Modern Error Control Codes and Applications to Distributed Source Coding

A Thesis
Presented to
The Academic Faculty

by

Mina Sartipi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2006

Modern Error Control Codes and Applications to Distributed Source Coding

Approved by:

Dr. Faramarz Fekri, Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Steven W McLaughlin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Guotong Zhou
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Christopher Heil
School of Mathematics
Georgia Institute of Technology

Date Approved: July 10, 2006

This dissertation is dedicated to my family.

Parivash, Hamid, Mojgan, and Maryam Sartipi:

thank you for your love, encouragement, and support.

ACKNOWLEDGEMENTS

Through the rewarding journey of graduate school, there were guides and companions. The first and most important guide is my advisor, Prof. Faramarz Fekri. From the young graduate student, he led me though many challenges and transformed me. He was always there to welcome my successes and problems alike, never failing to give me encouragement and suggestions.

The path on this journey is well tread with the footsteps of many predecessors. Prof. Steven McLaughlin, Prof. Raghupathy Sivakumar, Prof. Guotong Zhou, and Prof. Christopher Heil on my thesis committee have walked this path before. Their experience, support, and encouragement have truly made a difference on my journey.

Certainly, this journey was not made alone. Through my struggles and successes my fellow group members shared my disappointments and joy. Here, I would like to extend my gratitude to Badri Vellambi, Farshid Delgosha, Hossein Pishro Nik, Nazanin Rahnavard, and Kevin Chan. Extra special thanks must be given to Nazanin and Kevin for their friendship.

Friends have always been important to me. During my journey there was no shortage of great friendships. Some started as office-mates, others through adventures and friends. I would like to give thanks to my friends; Amer Abufadel, Kevin Chan, Houri Johari, Will Leven, Sam Li, Salman Mohagheghi, Maneli Noorkami, Sara Pashaie-Rad, Parisa Pooyan, Nazanin Rahnavard, Mina Rais Zadeh, Azin Sahabi, Nusha Safabakhsh, Pegah Zamani, and George Yu. They have added laughter, happiness, and comfort to my journey. I would like to give special thanks to Maneli. Four years of being roommates and friends, we have experienced so much together. Also, I would like to give extra special thanks to my close and special friend George. In the last year of my journey, the most difficult year, when the journey gets hard George was always there to encourage me, push me, and support me. George, thank you.

To many, paper works are done automatically, the garbage seems to disappear every night, and the floors magically stay clean. Certainly, that is not true. Everyday I come to work, I appreciate the hard work of the CSIP staff for making my work possible. Here, I would like to extend my gratitude to Kay Gilstrap, Christy Ellis, and Carla Zachery.

On this journey, it was difficult to be away from home for such a long time. Although, I was away from my family, my sisters, and my home, there was no shortage of love and support close to me. Here, I would like to express my thanks and love to my family specially my aunts, Sorahi Toloyan and Pouran Espahbodi, who loved and supported me like their daughter.

One must remember those who made this journey possible. The years of changing diapers, enduring cry for milk, stubborn outbursts, and requests for more money coupled with many hugs, kisses, and love have made this journey possible. My greatest thanks and love to my parents, Parivash and Hamid, and my sisters, Mojgan and Maryam. although, they were miles and miles away from me, but their love and support were always with me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation first studies two-dimensional wavelet codes (TDWCs). TDWCs are introduced as a solution to the problem of designing a 2-D code that has low decoding-complexity and has the maximum erasure-correcting property for rectangular burst erasures. The half-rate TDWCs of dimensions $N_1 \times N_2$ satisfy the Reiger bound with equality for burst erasures of dimensions $N_1 \times N_2/2$ and $N_1/2 \times N_2$, where $GCD(N_1, N_2) = 2$. Examples of TDWC are provided that recover any rectangular burst erasure of area $N_1 N_2/2$. These lattice-cyclic codes can recover burst erasures with a simple and efficient ML decoding.

This work then studies the problem of distributed source coding for two and three correlated signals using channel codes. We propose to model the distributed source coding problem with a set of parallel channel that simplifies the distributed source coding to designing non-uniform channel codes. This design criterion improves the performance of the source coding considerably. LDPC codes are used for lossless and lossy distributed source coding, when the correlation parameter is known or unknown at the time of code design. We show that distributed source coding at the corner point using LDPC codes is simplified to non-uniform LDPC code and semi-random punctured LDPC codes for a system of two and three correlated sources, respectively. We also investigate distributed source coding at any arbitrary rate on the Slepian-Wolf rate region. This problem is simplified to designing a rate-compatible LDPC code that has unequal error protection property. This dissertation finally studies the distributed source coding problem for applications whose wireless channel is an erasure channel with unknown erasure probability. For these application, rateless codes are better candidates than LDPC codes. Non-uniform rateless codes and improved decoding algorithm are proposed for this purpose. We introduce a reliable, rate-optimal, and energy-efficient multicast algorithm that uses distributed source coding and rateless coding. The proposed multicast algorithm performs very close to network coding, while it has lower complexity and higher adaptability.

CHAPTER I

INTRODUCTION

In this work, the theoretical and practical aspects of recently developed or rediscovered error control codes are studied. We consider wavelet codes, LDPC codes and fountain codes as modern error control codes. First, we introduce two-dimensional wavelet codes and investigate their application in recovering 2-D burst erasures. Then, we study the application of LDPC codes in distributed source coding. Finally, we explore fountain codes and their applications in distributed source coding and multicasting over lossy wireless networks.

1.1 Two-Dimensional Wavelet Codes

The first application of finite-field wavelet transforms to error-control coding was introduced in [18, 19]. We extend this modern error control code to the 2-D case. To construct two-dimensional wavelet codes (TDWCs), we employ two-channel maximally decimated orthogonal filter banks. The synthesis bank constructs the encoder of the TDWC and the corresponding analysis bank constructs the syndrome generator. By using a sampling matrix with determinant of two, a half-rate TDWC is generated. To find codes with desired properties such as good minimum distance, self duality, or burst erasure correctability, we add two pre-filters to the two channels of the synthesis bank. Minimum distance is an important property of error-control codes, which is defined as the minimum Hamming distance between all pairs of codewords. Using a proper prefilter, we generate a half-rate TDWC of length 24 that has minimum distance of eight. Self-dual codes attract a great deal of attention, mainly due to their intimate connections with important problems in algebra, combinatorics, and number theory [32]. Therefore, we develop criteria to construct 2-D self-dual codes and weakly self-dual codes. We also investigate the problem of designing a 2-D code that has low decoding complexity and has the maximum erasure-correcting

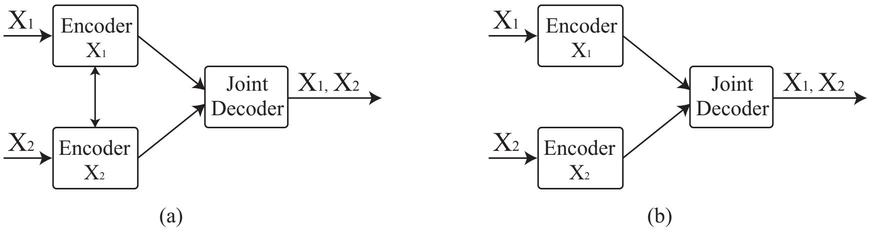

Figure 1: (a) Joint encoding of X_1 and X_2. (b) Distributed encoding of X_1 and X_2.

property for erasure bursts of dimension $b_1 \times b_2$. There are many data transmission and storage systems with two-dimensional (2-D) data structures that suffer from 2-D bursts of error and erasures. 2-D codes can be used to combat such errors and erasures. We design half-rate TDWCs of dimension $N_1 \times N_2$ that are capable of recovering erasure bursts of size $N_1/2 \times N_2$ and $N_1 \times N_2/2$. The only constraint on the parameters (N_1 and N_2) of our codes is that the greatest common devisor of them must be two, $GCD(N_1, N_2) = 2$. TDWCs are lattice cyclic that makes them to have a simple and efficient ML decoding for burst erasures. We present examples of TDWCs that satisfy the Reiger bound with equality, i.e., they are capable of correcting any burst of size $\frac{N_1 N_2}{2}$.

1.2 Distributed Source Coding

Consider a communication system that is consist of two sources X_1 and X_2. Let $\{X_{1_i}, X_{2_i}\}_{i=1}^{\inf}$ be a sequence of independent and identically distributed (i.i.d.) drawings of a pair of correlated discrete random variables X_1 and X_2. The compression techniques, exploiting this correlation, eliminate any redundancy in data transmission.

We know from Shannon's source coding theory [13] that a rate given by the joint entropy $h(X_1, X_2)$ of X_1 and X_2 is sufficient, if we are encoding them together as in Fig. 1a. For example, we can first compress X_1 into $h(X_1)$ bits per sample and based on the complete knowledge of X_1 at both the encoder and decoder, we then compress X_2 into $h(X_2|X_1)$ bits per sample. But what if X_1 and X_2 must be encoded separately for some user to reconstruct both of them, as in Fig. 1b? As shown in Fig. 1b, each source compresses its data without communicating with the other source and sends the compressed data to the sink. The decoder, on the other hand has access to both compressed data. This procedure is known as distributed source coding, since sources are distributed and being compressed independently.

To motivate the distributed source coding, as an application, consider a wireless sensor network. In recent years, wireless sensor networks have received a great deal of attention by researchers. Wireless sensor networks consist of several small nodes deployed throughout an area of interest to sense and collect data. The data observed by each sensor node is transmitted through multiple hops (the path nodes) to a base station. These tiny nodes have limited power recourses. Since applications involving wireless sensor networks require long system lifetimes, energy usage must be minimized. Our main goal is to reduce the transmission power usage in the wireless sensor networks. Since different sensor nodes partially monitor the same spatial region, data is often correlated. Thus, it is desirable to remove the redundant information among the nodes and hence reduce the amount of data to be transmitted.

According to the Slepian-Wolf theorem [69], the output of two correlated sources that do not communicate can be compressed with the same rate as if they were communicating. This is true when the decoder has access to both of the compressed outputs. In other words, knowledge of the joint entropy of X_1 and X_2, $h(X_1, X_2)$, is sufficient to optimally compress X_1 and X_2 in Fig. 1(b). The Slepian-Wolf rate region for two arbitrarily correlated sources X_1 and X_2, as shown in Fig. 2, is bounded by the following inequalities:

$$R_{X_1} \geq h(X_1|X_2), \quad R_{X_2} \geq h(X_2|X_1),$$

$$R_{X_1} + R_{X_2} \geq h(X_1, X_2). \tag{1}$$

For implementation, Wyner proposed to use channel coding for distributed source coding [75]. He proposed to partition the space of all possible source outcomes into disjoint sets that are the cosets of some good linear channel codes. The index of each coset constructs the compressed version of the elements of that coset. In distributed source coding via channel coding, this index is either the syndrome or the parity sequence corresponding to the

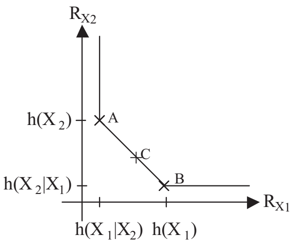

Figure 2: Slepian-Wolf rate region for two sources.

elements of the coset. We investigate the application of low-density parity-check (LDPC) codes and Bose, Ray-Chaudhuri, Hocquenghem (BCH) codes into distributed source coding. The simulation results show that for moderate-length sequences, a LDPC code corrects the same number of error bits/ packets as a BCH code does, while it has a lower redundancy. We also compare LDPC codes with BCH codes considering the decoding energy. As shown in [58], the energy-efficiency factor involves the energy efficiency and the reliability factor, where the energy efficiency is defined as the energy for communication of the information bits divided by the sum of total energy for communication of both the information bits and the redundant bits and decoding energy consumption. The simulation results in [59] confirm that LDPC codes are more energy-efficient than BCH codes. Therefore, the focus of our work is on distributed source coding using LDPC codes.

1.2.1 Distributed Source Coding Using LDPC Codes

LDPC codes were first discovered by Gallager [21] in 1962. Recently, there has been a tremendous amount of work on these codes which resulted in considerable improvement in this area. These codes have been shown to approach the theoretical Shannon limit, while having efficient decoding algorithms. We investigate the application of LDPC codes into distributed source coding. To design the LDPC code for such applications, we propose to use non-uniform LDPC codes for distributed source coding at the asymmetric rates. As opposed to previous work which design the LDPC code for the equivalent channel, we take a completely new approach. We take both the correlation channel and the wireless channel and formulate the source coding problem to the code design over parallel channels. We design a non-uniform LDPC code for this set of parallel channels. Our proposed design criteria improve the performance of the distributed source coding, significantly. We also extend our method to three sources. We study the distributed source coding of three sources that are pairwise correlated with arbitrary correlation probability. As a special case, we also study a simpler case, where sources are correlated with the same correlation probability. We show that this problem can be modeled with a set of parallel channels, which simplifies the problem to semi-random puncturing. This new model improves the performance of the distributed source coding significantly. Then, we study the distributed source coding of correlated sources that can achieve every arbitrary coding rate on the Slepian-Wolf rate region. We propose a scheme for this problem using a single systematic channel code. By modeling the distributed source coding with a set of parallel channels, we simplify the distributed source coding problem to the rate-compatible LDPC code design with an unequal error protection property.

1.2.2 Distributed Source Coding Using Rateless Codes

Rateless codes are a new class of codes that have been invented recently. LT codes [41], raptor codes [68], and Online codes [44] are examples of such codes. Unlike the traditional codes, rateless codes on lossy channels do not assume any knowledge about the channel. Therefore, rateless codes are very suitable candidates in the applications that the channel erasure probability is unknown, nonuniform, or time-varying. It was shown that rateless codes have very simple encoding and decoding algorithms. Asymptotically good degree distributions for them were also developed in introduced in [41,44,68]. We study distributed source coding using rateless codes. We show that our method is applicable to any wireless channel whose erasure probability is unknown in advance. A technique for designing LT-codes over a set of parallel subchannels is proposed. The simulation results show that our proposed method performs very close to distributed source coding using LDPC codes. However, the LT code-based distributed source coding has the additional advantage that we do not need to know the erasure probability. We also introduce an improved LT-decoding algorithm that improves the performance of the LT-codes, significantly.

1.3 Lossy Distributed Source Coding Using LDPC Codes

In many applications such as sensor networks, some distortion can be tolerated. In these cases, lossy distributed source coding results in compression with higher rates and consequently saves more energy. Wyner and Ziv were the first who studied the lossy distributed source coding and provided a rate-distortion function for lossy distributed source coding of two sources. We introduce a method for lossy distributed source coding based on parity bits using LDPC codes. LDPC codes are chosen because of their good performance and their practically feasible belief propagation decoding. We provide the design procedure for the LDPC code that guarantees performance close to the Wyner-Ziv limit for long-length LDPC codes. We show that there exists an LDPC code with specific column weight that attains the rate-distortion function asymptotically.

1.4 Energy-Efficient and Rate-optimal Multicasting Using Rateless Codes

We propose a Distributed Source Coding-based Multicast algorithm (DSCM) that is energy-efficient, rate-optimal, and reliable over multiple disjoint paths. Our proposed algorithm consists of rateless coding and distributed source coding. Rateless coding is used to provide reliability and rate optimality, while distributed source coding is used to ensure energy efficiency. First, we develop a unicast algorithm for a randomly deployed lossy wireless network with a single source and a single receiver. Then, we modify our proposed method for multicasting over lossy wireless networks with local information. For these networks, we also provide a multicast subgraph algorithm with the goal of delivering information to destinations by maximizing the use of common links. We show that distributed source coding coupled with the multicast subgraph algorithm generate a reliable, rate-optimal, and energy-efficient multicast algorithm. We compare our scheme with energy-efficient network coding. Our simulations reveal that our schemes perform very close to network coding, while having lower complexity and higher adaptability.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 introduces two-dimensional wavelet codes and investigates the properties of these codes and their burst erasure recovery. Chapter 3 studies distributed source coding using non-uniform LDPC codes at asymmetric rate. Chapter 4 extends the distributed source coding at asymmetric rate to any arbitrary rate on the Slepian-Wolf rate region using a single LDPC code. This chapter shows how this problem can be simplified to designing a rate-compatible LDPC code with non-equal error correction property. Chapter 5 introduces a technique for lossy distributed source coding using LDPC codes. This chapter also provides the design procedure for the LDPC code that guarantees a performance close to the Wyner-Ziv limit for long-length LDPC codes. Chapter 6 investigates two problems related to distributed source coding . 1: Distributed source codingwhen the correlation parameter in unknown in advance and 2: Distributed source codingusing rateless codes. The latter is suitable for applications when the noise of wireless channel is unknown in advance. Chapter 7 introduces a new multicast algorithm that is rate-optimal, energy-efficient, and reliable. Finally, Chapter 8 summarizes the complete work and points out some of the possible future research directions.

CHAPTER II

TWO-DIMENSIONAL WAVELET CODE

2.1 Introduction

This chapter introduces two-dimensional wavelet codes (TDWCs). The synthesis bank of a two-channel two-variable filter bank over the finite field is used to design a two-dimensional (2-D) code, and the corresponding analysis bank is used to generate the syndrome of the code. First, we study the encoder of half-rate TDWCs and show that these linear codes are lattice cyclic. It is proven that any 2-D lattice-cyclic code can also be generated by a 2-D wavelet transform. Second, we introduce a methodology to design TDWCs over binary erasure channels. These codes have simple and efficient maximum likelihood (ML) decoding for burst erasures. We show that half-rate TDWCs of dimensions $N_1 \times N_2$ can recover burst erasures of size up to $N_1 \times N_2/2$ and $N_1/2 \times N_2$ using the proposed simple decoding technique. Finally, we present examples of TDWCs that satisfy the Reiger bound with equality, i.e., they are capable of correcting any burst of size $\frac{N_1 N_2}{2}$.

2.1.1 Related Work

There are many data transmission and storage systems with 2-D data structures that suffer from 2-D bursts of error and erasures. 2-D codes can be used to combat such errors and erasures. Elspas [16] showed that the product of two cyclic codes is capable of correcting 2-D burst errors. Later, Imai [26] studied 2-D fire codes, which are cyclic and are capable of correcting bursts of size $b_1 \times b_2$ (for some b_1 and b_2). To evaluate the performance of a 2-D (n, k) code that can correct burst errors of the area up to b, the parameter burst-correcting efficiency is used. This parameter is denoted by z and defined as $z = \frac{2b}{n-k}$. It is clear that for erasure correcting, z is modified as $z = \frac{b}{n-k}$. According to the Reiger bound [56], $z \leq 1$. An important goal in the study of 2-D codes is introducing families of codes whose efficiency approaches one. One problem of the product codes is the unnecessary redundancies. However, it is worth noting that the capability of these codes is not limited to correcting burst errors. Although 2-D fire codes have less redundancy compared to the product codes, their burst-correcting efficiency is still less than 0.4. In [2, 7, 10], different 2-D codes were suggested to correct a single rectangular cluster of errors of size $b_1 \times b_2$. The 2-D codes introduced in [10] have the highest efficiency that is equal to 2/3.

