^N, s^{k-1}), & \text{otherwise.} \end{cases} \tag{5.130}$$

Also, note the following

$$Q_{X^N U^N V^N}(x^N, u^N, v^N) = Q_{X^N S^{2N}}(x^N, \pi_N(u^N, v^N)). \tag{5.131}$$

Lemma 11 (Total Variation Bound)**.** *Let probability measures P and Q be defined as in* (5.96) *and* (5.129)*, respectively. For $0 < \beta < 1/2$ and sufficiently large N, the total variation distance between P and Q is bounded as*

$$\sum_{s^{2N}, x^N} \left| P_{X^N S^{2N}}(x^N, s^{2N}) - Q_{X^N S^{2N}}(x^N, s^{2N}) \right| \leq 2^{-N^\beta}. \tag{5.132}$$

Proof. See Appendix C.2. □

5.2.1.5 Average Distortion

For a source sequence x^N, random encoding maps $(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2})$ and encoding rule (5.124), (u^N, v^N) appears with probability

$$\left(\prod_{i \in \mathcal{I}_1} P_{U_i | X^N U^{i-1} V^j}(u_i | x^N, u^{i-1}, v^j) \right) \left(\prod_{i \in \mathcal{F}_1} \mathbb{1}_{\left\{ \Lambda^{(1)}_i (u^{i-1}) = u_i \right\}} \right) \cdot \left(\prod_{j \in \mathcal{I}_2} P_{V_j | X^N, V^{j-1}, U^i}(v_j | y^N, v^{j-1}, u^i) \right) \left(\prod_{j \in \mathcal{F}_2} \mathbb{1}_{\left\{ \Lambda^{(2)}_j (v^{j-1}) = v_j \right\}} \right).$$

For random sets of maps $(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2})$, the average distortions are random quantities given by

$$\begin{aligned} D_0(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}) = \sum_{x^N} P_{X^N}(x^N) \sum_{u^N, v^N} \mathrm{d}_0(x^N, \phi_0(u^N G_N, v^N G_N)) \cdot \\ \left(\prod_{i \in \mathcal{I}_1} P_{U_i | X^N U^{i-1} V^j}(u_i | x^N, u^{i-1}, v^j) \right) \left(\prod_{i \in \mathcal{F}_1} \mathbb{1}_{\left\{ \Lambda^{(1)}_i (u^{i-1}) = u_i \right\}} \right) \cdot \\ \left(\prod_{j \in \mathcal{I}_2} P_{V_j | X^N, V^{j-1}, U^i}(v_j | y^N, v^{j-1}, u^i) \right) \left(\prod_{j \in \mathcal{F}_2} \mathbb{1}_{\left\{ \Lambda^{(2)}_j (v^{j-1}) = v_j \right\}} \right), \end{aligned} \tag{5.133}$$

$$D_1(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}) = \sum_{x^N} P_{X^N}(x^N) \sum_{u^N} \mathrm{d}_1(x^N, \phi_1(u^N G_N)) \cdot \sum_{v^N} \left(\prod_{i \in \mathcal{I}_1} P_{U_i|X^N U^{i-1} V^j}(u_i | x^N, u^{i-1}, v^j) \right) \left(\prod_{i \in \mathcal{F}_1} \mathbb{1}_{\left\{\Lambda^{(1)}_i(u^{i-1}) = u_i\right\}} \right) \cdot \left(\prod_{j \in \mathcal{I}_2} P_{V_j|X^N, V^{j-1}, U^i}(v_j | y^N, v^{j-1}, u^i) \right) \left(\prod_{j \in \mathcal{F}_2} \mathbb{1}_{\left\{\Lambda^{(2)}_j(v^{j-1}) = v_j\right\}} \right), \tag{5.134}$$

$$D_2(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}) = \sum_{x^N} P_{X^N}(x^N) \sum_{v^N} \mathrm{d}_2(x^N, \phi_2(v^N G_N)) \cdot \sum_{u^N} \left(\prod_{i \in \mathcal{I}_1} P_{U_i|X^N U^{i-1} V^j}(u_i | x^N, u^{i-1}, v^j) \right) \left(\prod_{i \in \mathcal{F}_1} \mathbb{1}_{\left\{\Lambda^{(1)}_i(u^{i-1}) = u_i\right\}} \right) \cdot \left(\prod_{j \in \mathcal{I}_2} P_{V_j|X^N, V^{j-1}, U^i}(v_j | y^N, v^{j-1}, u^i) \right) \left(\prod_{j \in \mathcal{F}_2} \mathbb{1}_{\left\{\Lambda^{(2)}_j(v^{j-1}) = v_j\right\}} \right). \tag{5.135}$$

The expectations over maps are

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}\}} \left[D_0(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}) \right] = \sum_{x^N} P_{X^N}(x^N) \sum_{u^N, v^N} \mathrm{d}_0(x^N, \phi_0(u^N G_N, v^N G_N)) \cdot \left(\prod_{i \in \mathcal{I}_1} P_{U_i|X^N U^{i-1} V^j}(u_i | x^N, u^{i-1}, v^j) \right) \left(\prod_{i \in \mathcal{F}_1} P_{U_i|U^{i-1}}(u_i | u^{i-1}) \right) \cdot \left(\prod_{j \in \mathcal{I}_2} P_{V_j|X^N, V^{j-1}, U^i}(v_j | y^N, v^{j-1}, u^i) \right) \left(\prod_{j \in \mathcal{F}_2} P_{V_j|V^{j-1}}(v_j | v^{j-1}) \right),$$