Array codes were introduced in [8]. These codes have simple structures and are able to correct multiple bursts along the diagonal with efficiency 1/2. A 2-D code was proposed in [6] that has efficiency 4/5 and can correct one diagonal error. These array codes have a simple decoding algorithm, but their efficiency is less than one. In [9], a new class of array codes was presented that is maximum distance separable (MDS). The dimension of these codes is $N_1 \times N_2$, where $N_1 + 1$ must be a prime number and N_2 must be less than N_1. The decoding complexity of these codes is less than that of regular Reed-Solomon (RS) codes. In [29], another class of 2-D codes was proposed whose lengths are longer than the codes in [9]. These constraints on the code length are quite restrictive. To overcome this problem, a family of binary array codes was introduced in [28] that is MDS and has lengths longer than the codes suggested in [29]. These codes also have a decoding algorithm that has lower complexity compared to RS codes. These codes are MDS over $GF(2^{N_1})$, where N_1 denotes the length of each column. In other words, using r redundant columns, r column erasures can be recovered. However, these codes are not necessarily capable of recovering erasure bursts of dimension $b_1 \times b_2$, even if $b_1 b_2 < rN_1$.

2.1.2 Contribution

We introduce two-dimensional wavelet codes (TDWCs). The main goal of our approach is to construct a code that has low decoding complexity and has the maximum erasure-recovering property for erasure bursts of dimension $b_1 \times b_2$. The first application of finite-field wavelet transforms to error-control coding was introduced in [18], [19]. Here, we extend wavelet coding to the 2-D case [64]. We employ two-channel maximally decimated orthogonal filter banks. The synthesis bank constructs the encoder of the TDWC and the corresponding analysis bank constructs the syndrome generator. We design half-rate TDWCs of dimension

$N_1 \times N_2$ that are capable of recovering erasure bursts of dimension $N_1/2 \times N_2$ and $N_1 \times N_2/2$. Therefore, TDWC has efficiency one for these burst patterns. The only constraint on the parameters N_1 and N_2 of our codes is that their greatest common devisor must be two, $GCD(N_1, N_2) = 2$. These codes are lattice cyclic. Thus, their encoding and decoding algorithms are significantly simplified. We also develop criteria to construct 2-D self-dual codes. By extending these codes, we generate weakly self-dual codes.

2.1.3 Notation

1. *Notation:* Boldfaced lowercase letters are used for vectors. Matrices and tensors are denoted by boldfaced uppercase letters. A two-dimensional (2-D) array x is represented by $x(\mathbf{n})$ in the time domain, where $\mathbf{n} = (n_1,\, n_2)$ is a vector with integer elements that denotes the index for the array x. As an example, the 2-D array $x = \begin{bmatrix} 1\;0\;4 \\ 2\;3\;6 \end{bmatrix}$ is represented by $x(n_1,\, n_2)$, where $x(1,\, 2) = 6$. In the polynomial representation, we use the vector $\mathbf{D} = (D_1,\, D_2)$ as the intermediate variables. For example, the polynomial representation of the above array is given by $X(\mathbf{D}) = 1 + 4D_2^2 + 2D_1 + 3D_1D_2 + 6D_1D_2^2$. The ring of integers is denoted by $\mathbb{Z}$. The symbol $\mathbb{F}$ is used for an arbitrary Galois field of characteristic two. If $\mathbf{m} = \left[m_1,\, m_2\right]^T \in \mathbb{Z}^2$, then we use the shorthand notation $\mathbf{D}^{\mathbf{m}}$ for $D_1^{m_1}D_2^{m_2}$. If $\mathbf{M} = \left[\mathbf{m}_1,\, \mathbf{m}_2\right]$, where $\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}^2$ are the columns of the matrix $\mathbf{M}$, then $\mathbf{D}^{\mathbf{M}} \triangleq (\mathbf{D}^{\mathbf{m}_1},\, \mathbf{D}^{\mathbf{m}_2})$. For the sampling matrix $\mathbf{M}$, the LAT($\mathbf{M}$) consists of all integral linear combinations of the column vectors of $\mathbf{M}$.

2. *Convolution Operators:* The symbol $\circledast$ is used for 2-D circular convolution. If $x(\mathbf{n})$ and $h(\mathbf{n})$ are 2-D arrays whose support is limited to the rectangle $0 \leq n_1 < N_1, 0 \leq n_2 < N_2$, then

$$\begin{aligned} y(\mathbf{n}) &= (h \circledast x)(\mathbf{n}) \\ &\triangleq \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} h(i,j)x(((n_1-i))_{N_1}, ((n_2-j))_{N_2}), \end{aligned} \tag{2}$$

where $((\cdot))_N$ denotes reduction modulo N. The symbol $\circledast_{\downarrow\mathbf{M}}$ denotes circular convolution followed by an $\mathbf{M}$-fold decimator, i.e.,

$$\begin{aligned} y(\mathbf{n}) &= (h \circledast_{\downarrow\mathbf{M}} x)(\mathbf{n}) \\ &\triangleq (h \circledast x)(\mathbf{M}\mathbf{n}). \end{aligned} \tag{3}$$

Similarly, the symbol ${}_{\uparrow\mathbf{M}}\circledast$ denotes the circular convolution preceded by an $\mathbf{M}$-fold expander, i.e.,

$$\begin{aligned} y(\mathbf{n}) &= (h \,{}_{\uparrow\mathbf{M}}\circledast\, x)(\mathbf{n}) \\ &\triangleq \begin{cases} (h \circledast x)(\mathbf{n}'), & \mathbf{n} = \mathbf{M}\mathbf{n}' \in LAT(\mathbf{M}), \quad \mathbf{n}' \in \mathbb{Z}^2 \\ 0, & \mathbf{n} \notin LAT(\mathbf{M}). \end{cases} \end{aligned} \tag{4}$$

3. *Unitary:* A matrix $\mathbf{A}$ for positive integers n and k is unitary if $\mathbf{A}^T\mathbf{A} = \mathbf{I}$.

4. *Paraunitary:* Let us define the involution tilde as follows:

$$\tilde{\mathbf{E}}(D_1) \triangleq \mathbf{E}^T(D_1^{-1}), \tag{5a}$$

$$\tilde{\mathbf{E}}(D_1, D_2) \triangleq \mathbf{E}^T(D_1^{-1}, D_2^{-1}). \tag{5b}$$

A bivariate polynomial matrix $\mathbf{E}(D_1, D_2)$ is called "paraunitary (PU) in $\mathbb{F}$" if [72]

$$\tilde{\mathbf{E}}(D_1, D_2)\mathbf{E}(D_1, D_2) \equiv \mathbf{I}, \tag{6}$$

and is called "PU in the ring $\mathbb{F}[x]$" if

$$\tilde{\mathbf{E}}(D_1, D_2)\mathbf{E}(D_1, D_2) \equiv \tilde{\alpha}(D_1)\alpha(D_1)\mathbf{I}, \qquad \alpha(D_1) \in \mathbb{F}[D_1]. \tag{7}$$

The term "paraunitary" is used in this correspondence interchangeably with "paraunitary in the field $\mathbb{F}$."

5. *FIR:* A finite impulse response filter is represented by a finite-degree laurant polynomial with only positive exponents of $\mathbf{D}$.

2.2 Brief Review of Filter Banks

In this section, we give a quick review of two-channel two-variable filter banks that are the realization of two-variable wavelets. A thorough introduction can be found in [27,31,71,72]. As shown in Fig. 3a, the two-channel two-variable filter bank consists of a bank of two analysis filters, $h_0(\mathbf{n})$ and $h_1(\mathbf{n})$, each followed by a 2×2 sampling matrix $\mathbf{M}$ such that $|\det \mathbf{M}| = 2$. These filters split the 2-D input signal $x(\mathbf{n})$ into two sub-band signals $y_1(\mathbf{n})$ and $y_2(\mathbf{n})$. At the receiver end, these signals are up-sampled by the same matrix $\mathbf{M}$ and passed through a bank of synthesis filters, $g_0(\mathbf{n})$ and $g_1(\mathbf{n})$. This process results in the signal $\hat{x}(\mathbf{n})$ that is aimed to be the best approximation for the original signal $x(\mathbf{n})$. If the filter bank is designed properly, $\hat{x}(\mathbf{n})$ becomes a replica of $x(\mathbf{n})$ up to a constant amplitude scale and time shift.

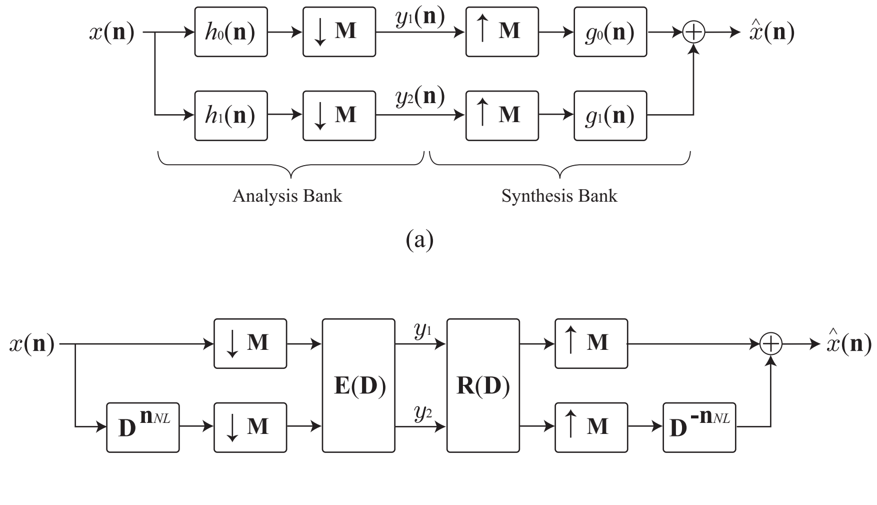

Figure 3: (a) Two-channel two-variable maximally decimated filter bank; (b) Polyphase representation of the system in (a).

The task of 2-D sampling by the matrix $\mathbf{M}$ is clarified by Fig. 4, where $\mathbf{M} = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}$. An integral linear combination of the columns of the matrix $\mathbf{M}$ generates LAT($\mathbf{M}$). In other words, a point with coordinates (n_1, n_2) belongs to LAT($\mathbf{M}$) if there exists $t_1, t_2 \in \mathbb{Z}$, such that $t_1\mathbf{m}_1 + t_2\mathbf{m}_2 = (n_1, n_2)$. The black points in Fig. 4 construct LAT($\mathbf{M}$). These points are called lattice points and the rest of the points are referred to as non-lattice points. Any lattice point can be reached from any other lattice points by vector $\mathbf{w} \in$ LAT($\mathbf{M}$), while the non-lattice points are not reachable from lattice points.

Down-sampling a signal $x(\mathbf{n})$ by the matrix $\mathbf{M}$ is defined as $x_d(\mathbf{n}) = x(\mathbf{Mn})$, which can be explained as follows. The non-lattice points in $x(\mathbf{n})$ are discarded and the lattice points with index $\mathbf{n}$ has the value of $x(\mathbf{Mn})$. The up-sampling of the signal $x(\mathbf{n})$ by the matrix $\mathbf{M}$ is defined as follows:

$$x_u(\mathbf{n}) = \begin{cases} x(\mathbf{k}), \text{ if } \mathbf{n} = \mathbf{Mk} \\ 0, \qquad \text{else} \end{cases}$$

Thus, by upsampling, the non-lattice points get the value of zero, while the lattice points with index $\mathbf{n}$ get the value of $x(\mathbf{M}^{-1}\mathbf{n})$.

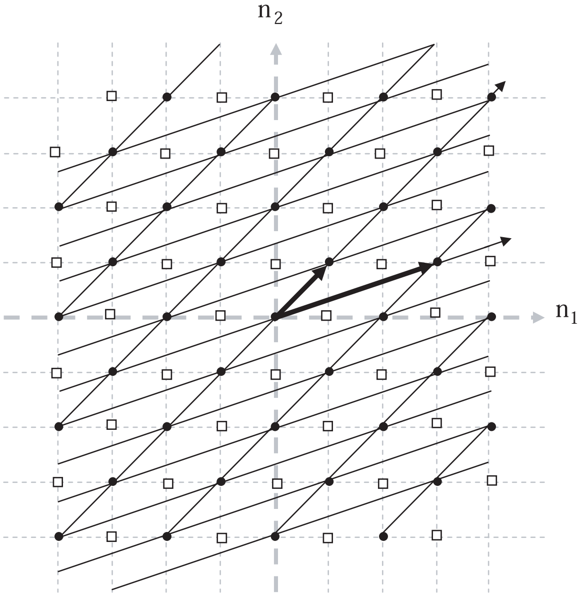

Figure 4: An example of LAT($\mathbf{M}$) and 2-D sampling. The black points construct the LAT($\mathbf{M}$) and are called lattice points and the white points that are not located on the LAT($\mathbf{M}$) are called non-lattice points.

Polyphase representations of the analysis and synthesis filters result in the equivalent system of Fig. 3b. The polyphase representation provides a technique to design the filter banks. In this representation, the 2 × 2 matrices $\mathbf{E}(\mathbf{D})$ and $\mathbf{R}(\mathbf{D})$ are polyphase matrices of the analysis and synthesis banks, respectively. The relation between polyphase matrices and impulse responses of the filters in analysis and synthesis banks is as follows:

$$H_i(\mathbf{D}) = E_{0i}(\mathbf{D}^{\mathbf{M}}) + \mathbf{D}^{\mathbf{n}_{NL}} E_{1i}(\mathbf{D}^{\mathbf{M}}), \tag{8}$$

$$G_i(\mathbf{D}) = R_{0i}(\mathbf{D}^{\mathbf{M}}) + \mathbf{D}^{-\mathbf{n}_{NL}} R_{1i}(\mathbf{D}^{\mathbf{M}}),$$

where $\mathbf{n}_{NL}$ is the index of an arbitrary non-lattice point. Here, $E_{ij}(\mathbf{D})$ and $R_{ij}(\mathbf{D})$ are the ij^{th} entries of the matrices $E(\mathbf{D})$ and $R(\mathbf{D})$, respectively. The perfect reconstruction (PR) property implies that

$$\mathbf{R}(\mathbf{D})\mathbf{E}(\mathbf{D}) = \mathbf{I}. \tag{9}$$

If the analysis bank consists of FIR filters, $\mathbf{E}(\mathbf{D})$ is a square matrix whose entries are bivariate polynomials. In this situation, $\mathbf{R}(\mathbf{D})$ the inverse system of $\mathbf{E}(\mathbf{D})$ is IIR. To avoid IIR synthesis filters, the class of FIR paraunitary (PU) matrices that have FIR inverses is used. In light of (6), with the choice of $\mathbf{R}(\mathbf{D}) = \tilde{\mathbf{E}}(\mathbf{D})$, the synthesis bank becomes FIR as well.

One way of constructing PU matrices is to represent them as a product of some elementary building blocks. In the following, we present two-variable elementary building blocks over the ring $\mathbb{F}[D_1^{\pm 1}]$. These building blocks are used to generate two-channel two-variable wavelets over finite fields.

The two-variable degree-one PU building block in $\mathbb{F}[D_1^{\pm 1}]$ is defined as [15]

$$\mathbf{U}\left(D_2; \mathbf{v}(D_1)\right) \triangleq \alpha(D_1)\mathbf{I} + \mathbf{v}(D_1)\tilde{\mathbf{v}}(D_1) + \mathbf{v}(D_1)\tilde{\mathbf{v}}(D_1)D_2^{-1}, \tag{10}$$

where $\mathbf{v}(D_1) \in (\mathbb{F}[D_1^{\pm 1}])^2$ is a none self-orthogonal polynomial vector and $\alpha(x) \triangleq \tilde{\mathbf{v}}(D_1)\mathbf{v}(D_1) \in \mathbb{F}[D_1^{\pm 1}] \setminus \{0\}$. A direct computation shows that $\mathbf{U}\left(D_2; \mathbf{v}(D_1)\right)$ is PU in $\mathbb{F}[D_1^{\pm 1}]$, i.e.,

$$\tilde{\mathbf{U}}\left(D_2; \mathbf{v}(D_1)\right)\mathbf{U}\left(D_2; \mathbf{v}(D_1)\right) \equiv \alpha^2(D_1)\mathbf{I}. \tag{11}$$

The two-variable degree-2τ PU building block in $\mathbb{F}[D_1^{\pm 1}]$ is defines as [15]

$$\mathbf{V}_\tau\left(D_2; \mathbf{u}(D_1), \mathbf{v}(D_1), \zeta(D_1)\right) \triangleq \begin{bmatrix} D_1^{m-r}p(D_1)q(D_1^{-1}) & 0 \\ 0 & p(D_1^{-1})q(D_1) \end{bmatrix} D_2^{-\tau}$$
$$+ \zeta(D_1)\mathbf{u}(D_1)\tilde{\mathbf{v}}(D_1) + \zeta(D_1)\mathbf{u}(D_1)\tilde{\mathbf{v}}(D_1)\, D_2^{-2\tau}. \tag{12}$$

Here, τ is a positive integer, $\zeta(D_1) \in \mathbb{F}[D_1^{\pm 1}]$ is a univariate polynomial such that $\tilde{\zeta}(D_1) = \zeta(D_1)$, and

$$\mathbf{u}(D_1) = \begin{bmatrix} D_1^m p(D_1) \\ p(D_1^{-1}) \end{bmatrix}, \qquad \mathbf{v}(D_1) = \begin{bmatrix} D_1^r q(D_1) \\ q(D_1^{-1}) \end{bmatrix}. \tag{13}$$

In (13), $m, r \in \mathbb{Z}$ and $p(D_1), q(D_1) \in \mathbb{F}[D_1^{\pm 1}]$. Note that $\mathbf{u}(D_1)$ and $\mathbf{v}(D_1)$ in (13) are both self-orthogonal polynomial vectors. Therefore, $\mathbf{V}_\tau\left(D_2;\, \mathbf{u}(D_1), \mathbf{v}(D_1), \zeta(D_1)\right)$ is not factorable into the product of degree-one building blocks. It is straightforward to show that $\mathbf{V}_\tau\left(D_2;\, \mathbf{u}(D_1), \mathbf{v}(D_1), \zeta(D_1)\right)$ is PU in $\mathbb{F}[D_1^{\pm 1}]$. In other words, we have

$$\tilde{\mathbf{V}}_\tau\left(D_2;\, \mathbf{u}(D_1), \mathbf{v}(D_1), \zeta(D_1)\right) \mathbf{V}_\tau\left(D_2;\, \mathbf{u}(D_1), \mathbf{v}(D_1), \zeta(D_1)\right) \equiv p(D_1)p(D_1^{-1})q(D_1)q(D_1^{-1})\mathbf{I}. \tag{14}$$

Thus, to construct a two-variable filter bank, first we use the above elementary building blocks to generate a paraunitary polyphase matrix in two variables over a finite field. Then, we use (8) to determine the filters in the synthesis and analysis banks and generate the corresponding two-channel two-variable maximally decimated orthogonal filter bank. In the following sections, we study the application of these filter banks in constructing and decoding 2-D error-control codes.