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}\}} \left[D_1(\Lambda^{(1)}_{\mathcal{F}_1}, \Lambda^{(2)}_{\mathcal{F}_2}) \right] = \sum_{x^N} P_{X^N}(x^N) \sum_{u^N, v^N} \mathrm{d}_1(x^N, \phi_1(u^N G_N)) \cdot \left(\prod_{i \in \mathcal{I}_1} P_{U_i|X^N U^{i-1} V^j}(u_i | x^N, u^{i-1}, v^j) \right) \left(\prod_{i \in \mathcal{F}_1} P_{U_i|U^{i-1}}(u_i | u^{i-1}) \right) \cdot \left(\prod_{j \in \mathcal{I}_2} P_{V_j|X^N, V^{j-1}, U^i}(v_j | y^N, v^{j-1}, u^i) \right) \left(\prod_{j \in \mathcal{F}_2} P_{V_j|V^{j-1}}(v_j | v^{j-1}) \right),$$

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_2(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] = \sum_{x^N} P_{X^N}(x^N) \sum_{u^N,v^N} \mathrm{d}_2(x^N,\phi_2(v^N G_N))\cdot$$
$$\left(\prod_{i\in\mathcal{I}_1} P_{U_i|X^N U^{i-1}V^j}(u_i|x^N,u^{i-1},v^j)\right)\left(\prod_{i\in\mathcal{F}_1} P_{U_i|U^{i-1}}(u_i|u^{i-1})\right)\cdot$$
$$\left(\prod_{j\in\mathcal{I}_2} P_{V_j|X^N,V^{j-1},U^i}(v_j|y^N,v^{j-1},u^i)\right)\left(\prod_{j\in\mathcal{F}_2} P_{V_j|V^{j-1}}(v_j|v^{j-1})\right)$$

Using the probability distribution Q defined in (5.129) we can write the expectations as

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_0(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] = \mathbb{E}_Q\left[\mathrm{d}_0(X^N,\phi_0(U^N G_N,V^N G_N))\right], \tag{5.136}$$
$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_1(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] = \mathbb{E}_Q\left[\mathrm{d}_1(X^N,\phi_1(U^N G_N))\right], \tag{5.137}$$
$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_2(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] = \mathbb{E}_Q\left[\mathrm{d}_2(X^N,\phi_2(V^N G_N))\right]. \tag{5.138}$$

Therefore, we get

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_0(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] \leq \mathbb{E}_P\left[\mathrm{d}_0(X^N,\phi_0(U^N G_N,V^N G_N))\right] + \mathrm{d}_{\max}||P_{X^N S^{2N}} - Q_{X^N S^{2N}}||, \tag{5.139}$$

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_1(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] \leq \mathbb{E}_P\left[\mathrm{d}_1(X^N,\phi_1(U^N G_N))\right] + \mathrm{d}_{\max}||P_{X^N S^{2N}} - Q_{X^N S^{2N}}||, \tag{5.140}$$

$$\mathbb{E}_{\{\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2}\}}\left[D_2(\Lambda^{(1)}_{\mathcal{F}_1},\Lambda^{(2)}_{\mathcal{F}_2})\right] \leq \mathbb{E}_P\left[\mathrm{d}_2(X^N,\phi_2(V^N G_N))\right] + \mathrm{d}_{\max}||P_{X^N S^{2N}} - Q_{X^N S^{2N}}||. \tag{5.141}$$

Lemma 11 shows that second term of the sum is $O(2^{-N^\beta})$. Therefore, there exist deterministic sets of maps $\lambda^{(1)}_{\mathcal{F}_1}$ and $\lambda^{(2)}_{\mathcal{F}_2}$ such that $D_0(\lambda^{(1)}_{\mathcal{F}_1},\lambda^{(2)}_{\mathcal{F}_2}) = D_0^* + O(2^{-N^\beta})$, $D_1(\lambda^{(1)}_{\mathcal{F}_1},\lambda^{(2)}_{\mathcal{F}_2}) = D_1^* + O(2^{-N^\beta})$ and $D_2(\lambda^{(1)}_{\mathcal{F}_1},\lambda^{(2)}_{\mathcal{F}_2}) = D_2^* + O(2^{-N^\beta})$.

Chapter 6

Conclusion

Although polar codes were invented as channel codes that can achieve capacity of binary-input symmetric channels, their underlying polarization principle was shown to be applicable to various channel and source coding problems. In this thesis, we extended previous results on polar codes and devised methods that can achieve known bounds for diverse distributed lossless and lossy source coding problems.

In Chapter 3, we considered a restricted version of the Slepian-Wolf (SW) problem. We showed how single-user polar codes can be used to achieve any point on the SW region for binary symmetric sources (BSS) without time-sharing. In exchange for this special source distribution, it was possible to use single-user polar successive-cancellation decoder as source decoder. Thus, the complexity of the method was very low.

In Chapter 4, we first discussed "monotone chain rule based" polarization approach which extends polar coding to multi-user settings. This method was introduced by Arıkan [28]. We presented an extended treatment of the method which consisted of two sources with prime-sized alphabets and a side-information with an arbitrary alphabet. This treatment formed the basis of our other polar coding schemes for distributed settings in later sections. We derived recursive

formulas for *two-user* polar decoder implementation and gave explicit algorithms for implementing successive-cancellation *list* (SCL) decoder based on single-user SCL decoder [40]. This two-user SC decoder was used as the source decoder in SW setting. We also presented the performance of the decoder by giving experimental results. The method in this chapter solves the *general* SW problem with arbitrary source distributions, and its complexity is higher compared to the method in Chapter 3 in exchange. Lastly, we moved on to multiple-access channel (MAC) problem, which is considered as the dual of the SW problem. We devised polarization sets, encoding and decoding methods for this problem, too. We proved that our method can achieve not only the uniform capacity region but the whole MAC capacity region. By uniform capacity region, we mean that the capacity region when distributions of variables at the inputs of the MAC are uniform. We used *randomized* methods to prove this extended result. Then, we gave simulation results presenting the performance of the devised polar coding scheme for MAC.