2.3 2-D Codes Using Finite-Field Wavelets

A 2-D binary code of dimension $N_1 \times N_2$ is a set of $N_1 \times N_2$ binary arrays whose elements are called codewords. A 2-D codeword c represented by $c(\mathbf{n})$ is an $N_1 \times N_2$ array such as

$$c(\mathbf{n}) = [c(i,j)] = \begin{bmatrix} c(0,0) & c(0,1) & \cdots & c(0,N_2-1) \\ c(1,0) & c(1,1) & \cdots & c(1,N_2-1) \\ \vdots & \vdots & \vdots & \vdots \\ c(N_1-1,0) & c(N_1-1,1) & \cdots & c(N_1-1,N_2-1) \end{bmatrix}, \tag{15}$$

where $\mathbf{n} = (n_1,\, n_2)$ is a vector with integer elements that denotes the index for the codeword c. The set of all $N_1 \times N_2$ arrays over the binary field forms an (N_1N_2)-dimensional vector space over $\mathbb{F}$. Any subvector space $\mathcal{C}$ is called a 2-D linear code. We assume a 2-D message array $m(\mathbf{n})$ with area k is fed into the encoder and the 2-D codeword array $c(\mathbf{n})$ is generated. All 2-D linear codes can be represented by generator tensors or parity-check tensors. The parity-check tensor $\mathbf{H}$ consists of r layers of $N_1 \times N_2$ arrays $h_s(\mathbf{n})$, $1 \leq s \leq r$, where $r = N_1N_2 - k$. Codeword array $c(\mathbf{n})$ of area $N_1 \times N_2$ satisfies the equation

$$\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} c(i,j)h_s(i,j) = 0, \qquad s = 1, \ldots, r \tag{16}$$

We propose to use finite-field wavelets to construct 2-D codes [64]. In the proposed two-dimensional wavelet code (TDWC), we use the synthesis filter bank of the two-channel two-variable filter bank in Fig. 3a. First, the 2-D message array $m(\mathbf{n})$ passes through an $\mathbf{M}$-fold expander whose determinant is two and generates the expanded message array $m_{\uparrow\mathbf{M}}(\mathbf{n})$. As shown later, the area of $m_{\uparrow\mathbf{M}}(\mathbf{n})$ is twice as much as that of array $m(\mathbf{n})$. The fact that signal $m_{\uparrow\mathbf{M}}(\mathbf{n})$ is expanded by a factor of two results in a code rate of 1/2. Then, array $m_{\uparrow\mathbf{M}}(\mathbf{n})$ passes through the bank of synthesis filters $g_0(\mathbf{n})$ and $g_1(\mathbf{n})$, whose region of support $N_1 \times N_2$ is identical to the size of the codeword array. The summation of the outputs of the synthesis filters generates the codeword array.

The 2-D upsampling is explained by pursuing a specific example. Suppose the area of the 2-D input signal is 12. Label the elements in the 2-D signal by numbers one through 12 and rearrange them to form the array $m(\mathbf{n})$ in Fig. 5. As shown in Fig. 5, applying the upsampling by the matrix $\mathbf{M}$ to the signal $m(\mathbf{n})$, the signal $m_{\uparrow\mathbf{M}}(\mathbf{n})$ of dimension 4×6 is generated. Half of the elements of the output of the expander are zero. The inserted zeros are non-lattice points, while the rest of the points are lattice points.

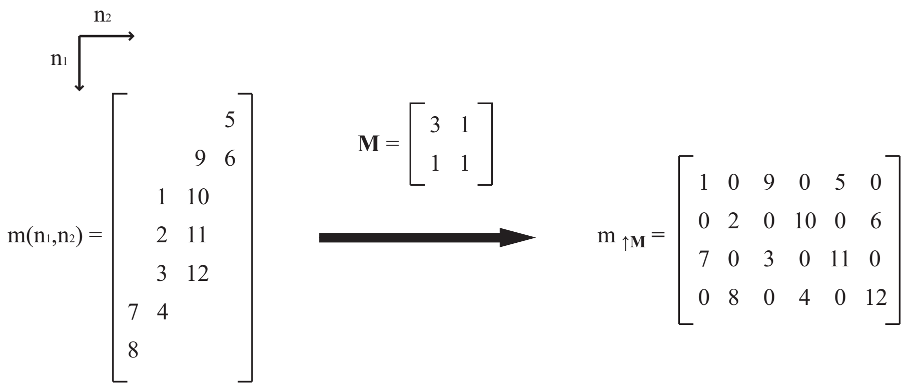

Figure 5: An example for illustration of 2-D upsampling.

We note that the TDWC is lattice cyclic. Code $\mathcal{C}$ is lattice cyclic, if $\Big(\mathbf{D}^{\mathbf{w}}C(\mathbf{D}) \mod (D_1^{N_1}-1, D_2^{N_2}-1)\Big) \in \mathcal{C}$ for all $C(\mathbf{D}) \in \mathcal{C}$ and all $\mathbf{w} \in \mathrm{LAT}(\mathbf{M})$. In other words, the cyclic shift of any codeword array in the direction of an arbitrary vector $\mathbf{w} \in \mathrm{LAT}(\mathbf{M})$ is itself a codeword. Because of the lattice-cyclic property of the TDWC, the generator tensor consists of layers that are cyclic shift of each other. The same is true about the parity-check tensor. This property simplifies the encoding and the syndrome computing. These operations are simplified to calculate 2-D circular convolutions or equivalently to polynomial operations in the quotient ring $\mathbb{F}[\mathbf{D}]/\left(D_1^{N_1}-1, D_2^{N_2}-1\right)$. As an example, (16) can be rewritten as $(c \circledast_{\downarrow \mathbf{M}} h_s)(\mathbf{n}) = 0$. Note that $\circledast_{\downarrow \mathbf{M}}$ is defined as in (3) and the array $h_s(\mathbf{n})$ is one layer of the parity-check tensor $\mathbf{H}$ and the rest of the layers are cyclic shifts of h_s in directions of the points in LAT($\mathbf{M}$).

To find codes with the desired properties such as good minimum distance, self duality, or burst erasure correctability, we add two prefilters to the two channels of the synthesis bank. As shown later, the only constraint on the prefilters is that at least one of them needs to be circularly invertible. Figure 6 shows the encoder of the TDWC with two prefilters $\lambda_0(\mathbf{n})$ and $\lambda_1(\mathbf{n})$. The properties of the proposed TDWC are summarized in the following lemma.

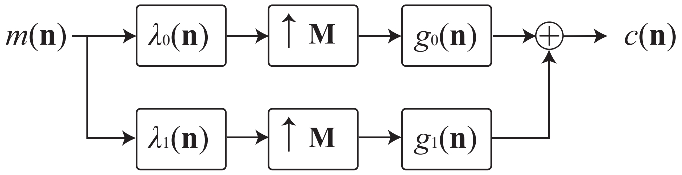

Figure 6: The filter bank structure of the half-rate encoder of the TDWC.

Lemma 1. *Consider the encoder in Fig. 6. We assume that at least one of the prefilters is invertible. The generated TDWC has the following properties:*

i. Let $c(\mathbf{n})$ be a 2-D codeword with dimension $N_1 \times N_2$. If N_1 and N_2 are both even integers, then the generated code is lattice cyclic.

ii. There is a one-to-one mapping between the message and the codeword.

iii. The code generated by the system in Fig. 7 is equivalent to the one generated by the system in Fig. 6.

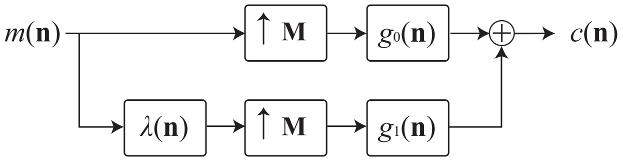

Figure 7: The equivalent filter bank structure of the half-rate encoder of the TDWC.

Proof. As stated earlier, a binary 2-D code is lattice cyclic if any cyclic shift of the codeword in the direction of points in $\mathrm{LAT}(\mathbf{M})$ is still a codeword. Up-sampling the codeword with the matrix $\mathbf{M}$ gives an array of size $N_1 \times N_2$ with message bits on the lattice points and zeros on non-lattice points. Since N_1 and N_2 are both even, the zeros in the non-lattice points are preserved by shifting in the direction of the points in $\mathrm{LAT}(\mathbf{M})$. Therefore, any such cyclic shift generates another valid message sequence. The corresponding codeword is a cyclic shift of the original codeword in the direction of the points in $\mathrm{LAT}(\mathbf{M})$. Hence, the TDWC is lattice cyclic.

To show property ii), notice that the filter bank is designed using a PU matrix that is generated by multiplying the PU elementary building blocks introduced in Section 2.2. Therefore, applying $c(\mathbf{n})$ to the analysis bank gives back the message if one of the prefilters is invertible. Since we assume that at least one of the prefilters is invertible, the generated TDWC has this property.

Finally, to show property iii, assume the prefilter λ_0 in Fig. 6 is invertible. Choose $\lambda(\mathbf{n}) = (\lambda_0^{-1} \circledast \lambda_1)(\mathbf{n})$ in Fig. 7. It can be verified that the codeword generated for the message $m(\mathbf{n})$ in Fig. 6 is the same as the codeword generated for the message $(m \circledast \lambda_0)(\mathbf{n})$ in Fig. 7. Hence, the encoders in these two figures are equivalent. □

The following lemma shows that the encoder in Fig. 6 is able to generate an arbitrary lattice-cyclic 2-D code.

Lemma 2. *Let $\mathcal{C}_{\mathbf{G}}$ be an arbitrary lattice-cyclic 2-D code with generator tensor $\mathbf{G}$ and dimension $N_1 \times N_2$, where N_1 and N_2 are both even integers. Then, the encoder of Fig. 6 generates a code equivalent to $\mathcal{C}_{\mathbf{G}}$.*

Proof. Consider the TDWC encoder in Fig. 6. We need to show that there exist prefilters $\lambda_0(\mathbf{n})$ and $\lambda_1(\mathbf{n})$ such that the encoder in Fig. 6 generates a code equivalent to the code $\mathcal{C}_{\mathbf{G}}$. As stated in Section 2.3, the generator tensor $\mathbf{G}$ consists of 2-D array $g(\mathbf{n})$ and its cyclic shifts. We write $g(\mathbf{n})$ as the summation of two arrays $g(\mathbf{n}) = i_g(\mathbf{n}) + u_g(\mathbf{n})$, where $i_g(\mathbf{n})$ is invertible. We need to find prefilters $\lambda_0(\mathbf{n})$ and $\lambda_1(\mathbf{n})$ in Fig. 6 such that

$$\begin{aligned} i_g(\mathbf{n}) &= \left(\hat{\lambda}_0 \circledast g_0\right)(\mathbf{n}) \quad \text{and} \\ u_g(\mathbf{n}) &= \left(\hat{\lambda}_1 \circledast g_1\right)(\mathbf{n}), \end{aligned} \tag{17}$$

where $\hat{\lambda}_i(\mathbf{n})$ is the $\mathbf{M}$-fold expansion of $\lambda_i(\mathbf{n})$ for $i = 1, 2$. This choice of prefilters guarantees the invertibility of prefilter $\hat{\lambda}_0(\mathbf{n})$. Now, we need to find the solutions for (17). We show that the following prefilters are valid solutions for the equations in (17).

$$\begin{aligned} \lambda_0(\mathbf{n}) &= \left(g \circledast_{\downarrow \mathbf{M}} h_0\right)(\mathbf{n}) \quad \text{and} \\ \lambda_1(\mathbf{n}) &= \left(g \circledast_{\downarrow \mathbf{M}} h_1\right)(\mathbf{n}), \end{aligned} \tag{18}$$

where $h_0(\mathbf{n})$ and $h_1(\mathbf{n})$ are the analysis filters of the two-channel filter bank used in the encoder of Fig. 6.

$$\begin{aligned} \left(g \circledast_{\downarrow \mathbf{M}} h_0\right)(\mathbf{n}) &= \left((\hat{\lambda}_0 \circledast g_0 + \hat{\lambda}_1 \circledast g_1) \circledast_{\downarrow \mathbf{M}} h_0\right)(\mathbf{n}) \\ &= \left(\hat{\lambda}_0 \circledast_{\downarrow \mathbf{M}} (g_0 \circledast h_0) + \hat{\lambda}_1 \circledast_{\downarrow \mathbf{M}} (g_1 \circledast h_0)\right)(\mathbf{n}) \\ &= \lambda_0(\mathbf{n}), \end{aligned} \tag{19}$$

where the last equality is due to the perfect reconstruction property of the filter bank. In summary, we show that any 2-D lattice-cyclic code $\mathcal{C}_{\mathbf{G}}$ can also be generated by the encoder of TDWCs. □

The TDWC offers an efficient implementation by using a simplified structure. It can be easily shown that the system in Fig. 8 is equivalent to the encoder in Fig. 7, where $G_{eq}(\mathbf{D}) = G_0(\mathbf{D}) + G_1(\mathbf{D})\Lambda(\mathbf{D}^{\mathbf{M}})$. Equivalently, $g_{eq}(\mathbf{n}) = g_0(\mathbf{n}) + (\hat{\lambda} \circledast g_1)(\mathbf{n})$, where $\hat{\lambda}(\mathbf{n})$ is the $\mathbf{M}$-fold expansion of λ. Hence, the codeword $c(\mathbf{n})$ can be computed as follows:

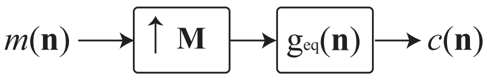

Figure 8: Simplified structure of the wavelet encoder.

$$c(\mathbf{n}) = (m \,{}_{\uparrow \mathbf{M}}\!\circledast g_{eq})(\mathbf{n}), \tag{20}$$

where ${}_{\uparrow \mathbf{M}}\circledast$ is introduced in (4). Example 1 shows how the TDWC is constructed.

Example 1. *We construct a TDWC of dimension* 4×6 *using the encoder of Fig. 7. The filters* $g_0(\mathbf{n}), g_1(\mathbf{n}), \hat{\lambda}(\mathbf{n})$*, and the up-sampling matrix* $\mathbf{M}$ *are chosen as follows:*

$$g_0(\mathbf{n}) = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&1&1 \end{bmatrix} \qquad g_1(\mathbf{n}) = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&1 \end{bmatrix}$$

$$\hat{\lambda}(\mathbf{n}) = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ 1&0&1&0&1&0 \\ 0&1&0&1&0&1 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 3&1 \\ 1&1 \end{bmatrix}.$$

An important property of error-control codes is minimum distance, which is defined as the minimum Hamming distance between all pairs of codewords. It can be verified that the minimum distance of this code is eight.

In the next section, we study the conditions on the synthesis filters in Fig. 3a under which the TDWC becomes self dual or weakly self dual.

2.4 Self-Dual Two-Dimensional Wavelet Code

Now, we give construction for 2-D self-dual codes. Let $x = [x_{ij}]$ and $y = [y_{ij}]$ be 2-D arrays of size $N_1 \times N_2$. Their inner product is defined by

$$\begin{aligned} < x, y > &\triangleq \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} x_{ij} y_{ij} \\ &= \mathrm{Tr}(xy^T). \end{aligned} \tag{21}$$

The dual of a code $\mathcal{C}$ is a subset of all 2-D arrays of size $N_1 \times N_2$ that is defined as

$$\mathcal{C}^{\perp} = \{x :< x, y >= 0, \quad \forall y \in \mathcal{C}\}.$$

If $\mathcal{C} = \mathcal{C}^{\perp}$, then $\mathcal{C}^{\perp}$ is said to be self dual. If $\mathcal{C} \subseteq \mathcal{C}^{\perp}$, then $\mathcal{C}^{\perp}$ is said to be weakly self dual. The following lemma states the necessary and sufficient condition on $g_{eq}(\mathbf{n})$ (in Fig. 8) by which the generated TDWC is self dual.

Lemma 3. *The TDWC is self dual if and only if the inner product of $g_{eq}(\mathbf{n})$ and any cyclic shift of $g_{eq}(\mathbf{n})$ in the direction of any lattice point is zero.*

Proof. Lemma can be proved using the property that, in any TDWC, we have $< x \,_{\uparrow\mathbf{M}}\circledast g_{eq}, y \,_{\uparrow\mathbf{M}}\circledast g_{eq} >= 0$ for all possible distinct message sequences x and y. The complete proof of Lemma 3 is given in Appendix A. □

The code designed in Example 1 is self dual, since $g_{eq}(\mathbf{n})$ satisfies the condition of Lemma 3. We also extend the self-dual codes generated by Fig. 7 to construct new codes. The following lemma gives the extension.

Lemma 4. *Let $\mathcal{C}_1$ be an $N_1 \times N_2$ self-dual TDWC. Let $\mathcal{C}_2$ be the code constructed from $\mathcal{C}_1$ by adding a new column to all codeword arrays in $\mathcal{C}_1$ such that the parity of each row becomes zero. If all rows of the codewords in $\mathcal{C}_1$ have the same sum, then the minimum distance of $\mathcal{C}_2$ is at least the same as that of $\mathcal{C}_1$. Moreover, $\mathcal{C}_2$ is weakly self dual.*

Proof. Since all rows of the original codewords have the same sum, the entries of the added column are all either one or zero. Therefore, the minimum distance of the new code remains either unchanged or increased by N_1 with respect to the original code. To show $\mathcal{C}_2$ is weakly self dual, we first note that for two distinct extended codewords x and y, we have

$$< x, y > = \underbrace{\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} x(i,j)y(i,j)}_{P_1} + \underbrace{\sum_{i=0}^{N_1-1} x(i,N_2)y(i,N_2)}_{P_2}. \tag{22}$$

The expression P_1 is equal to zero, since the original code is self dual. The expression P_2 is either equal to zero or N_1. Since N_1 is an even number, it is equivalent to zero in $\mathbb{F}$. Therefore, the new code is weakly self dual. □

Specifically, we have the following lemma.

Lemma 5. *If the rows of $g_{eq}(\mathbf{n})$ have the same weight, then the extended self-dual code is weakly self dual.*

Proof. According to Lemma 4, we only need to show that the sum of all rows of the original codewords generated by g_{eq} are the same. We note that

$$\begin{aligned}\sum_{n_2} c(n_1, n_2) &= \sum_{n_2=0}^{N_2-1} \sum_{(i,j)\in LAT(\mathbf{M})} x(i,j) g_{eq}(((n_1 - i))_{N_1}, ((n_2 - j))_{N_2}) \\ &= \sum_{(i,j)\in LAT(\mathbf{M})} x(i,j) \sum_{n_2} g_{eq}(((n_1 - i))_{N_1}, ((n_2 - j))_{N_2}). \qquad (23)\end{aligned}$$

Since the row weights of g_{eq} are the same, we can conclude that the row-weights of the codewords are also the same by (23). □

Example 2. *The filter $g_{eq}(\mathbf{n})$ of the code in Example 1 is*

$$g_{eq}(\mathbf{n}) = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \qquad (24)$$

where all rows have weight three. Therefore, the resulting extended code is weakly self dual by Lemma 4 and has minimum distance of eight.

2.5 The Syndrome Generator of Two-Dimensional Wavelet Code

The wavelet construction of the 2-D code has the benefit of simple syndrome generation. It can be shown that the syndrome generator consists of the analysis bank of the two-channel filter bank used in the encoder. Figure 9a shows the syndrome generator for the encoder in Fig. 6. The output $s(\mathbf{n})$ of the system in Fig. 9a is obtained by

$$s(\mathbf{n}) = \left(c \circledast_{\downarrow\mathbf{M}} \left(h_1 + \hat{\lambda} \circledast h_0\right)\right)(\mathbf{n}), \qquad (25)$$

where $c(\mathbf{n})$ is the received word. To reduce the computation further, we simplify the syndrome generator in Fig. 9a using the mentioned filter properties. The simplified syndrome generator of the TDWC is shown in Fig. 9b, where $h_{eq}(\mathbf{n}) = \left(h_1 + \hat{\lambda} \circledast h_0\right)(\mathbf{n})$.

Since the filter bank is designed to have the PR property, the expression in the right side of (25) is zero for a valid codeword. If the received array contains an error, then $s(\mathbf{n})$ gives the syndrome corresponding to that error. Hence, the following lemma can be stated.