In Chapter 5, we considered two different lossy source coding problems in distributed settings. The treatment of distributed polar codes in Chapter 4 comprised the basis of analysis in this chapter. The first problem we considered was the distributed lossy source coding which is the lossy version of the SW problem. We devised a polar coding method for the problem and showed that it can achieve the whole dominant face of the Berger-Tung (BT) region, which is the best known capacity region for this problem. Then, we presented simulation results on the performance of our method for distributed lossy source coding. The second problem we considered in this chapter was the multiple description coding problem. The setting consists of a single source and two different representations of the source generated by two encoders. Three different decoders that has access to first, second and both representations, generate three different reconstructions. Each reconstruction has a different distortion constraint. We considered the El Gamal-Cover (EGC) inner bound which is the best known bound for this problem. We constructed a polar coding method that can achieve the whole dominant face of the for EGC region. Similar to the MAC problem in Chapter 4, we used randomized approach for proving that our polarization based

encoding and decoding methods achieve the capacity bounds of the problems in this chapter.

Our treatment in this thesis shows that polar codes based on monotone chain rules can achieve known bounds of diverse lossless and lossy distributed source coding problems. Similar proof principles were used to achieve the results for each problem considered in this thesis. It seems that the known bounds of many distributed channel and source coding problems in information theory may be achieved using similar techniques. Investigating this conjecture and possibly devising a general polarization proof framework is left for future study.

Appendix A

Chapter 2

A.1 Useful Lemmas

Lemma 12 (Pinsker's Inequality)**.** *Let $P(y)$ and $Q(y)$ be two discrete probability measures where $y \in \mathcal{Y}$. The following inequality holds*

$$\sum_{y \in \mathcal{Y}} |P(y) - Q(y)| \leq \sqrt{\kappa \, D(P(y)||Q(y))},$$

where $D(\cdot||\cdot)$ is the Kullback-Leibler distance and $\kappa \triangleq 2 \ln 2$.

A.2 Proof of Lemma 4

$$
\begin{aligned}
Z(X|Y_2) &\triangleq \frac{1}{q-1} \sum_{\substack{x,x': \\ x\neq x'}} \sum_{y_2} \sqrt{p_{XY_2}(x,y_2)p_{XY_2}(x',y_2)} \\
&= \frac{1}{q-1} \sum_{\substack{x,x': \\ x\neq x'}} \sqrt{p_X(x)p_X(x')} \sum_{y_2} \sqrt{p_{Y_2|X}(y_2|x)p_{Y_2|X}(y_2|x')} \\
&= \frac{1}{q-1} \sum_{\substack{x,x': \\ x\neq x'}} \sqrt{p_X(x)p_X(x')} \sum_{y_2} \left[\sqrt{\sum_{y_1} p_{Y_1|X}(y_1|x)p_{Y_2|Y_1}(y_2|y_1)} \right. \\
&\qquad \left. \cdot \sqrt{\sum_{y_1} p_{Y_1|X}(y_1|x')p_{Y_2|Y_1}(y_2|y_1)} \right].
\end{aligned}
$$

Using Cauchy-Schwartz inequality gives

$$
\begin{aligned}
Z(X|Y_2) &\geq \frac{1}{q-1} \sum_{\substack{x,x': \\ x\neq x'}} \sqrt{p_X(x)p_X(x')} \sum_{y_2} \left[\sum_{y_1} \sqrt{p_{Y_1|X}(y_1|x)p_{Y_2|Y_1}(y_2|y_1)} \right. \\
&\qquad \left. \cdot \sum_{y_1} \sqrt{p_{Y_1|X}(y_1|x')p_{Y_2|Y_1}(y_2|y_1)} \right] \\
&\geq \frac{1}{q-1} \sum_{\substack{x,x': \\ x\neq x'}} \sqrt{p_X(x)p_X(x')} \sum_{y_2} \left[\sum_{y_1} p_{Y_2|Y_1}(y_2|y_1) \sqrt{p_{Y_1|X}(y_1|x)p_{Y_1|X}(y_1|x')} \right] \\
&= \frac{1}{q-1} \sum_{\substack{x,x': \\ x\neq x'}} \sqrt{p_X(x)p_X(x')} \sum_{y_1} \sqrt{p_{Y_1|X}(y_1|x)p_{Y_1|X}(y_1|x')} \\
&= Z(X|Y_1).
\end{aligned}
$$

□

A.3 Proof of Lemma 5

By Lemma 12, it is enough to bound the Kullback-Leibler distance between P and Q. Note that the Kullback-Leibler distance between two discrete probability measures can be expanded using chain-rule as

$$D(P(x^N)||Q(x^N)) = \sum_{i=1}^{N} D(P(x_i|x^{i-1})||Q(x_i|x^{i-1})).$$

Using the chain rule of Kullback-Leibler distance, we may write

$$\begin{aligned} D(P_{U^N}(u^N)||Q(u^N)) &= \sum_{i=1}^{N} D(P_{U_i|U^{i-1}}(u_i|u^{i-1})||Q(u_i|u^{i-1})), \\ &\overset{(a)}{=} \sum_{i\in\mathcal{I}} D(P_{U_i|U^{i-1}}(u_i|u^{i-1})||Q(u_i|u^{i-1})), \\ &\overset{(b)}{=} \sum_{i\in\mathcal{I}} \left[1 - H(U_i|U^{i-1})\right], \\ &\overset{(c)}{\leq} 2|\mathcal{I}|\delta_N. \end{aligned}$$

(a) is due to the fact that $Q(u_i|u^{i-1}) = P_{U_i|U^{i-1}}(u_i|u^{i-1})$ for $i \notin \mathcal{I}$ by definition in (2.47). (b) follows from the standard definition of the Kullback-Leibler distance:

$$\begin{aligned} D(P_{U_i|U^{i-1}}(u_i|u^{i-1})||Q(u_i|u^{i-1})) &= \sum_{u^i} P_{U^i}(u^i)\ \log \frac{P_{U_i|U^{i-1}}(u_i|u^{i-1})}{Q(u_i|u^{i-1})}, \\ &= \sum_{u^i} P_{U^i}(u^i)\ \log \frac{1}{Q(u_i|u^{i-1})} - H(U_i|U^{i-1}), \\ &= 1 - H(U_i|U^{i-1}). \end{aligned}$$

The first equality is the definition of the Kullback-Leibler distance. The second equality is from the definition of entropy. Last equality follows from the fact that $Q(u_i|u^{i-1}) = 1/q$ for $i \in \mathcal{I}$. (c) follows from Proposition 2 and the fact that $Z(U_i|U^{i-1}) \geq 1 - \delta_N$ for $i \in \mathcal{I}$ by definition.

Now, the proof of Lemma 5 may be completed as

$$\begin{aligned}\sum_{u^N \in \mathcal{X}^N} \left| P_{U^N}(u^N) - Q(u^N) \right| &\overset{(a)}{\leq} \sqrt{\kappa \; D(P_{U^N}(u^N)||Q(u^N))}, \\ &\overset{(b)}{\leq} \sqrt{2\kappa \cdot |\mathcal{I}| \cdot \delta_N}, \\ &\leq \sqrt{2\kappa \cdot N \cdot 2^{-N^{\beta'}}}.\end{aligned}$$

(a) is due to Pinsker's inequality in Lemma 12. (b) was proven above. $\sqrt{2\kappa \cdot N \cdot 2^{-N^{\beta'}}} < 2^{-N^{\beta}}$ is true for $\beta' \in (\beta, \frac{1}{2})$ and sufficiently large N. Thus, the total variation distance is bounded by $O(2^{-N^{\beta}})$ for any $0 < \beta < 1/2$. $\square$

Appendix B

Chapter 4

B.1 Recursive Formulas for SC Decoder

We use the following notation for joint probability for a block size of $2N$:

$$P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2}) \triangleq \\ \Pr\left[U_{2i-1} = u_{2i-1}, U_{2i} = u_{2i}, V_{2j-1} = v_{2j-1}, V_{2j} = v_{2j}| \right. \\ \left. Z^N = z^n, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right].$$

Note that we can write the following from the structure of polar transform:

$$P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2}) = \\ P_N^{(i,j)}(u_{2i-1} + u_{2i}, v_{2j-1} + v_{2j}|z_1^N, u_{1,o}^{2i-2} + u_{1,e}^{2i-2}, v_{1,o}^{2j-2} + v_{1,e}^{2j-2}) \cdot \\ P_N^{(i,j)}(u_{2i}, v_{2j}|z_{N+1}^{2N}, u_{1,e}^{2i-2}, v_{1,e}^{2j-2}).$$

Proof of (4.22).

$$P_{2N}^{(2i-1,2j-1)}(u_{2i-1}, v_{2j-1}|z^{2N}, u^{2i-2}, v^{2j-2}) = \sum_{u_{2i}, v_{2j}} P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2}).$$

□

Proof of (4.23).

$$P_{2N}^{(2i,2j-1)}(u_{2i}, v_{2j-1}|z^{2N}, u^{2i-1}, v^{2j-2}) = \frac{\sum_{v_{2j}} P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2})}{\Pr\left[U_{2i-1} = u_{2i-1}|Z^{2N} = z^{2N}, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right]}.$$

The denominator is expanded as:

$$\Pr\left[U_{2i-1} = u_{2i-1}|Z^N = z^n, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right] = \sum_{u_{2i}, v_{2j-1}, v_{2j}} P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2}).$$

Noting that this is equal to constant C_2 in (4.19) completes the proof. □

Proof of (4.24).

$$P_{2N}^{(2i-1,2j)}(u_{2i-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-1}) = \frac{\sum_{u_{2i}} P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2})}{\Pr\left[V_{2j-1} = v_{2j-1}|Z^N = z^n, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right]}.$$

The denominator is expanded as:

$$\Pr\left[V_{2j-1} = v_{2j-1}|Z^N = z^n, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right] = \sum_{u_{2i-1}, u_{2i}, v_{2j}} P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2}).$$

Noting that this is equal to constant C_3 in (4.20) completes the proof. □

Proof of (4.25).

$$P_{2N}^{(2i,2j)}(u_{2i}, v_{2j}|z^{2N}, u^{2i-1}, v^{2j-1}) = \frac{P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2})}{\Pr\left[U_{2i-1} = u_{2i-1}, V_{2j-1} = v_{2j-1}|Z^N = z^n, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right]}.$$

The denominator is expanded as:

$$\Pr\left[U_{2i-1} = u_{2i-1}, V_{2j-1} = v_{2j-1}|Z^N = z^n, U^{2i-2} = u^{2i-2}, V^{2j-2} = v^{2j-2}\right] = \sum_{u_{2i}, v_{2j}} P_{2N}(u_{2i-1}, u_{2i}, v_{2j-1}, v_{2j}|z^{2N}, u^{2i-2}, v^{2j-2}).$$

Noting that this is equal to constant C_4 in (4.21) completes the proof. □

B.2 Proof of Lemma 7

In the following, we make use of Definition 10 when we talk about vectors U^N, V^N, S^{2N} and their respective indices i, j, k under assumed path b^{2N}. By Lemma 12 it is enough to bound the Kullback-Leibler distance. Using the chain rule of Kullback-Leibler distance, we may decompose the total term into sets as follows

$$\begin{aligned} D(P(s^{2N})||Q(s^{2N})) &= \sum_{k=1}^{2N} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1})) \\ &= \sum_{k\in\tilde{\mathcal{I}}} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1}))+ \qquad \text{(B.1)} \\ &\quad \sum_{k\in\tilde{\mathcal{F}}_1} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1}))+ \qquad \text{(B.2)} \\ &\quad \sum_{k\in\tilde{\mathcal{F}}_2} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1})). \qquad \text{(B.3)} \end{aligned}$$