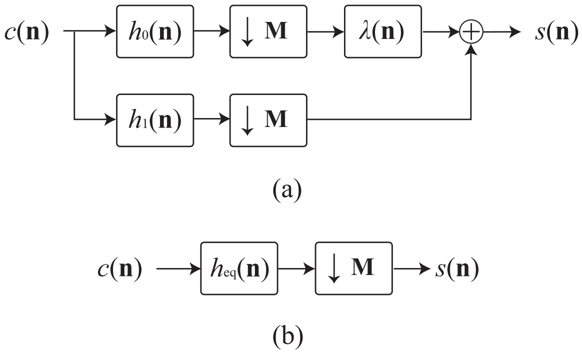

Figure 9: (a) The syndrome generator of the TDWC, (b) Simplified structure of the syndrome generator of the TDWC.

Lemma 6. *Assume there exists a parity-check tensor* $\mathbf{H}$ *for an arbitrary 2-D lattice-cyclic code* $\mathcal{C}_{\mathbf{H}}$ *such that* $(c \circledast_{\downarrow M} h)(\mathbf{n}) = 0$, *for all* $c \in \mathcal{C}_{\mathbf{H}}$, *where* $h(\mathbf{n})$ *is one layer of the parity-check tensor* $\mathbf{H}$. *Then, there exists a TDWC equivalent to* $\mathcal{C}_{\mathbf{H}}$ *with analysis filters* $h_0(\mathbf{n})$ *and* $h_1(\mathbf{n})$ *and prefilter* λ *such that* $\left(h_1 + \hat{\lambda} \circledast h_0\right)(\mathbf{n}) = h(\mathbf{n})$.

In the following section, we study the erasure decoding and burst erasure correcting capability of the TDWC.

2.6 TDWC for Burst Erasure

In this section, we consider 2-D codes for burst erasure recovering. The area of a 2-D burst $\mathbf{E}$ is defined by the region of the support of the smallest rectangle that contains the nonzero elements of $\mathbf{E}$. A code $\mathcal{C}$ can correct a burst of erasures of size $b_1 \times b_2$, if it corrects every 2-D burst of size $b'_1 \times b'_2$, where $b'_1 \leq b_1$ and $b'_2 \leq b_2$. A 2-D linear code is able to correct burst erasures of dimension $b_1 \times b_2$ if and only if no burst of area $b_1 \times b_2$ or smaller can be a codeword [26].

Let bivariate polynomial $b(\mathbf{D})$ represent the burst pattern, whose upper left corner is on the lattice and satisfies

$$\begin{aligned} \deg_{D_1} b(\mathbf{D}) &= b_1 - 1, \\ \deg_{D_2} b(\mathbf{D}) &= b_2 - 1. \end{aligned} \tag{26}$$

Also let polynomial $b'(\mathbf{D})$ represent a burst that has the same pattern as $b(\mathbf{D})$, but its upper left corner is a non-lattice point. Then, because of the lattice-cyclic property of the

TDWC, any burst of size $b_1 \times b_2$ can be expressed as

$$\mathbf{D}^{\mathbf{w}} b(\mathbf{D}) \qquad \text{or} \qquad \mathbf{D}^{\mathbf{w}} b'(\mathbf{D}), \tag{27}$$

where $\mathbf{w} \in \mathrm{LAT}(\mathbf{M})$. It is clear that a TDWC with the capability of correcting burst-erasures of dimension $b_1 \times b_2$ also corrects any bursts of the pattern expressed in (27).

In this work, we consider bursts of dimensions $N_1 \times N_2/2$ and $N_1/2 \times N_2$. The Reiger bound states that an (n, k) block code must have more than b parity bits to be able to correct an erasure burst of length b, i.e., $n - k \geq b$. Later, we show that we can design a half-rate TDWC that corrects any burst erasure of the mentioned sizes. Therefore, the TDWC satisfies the Reiger bound with equality [64].

We use a simple and efficient ML decoding to recover the erased bits. In other words, we find the unknowns by solving the equation $(c \circledast_{\downarrow\mathbf{M}} h_{eq})(\mathbf{n}) = 0$, where $h_{eq}(\mathbf{n})$ is introduced in Section 2.5. We represent this equation by $\mathbf{Hc} = 0$, where $\mathbf{H}$ is an $N/2 \times N$ matrix, $\mathbf{c}$ is an $N \times 1$ vector whose first $N/2$ elements are values of $c(\mathbf{n})$ at the lattice points and the second $N/2$ elements are values of the non-lattice points of $c(\mathbf{n})$, and $N = N_1 N_2$. The matrix $\mathbf{H}$ is constructed using two $N/2 \times N/2$ circulant matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ as $[\mathbf{A}_1|\mathbf{A}_2]$. The first column of the matrix $\mathbf{A}_1$ is a vector of size $N/2$ whose components are values of $h_{eq}(\mathbf{n})$ at the lattice points. Matrix $\mathbf{A}_2$ is constructed similarly, with the exception that it contains the values of the non-lattice points of $h_{eq}(\mathbf{n})$.

Now we need to specify the order in which the lattice points and non-lattice points are read. Let the lattice and non-lattice points be x_l and x'_l for $l = 1, 2, \ldots, \frac{N_1 N_2}{2}$, respectively. It can be verified that the vector $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$ is in $\mathrm{LAT}(\mathbf{M})$ for any 2×2 matrix $\mathbf{M}$ whose determinant is two. Therefore, ordering the lattice points and non-lattice points can be defined as

$$a = (i, j) \rightarrow b = (((i+2))_{N_1}, ((j+2))_{N_2}),$$

where $a \rightarrow b$ denotes that point a is followed by point b. Since $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$ is in $\mathrm{LAT}(\mathbf{M})$, a and b are both lattice points or non-lattice points. The process of enumerating lattice and non-lattice points is clarified by the following example. Let assume we have an array of dimension

4×6, as shown in Fig. 10, where $(1, 1)$ and $(1, 2)$ are coordinates for lattice and non-lattice points, respectively. As stated earlier, $(2, 2)$ is picked as the traveling vector. Thus, the second lattice point x_2 has the value of the element at the position $(((3))_{N_1}, ((3))_{N_2})$ and so on. As we can see, after x_6 we reach to x_1 again. However, not all the lattice points are enumerated yet. In such cases, an unlabeled lattice point that is the closest to the point $(1, 1)$ is used to continue. The procedure is continued till all lattice points are filled. The same method is used to enumerate the non-lattice points.

x_1	x'_1	x_5	x'_5	x_3	x'_3
x'_7	x_7	x'_{11}	x_{11}	x'_9	x_9
x_4	x'_4	x_2	x'_2	x_6	x'_6
x'_{10}	x_{10}	x'_8	x_8	x'_{12}	x_{12}

Figure 10: An example for enumerating lattice and non-lattice points.

Let $[\mathbf{E}]_{(b_1 b_2) \times 1}$ be the set of erasure bits and $[\mathbf{c} \backslash \mathbf{E}]_{(N/2 - b_1 b_2) \times 1}$ be the bits received excluding the erased bits. Additionally, let us denote $[h_{\mathbf{E}}]_{(N/2) \times (b_1 b_2)}$ and $[h_{\mathbf{c} \backslash \mathbf{E}}]_{(N/2) \times (N/2 - b_1 b_2)}$ as the columns of $h_{eq}(\mathbf{n})$ that correspond to the erased bits and correct bits, respectively. Using this notation, the maximum likelihood decoding is simplified to solving

$$[h_{\mathbf{E}}][\mathbf{E}] = [h_{\mathbf{c} \backslash \mathbf{E}}][\mathbf{c} \setminus \mathbf{E}]. \tag{28}$$

Since the TDWC is lattice cyclic, any burst erasure can be represented by (27). Therefore, prior to solving (28), we can shift the received sequence in the direction of a vector in LAT($\mathbf{M}$) such that we get a burst of the form $b(\mathbf{D})$ or $b'(\mathbf{D})$ in (27). Thus, only two possible choices exist for $[\mathbf{E}]$. The cyclic property of the TDWC simplifies the erasure decoding, because the inverse of $[h_{\mathbf{E}}]$ for those two configurations can be computed off-line. Hence, only XOR and AND operations are performed in the decoding process. Since C is a valid codeword, (28) has at least one solution. The ML decoder can recover the codeword uniquely, if and only if $[h_{\mathbf{E}}]$ is full rank. The following lemma shows the existence of a half-rate TDWC whose corresponding matrices $[h_{\mathbf{E}}]$ are full rank and invertible for the specified burst erasure dimensions.

Lemma 7. *There exist half-rate TDWCs capable of correcting erasure bursts of size $b_1 \times b_2$, where either $0 < b_1 \leq N_1/2$ and $0 < b_2 \leq N_2$ or $0 < b_1 \leq N_1$ and $0 < b_2 \leq N_2/2$.*

Proof. We prove this lemma for the following two cases:

1. $GCD(N_1, N_2) = 2$ and either N_1 or N_2 is a multiple of four.

2. $GCD(N_1, N_2) = 2$ and neither N_1 nor N_2 is a multiple of four.

First, we prove this lemma for Case 1 using the following lemma.

Lemma 8. *Let A be a 2-D array of size $N_1 \times N_2$, where $GCD(N_1, N_2) = 2$ and either N_1 or N_2 is a multiple of four. Also let M be a 2×2 matrix whose determinant is two. If the element at position $(1, 1)$ is a lattice point, then the element at position $(N_1/2+1, N_2/2+1)$ is a non-lattice point.*

The proof of Lemma 8 is given in Appendix A. To complete the proof of Lemma 7 for Case 1, we need to show that there exists an arbitrary lattice-cyclic 2-D code $\mathcal{C}_{\mathbf{G}}$ of dimension $N_1 \times N_2$, whose matrix $[h_{\mathbf{E}}]$ is full rank for erasure bursts of sizes $N_1 \times N_2/2$ and $N_1/2 \times N_2$. The codeword array $c \in \mathcal{C}_{\mathbf{G}}$ satisfies $\mathbf{Hc} = 0$, where the matrix $\mathbf{H}$ and the vector $\mathbf{c}$ are obtained as explained before. According to Lemma 8, the point at the position $(N_1/2, N_2/2)$ is non-lattice. Assume that the number associated to this non-lattice point is k. $\mathbf{A}_1$ is chosen to be a full rank circulant matrix. Then, $\mathbf{A}_2$ is constructed as follows: The k^{th} column of the matrix A_2 is chosen to be the same as the first column of the matrix A_1. It can be verified that the matrix $[h_{\mathbf{E}}]$ generated by matrices A_1 and A_2 is full rank. Thus, this code can recover the specified erasure bursts.

The proof for Case 2 is similar to that of Case 1, with a slight change in the construction of the matrix A_2. Matrix A_2 is obtained such that its k^{th} column is summation of columns i and j of matrix A_1, where k is the number associated to the non-lattice point at position $(1, 2)$ and i and j correspond to lattice-pints at positions $(1, N_2/2)$ and $(N_1/2, N_2/2)$. It can be verified that the matrix $[h_{\mathbf{E}}]$ generated by matrices A_1 and A_2 is full rank. Thus, this code can recover the specified erasure bursts. By Lemma 6, we conclude that there always exists a TDWC that can recover burst erasures of dimensions $N_1 \times N_2/2$ and $N_1/2 \times N_2$, where $GCD(N_1, N_2) = 2$. □

Collectively, to design a TDWC with dimension $N_1 \times N_2$ that is capable of recovering erasure bursts of sizes $N_1 \times N_2/2$ and $N_1/2 \times N_2$, we proceed as follows: First, we design a two-channel maximally decimated filter bank whose synthesis and analysis filters have impulse responses with a region of support of $N_1 \times N_2$. Then, we search for the prefilter $\lambda(n)$ such that the corresponding columns in matrix $[h_E]$ are independent for burst erasures of sizes $N_1 \times N_2/2$ and $N_1/2 \times N_2$. According to Lemma 7, we are guaranteed that such a prefilter exists.

Example 3. *We construct two codes of dimensions* 4×6 *and* 10×4*. The generator filters and the burst correction capabilities of the designed codes are as follows.*

- *Code with dimension* 4×6 *corrects burst patterns of sizes* 2×6, 3×4, *and* 4×3*. As we can see, this code has the efficiency of one for any rectangular burst erasure of area* $\frac{N_1 N_2}{2}$*. The synthesis bank filters and the prefilter are given as*

$$G_0(\mathbf{D}) = 1 + D_1 + D_1^2 D_2 + D_1^2 D_2^2 + D_1^3 D_2^2,$$

$$G_1(\mathbf{D}) = 1 + D_1 + D_1 D_2 + D_1^2 D_2^2 + D_1^3 D_2^2,$$

$$\hat{\lambda}(\mathbf{D}) = D_1^3 y + D_1^4 D_2^2.$$

In designing this code, one degree-2τ *building block as in (12) with the parameters*

$$\mathbf{u}(D_1) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{v}(D_1) = \begin{bmatrix} D_1 \\ 1 \end{bmatrix} \qquad \zeta(D_1) = 1$$

is used. With these parameters, the building block is as follows

$$\mathbf{V}_1 \left(D_2; \mathbf{u}(D_1), \mathbf{v}(D_1), \zeta(D_1)\right) = \begin{bmatrix} D_1(1 + D_2 + D_2^2) & 1 + D_2^2 \\ D_1(1 + D_2^2) & 1 + D_2 + D_2^2 \end{bmatrix}.$$

- *Code with dimension* 10×4 *corrects burst patterns of sizes* 5×4 *and* 4×5*. The synthesis bank filters and the prefilter are as follows*

$$G_0(\mathbf{D}) = D_1 D_2 + D_1^7 D_2^3 + D_1^2 D_2 + D_1^5 D_2^2 + D_1^8 D_2^3,$$

$$G_1(\mathbf{D}) = D_1 D_2 + D_1^7 D_2^3 + D_1^4 D_2^2 + D_1^2 D_2 + D_1^8 D_2^3,$$

$$\Lambda(\mathbf{D^M}) = D_1^9 D_2^3 + D_2^2.$$

In designing this code, the building block of previous example is used.

2.7 Conclusion

We introduced half-rate two-dimensional wavelet codes (TDWCs). The encoder consists of a synthesis bank of a two-channel two-variable filter bank. The corresponding analysis bank generates the syndrome of the code. We investigated properties of these codes. We showed that these linear codes are lattice cyclic. This property simplified the encoding and the erasure decoding of TDWCs. We also introduced a methodology to design TDWCs over binary erasure channels. We showed that the half-rate TDWCs of dimensions $N_1 \times N_2$ satisfy the Reiger bound with equality for burst erasures of dimensions to $N_1 \times N_2/2$ and $N_1/2 \times N_2$. These codes can recover burst erasures with a simple and efficient ML decoding.

CHAPTER III

DISTRIBUTED SOURCE CODING AT THE ASYMMETRIC RATE

3.1 Introduction

In this chapter, we propose a scheme for distributed source coding of correlated sources at the asymmetric rate. We show that our approach reaches the Slepian-Wolf limit, if we use a channel code that achieves the capacity of the equivalent channel. For coding, we use LDPC codes, because of their potentially capacity-approaching performance. This is true for infinite-length sequences. For finite-length sources, we take a new approach. For the first time, we take both the correlation channel and the actual channel and formulate the source coding problem to the code design over non-uniform channels. This is in contrast with the previous works, which design the LDPC code for the equivalent channel. As shown later, our design significantly benefits from our framework to design optimal LDPC codes for parallel channels. This benefit is more stressed for applications with finite-length sequences.

First, we investigate a system of two correlated sources. We present a methodology that involves construction of LDPC codes for non-uniform channels. We further study an extension of our approach to three correlated sources. We investigate two cases: a more general case, where sources are pairwise correlated with arbitrary correlation probabilities and a special case, where sources are pairwise correlated with the same correlation probability. We show that the former is simplified to non-uniform LDPC codes and the latter is simplified to semi-random punctured LDPC codes. This is a new formulation for distributed source coding using LDPC codes. We note that our proposed distributed source coding is well suited for sensor networks (where short length codes are used) and many other distributed source coding applications.

3.1.1 Related Work

The link between conventional channel coding and lossless distributed source coding with side information available at the decoder was established based on Wyner's scheme [75]. The basic idea was to partition the space of all possible source outcomes into disjoint bins (sets) that are the cosets of some good linear channel codes for the specific correlation model. If the correlation between X_1 and X_2 can be modeled by a binary channel, Wyner's syndrome concept can be extended to all binary linear codes; and state-of-the-art near-capacity channel codes such as turbo [5] and low-density parity-check (LDPC) codes [20,43] can be employed to approach the Slepian-Wolf limit.

In [51], the first practical framework for distributed source coding using syndromes (DISCUS) was introduced. Advanced schemes based on turbo codes and LDPC codes were proposed in [1, 4, 23, 38, 46] and [36, 37, 59, 65], respectively. Irregular repeat accumulate (IRA) codes were also used for distributed source coding in [39] and [70]. Extension of distributed source coding to three correlated sources is studied in [35].

3.1.2 Contribution

In this chapter, we propose to apply non-uniform LDPC codes for distributed source coding at the asymmetric rates. As opposed to previous works, which design the LDPC code for the equivalent channel, we take a completely new approach. We take both the correlation channel and the wireless channel and formulate the source coding problem to the code design over parallel channels. We design a non-uniform LDPC code for this set of parallel channels. Our proposed design criteria improve the performance of the distributed source coding significantly for finite-length cases [59, 60].

We also extend our method to three sources. We study the distributed source coding of three sources that are pairwise correlated with arbitrary correlation probability. As a special case, we also study a simpler case, where sources are correlated with the same correlation probability. This problem is also studied in [35]. However, we take a new approach based on LDPC codes with emphasis on finite-length sequences. We model this problem with a set of parallel channels that simplifies the problem to semi-random puncturing. This new model improves the performance of the distributed source coding significantly.

3.2 Distributed Source Coding of Two Correlated Sources at The Asymmetric Rate

Consider a communication system of two sequences X_1 and X_2. Let $\{X_{1_i}, X_{2_i}\}_{i=1}^{\inf}$ be a sequence of independent and identically distributed (i.i.d.) drawings of a pair of correlated discrete random variables X_1 and X_2. In this chapter, we assume that the correlation value is known in advance. Each node compresses its data without communicating with the other node and sends the compressed data to the sink. A suitable source encoder that removes the redundant bits, reduces both the length of the transmission information and power consumption. Such a system requires distributed source coding. In this chapter, we focus on distributed source coding at the asymmetric rate. Distributed source coding at the asymmetric rate corresponds to the case that one of the sources, X_1, is compressed conventionally at the rate $R_{X_1} = h(X_1)$ and is recovered perfectly at the decoder, while the other signal, X_2, is compressed as close as possible to the Slepian-Wolf limit $R_{X_2} = h(X_2|X_1)$ to achieve the overall rate $R_{X_1} + R_{X_2}$ close to $h(X_1, X_2)$. This procedure is also known as distributed source coding X_2 with the side information X_1 available at the decoder.

To study the distributed source coding of two sources, first we need to model the correlation between sources. In this work, we consider the system of statistically dependent sources whose dependency can be fully explained by their conditional probability mass function $P[X_2|X_1]$. This correlation can also be modeled by a BSC whose input and output are X_1 and X_2 and has a crossover probability of $P[X_1 \neq X_2|X_1] = p$. In this case $h(X_2|X_1) = h(p) = -p \log_2 p - (1-p)\log_2(1-p)$.

Source coding of X_2 with side information X_1 available at the decoder is performed as follows: X_2 is fed into a rate R systematic LDPC encoder. At the output of the encoder, we only send the corresponding parity bits, P_{X_2}, of the codewords while disregarding the information bits. This results in an encoding rate of $R_{X_2} = \frac{1}{R} - 1$ bits per input bit. The rate R is chosen such that the compression rate R_{X_2} gets as close as possible to the

Slepian-Wolf limit. The decoder tries to recover X_2 by using P_{X_2} and X_1, which are also available at the decoder. The only difference between this decoding and the original LDPC decoding is the initialization of the log-likelihood ratios (LLRs) which will be discussed more in Section 3.2.2.

3.2.1 Non-uniform LDPC Code Design

We propose to model the procedure for distributed source coding of X_2 at the asymmetric rate with a set of parallel channels as shown in Fig. 11. Suppose the length of the LDPC code of rate R is n. Information bits (Rn bits) are transmitted over the dependency channel and parity bits ($(1-R)n$ bits) are transmitted over the wireless channel, which is assumed to be an ideal channel, unless otherwise stated. The solid line in this figure shows the bits that are actually transmitted to the decoder and dashed lines show the bits that are not sent. Let Z_i, $i = 1, 2$, denotes the random variable that is equal to the LLR of the received bit for the i^{th} channel. The set of two channels can be modeled as a single channel having the LLR distribution of $P_Z(z) = (1 - R)P_{Z_1}(z) + RP_{Z_2}(z)$. Using the methods in [57], a good degree distribution for LDPC code is set up for the equivalent channel. Designing the LDPC code for the equivalent channel works well for infinite lengths, which may not be the case for some applications such as sensor networks. In this work, we consider LDPC codes with finite lengths. Thus, instead of designing the best LDPC code for the equivalent channel, we propose to design a non-uniform LDPC code by considering the fact that different bits are subject to different sources of noise [59]. In [50] the authors introduced the ensemble of graphs that is used over parallel channels and they showed that better results for parallel channels are achievable by designing non-uniform LDPC codes, especially for short and

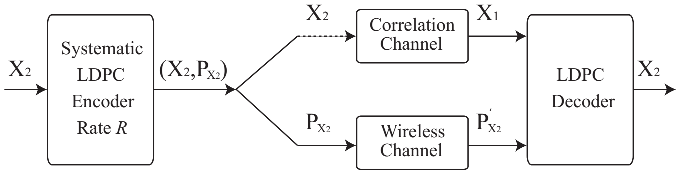

Figure 11: Source coding X_2 with side information X_1 available at the decoder.

moderate length cases. In this section, we discuss the design methodology for the non-uniform LDPC codes.

To encode X_2, we have two sets of bits, the first set is passing through the ideal channel and the second set is passing through the correlated channel. We introduce an ensemble $g(\Lambda, \rho)$ of bipartite graphs for non-uniform error correction. Distinct degree distributions are chosen for each set. Let $\Lambda = \{\lambda^1(x), \lambda^2(x)\}$, where $\lambda^i(x)$ is the variable node degree distribution of each set. Also, let $\rho(x)$ be the check node degree distribution. We choose Λ and ρ according to the guideline of [50] for non-uniform channels. This guideline in the nutshell suggests higher degree for the bits that experience more error prone channels. Then, for a codeword of length n and a degree distribution $(\Lambda(x), \rho(x))$, a random realization of H is generated. Using Gaussian elimination, the parity-check matrix H' for the equivalent systematic LDPC code is constructed. Matrix H' is used to find the generator matrix G for the systematic LDPC code. For encoding a sequence, we compute the parity bits of this sequence using the matrix G.

3.2.2 Decoding

The decoder needs to determine an n-length sequence of X_2 from its parity bits P_{X_2} and its correlated sequence X_1. To apply the message passing algorithm, LLR values of all bits are required to be known. LLRs of the parity bits and the information bits are infinity and $\ln\left(\frac{1-p}{p}\right)$, respectively, since these bits are transmitted through ideal channel and BSC with crossover probability p. Knowing the LLRs of all bits and the value of the signal X_1, the message passing algorithm can be executed to decode X_2.

In this section, we investigated the problem of distributed source coding of two correlated sources at the asymmetric rate. In the following section, we extend our method to three correlated sources. Exploiting the correlation between three sources simultaneously obtains rate savings compared to the two-source setup.

3.3 Distributed Source Coding of Three Correlated Sources at the Asymmetric Rate

Consider a communication system of three sources X_1, X_2, and X_3 each of which is i.i.d. binary sequences of length k. The bits of these sources are assumed to be zero and one, with equal probability. Sources X_1, X_2 and X_3 are statistically dependent to each other. As stated in the previous section, the correlation between sources can be fully explained by their conditional probability mass function. First, we consider a more general case in which $P[X_k \neq X_i|X_i] = p_{ik}$, $\forall i, k \in \{1, 2, 3\}$ and $i \neq k$. Then, we study a simpler case, where sources are pairwise correlated with the same correlation probability. The correlation values are known in advance. Since the dependency of two correlated sources X_i and X_k can be represented by their conditional probability mass function $P[X_k \neq X_i|X_i] = p_{ik}$, we can model this dependency with a BSC whose cross over probability is p_{ik}. The dependency between X_i and X_k while X_j is also available at the decoder cannot be simply modeled by a single BSC with a fixed crossover probability. The crossover probability depends on the values of X_i and X_j. The correlation between X_i and X_k given that X_j is also present error free at the decoder can be computed as

$$P[X_k \neq X_i|X_i = X_j, X_j] = \frac{p_{ik} + p_{jk} - p_{ij}}{2(1 - p_{ij})}, \tag{29}$$

$$P[X_k \neq X_i|X_i \neq X_j, X_j] = \frac{p_{ik} - p_{jk} + p_{ij}}{2p_{ij}}.$$

Figure 12 represents the model of the correlation between source X_i and X_k. As we can see in Fig. 12, with probability p_{ij} and $(1 - p_{ij})$, X_k is the input to a BSC with cross over probability $\frac{p_{ik}-p_{jk}+p_{ij}}{2p_{ij}}$ and $\frac{p_{ik}+p_{jk}-p_{ij}}{2(1-p_{ij})}$, respectively. The BSC with input X_k whose crossover probability depends on the values of X_i and X_j is denoted by $Q_{ij}(k)$.

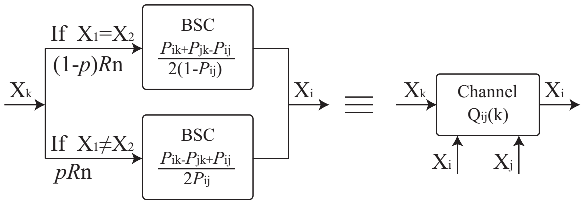

Figure 12: The correlation model between X_k and both X_i and X_j.

In this section, we study on distributed source coding of three correlated sources at the asymmetric rate. The achievable rate for compression of three correlated sources X_1, X_2, and X_3 at the asymmetric rate must satisfy the following inequalities [74]:

$$R_{X_1} \geq h(X_1), \quad R_{X_2} \geq h(X_2|X_1), \quad R_{X_3} \geq h(X_3|X_1, X_2),$$

$$R_{X_1} + R_{X_2} + R_{X_3} \geq h(X_1, X_2, X_3).$$

The source coding of sources X_1 and X_2 is done with the same method described in Section 3.2, and the third one is compressed with the rate as close as possible to the theoretical limit $h(X_3|X_1, X_2)$. Since $h(X_3|X_1, X_2) \leq h(X_3|X_1)$, exploiting the correlation between all sources simultaneously obtains rate savings compared to the two-source setup. The source coding of X_3 is done as follows: source X_3 is fed into a systematic LDPC code of rate $R = \frac{1}{1+h(X3|X_1,X_2)}$. The distributed source encoder sends only the corresponding parity bits P_{X_3}. As shown in Fig. 13, the distributed source coding model involves a set of parallel channels. Therefore, we require a non-uniform LDPC code for this set of parallel channels.

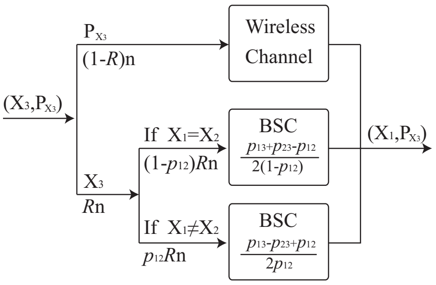

Figure 13: Source coding of X_3 using the side information X_1 and X_2 available at the decoder.

3.3.1 Decoding

We note that X_1 and X_2 are recovered as in Section 3.2. However, X_3 is recovered by using P_{X_3} and the side information provided by both X_1 and X_2. We assume that X_1 and X_2 are known error free at the decoder. The decoder requires to have the LLR values of all bits. Identical and different bits of X_1 and X_2 indicate that X_3 is passed through a BSC with crossover probability $\frac{p_{13}+p_{23}-p_{12}}{2(1-p_{12})}$ and $\frac{p_{13}-p_{23}+p_{12}}{2p_{12}}$, respectively. The latter probability is greater than 1/2 if $P_{13} > P_{23}$. Therefore, based on the values of these correlation probabilities, we choose one of the following decoding procedures:

- $P_{13} < P_{23}$: First, we compare X_1 and X_2 bitwise. The LLR values of bits in X_3 that correspond to the identical bits are set to $\ln(\frac{2-p_{12}-p_{13}-p_{23}}{p_{13}+p_{23}-p_{12}})$. Otherwise, their values are set to $\ln(\frac{p_{12}-p_{13}+p_{23}}{p_{13}-p_{23}+p_{12}})$. Having the LLR values of all the bits and X_1, the message passing algorithm can be executed to recover X_3.
- $P_{13} > P_{23}$: First, we compare X_1 and X_2 bitwise. The LLR values of bits in X_3 are set to $\ln(\frac{2-p_{12}-p_{13}-p23}{p_{13}+p_{23}-p_{12}})$, if the corresponding bits in X_1 and X_2 are identical. Otherwise, their LLR are equal to $\ln(\frac{p_{13}-p_{23}+p_{12}}{p_{12}-p_{13}+p_{23}})$. The value of X_1 is modified as follows to compensate for error probability greater than 1/2:

$$X_1 = \begin{cases} X_1, & \text{if } X_1 = X_2 \\ -X_1, & \text{if } X_1 \neq X_2 \end{cases}$$

Having the LLR values of all the bits and the modified value of X_1, the message passing algorithm can be executed to recover X_3.

3.4 Distributed Source Coding of Three Sources- A Special Case

As a special case, we study the problem of distributed source coding of three sources that are pairwise correlated with the same correlation probability p. The correlation between X_i and X_k given that the signal X_j is also present error free at the decoder can be computed as

$$P[X_k \neq X_i | X_i = X_j] = \frac{p/2}{1-p}, \;\; P[X_k \neq X_i | X_i \neq X_j] = \frac{1}{2}. \tag{30}$$

Using (30), we can say that with probability p, X_k is the input to a BSC with cross over probability $\frac{p/2}{1-p}$, and with probability $1-p$ it is the input to a BSC channel with cross over probability 1/2. The output of the channel with cross over probability 1/2 can be considered as erased (punctured) bits. Therefore, X_1 is punctured X_3 that has passed through a BSC with cross over probability $\frac{p/2}{1-p}$. This is a semi-random puncturing, since we only know a fraction p of the information bits is erased but we have no further information about the positions of these bits. Thus, the distribution source coding is reduced to the design of an LDPC code with semi-random puncturing.

The compression rate of X_3 with both X_1 and X_2 present at the decoder is computed by $h(X_3|X_1, X_2) = h(q) = (1-p)h(\frac{1-3p/2}{1-p}) + p$, while the compression rate for the case that only one signal is present at the decoder can be found from $h(X_i|X_j) = h(p)$. As we can see, the compression rate for the case of three correlated sources is lower. Therefore, it is more efficient to consider the three correlated sources together instead of the pairwise study.

In the following section, we investigate the problem of designing an LDPC code of rate $R = \frac{1}{1+h(X_3|X1,X2)}$ whose performance does not degrade after puncturing (here, by performance we mean the ratio $\frac{R}{C}$, where C is the capacity of the channel and R is the maximum rate of the code for which the error probability is less than a required value). To design the LDPC code, we consider the fact that some bits are punctured randomly. It is worth noting that in [35], the authors designed an LDPC code by applying the density evolution to the initial distribution of the equivalent channel. However, we show that our non-uniform formulation and the puncturing model perform considerably better than previous methods for finite-length codes.

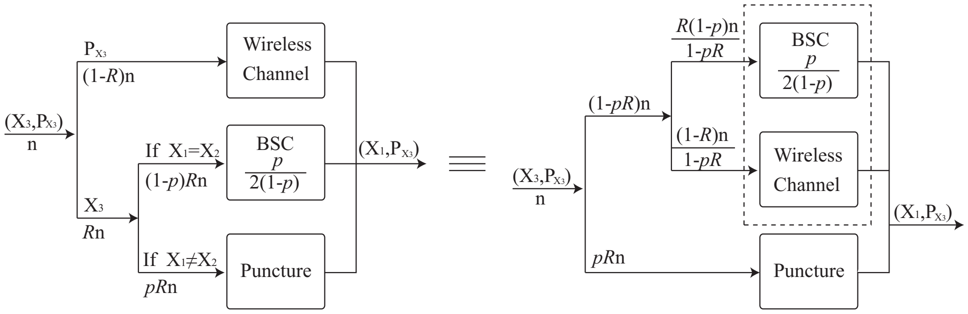

Figure 14: Source coding of X_3 for three correlated sources with identical correlation probabilities.

3.4.1 Rate-Compatible LDPC Code Design

Figure 14 shows that a fraction $1-p$ of the information bits are passing through the BSC with crossover probability $\frac{p/2}{1-p}$ whose capacity is denoted by c^*, while the rest of the information bits are punctured. To design an appropriate LDPC code for this channel, first we design a parent LDPC code for the case that no puncturing has occurred, as shown in Fig. 15. We can then apply the puncturing to revert to the original system of Fig. 14. The goal is to construct a parent LDPC code, such that its performance does not degrade after puncturing. The channel c_1 in Fig. 15 has a capacity lower than c^*, which requires more bits to be transmitted. To compute the value of c_1 in Fig. 15, we use the results in [48], which states that the performance of an arbitrary code does not change after random puncturing if and only if the ratio of the *rate/capacity* for the channels of dotted section in Fig. 14b and the channels in Fig. 15 are identical. The overall capacity of the dotted section in Fig. 14b and the overall capacity of the channel in Fig. 15 are equal to $C_{eq} = \frac{1-R}{1-pR} + \frac{R(1-p)c^*}{1-pR}$ and $C = 1-R+Rc_1$, respectively. The rate of the LDPC code for the dotted section in Fig. 14b and the rate of the LDPC code for channels in Fig. 15 are $\frac{R}{1-pR}$ and R, respectively.

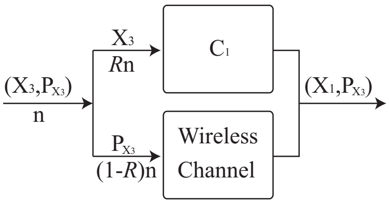

Figure 15: A model that describes the channel of Fig. 14 before puncturing.

Equating the ratios of *rate/capacity* for these two channels, we obtain $c_1 = (1-p)c^*$. Therefore, if the parent LDPC code reaches the overall capacity of the channels in Fig. 15, we can conclude that the punctured parent code also reaches the capacity of Fig. 14. To design the best LDPC code of rate R for the channel of Fig. 15, we suggest using a non-uniform framework introduced in [50].

3.4.2 Decoding

At the decoder, X_3 is recovered by using P_{X_3} and the side information provided by both X_1 and X_2. We note that X_1 and X_2 are assumed to be known at the decoder error free. The decoder requires to have the LLR values of all bits. Equation (29) shows that identical bits of X_1 and X_2 indicate that those bits are passed through a BSC with crossover probability $\frac{p/2}{1-p}$ but different bits indicate that those bits must be considered punctured bits. Therefore, by comparing the values of X_1 and X_2 at each index, we can compute the LLR value of

X_3 at the associated index. If X_1 and X_2 are identical, then the LLR value of X_3 equal to $\ln(\frac{1-3p/2}{p/2})$. Otherwise the LLR value is set to zero. Having LLR values of all bits and the values of the side information, the message passing algorithm is executed to recover X_3.

3.5 Simulation Results

First, we investigate the application of LDPC codes and BCH codes to distributed source coding. The simulation results show that for moderate-length sequences, a LDPC code corrects the same number of error bits/ packets as a BCH code does, while it has a lower redundancy. As an example, simulation results show that LDPC code of rate 0.55 has packet error rate (PER) of $PER < 10^{-4}$ for $p = 0.045$. The performance of the BCH code is given by

$$1 - PER = \sum_{j=0}^{t} \binom{n}{k} p^j (1-p)^{n-j},$$

where t is the error correcting capability of the BCH code. For $p = 0.045$ and $1 - PER = 1 - 10^{-4}$, t needs to be equal to 73. BCH code of length 1023 and error correcting capability of 73 has rate 0.43 [34]. As we can see, LDPC code has lower redundancy compared with BCH code. We also compare LDPC codes with BCH codes considering the decoding energy. Similar to [58], we define the energy-efficiency factor as follows:

$$\eta = \eta_e(1 - PER), \tag{31}$$

where η_e, energy efficiency, is defined as the energy for communication of the information bits divided by the sum of total energy for communication of both the information bits and the redundant bits and decoding energy consumption. The expression $(1 - PER)$ indicates the reliability. For $p = 0.045$, energy efficiency factor is equal to 0.5427 and 0.4231 for LDPC code of (n=1000,R=0.55) and BCH code of (n=1023,R=0.43), respectively. The following figure compare the energy efficiency of LDPC codec and BCH codes. In design of these codes the PER of less than 10^{-4} was sought. The simulation results confirm that LDPC codes are more energy-efficient than BCH codes [59]. Therefore, the focus of our work is on distributed source coding using LDPC codes.

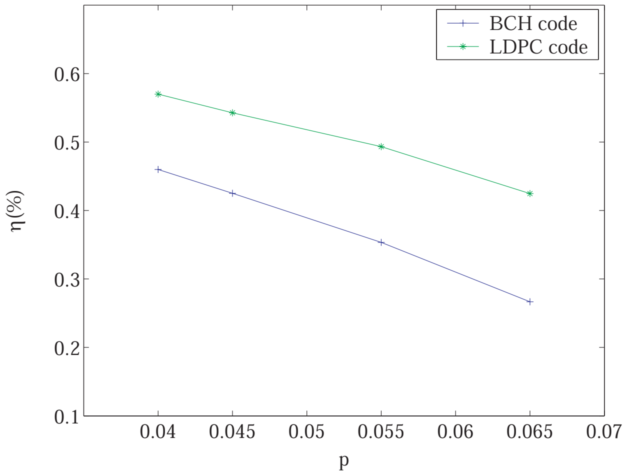

Figure 16: Energy efficiency of BCH codes of length 1023 and LDPC codes of length 1000

3.5.1 Two Correlated Sources

We simulate and test our proposed model for distributed source coding of two correlated sources using LDPC codes. We choose $p = 0.11$, which results in the joint entropy of $h(X_1, X_2) = 1.5$. In distributed source coding at the asymmetric rate, X_1 is compressed conventionally and sent at the full rate of $R_{X_1} = h(X_1) = 1$ and is recovered perfectly at the decoder. The signal X_2 is compressed as close as possible to the Slepian-Wolf limit which is equal to $R_{X_2} = h(X_2|X_1) = 0.5$. Hence, we design an LDPC code of rate $R = 2/3$ and length $n = 1000$. This code was chosen randomly from the ensemble that is defined by the following degree distribution to achieve nonuniform property.

$$\Lambda(x) = \{x^2, 0.7585x^3 + 0.1422x^4 + 0.0993x^8\},$$

$$\rho(x) = \{0.5x^{10}, 0.5x^{11}\}.$$

The Slepian-Wolf theoretical limit for this code is 0.5 bits for ideal channels. Figure 17 compares the performance of the distributed source coding of the non-uniform LDPC code with the results of [65]. We note that the nonuniform LDPC code of length 1000 outperforms the DISCUS using LDPC code of the same length by more than 60%.