The first term (B.1) can be bounded by the standard definition of the Kullback-Leibler distance:

$$\begin{aligned} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1})) &= \sum_{s^k} P(s^k)\ \log \frac{P(s_k|s^{k-1})}{Q(s_k|s^{k-1})}, \\ &= \sum_{s^k} P(s^k)\ \log \frac{1}{Q(s_k|s^{k-1})} - H(S_k|S^{k-1}), \\ &= 1 - H(S_k|S^{k-1}), \\ &\leq 2\delta_N. \end{aligned}$$

The first equality is the definition of the Kullback-Leibler distance. The second equality is from the definition of entropy. The last equality follows from the fact that $Q(s_k|s^{k-1}) = 1/q$ for $k \in \tilde{\mathcal{I}}$. The last inequality is from Proposition 2 and the fact that $Z(U_i|U^{i-1}) \geq 1 - \delta_N$ for $k \in \tilde{\mathcal{I}}$ by definition. Then we have

$$\begin{aligned} \sum_{k \in \tilde{\mathcal{I}}} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1})) &\leq 2|\mathcal{I}| \cdot 2^{-N^{\beta'}}, \\ &\leq 2N \cdot 2^{-N^{\beta'}}. \end{aligned}$$

Thus, (B.1) is bounded by $O(2^{-N^{\beta}})$.

Now, we upper bound the second term (B.2). Note that the following is true for $k \in \tilde{\mathcal{F}}_1$ ($i \in \mathcal{F}_1$ and $b_k = 0$):

$$\begin{aligned} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1})) &= \sum_{s^k} P(s^k)\ \log \frac{P(s_k|s^{k-1})}{Q(s_k|s^{k-1})}, \\ &\overset{(a)}{=} \sum_{u^i, v^j} P(u^i, v^j)\ \log \frac{P(u_i|u^{i-1}, v^j)}{P(u_i|u^{i-1})}, \\ &= H(U_i|U^{i-1}) - H(U_i|U^{i-1}, V^j). \end{aligned}$$

(a) is from the definition of probability measure Q in (4.72). Then, observe that

for $i \in \mathcal{F}_1$ we have

$$\begin{aligned} H(U_i|U^{i-1}) - H(U_i|U^{i-1}, V^j) &\overset{(a)}{\leq} \min\{\kappa Z(U_i|U^{i-1}), 1 - Z(U_i|U^{i-1}, V^j)^2\} \\ &\overset{(b)}{\leq} \max\{2, \kappa\} \cdot \delta_N, \end{aligned}$$

where $\kappa = (q-1)/\ln q$. (a) is from Proposition 2 and due to the fact that $H(\cdot|\cdot)$, $Z(\cdot|\cdot) \in [0,1]$. (b) is because of definition of $\mathcal{F}_1$ in (4.64). Defining $\kappa' \triangleq \max\{2, \kappa\}$ we get

$$\begin{aligned} \sum_{k \in \tilde{\mathcal{F}}_1} D(P(s_k|s^{k-1})||Q(s_k|s^{k-1})) &\leq \kappa' |\tilde{\mathcal{F}}_1| \cdot 2^{-N^{\beta'}}, \\ &\leq \kappa' N \cdot 2^{-N^{\beta'}}. \end{aligned}$$

Thus, (B.2) is also bounded by $O(2^{-N^\beta})$.

The last term (B.3) in the summation can be proven similarly. Combining the result with Lemma 12 gives the desired result that the total variation distance is bounded by $O(2^{-N^\beta})$ for any $0 < \beta < 1/2$. $\square$

Appendix C

Chapter 5

C.1 Proof of Lemma 9

In the following, we make use of Definition 10 when we talk about vectors U^N, V^N, S^{2N} and their corresponding indices i, j, k under assumed path b^{2N}. By Lemma 12, it is enough to bound the Kullback-Leibler distance. First note that since $Q(x^N, y^N) = P(x^N, y^N)$,

$$D(P(s^{2N}, x^N, y^N)||Q(s^{2N}, x^N, y^N)) = D(P(s^{2N}|x^N, y^N)||Q(s^{2N}|x^N, y^N)).$$

Furthermore, we can use the chain rule for the Kullback-Leibler distance to write

$$D(P(s^{2N}|x^N, y^N)||Q(s^{2N}|x^N, y^N)) = \sum_{k=1}^{2N} D(P(s_k|s^{k-1}, x^N, y^N)||Q(s_k|s^{k-1}, x^N, y^N)).$$

Using the chain rule of the Kullback-Leibler distance, we may decompose the total term into sets as follows

$$
\begin{aligned}
&D(P(s^{2N},x^N,y^N)||Q\ (s^{2N},x^N,y^N)) \\
&\qquad = \sum_{k=1}^{2N} D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N)) \\
&\qquad = \sum_{k\in\tilde{\mathcal{F}}_{\bar{X}}} D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N))+ \qquad \text{(C.1)} \\
&\qquad\quad \sum_{k\in\tilde{\mathcal{I}}_{\bar{X}}} D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N))+ \qquad \text{(C.2)} \\
&\qquad\quad \sum_{k\in\tilde{\mathcal{F}}_{\bar{Y}}} D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N))+ \qquad \text{(C.3)} \\
&\qquad\quad \sum_{k\in\tilde{\mathcal{I}}_{\bar{Y}}} D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N)). \qquad \text{(C.4)}
\end{aligned}
$$

We upper bound the first term (C.1) as follows. Note that the following is true for $k\in\tilde{\mathcal{F}}_{\bar{X}}$ ($i\in\mathcal{F}_{\bar{X}}$ and $b_k=0$):

$$
\begin{aligned}
&D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N)) \\
&\qquad = \sum_{s^k,x^N,y^N} P(s^k,x^N,y^N)\ \log\frac{P(s_k|s^{k-1},x^N,y^N)}{Q(s_k|s^{k-1},x^N,y^N)}, \\
&\qquad \overset{(a)}{=} \sum_{u^i,v^j,x^N,y^N} P(u^i,v^j,x^N,y^N)\ \log\frac{P(u_i|u^{i-1},v^j,x^N,y^N)}{P(u_i|u^{i-1})}, \\
&\qquad \overset{(b)}{=} H(U_i|U^{i-1}) - H(U_i|U^{i-1},V^j,X^N,Y^N), \\
&\qquad \overset{(c)}{=} H(U_i|U^{i-1}) - H(U_i|U^{i-1},X^N).
\end{aligned}
$$

(a) is from the definition of probability measure Q in (5.66). (b) is from the definition of entropy. (c) is due to the special Markov distribution of the random variables. Then, observe that for $i\in\mathcal{F}_{\bar{X}}$ we have

$$
\begin{aligned}
H(U_i|U^{i-1}) - H(U_i|U^{i-1},X^N) &\overset{(a)}{\leq} \kappa Z(U_i|U^{i-1}) \\
&\overset{(b)}{\leq} \kappa\cdot\delta_N,
\end{aligned}
$$

where $\kappa = (q-1)/\ln q$. (a) is from Proposition 2 and due to the fact that $H(\cdot|\cdot)$, $Z(\cdot|\cdot) \in [0,1]$. Also, the term is positive from the fact that $H(U_i|U^{i-1}) \geq H(U_i|U^{i-1}, X^N)$. (b) is because of definition of $\mathcal{F}_{\bar{X}}$ in (5.49). Thus, we get

$$\begin{aligned}\sum_{k\in\tilde{\mathcal{F}}_{\bar{X}}} D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N)) &\leq \kappa|\tilde{\mathcal{F}}_{\bar{X}}|\cdot 2^{-N^{\beta'}},\\ &\leq \kappa N\cdot 2^{-N^{\beta'}}.\end{aligned}$$

Thus, (C.1) is bounded by $O(2^{-N^{\beta}})$.

For the second term (C.2), the following is true

$$\begin{aligned}&D(P(s_k|s^{k-1},x^N,y^N)||Q(s_k|s^{k-1},x^N,y^N))\\ &\qquad= \sum_{s^k,x^N,y^N} P(s^k,x^N,y^N)\ \log\frac{P(s_k|s^{k-1},x^N,y^N)}{Q(s_k|s^{k-1},x^N,y^N)},\\ &\qquad\overset{(a)}{=} \sum_{u^i,v^j,x^N,y^N} P(u^i,v^j,x^N,y^N)\ \log\frac{P(u_i|u^{i-1},v^j,x^N,y^N)}{P(u_i|u^{i-1},x^N)},\\ &\qquad\overset{(b)}{=} \sum_{u^i,x^N} P(u^i,x^N)\ \log\frac{P(u_i|u^{i-1},x^N)}{P(u_i|u^{i-1},x^N)},\\ &\qquad= 0.\end{aligned}$$

(a) is from the definition of probability measure Q in (5.66). (b) is from the fact that $P(u_i|u^{i-1},v^j,x^N,y^N) = P(u_i|u^{i-1},x^N)$ which is due to the Markov chain probability distribution of the problem.

Other terms (C.3) and (C.4) in the summation can be proven similar to (C.1) and (C.2), respectively. Combining the result with Lemma 12 gives the desired result that the total variation distance is bounded by $O(2^{-N^{\beta}})$ for any $0 < \beta < 1/2$. $\square$

C.2 Proof of Lemma 11

In the following, we make use of Definition 10 when we talk about vectors U^N, V^N, S^{2N} and their corresponding indices i, j, k under assumed path b^{2N}. By Lemma 12, it is enough to bound the Kullback-Leibler distance. First note that since $Q_{X^N}(x^N) = P_{X^N}(x^N)$,

$$D(P(x^N, s^{2N})||Q(x^N, s^{2N})) = D(P(s^{2N}|x^N)||Q(s^{2N}|x^N)).$$

Furthermore, we can use the chain rule for Kullback-Leibler distance to write

$$D(P(s^{2N}|x^N)||Q(s^{2N}|x^N)) = \sum_{k=1}^{2N} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1})).$$

Using the chain rule of the Kullback-Leibler distance, we may decompose the total term into sets as follows:

$$\begin{aligned}
&D(P(s^{2N}, x^N)||Q\ (s^{2N}, x^N)) \\
&\qquad = \sum_{k=1}^{2N} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1})) \\
&\qquad = \sum_{k\in\tilde{\mathcal{I}}_1} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1}))+ && \text{(C.5)} \\
&\qquad\quad \sum_{k\in\tilde{\mathcal{F}}_1} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1}))+ && \text{(C.6)} \\
&\qquad\quad \sum_{k\in\tilde{\mathcal{I}}_2} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1}))+ && \text{(C.7)} \\
&\qquad\quad \sum_{k\in\tilde{\mathcal{F}}_2} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1})). && \text{(C.8)}
\end{aligned}$$

Note that, since $Q(s_k|x^N, s^{k-1}) = P(s_k|x^N, s^{k-1})$ for $k \in \tilde{\mathcal{I}}_1$ and $k \in \tilde{\mathcal{I}}_2$, the first (C.5) and third (C.7) terms are zero.