To show that the performance gap of our proposed method from the Slepian-Wolf limit reduces as the code length increases, we run the simulation for LDPC code of (34) with lengths 2000 and 4000. The following table summarizes the gap to the Slepian-Wolf theoretical limit for code lengths 1000, 2000, and 4000. As we can see, the gap from the theoretical

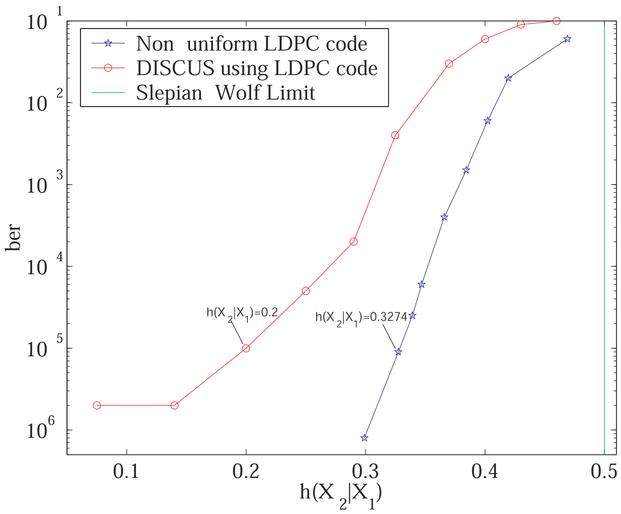

Figure 17: Comparison of distributed source coding using non-uniform LDPC codes and the method in [65] for $n = 1000$.

limit decreases as the code length increases.

Table 1: Gap from the Slepian-Wolf theoretical limit for code lengths 1000, 2000, and 4000.

Code length	1000	2000	4000
Gap from the theoretical limit	0.1726	0.1416	0.1267

3.5.2 Three Correlated Sources

We simulate and test our proposed model at the asymmetric rate. Similar to Section 3.5.1, we choose $p = 0.11$ that results in the compression rates of $R_{X_2} = 0.5$ and $R_{X_3} = 0.4$ for sources X_2 and X_3, respectively. The desired rate of LDPC code for distributed source coding of X_3 is equal to $R = \frac{1}{1+R_{X_3}} = 0.71$. Therefore, we design a parent nonuniform LDPC code of rate R for the channel of Fig. 15. The resulting code has the following degree distributions:

$$\Lambda(x) = \{x^2, 0.7054x^3 + 0.1793x^4 + 0.1153x^8\}, \quad \rho(x) = \{0.5x^{10} + 0.5x^{11}\}.$$

The simulation results show that for length 1000, convergence of the nonuniform LDPC code is achieved by $p^* = 0.08$ for which the compression rate is $R_{X_3} = H(X_3|X_1, X_2) = 0.3174$. In other words, this code is about 0.0826 bits away from the Slepian-Wolf limit. In [35], the authors report the simulation results of an LDPC code of length 5×10^5. To compare our method with the method of [35], we generate an LDPC code of length 1000 with the degree distribution reported in [35]. We find the value of p^* at which the convergence of this LDPC code is achieved. The following table summarizes the gap between the Slepian-Wolf limit and the convergence of the LDPC code of length 1000 for both the proposed method and the method in [35]. As we can see, the gap from the theoretical limit in [35] is twice as much as our proposed method.

Table 2: Gap of $H(p^*)$ from the Slepian-Wolf limit.

	p^*	S-W limit	Gap
Proposed method	0.08	0.4	0.0826
Method of [35]	0.21	0.8247	0.1712

3.6 Conclusion

We studied the problem of source coding for two and three correlated sources. We proposed to use nonuniform LDPC codes for distributed source coding of correlated sources. This design criterion improved the performance of the source coding considerably. The convergence of the nonuniform LDPC code of our method is almost 60% closer to the Slepian-Wolf limit than the uniform design for two sources. Through simulation results, we also verified that the gap from the Slepian-Wolf theoretical limit decreases significantly as we increase the code length. For three correlated sources, first we studied the case where sources are correlated with arbitrary correlation probabilities and then we studied a special case, where sources are pairwise correlated with the same correlation probability. We showed that these problems simplified to the design procedure of nonuniform LDPC codes and randomly punctured LDPC codes, respectively. The simulation results of our approach shows that the convergence of the LDPC code of length 1000 is only 0.08 away from the Slepian-Wolf limit.

CHAPTER IV

DISTRIBUTED SOURCE CODING AT ARBITRARY RATE

4.1 Introduction

In this chapter, we study the distributed source coding of correlated sources that can achieve every arbitrary coding rate on the Slepian-Wolf rate region. We propose a scheme for this problem using a single systematic channel code. Similar to Chapter 3, we are interested in applications with finite-length sequences. By modeling the distributed source coding with a set of parallel channels, we simplify the distributed source coding problem to the rate-compatible LDPC code design with an unequal error protection property. At the decoder, each source is decoded independently (only part of information bits are exchanged) which prevents the propagation of errors. The simulation results confirm that the gap from the theoretical limit remains almost the same for different rates on the Slepian-Wolf rate region. In summary, our design significantly benefits from our framework to design optimal LDPC codes for nonuniform channels, rate-compatible punctured LDPC codes, and unequal error protection codes.

4.1.1 Related Work

Most of the previous works on distributed source coding are based on compressing at a corner point (points A and B in Fig. 2), i.e. the signal X_1 is compressed conventionally and sent at the full rate of $R_{X_1} = h(X_1)$ and is recovered perfectly at the decoder, while the signal X_2 is compressed as close as possible to the Slepian-Wolf limit $h(X_2|X_1)$. However, for some applications, it is required to operate at rates other than corner points. For example, at the midpoint of the Slepian-Wolf rate region, the sources are compressed at the same rates. This is desirable in sensor networks because of the uniform energy consumption among sources. Distributed source coding at arbitrary rates has been studied as an important problem in recent years [12, 22, 52, 61, 62, 66, 70]. In [52], Pradhan et al. studied the problem of distributed source coding of two sources at the symmetric rate (point C in Fig. 2) using trellis and lattice codes. They showed that the performance of these symmetric methods are the same as that of asymmetric ones. Garcia-Frias also considered the compression of two sources at the symmetric rate using turbo codes for the joint source-channel coding [22]. The extension of this work is proposed recently in [12]. In this paper, the authors studied distributed source coding of two correlated sources at any arbitrary rate on the Slepian-Wolf rate region using turbo codes. In [66] and [70], the authors have considered not only the symmetric rate, but also any arbitrary rates on the Slepian-Wolf rate region. We note that both [66] and [70] are based on sending syndromes. In [70], using IRA and turbo codes, the authors designed distributed source coding by channel code partitioning. Recently, Gehrig and Dragotti proposed an approach using systematic and non-systematic channel codes [24]. Their method is based on sending partial information bits along with the syndromes generated by the channel encoder.

4.1.2 Contribution

To clarify the significance of our work, we want to stress on two points. First, this work studies the distributed source coding of applications with finite-length sequences. Distributed source coding using finite-length channel codes introduces new challenges, since the assumption of capacity-approaching channel code is not valid anymore. We model the distributed source coding problem with code design for a set of parallel channels. This model for short-length sequences provides the advantage of using nonuniform, rate adaptive and unequal error protection codes. Second, our work focuses on distributed source coding based on sending parity bits. This is a substantial diversion over the other possible method which is based on syndromes. If the wireless channel is ideal, both syndrome and parity methods can be used. However, in applications where wireless channels are not error free, the syndrome-based method needs to be modified to be applied for distributed source coding. The methods in [24, 66, 70] are based on sending syndromes.

- The decoding procedure in [66] consists of decoding the difference between sources before decoding either source. Once the first decoding is complete, the second decoding can be done to either source. The second decoding is in essence an entropy decoding. The code is designed to be strong enough to decode the difference. However, this design procedure does not guarantee a successful entropy decoding. The simulation results reported in [66] shows that the gap from theoretical limit increases as compression rate raises.

- The decoding procedure in [70] consists of a channel decoding and source encodings. Therefore, the decoding of sources are dependent to each other, i.e., both sources are either recovered perfectly or corrupted. The authors also extend their method to more than two sources. The method works when the summation of the sources has a Bernoulli distribution.

- In [24], the decoder first decodes the difference between sources. Then, using the difference and the invertible section of the parity check matrix, both sources are decoded. This decoding algorithm propagates the error. The authors extend their technique to more than two sources. However, their method only considers two correlated sources at a time. Therefore, for cases with more than two sources, we expect that the scheme to be bounded away from the theoretical limit.

As we see later, the decoding procedure we proposed avoids dependent decoding. Hence, it prevents error propagation. Moreover, for more than two sources, no particular correlation model among sources is necessary for our approach and we consider the correlation between all sources simultaneously which results in rate savings in comparison with the two-source setup.

The authors in [12] studied distributed source coding of two sources using the parity bits generated by turbo codes. Our work differs from this paper in the sense that we design the LDPC code for the set of parallel channels, which improves the performance significantly for short-length sequences. As we see in Section 6.6, our method has a gap of 0.173 bits and 0.1416 bits from the Slepian-Wolf limit for LDPC code of length only 1000 and 2000, respectively. The simulation results in [12] reports a gap of 0.13 for turbo code of length 16384. We note that our method performs almost the same as [12] for much shorter code lengths. We also study the problem of distributed source coding, when correlation parameter is unknown in advance.

4.2 Distributed Source Coding of Two Sources at Arbitrary Rates

First, we study source coding of two correlated sources and then we apply our approach to three or more correlated sources. Consider a communication system consisted of two correlated binary sequences X_1 and X_2, where $P[X_1 = 0] = P[X_2 = 0] = 1/2$. The correlation between X_1 and X_2 is modeled as the input and output of a BSC with crossover probability $P[X_2 \neq X_1|X_1] = p$.

We propose a method for compression of two correlated sources at every arbitrary rate on the Slepian-Wolf rate region using a single channel code. To describe the procedure, first we assume that each source uses a separate systematic LDPC code of rate R_i, where $i \in \{1, 2\}$. Then, we show how the same procedure compresses both sources at rates close to the theoretical limit using only a single systematic channel code. The encoding procedure for the k-bit sequence of source X_1 is done as follows: source X_1 is fed into a systematic code of rate $R_1 = \frac{k}{k+|P_{X_1}|}$, where $|P_{X_1}|$ denotes the cardinality of the parity bits P_{X_1}. The distributed-source encoder sends the corresponding parity bits P_{X_1} and the first ak bits of the information bits in X_1. Thus, the compression rate can be computed by

$$R_{X_1} = \frac{ak + |P_{X_1}|}{k}. \tag{32}$$

The same procedure with a few modification is applied to encode source X_2. For source X_2, we assume a systematic channel code of rate R_2. The source encoder sends the corresponding parity bits P_{X_2} and the remaining $(1 - a)$ fraction of the information bits in X_2. The distributed source coding procedure is demonstrated in Fig. 18.

We show that the encoding procedure in Fig. 18 can be demonstrated as bits passing through a set of parallel channels. As an example in Fig. 18, a fraction $(1 - a)$ of the information bits in X_2 and the parity bits P_{X_2} are transmitted through the wireless channel,

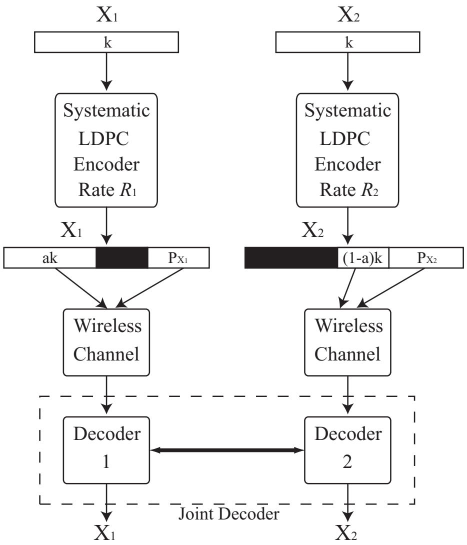

Figure 18: Distributed source coding of two correlated sources at an arbitrary rate.

and the remaining a fraction of the information bits are sent through a BSC with crossover probability p, which models the correlation between the sources X_1 and X_2. As we can see, bits are sent over two different independent channels. This process is shown in Fig. 19, where $n_2 = k + |P_{X_2}|$. Wireless channel is assumed to be an ideal channel unless indicated otherwise. The following lemma computes P_{X_1} and P_{X_2} using the parallel channel model in Fig. 19.

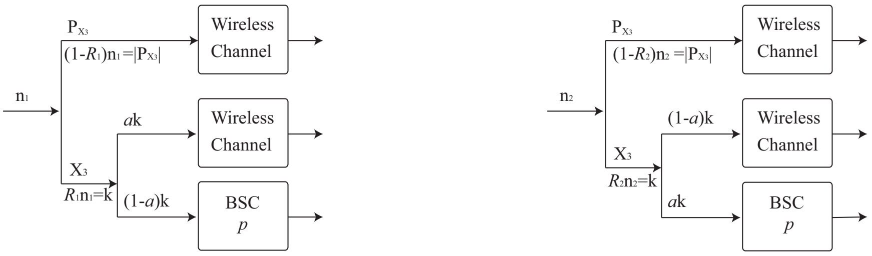

Figure 19: Parallel channel model for distributed source coding of two sources (n_i is the length of the LDPC code): source X_i.

Lemma 9. *Let P_{X_1} and P_{X_2} be the parity bits generated by the encoders of Fig. 18. Then, to recover both sources with low bit error rates at the decoder, the number of the parity bits $|P_{X_1}|$ and $|P_{X_2}|$ are at least equal to $(1-a)kh(p)$ and $akh(p)$, respectively.*

Proof. We prove the lemma for the source X_2. The overall channel capacity in Fig. 19b is computed from $C_{eq} = (1-a)R_2 + aR_2c(p) + (1-R_2)$, where $c(p) = 1 - h(p)$ denotes the capacity of BSC with cross over probability p. The highest code rate for this channel is $R_2 \leq C_{eq}$, that results in $R_2 = \frac{k}{k+|P_{X_2}|} \leq \frac{1}{1+ah(p)}$. This implies $|P_{X_2}| \geq akh(p)$. □

The procedure we discussed above involves two channel codes. However, we are interested in a method that distributed source codes two sources at every valid rate on the Slepian-Wolf rate region using a single channel code. First, we need to identify the rate of such a code that is capable of encoding all sources. The rate of the LDPC code in Fig. 19b is $R_2 = \frac{1}{1+ah(p)}$. As we can see R_2 increases from $R = \frac{1}{1+h(p)}$ to 1 as a varies from one to zero. Similarly, the rate of the LDPC code for X_1 is $R_1 = \frac{1}{1+(1-a)h(p)}$ that decreases from 1 to R as a varies from one to zero. The range of the rates that needs to be covered for both sources are identical. Therefore, for code design, it is sufficient to consider only one of the sources and the other one will be automatically covered.

Using a rate-compatible code, we design a parent code for the lowest rate, R. Puncturing the parity bits of the parent code, higher rates are achievable. The parent code corresponds to the case that no information bits but all parity bits are sent ($a = 1$). The corresponding parallel channel model is demonstrated in Fig. 20. For this set of parallel channels, we design a parent code whose performance, ideally, does not degrade by puncturing (here, by performance we mean the ratio $\frac{R}{C}$, where C is the capacity of the channel and R is the maximum rate of the code for which the error probability is less than a required value).

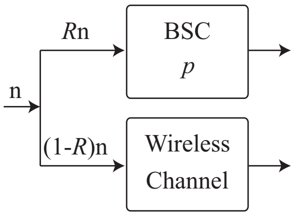

Figure 20: An equivalent model for the channel of Fig. 19 when $a = 1$.

The resulting distributed source coding for source X_2 at every arbitrary rate is demonstrated by Fig. 21. Source X_2 is fed into a systematic channel code of rate $R = \frac{1}{1+h(p)}$, fraction $(1-a)$ of the information bits in X_2 along with the fraction a of the generated parity bits are transmitted. The same procedure is applied to X_1 with the exception that the remaining fraction of information bits in X_1 and parity bits in P are transmitted. The

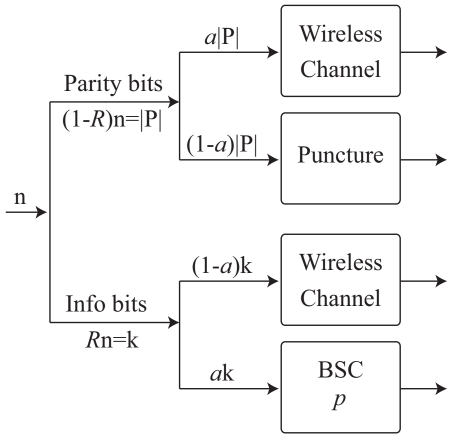

Figure 21: Parallel channel model for distributed source coding of two sources (n in the length of the LDPC code): source X_2.

corresponding compression rates for X_1 and X_2 are equal to $R_{X_1} = a + (1-a)h(p)$ and $R_{X_2} = (1-a) + ah(p)$, respectively. It can be easily verified that these compression rates satisfy (1) with equality. By changing the value of a, every valid rate on the Slepian-Wolf rate region is achievable by using a single systematic channel code. The total compression rate for sources X_1 and X_2 is at least $1 + h(p)$. We note that although we considered Bernoulli sources with parameter 1/2, a similar method can be used for other choices of the Bernoulli parameter Q. It suffices to modify the scheme such that the total compression rate be at least $h(Q) + h(p)$.

The following lemma asserts that if the parent code has performance very close to the overall capacity of the channel in Fig. 20, then the distributed source coding at every arbitrary rate on the Slepian-Wolf limit approaches its theoretical limit. More precisely, the lemma shows that by using a single systematic rate-compatible parent code that approaches the Shannon limit, not only every arbitrary compression rate is achievable, but the compression rates are very close to the Slepian-Wolf limits.

Lemma 10. *If a rate-compatible channel code of parent rate R achieves the capacity of the channel in Fig. 20, then it also approaches the capacity of the channel in Fig. 21.*

Proof. Let R and R_2 denote the rates of the codes and C and C_p be the overall capacities of the channels in Fig. 20 and Fig. 21, respectively. It can be easily verified that R/C and R_2/C_p are identical. Using [49], we are assured that the code for Fig. 20 also performs well for channels in Fig. 21. □

We conclude that, if the parent code reaches the overall capacity of the channels in Fig. 20, then the punctured parent code will achieve the capacity of the Fig. 21. Thus, the compression rate reaches the Slepian-Wolf limit. A long-length LDPC code performs very close to the Shannon capacity and is resistant to random puncturing. Therefore, the scheme in Fig. 21 with an infinite-length LDPC code performs close to the theoretical limit. However, as stated earlier, we are interested in applications such as sensor networks where the length of the code is finite. For finite-length LDPC codes, there is a performance loss due to puncturing. In the following section, we study the design procedure for finite-length rate-compatible LDPC codes that result in performance close to the Slepian-Wolf limit.