We upper bound the second term (C.6) as follows. Note that the following is

true for $k \in \tilde{\mathcal{F}}_1$ ($i \in \mathcal{F}_1$ and $b_k = 0$):

$$\begin{aligned} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1})) & \\ &= \sum_{s^k, x^N} P(s^k, x^N) \log \frac{P(s_k|x^N, s^{k-1})}{Q(s_k|x^N, s^{k-1})}, \\ &\overset{(a)}{=} \sum_{u^i, v^j, x^N} P(u^i, v^j, x^N) \log \frac{P(u_i|x^N, u^{i-1}, v^j)}{P(u_i|u^{i-1})}, \\ &= H(U_i|U^{i-1}) - H(U_i|X^N, U^{i-1}, V^j). \end{aligned}$$

(a) is from the definition of probability measure Q in (5.129). Then, observe that for $i \in \mathcal{F}_1$ we have

$$\begin{aligned} H(U_i|U^{i-1}) - H(U_i|X^N, U^{i-1}, V^j) &\overset{(a)}{\leq} \min\{\kappa Z(U_i|U^{i-1}), 1 - Z(U_i|X^N, U^{i-1}, V^j)^2\} \\ &\overset{(b)}{\leq} \max\{2, \kappa\} \cdot \delta_N, \end{aligned}$$

where $\kappa = (q-1)/\ln q$. (a) is from Proposition 2 and due to the fact that $H(\cdot|\cdot)$, $Z(\cdot|\cdot) \in [0,1]$. (b) is because of definition of $\mathcal{F}_1$ in (5.111). Defining $\kappa' \triangleq \max\{2, \kappa\}$, we get

$$\begin{aligned} \sum_{k \in \tilde{\mathcal{F}}_1} D(P(s_k|x^N, s^{k-1})||Q(s_k|x^N, s^{k-1})) &\leq \kappa' |\tilde{\mathcal{F}}_1| \cdot 2^{-N^{\beta'}}, \\ &\leq \kappa' N \cdot 2^{-N^{\beta'}}. \end{aligned}$$

Thus, (C.6) is also bounded by $O(2^{-N^\beta})$.

The fourth term (C.8) in the summation can be proven similarly. Combining the result with Lemma 12 gives the desired result that the total variation distance is bounded by $O(2^{-N^\beta})$ for any $0 < \beta < 1/2$. □

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[2] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, Apr. 1950.

[3] M. J. E. Golay, "Notes on digital coding," *Proceedings of the IRE*, vol. 37, p. 657, June 1949.

[4] D. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *Transactions of the IRE Professional Group on Electronic Computers*, vol. EC-3, pp. 6–12, Sept. 1954.

[5] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, pp. 38–49, Sept. 1954.

[6] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error-correcting binary group codes," *Inform. Contr.*, vol. 3, pp. 68–79, Mar. 1960.

[7] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, pp. 300–304, June 1960.

[8] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, vol. 4, pp. 37–46, Mar. 1955.

[9] R. G. Gallager, *Low-Density Parity-Check Codes*. PhD thesis, M.I.T., 1963.

[10] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. of the IEEE International Conference on Communications (ICC '93)*, vol. 2, pp. 1064–1070, May 1993.

[11] D. J. Costello and G. D. Forney, "Channel Coding: The Road to Channel Capacity," *Proc. of the IEEE*, vol. 95, pp. 1150–1177, June 2007.

[12] E. Arıkan, "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.

[13] E. Arıkan, "Channel combining and splitting for cutoff rate improvement," *IEEE Transactions on Information Theory*, vol. 52, pp. 628–639, Feb. 2006.

[14] E. Arıkan, "A survey of Reed-Muller codes from polar coding perspective," in *Proc. of the IEEE Information Theory Workshop (ITW 2010)*, pp. 1–5, 2010.

[15] E. Arıkan and E. Telatar, "On the rate of channel polarization," in *Proc. of the IEEE International Symposium Information Theory (ISIT 2009)*, pp. 1493–1495, July 2009.

[16] S. B. Korada, *Polar codes for channel and source coding*. PhD thesis, EPFL, Lausanne, Switzerland, July 2009.

[17] E. Şaşoğlu, E. Telatar, and E. Arıkan, "Polarization for arbitrary discrete memoryless channels," in *Proc. of the IEEE Information Theory Workshop (ITW 2009)*, Oct. 2009.

[18] E. Şaşoğlu and E. Telatar, "Polar codes for the two-user binary-input multiple-access channel," in *Proc. of the IEEE Information Theory Workshop (ITW 2010)*, pp. 1–5, 2010.

[19] E. Şaşoğlu, "An entropy inequality for q-ary random variables and its application to channel polarization," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2010)*, pp. 1360–1363, June 2010.

[20] E. Şaşoglu, "Polarization in the presence of memory," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2011)*, pp. 189–193, July 2011.

[21] E. Şaşoglu, *Polar Coding Theorems for Discrete Systems*. PhD thesis, EPFL, 2011.

[22] E. Arıkan, "Source Polarization," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2010)*, pp. 899–903, June 2010.

[23] E. Arıkan, "Systematic Polar Coding," *IEEE Communications Letters*, vol. 15, pp. 860 –862, Aug. 2011.

[24] I. Tal and A. Vardy, "How to Construct Polar Codes," *arXiv:1105.6164*, May 2011.

[25] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2011)*, pp. 1 –5, Aug. 2011.

[26] N. Goela, E. Abbe, and M. Gastpar, "Polar codes for broadcast channels," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2013)*, pp. 1127–1131, July 2013.

[27] J. Honda and H. Yamamoto, "Polar Coding Without Alphabet Extension for Asymmetric Models," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 7829–7838, 2013.

[28] E. Arıkan, "Polar coding for the Slepian-Wolf problem based on monotone chain rules," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2012)*, pp. 566 –570, July 2012.

[29] S. Önay, "Successive cancellation decoding of polar codes for the two-user binary-input MAC," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2013)*, pp. 1122–1126, July 2013.

[30] Q. Shi, *Polar Codes for Multiple Descriptions*. PhD thesis, McMaster University, Oct. 2013.

[31] M. Valipour and S. Yousefi, "Multiple description coding with polar codes," in *Proc. of the 27th Biennial Symposium on Communications (QBSC)*, pp. 109–112, June 2014.

[32] C. Leroux, I. Tal, A. Vardy, and W. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1665–1668, 2011.

[33] A. Pamuk, "An FPGA Implementation Architecture for Decoding of Polar Codes," in *Proc. of the 8th International Symposium on Wireless Communication Systems*, 2011.

[34] J. Lin and Z. Yan, "Efficient list decoder architecture for polar codes," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1022–1025, June 2014.

[35] A. Balatsoukas-Stimming, A. Raymond, W. Gross, and A. Burg, "Hardware Architecture for List Successive Cancellation Decoding of Polar Codes," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, pp. 609–613, Aug. 2014.

[36] B. Yuan and K. Parhi, "Low-Latency Successive-Cancellation List Decoders for Polar Codes With Multibit Decision," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–1, 2014.

[37] C. Zhang, X. You, and J. Sha, "Hardware architecture for list successive cancellation polar decoder," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 209–212, June 2014.

[38] J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, pp. 471–480, July 1973.

[39] A. Wyner, "Recent results in the Shannon theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, 1974.

[40] I. Tal and A. Vardy, "List Decoding of Polar Codes," *arXiv:1206.0050*, May 2012.

[41] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," in *Proc. of the Data Compression Conference (DCC 1999)*, pp. 158 –167, Mar. 1999.

[42] P. L. Dragotti and M. Gastpar, *Distributed Source Coding*. Burlington, MA: Academic, 2008.

[43] Z. Tu, J. Li, and R. S. Blum, "An efficient SF–ISF approach for the Slepian–Wolf source coding problem," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 961–971, May 2005.

[44] A. Roumy, K. Lajnef, and C. Guillemot, "Rate–adaptive turbo–syndrome scheme for Slepian–Wolf coding," in *Proc. of the IEEE Asilomar Conference on Signals, Systems, and Computers*, pp. 545–549, Nov. 2007.

[45] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, pp. 76–80, Jan. 1978.

[46] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Transactions on Information Theory*, vol. 40, pp. 1663–1666, Sept. 1994.

[47] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information using low-density parity-check codes," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 2, pp. 1300– 1304 vol.2, Nov. 2002.

[48] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. of the IEEE International Data Compression Conference (DCC 2002)*, pp. 252–261, Apr. 2002.

[49] J. Bajcsy and P. Mitran, "Coding for the Slepian–Wolf problem with turbo codes," in *Proc. of the IEEE International Global Communications Conference (GLOBECOM '01)*, pp. 1400–1404, Dec. 2001.

[50] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, pp. 417–419, Oct. 2001.

[51] F. Cabarcas and J. Garcia-Frias, "Approaching the Slepian–Wolf boundary using practical channel codes," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2004)*, p. 330, June 2004.

[52] M. Sartipi and F. Fekri, "Distributed source coding in wireless sensor networks using LDPC coding: The entire Slepian–Wolf rate region," in *Proc. of the IEEE Wireless Communications and Networking Conference*, vol. 4, pp. 1939–1944, Mar. 2005.

[53] M. Sartipi and F. Fekri, "Distributed source coding using short to moderate length rate–compatible LDPC codes: The entire Slepian–Wolf rate region," *IEEE Transactions on Communications*, vol. 56, pp. 400–411, Mar. 2008.

[54] S. Pradhan and K. Ramchandran, "Distributed source coding: symmetric rates and applications to sensor networks," in *Proc. of the Data Compression Conference (DCC 2000)*, pp. 363 –372, 2000.

[55] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Proc. of the Data Compression Conference (DCC 2004)*, pp. 292–301, Mar. 2004.

[56] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Design of Slepian–Wolf codes by channel code partitioning," in *Proc. of the IEEE International Data Compression Conference (DCC 2004)*, pp. 302–311, Mar. 2004.

[57] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 1495 –1507, Apr. 2006.

[58] N. Gehrig and P. Dragotti, "Symmetric and asymmetric Slepian-Wolf codes with systematic and nonsystematic linear codes," *IEEE Communications Letters*, vol. 9, pp. 61 – 63, Jan. 2005.

[59] S. Önay, "Polar codes for distributed source coding," *Electronics Letters*, vol. 49, pp. 346–348, Feb. 2013.

[60] E. Abbe and E. Telatar, "Polar Codes for the m-User Multiple Access Channel," *IEEE Transactions on Information Theory*, vol. 58, pp. 5437 –5448, Aug. 2012.

[61] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Achieving the Uniform Rate Region of Multiple Access Channels Using Polar Codes," *arXiv:1307.2889 [cs, math]*, July 2013.

[62] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Techniques for polar coding over multiple access channels," in *Proc. of the 48th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, Mar. 2014.

[63] I. Tal and A. Vardy, "How to Construct Polar Codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

[64] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2 ed., July 2006.

[65] A. G. Sahebi and S. Pradhan, "Polar codes for some multi-terminal communications problems," in *Proc. of the IEEE International Symposium on Information Theory (ISIT 2014)*, pp. 316–320, June 2014.

[66] A. G. Sahebi and S. S. Pradhan, "Nested Polar Codes Achieve the Shannon Rate-Distortion Function and the Shannon Capacity," *arXiv:1401.6482 [cs, math]*, Jan. 2014.

[67] Y. Zhang, S. Dumitrescu, J. Chen, and Z. Sun, "LDGM-Based Multiple Description Coding for Finite Alphabet Sources," *IEEE Transactions on Communications*, vol. 60, no. 12, pp. 3671–3682, 2012.

[68] A. A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, UK; New York: Cambridge University Press, 2011.

[69] J. Wang, J. Chen, L. Zhao, P. Cuff, and H. Permuter, "A random variable substitution lemma with applications to multiple description coding," *arXiv preprint arXiv:0909.3135*, 2009.