4.2.1 Encoding

In the design procedure, we consider the fact that not all data bits have equal importance, because certain bits may need a higher level of protection against error than the other parts. In [53], the authors showed that unequal error protection using LDPC codes can be utilized for such applications. Information bits are the most important bits, i.e., the parity bits are only used to recover the information bits and their bit value are not of our interest in the systematic LDPC code. Therefore, we divide the codeword bits into two groups and consider two levels of protection. One group consists of information bits which are the most important bits (MIB). The other group contains the parity bits which are the least important bits (LIB). Thus, we design a systematic parent rate-compatible LDPC code of rate R with distinct variable node degree distributions for its information bits and parity bits.

We introduce an ensemble $g(\Lambda, \rho)$ of bipartite graphs for unequal error protection, where $\Lambda(x)$ and $\rho(x)$ denotes the degree distribution for variable nodes and check nodes, respectively. For unequal error protection, we use two sets of degree distributions $\lambda^1(x)$ and $\lambda^2(x)$ correspond to information bits and parity bits, respectively. A random realization of H is generated based on the degree distribution $(\Lambda(x), \rho(x))$. Using Gaussian elimination, the parity-check matrix H' for the equivalent systematic LDPC code is constructed. Matrix

H' is used to find the generator matrix G for the systematic LDPC code. For encoding a sequence, we compute the parity bits of this sequence using the matrix G. The same matrix G is used to encode both sources X_1 and X_2.

4.2.2 Decoding

The decoder needs to determine a length-n sequence of X_1 from its available parity bits P_{X_1}, its partially available information bits, and its correlated sequence X_2. As shown in Fig. 21, the decoder has three sets of bits available: the error free set which is the fraction $(1-a)$ of information bits in X_2 and fraction a of parity bits, punctured set which is the fraction $(1-a)$ of parity bits, and the noisy set which is the fraction a of information bits in X_1. In order to apply the message passing algorithm, LLR values of all bits are required to be known. LLRs of the fraction of the bits that passes through the ideal channel are infinity, the LLR's of the fraction that passes through the BSC are $\ln(\frac{1-p}{p})$, and the punctured bits have zero LLRs. Then, the message passing algorithm can be executed to decode the source message X_2. As we can see, the fraction $(1-a)$ of the parity bits in X_2 is considered to be punctured. However, the same fraction of the parity bits in X_1 is available at the decoder error free. We note that we do not use these bits of X_1 in the decoding process of X_2. To explain the rationale, we need to find the correlation probability between the parity bits of sources X_1 and X_2 given that the correlation probability between information bits in X_1 and X_2 is p. More precisely, we need to compute $P[P_{X_2} \neq P_{X_1}|P_{X_1}]$ subject to $P[X_2 \neq X_1|X_1] = p$. It can be verified that the quantity is given by

$$P[P_{X_2} \neq P_{X_1}|P_{X_1}] = \frac{1}{2} - \frac{1}{2}(1-2p)^k,$$

where k is the average degree of the generator matrix. Since the correlation probability is very close to 1/2, the missing fraction of the parity bits in X_2 may be viewed as punctured bits. The source X_1 is decoded similarly. We note that, the decoders communicate to each other to exchange some of the information bits prior to decoding. However, thereafter, each performs the decoding independently. Therefore, errors in one decoder does not affect the other one.

In this section, we investigated the problem of distributed source coding of two correlated sources at any arbitrary rate on the Slepian-Wolf limit. In the following section, we extend our method to three or more correlated sources and we show how our method exploits the correlation between all sources simultaneously as opposed to the previous work that only considers the pair wise correlation, separately.

4.3 Distributed Source Coding of Three Sources at Arbitrary Rates

Consider a communication system of three sources X_1, X_2, and X_3 each of which is i.i.d. binary sequences of length k. Sources X_1, X_2 and X_3 are statistically dependent to each other such that $P[X_k \neq X_i | X_i] = p_{ik}$, $\forall i, k \in \{1, 2, 3\}$ and $i \neq k$. The correlation between source X_i and X_k, while X_j is present at the decoder is modeled with $Q_{ij}(k)$ as shown in Fig. 12.

The source coding of X_i is done as follows: source X_i is fed into a systematic LDPC code of rate R_i. The distributed source encoder sends the corresponding parity bits along with only a_i fraction of the information bits in X_i. This results in compression rate of $R_{X_i} = \frac{a_i k + P_i}{k}$, where P_{X_i} is the number of the parity bits associated with the LDPC code of rate R_i. By changing the values of a_is, every point on the Slepian-Wolf rate region is achievable by this method. To apply the distributed source coding of three sources by using a single systematic LDPC code, we need to know the rate of the parent LDPC code. Then, arbitrary compression rate on the Slepian-Wolf limit is achieved by puncturing specific fraction of the parity bits of the parent code. In order to know the rate of the parent LDPC code we need to know the number of the required parity bits. As stated earlier, source X_i needs P_{X_i} parity bits. The following lemma shows that the total number of the required parity bits $|P| = |P_{X_1}| + |P_{X_2}| + |P_{X_3}|$ is fixed and is independent of the values of parameter a_is.

Lemma 11. *The total number of the parity bits required for source coding of three sources with the method explained above is at least equal to $k(h(X_1, X_2, X_3) - 1)$.*

Proof. According to Slepian-Wolf theorem, the lowest compression rates for three sources must satisfy the following:

$$R_{X_1} + R_{X_2} + R_{X_3} = h(X_1, X_2, X_3).$$

Using the chain rule, we have

$$\frac{a_1k + P_{X_1} + a_2k + P_{X_2} + a_3k + P_{X_3}}{k} = 1 + h(X_2|X_1) + h(X_3|X_2, X_1).$$

Thus, we have $P = P_{X_1} + P_{X_2} + P_{X_3} = k(h(X_1, X_2, X_3) - 1)$. We note that P depends only on the correlation probabilities of sources. The rate of the required LDPC code is computed from $R = \frac{1}{h(X_1,X_2,X_3)}$. □

Similar to the two-source case, we use a parallel channel model for sources to find the value of P_{X_i} with respect to P. The model is obtained as follows: At the receiver, the bitstreams of three sources need to be combined for full recovery. The decoder of X_1 has a_1k information bits and P_{X_1} parity bits perfectly. To construct the whole sequence, the decoder tries to recover the remaining bits from the fraction a_2 of the information bits of X_2 and the fraction a_3 of the information bits of X_3 which are available at the decoder. The decoder considers those bits as the output of BSC with crossover probability p_{12} and p_{13}, respectively. Therefore, the decoder would access parity bits P_{X_1} and three sets of information bits: error free bits and the bits passed through BSC with cross over probability p_{12} and p_{13}. The decoding procedure of the source X_2 is different from that of source X_1. In addition to parity bits P_{X_2}, the decoder has fraction a_2 of the information bits error free. For the fraction a_1 of the information bits of X_2, the decoder considers those bits of source X_1 (that is available error free) as the output of the BSC with cross over probability p_{12}. For the fraction a_3 of the information bits, the procedure is a little bit different. For this set of bits, we have the values of both sources X_1 and X_3 perfectly. Therefore, the fraction a_3 of the information bits of X_2 are received from the output of the channel Q_{13}. Therefore, to recover the source X_2, the decoder uses the parity bits and three different sets of information bits: error free bits, bits passed through BSC with cross over probability p_{12}, and bits passed through the channel Q_{13}. To recover the source X_3, the decoder has the fraction a_3 of the information bits error free. The decoder utilizes the fraction $(1 - a_3)$

of the information bits of both X_1 and X_2. The decoder considers those bits of X_3 as the output of channel Q_{12}. Therefore, the decoder has two sets of information bits: error free bits and the bits passed through channel Q_{12}. As an example, distributed source coding of source X_2 can be demonstrated in Fig. 22b.

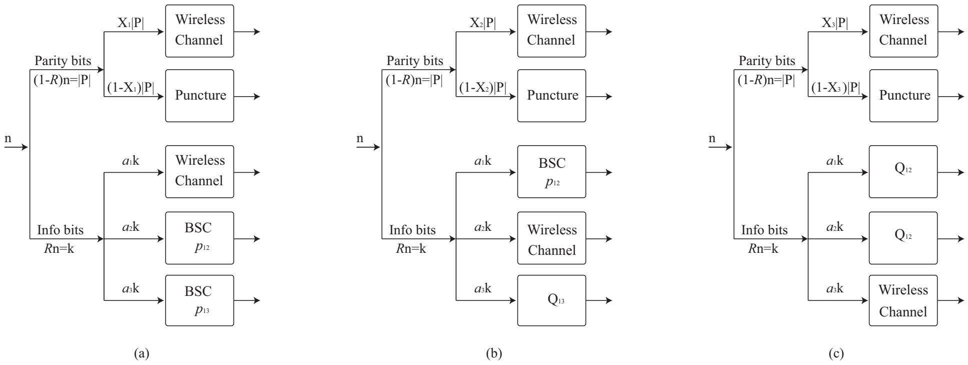

Figure 22: Parallel channel model for distributed source coding of three correlated sources: (a) source X_1, (b) source X_2, and (c) source X_3.

The following lemma indicates the number of the required parity bits for each source based on the value of a_is.

Lemma 12. *For distributed source coding of X_1, X_2, and X_3 as described in Section 4.3, we must have $|P_{X_1}| \geq k(a_2 h(p_{12}) + a_3 h(p_{13}))$, $|P_{X_2}| \geq k(a_1 h(p_{12}) + a_3 h(Q_{13}))$, and $|P_{X_3}| \geq k(1 - a_3)h(Q_{12})$, where $h(p_{ij}) = h(X_j|X_i)$ and $h(Q_{ij}) = h(X_k|X_i, X_j)$.*

Proof. The proof for P_{X_1} and P_{X_3} is omitted because of the similarity to Lemma 15. The overall capacity for the channel of Fig. 23 is $C_{eq} = R_2(a_1 c(p_{12}) + a_2 + a_3 c_{Q_{13}}) + (1 - R_2)$, where $c_{Q_{13}}$ is the capacity of the channel Q_{13}. The highest code rate for this channel is $R_2 \leq C_{eq}$ which results in the code rate of $R_2 \leq \frac{1}{1+a_1 h(p_{12})+a_3 h(Q_{13})}$. This implies $|P_{X_2}| \geq k(a_1 h(p_{12}) + a_3 h(Q_{13}))$. □

Therefore, by using a systematic rate-compatible parent LDPC code of rate $R = \frac{1}{h(X_1,X_2,X_3)}$, and puncturing fraction of the parity bits, every arbitrary rate on the Slepian-Wolf rate region is achieved. To decode sources, we apply message passing algorithm to the bits observed from the channel and the bits available at the decoder. The decoder of X_i has its available parity bits P_{X_i}, its partially available information bits, and its available

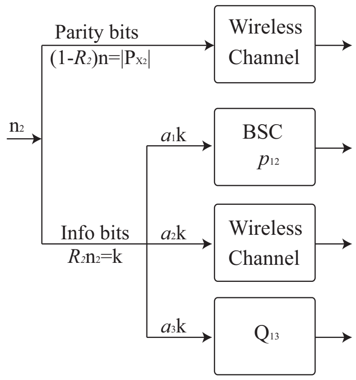

Figure 23: An equivalent model for the channel of Fig. 22b.

correlated sequences. In order to apply the message passing algorithm, LLR values of all bits are required. LLRs of the fraction of the bits that are passed through the ideal channel are infinity, the fraction that is passed through the BSC with cross over probability p_{ij} has LLR of $\ln(\frac{1-p_{ij}}{p_{ij}})$, and the fraction that passed through the channel Q_{ij} has LLR that is equal to $\ln(\frac{2-p_{ij}-p_{ik}-p_{jk}}{p_{ik}+p_{jk}-p_{ij}})$, if X_i and X_j are identical and to $\ln(\frac{p_{ij}-p_{ik}+p_{jk}}{p_{ik}-p_{jk}+p_{ij}})$ otherwise. Then, we decode X_i by applying the message passing algorithm.

4.4 Distributed Source Coding of M Sources at General Rates

Consider a communication system of M sources X_1, X_2, $\cdots$, and X_M which are identically distributed binary sequences of length k. Each bit is assumed to be zero and one with equal probability. These sources are statistically dependent to each other in which the dependency between each pair can be fully described by the conditional probability mass function $P[X_i|X_j] = p_{ij}$. Let source X_i be compressed at rate R_{X_i}, then the Slepian-Wolf rate region for distributed source coding of these M sources is given by [74]:

$$R_{X_{i_1}} + R_{X_{i_2}} + \cdots + R_{X_{i_l}} \geq h(X_{i_1}, \cdots, X_{i_l}|X_{j_1}, \cdots, X_{j_{M-l}}), \tag{33}$$

where $l \leq M$, $\{i_1, i_2, \cdots, i_l\} \subseteq \{1, \cdots, M\}$ and $\{j_1, j_2, \cdots, j_{M-l}\} = \{1, \cdots, M\}/\{i_1, i_2, \cdots, i_l\}$.

Similar to three-correlated sources case, total number of the parity bits required for source coding of M sources is equal to $k(h(X_1, \cdots, X_M) - 1)$. Thus, a parent LDPC code of rate $R = \frac{1}{h(X_1, \cdots, X_M)}$ is required. Here, we consider the encoding of source X_i. The same procedure with a few modification is applied to encode the rest of the sources. After encoding by the systematic parent LDPC code of rate R, the fraction a_i of the information bits in X_i and the fraction x_i of the generated parity bits are transmitted. The fractions of the information bits in each source are chosen such that $\sum_{i=1}^{M} a_i = 1$. Moreover, the bit positions are complimentary of each other (i.e., each bit position is sent from only one encoder). By changing the value of a_is, any arbitrary rate on the Slepian-Wolf rate region can be achieved. Similar to Lemma 15 and Lemma 12, the values of x_is are determined based on the values of a_is.

At the receiver, the bitstreams of all sources need to be combined for full recovery. The decoder needs to determine a length-n sequence X_i from its partially available parity bits, $a_i k$ information bits, and its $(1 - a_i)k$ information bits in X_js, where $j \in \{1, 2, \cdots M\}$ and $j \neq i$.

4.5 Simulation Results

We give experimental results for distributed source coding of two and three correlated sources at different rates. First, we simulate the distributed source coding of two sources at an asymmetric rate ($a = 1$), a symmetric rate ($a = 1/2$), and two arbitrary rates on the Slepian-Wolf rate region ($a = 1/3$ and $a = 2/3$). Next, we provide the distributed source coding results of three sources at an asymmetric rate ($a_1 = 1$, $a_2 = 0$, and $a_3 = 0$), a symmetric rate ($a_1 = 0.28$, $a_2 = 0.34$, and $a_3 = 0.38$), and an arbitrary rate on the Slepian-Wolf region ($a_1 = 1/3$, $a_2 = 1/3$, and $a_3 = 1/3$).

The parameter p is assumed to be equal to 0.11, which is assumed to be known at the encoder. Using $p = 0.11$ and the mentioned distribution for X_1 and X_2 , we obtain the joint entropy $h(X_1, X_2) = 1.5$ and the conditional entropy $h(X_2|X_1) = h(p) = 0.5$. We design a rate-compatible LDPC code of rate $R = \frac{1}{1+h(p)} = 2/3$ with an unequal error protection property whose length is $n = 1000$. The same code is used to distributed source code a system of any number of correlated sources at any arbitrary compression rate. This code was chosen randomly from the ensemble that is defined by the following degree distribution

$$\Lambda(x) = \{\lambda^1(x), \lambda^2(x)\} = \{0.6406x + 0.0319x^6 + 0.3043x^8, 0.1784x + 0.6372x^4 + 0.1845x^6\} \tag{34}$$

$$\rho(x) = \{0.52x^{12} + 0.48x^{13}\}.$$

Here $\lambda^1(x)$ and $\lambda^2(x)$ denote the variable-node degree distribution of the parity bits and information bits, respectively. We utilize two sets of degree distribution for the variable nodes to achieve unequal error protection property. Note that $\rho(x)$ is the check-node degree distribution.

4.5.1 Two Correlated Sources

To study performance for different rates on the Slepian-Wolf rate region, we consider the following rates:

$$\begin{aligned} R_{X_1} &= 2/3,\ R_{X_2} = 5/6 \text{ (point } a \text{)} \\ R_{X_1} &= 3/4,\ R_{X_2} = 3/4 \text{ (point } b \text{)} \\ R_{X_1} &= 5/6,\ R_{X_2} = 2/3 \text{ (point } c \text{)} \end{aligned} \tag{35}$$

The above rates are taken when parameter a has values $1/3, 1/2$, and $2/3$, respectively. To achieve the compression rates in (49), puncturing is applied to the LDPC code with the degree distribution given in (34).

Figure 32 shows the results of the bit error rate averaged over the two sources X_1 and X_2 as a function of the joint entropy for the rates in (49). In addition to the rates in (49), the result of the asymmetric rate is also presented. As we can see, there is a subtle performance degradation in comparison with the asymmetric rate. For the length of 1000, convergence of the LDPC codes is achieved at $h(X_1, X_2) = 1.327$, $h(X_1, X_2) = 1.317$, and $h(X_1, X_2) = 1.317$, for the asymmetric rates, point a, and point b, respectively.

In this experiment, bit error rates (BER) lower than 10^{-5} are desired. For the length of 1000, convergence of the LDPC codes is achieved at $h(X_1, X_2) = 1.327$, $h(X_1, X_2) = 1.317$, and $h(X_1, X_2) = 1.317$, for the asymmetric rates, point a, and point b, respectively. As we can see, there is a subtle performance degradation in comparison with the asymmetric rate.

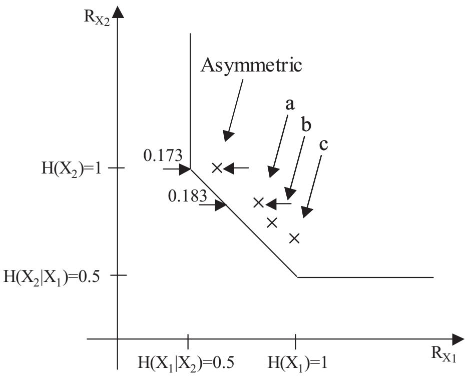

Figure 24: The gap between Slepian-Wolf limit and the convergence of the LDPC code and length 1000 for different rates of distributed source coding.

4.5.2 Three Correlated Sources

In this section, we provide simulation results for distributed source coding of three correlated sources. For simplicity, we assume sources to be correlated with the same correlation probability $p = 0.11$, which results in $R_{X_1} + R_{X_2} + R_{X_3} = h(X_1, X_2, X_3) = 1.92$. To study performance for different rates on the Slepian-Wolf rate region, we considered the following rates:

- Point a: Asymmetric rate (i.e., $R_{X_1} = 1$, $R_{X_2} = 0.5$, and $R_{X_3} = 0.42$) resulting from: $a_1 = 1$, $a_2 = 0$, and $a_3 = 0$.
- Point b: Symmetric rate (i.e., $R_{X_i} = 0.64$, for $i \in \{1, 2, 3\}$) resulting from: $a_1 = 0.28$, $a_2 = 0.34$, and $a_3 = 0.38$.
- Point c: $R_{X_1} = 0.67$, $R_{X_2} = 0.64$, and $R_{X_3} = 0.61$ resulting from: $a_i = 1/3$, for $i \in \{1, 2, 3\}$.

The following table summarizes the results for three sources at three different rates using LDPC code of length 1000. In this experiment, BER lower than 10^{-5} were desired.

Table 3: Total compression rate achieved using a rate-compatible systematic LDPC code with the unequal error protection property and length 1000.

Point	$h(X_1, X_2, X_3)$	Gap from the theoretical limit
a	1.61	0.3
b	1.59	0.32
c	1.59	0.32

The simulation results suggest that there is a very small performance degradation for points b and c in comparison with the asymmetric rate (point a). For the length of 1000, convergence of the LDPC codes for point a, point b, and point c are achieved at $h(X_1, X_2, X_3) = 1.61$, $h(X_1, X_2, X_3) = 1.59$, and $h(X_1, X_2, X_3) = 1.59$, respectively.

4.6 Conclusion

We proposed a scheme for distributed source coding of correlated sources that achieves every arbitrary rate on the Slepian-Wolf rate region. This method is based on sending a fraction of the information bits along with a fraction of the parity bits generated by a systematic LDPC code. We showed that the distributed source coding problem is simplified to designing a rate-compatible LDPC code that has unequal error protection property. We proved that there is no rate loss across the Slepian-Wolf rate region for every arbitrary compression rate. This was also confirmed by simulation results. The simulation results showed that performance improvement is achieved by our proposed scheme. First, we considered two correlated sources. Since each source is decoded independently (only part of information bits is exchanged between the decoders), the system does not suffer from the problem of error propagation. Then, we studied distributed source coding of three sources. No particular correlation model among sources is necessary for this approach. The simulation results confirm that the gap from the theoretical limit remains almost intact for different compression rates.

CHAPTER V

LOSSY DISTRIBUTED SOURCE CODING

5.1 Introduction

So far, we considered the problem of lossless distributed source coding. In many applications such as sensor networks, some distortion can be tolerated. In these cases, lossy distributed source coding results in compression with higher rates and consequently saves more energy. In this chapter, we propose a scheme for lossy distributed source coding with side information available at the decoder based on sending parity bits using LDPC codes. LDPC codes are chosen because of their good performance and their practically feasible belief propagation decoding. We provide the design procedure for the LDPC code that guarantees performance close to the Wyner-Ziv limit for long-length LDPC codes. Simulation results are obtained for short-length sequences.

Wyner and Ziv studied the lossy distributed source coding with side information available at the decoder [76]. This is a generalization to the setup of Slepian-Wolf theorem [69] in that coding is with respect to a fidelity criterion rather than lossless. A theoretical study of this problem in the binary case was recently presented by Shamai, Verdu and Zamir [67]. The work can be viewed as an extension of Wyner's scheme to the lossy case. The authors suggested to use nested linear codes and send the corresponding syndromes. They showed that lossy distributed source coding using a set of good linear codes can perform close to the Wyner-Ziv limit. The authors did not provide any practical designs for nested codes. Liveris *et al.* provide a practical designs for nested codes using convolutional codes [40]. Our proposed lossy distributed source coding is based on sending parity bits that is a substantial diversion over the previous work which is based on syndromes [67]. As mentioned in Chapter 4, in applications where wireless channels is not error free, the syndrome-based method cannot be applied for distributed source coding, while the parity-based method can be used.

5.2 Lossy Distributed Source Coding

Wyner and Ziv were the first who studied the lossy distributed source coding . The result of their work is Wyner-Ziv theorem that is a generalization to the Slepian-Wolf theorem and provides a rate-distortion function for lossy distributed source coding of two sources. Consider a communication system of two statistically Bernoulli sequences X and Y (with parameter 0.5). The dependency between these two sources can be fully explained by their conditional probability mass function $P[X|Y \neq X] = p$. In lossy distributed source coding with side information available at the decoder , one of the signals, X, is compressed conventionally and recovered perfectly at the decoder, while the other signal, Y, is compressed at a rate close to the Wyner-Ziv theoretical limit $R_{WZ}(d)$. The goal is to reproduce the source Y by Y_d at the receiver such that $\rho(Y, Y_d)$ does not exceed d, where $\rho(Y, Y_d)$ is the distortion measure between Y and its recovered sequence Y_d. It is convention to use Euclidean distance as distortion measure. However, instead, the Hamming distance can be used for binary sources. Rate-distortion function $R_{WZ}(d)$ is defined as [76]

$$R_{WZ}(d) = \begin{cases} g(d) & 0 \leq d \leq d_c, \\ g(d_c)\left(1 - \frac{d-d_c}{p-d_c}\right) & d_c \leq d \leq p, \end{cases}$$

where

$$g(d) = \begin{cases} h(p * d) - h(d) & 0 \leq d < p, \\ 0 & d = p, \end{cases}$$

$h(x)$ is the entropy function of x, $p * d \triangleq p(1-d) + (1-p)d$, and d_c is the solution to $\frac{g(d_c)}{d_c - p} = g'(d_c)$. In Fig. 25, $R_{WZ}(d)$ is plotted for $p = 0.27$. In this figure $R_{Y|X}(d)$ is also plotted, where $R_{Y|X}(d)$ denotes the rate-distortion function of lossy coding of Y given X available at both the encoder and the decoder. $R_{Y|X}(d)$ is computed from

$$R_{Y|X}(d) = \begin{cases} h(p) - h(d) & 0 \leq d \leq \min\{p, 1-p\}, \\ 0 & d > \min\{p, 1-p\}. \end{cases}$$

As we can see, unlike the Slepian-Wolf region, the Wyner-Ziv function suffers a loss in rate relative to the case, where both the encoder and the decoder have access to X.

Lossy source coding of Y with length k and the side information X available at the decoder is studied in [67]. Their method is based on nested linear codes. Two linear codes

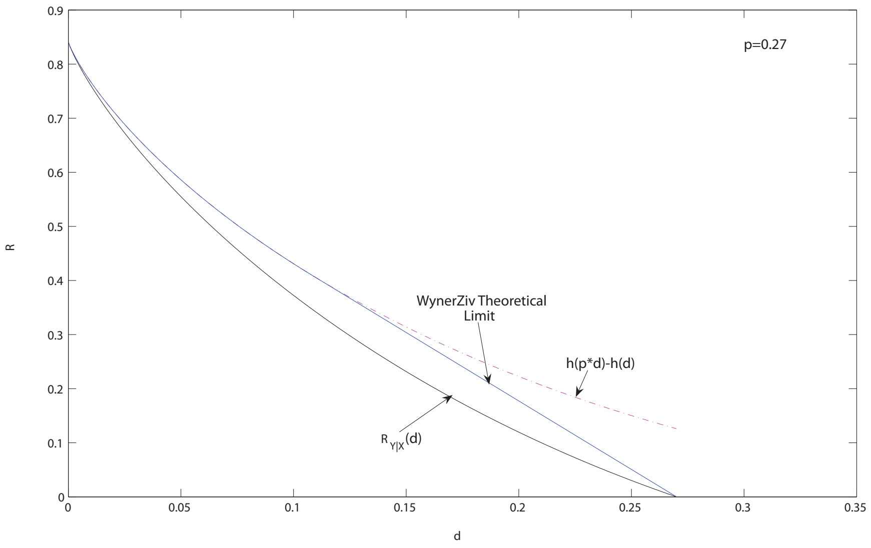

Figure 25: Wyner-Ziv limit and $R_{Y|X}(d)$ for $p = 0.27$ as a function d.

defined by parity-check matrices $H_{1(m_1 \times n)}$ and $H_{2(m_2 \times n)}$ are chosen such that $\frac{m_1}{n} = h(d)$ and $\frac{m_2}{n} = h(p * d)$, and code 2 is a subcode of code 1. Therefore, the parity-check matrices H_1 and H_2 satisfy the following:

$$H_2 = \left[\frac{H_1}{H_a}\right]. \tag{36}$$

Every codeword Y_d of H_1 satisfies $H_1 Y_d^T = 0$. The Wyner-Ziv encoding of two binary sources consists of the following steps:

1. Among the codewords of H_1, select Y_d which is the closest to Y in the Hamming distance.

2. $H_a Y_d^T$ is sent to the receiver.

The compression rate of the Wyner-Ziv encoder is $R = \frac{m_2 - m_1}{n} = h(p * d) - h(d)$ as desired. The decoder receives $H_a Y_d^T$ and has X as the side information. Since Y_d is a codeword of H_1, $H_1 Y_d^T = 0$ and

$$H_2 Y_d^T = \left[\frac{0}{H_a Y_d^T}\right]. \tag{37}$$

We know that $Y_d = Y + N_1 = X + N_2 + N_1$, where N_1 and N_2 are Bernoulli(d) and Bernoulli(p), respectively. Therefore, $Y_d = X \oplus N$, where $N = N_1 + N_2$ is Bernoulli with parameter $p*d$. Using a set of good linear codes, Y_d is recovered using X and the syndromes of (37). In [67], the authors show that lossy distributed source coding with side information available at the decoder using a set of good linear codes can perform close to Wyner-Ziv limit. However, the authors did not provide any practical designs for these nested codes. In [40], the authors used nested convolutional codes whose performance was 0.09 bits away from the theoretical limit for code of length 5×10^5.

5.2.1 Proposed Method for Lossy Distributed Source Coding

In this section, we introduce a method for lossy distributed source coding based on parity bits using low-density parity-check (LDPC) codes [63]. LDPC codes are chosen because of their good performance and their practically feasible belief propagation decoding. We provide the design procedure for the LDPC code that guarantees performance close to the Wyner-Ziv limit for long-length LDPC codes. We show that there exists an LDPC code with specific column weight that attains the rate-distortion function asymptotically.

Lossy coding of Y with length k and the side information X available at the decoder is done as follows: First, Y is mapped to a sequence Y_d, where $\rho(Y, Y_d) < d$. The mapping of Y to Y_d will be discussed later in details. Then, Y_d is fed into a rate R systematic channel encoder. At the output of the encoder, we only send a fraction of the generated parity bits. This results in compression rate of less than $\frac{1}{R} - 1$.

Suppose the generator matrix of the channel code is $G = [I|P_1|P_2]$, where I is the identity matrix of dimension $k \times k$, and P_1 and P_2 are matrices of dimensions $k \times kh(d)$ and $k \times k(h(p * d) - h(d))$, respectively. For signal Y_d, the corresponding codeword is $[Y_d|Y_dP_1|Y_dP_2]$. The Wyner-Ziv encoding of the binary source Y consists of two steps:

1. Among the codewords associated with P_1^T, select Y_d which is the closest to Y in Hamming distance.

2. Y_d is fed into the systematic generator and the corresponding parity bits are sent to the receiver. The generated parity bits are $[Y_dP_1 = 0|Y_dP_2]$. We only need to send parity bits Y_dP_2.

Lemma 13. *The procedure explained above results in compression rate of $h(p * d) - h(d)$.*

Proof. The correlation between Y_d and X can be modeled by a BSC with crossover probability $p * d$. As shown in Chapter 3, for lossless distributed source coding of Y_d with side information X available at the decoder, a code of rate $\frac{1}{1+h(p*d)}$ is required. Therefore, the length of the generated parity bits is $kh(p * d)$, from which $kh(d)$ bits are zero. Therefore, $k(h(p * d) - h(d))$ parity bits are adequate to be sent, which results in compression rate of $h(p * d) - h(d)$. □

Now, we need to explain how Y_d can be generated from Y. In other words, we need to determine Y_d as a codeword of the parity-check matrix P_1^T such that $\rho(Y, Y_d) < d$. The connection between Y and Y_d can be modeled by a BSC with crossover probability d, as shown in Fig. 26. Applying the decoding algorithm to Y, the desired Y_d is generated. Having Y_d, the parity bits Y_dP_2 are generated and sent to the decoder. Lossy source coding of Y with side information X is shown in Fig. 27, where the solid line shows the bits that are sent to the decoder and dashed lines show the bits that are not sent. The decoder determines the sequence Y_d using parity bits and the side information available at the decoder. As stated before, we assume that the wireless channel is an ideal channel.

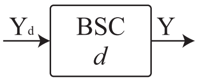

Figure 26: Correlation between Y and its distorted version Y_d.

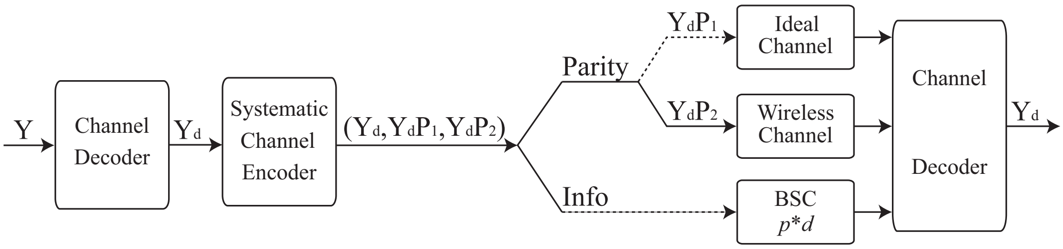

Figure 27: Lossy source coding of Y with side information X at the decoder.

5.2.2 Code Design

In this section, we study the design procedure of the systematic LDPC code that achieves the Wyner-Ziv theoretical limit. The parity-check matrix associated with the generator matrix G is of the form

$$H = \left[\frac{P_1^T}{P_2^T} \middle| I \right]. \tag{38}$$

First, we design the equivalent LDPC code with parity-check matrix $H = \left[\frac{C_1}{C_2}\middle|C_3\right]$. Then, we use Gaussian elimination to derive an equivalent parity-check matrix in the systematic form of (38). The matrix H should be designed such that the following two conditions are satisfied:

1. C_1 must be designed such that Y can be mapped to Y_d.
2. The matrix H must be designed such that Y_d can be recovered from the available parity bits and the correlated source X.

We use the MacKay LDPC code to design H. MacKay's ensemble is an ensemble of LDPC codes with $m \times n$ parity-check matrices, each is constructed as follows: for every column of H, first, set the column to all 0s and repeat the following procedure t times. Choose an index uniformly at random and independently from $\{1, 2, \ldots, m\}$ and flip the corresponding bit of the column. If an index is chosen an odd number of times in t repetitions, the corresponding element of the column becomes one, and otherwise, the element becomes zero [43]. The generated code is called (m, n, t)-LDPC codes. In this section, we show how the value of t must be chosen to ensure a lossy distributed source coding with performance close to the theoretical limit.

To design C_1, we consider the fact that the corresponding codeword is passing through BSC with crossover probability of d. Thus, a code rate of at most $1 - h(d)$ is required to ensure a reliable recovery for Y_d. We know that C_1 is parity-check matrix of a code with rate $1 - h(d)$. According to [43], given a noise distribution $P(n)$ with entropy $h(n)$ and a ratio $\lambda = \frac{1}{h(d)} < \frac{1}{h(n)}$, there exists an integer t such that for any desired error rate $\epsilon > 0$, there is an integer m such that (m, k, t)-LDPC code recovers Y_d from Y with probability of error less than ϵ. Based on this theorem we choose C_1 as (m, k, t)-LDPC code, where $m \geq kh(d)$.

To design the matrix H, we consider the fact that different bits are subject to different noise as shown in Fig. 27. Thus, $C = \left[\frac{C_1}{C_2}\right]$ and C_3 are designed differently. These codes are designed for $BSC(p*d)$ and ideal channel, respectively. The following procedure shows how t_1 and t_2 must be chosen for C and C_3 to ensure a reliable recovery for Y_d.

We have two sets of noise, n_1 with probability distribution $P(n_1)$ and length $l_1 = k$ and n_2 with probability distribution $P(n_2)$ and length $l_2 = kh(p*d)$. The typical set for these two sets of bits is defined as follows:

$$T_i = \{n \in \{0,1\}^{l_i} : \Big|\frac{1}{l_i}\log_2\frac{1}{P(n_i)} - h(n_i)\Big| \leq \eta_i\}, \tag{39}$$

where $i \in 1, 2$. The probability that decoder fails is denoted by P_{err} and bounded as follows:

$$\begin{aligned} P_{err}([C|C_3]) \leq \Big[& \sum_{n_1 \in T_1} P(n_1) \sum_{\substack{n_1' \in T_1 \\ n_1' \neq n_1}} \delta(C(n_1 - n_1') = 0 \mod 2) + \\ & \sum_{n_2 \in T_2} P(n_2) \sum_{\substack{n_2' \in T_2 \\ n_2' \neq n_2}} \delta(C_3(n_2 - n_2') = 0 \mod 2)\Big], \end{aligned} \tag{40}$$

where $\delta(s)$ denotes the indicator function which has value 1 if the statement s is true and has value 0, otherwise. We now average (40) over codes $[C|C_3]$:

$$\begin{aligned} \overline{P_{err}} \leq \Big[& \sum_{\substack{n_1, n_1' \in T_1 \\ n_1' \neq n_1}} P(n_1) \sum_{C} prob(C)\delta(C(n_1 - n_1') = 0) + \\ & \sum_{\substack{n_2, n_2' \in T_2 \\ n_2' \neq n_2}} P(n_2) \sum_{C_3} prob(C_3)\delta(C_3(n_2 - n_2') = 0)\Big]. \end{aligned} \tag{41}$$

Following the same procedure as in [43], the condition for vanishing $\overline{P_{err}}$ is

$$\exp[-2H^{-1}(h(n_i) + \eta_i')\lambda_i t_i] < \log_2[1 - \lambda_i(h(n_i) + \eta_i')], \tag{42}$$

where $\lambda_i = \frac{l_i}{kh(p*d)}$ and $\eta_i' \geq \eta_i + \frac{1}{l_i}\log_2(l_i + 1)$. The parameter t_i can be found from (42). Based on these values of t_i, matrix H is generated.

5.2.3 Simulation Results

In this section, we give experimental results for lossy distributed source coding with side information available at the decoder . The correlation p and the distortion d are assumed to be equal to 0.27 and 0.11, respectively. These parameters result in $R_{WZ}(0.11) = 0.405$.

The required LDPC code has rate $\frac{1}{1+h(p*d)} = 0.525$. Code C_1 which maps Y to Y_d has rate $R_{C_1} = 1 - h(d) = 0.5$. This code is generated from the ensemble $(250, 500, 3)$. The matrix H recovers Y_d from the parity bits received from the channel and the parity and information bits available at the decoder. For this purpose, C_3 is chosen from ensemble $(452, 452, 8)$ and C_2 is chosen such that C belongs to the ensemble $(452, 500, 5)$, considering the fact that C_1 belongs to ensemble $(250, 500, 3)$). The values for t_1 and t_2 are obtained using (42). The simulation results show that at $d = 0.063$, Y is mapped to Y_d such that on the average $\rho(Y - Y_d) < d$. Moreover, our distributed source code can recover Y_d at $p = 0.14$ and $d = 0.063$ with negligible error. In other words, our scheme performs 0.2 bits away from the Wyner-Ziv theoretical limit for the LDPC code of length 952. We also run the simulation results for LDPC code of length 1905. The gap from the theoretical limit decreases to 0.18. We expect that if longer LDPC codes are used, the performance would approach to the theoretical limit.

5.3 Conclusion

We proposed a scheme for lossy distributed source coding with side information available at the decoder using LDPC codes. Our proposed lossy distributed source coding is based on sending parity bits that is a substantial diversion over the previous work which is based on syndromes. The design methodology for the LDPC code that guarantees performance close to the Wyner-Ziv limit for long-length LDPC codes is provided. Simulation results for short-length sequences are obtained.

CHAPTER VI

ADDITIONAL PROBLEMS RELATED TO DISTRIBUTED SOURCE CODING

6.1 Introduction

In this chapter, we study two additional problems related to distributed source coding. So far, we assumed the correlation parameter to be known in advance. Here, we investigate the distributed source coding of correlated sources, when there is no prior knowledge about the correlation parameter at the time of code design (prior to the network deployment). Although, the emphasize in on lossless distributed source coding, the proposed method can be easily extended to lossy distributed source coding. Instead of designing the channel code for the lowest correlation, we introduce a method that involves rate adaptive channel coding. The simulation results confirm that our compression technique is superior to the design of the code for the lowest possible correlation probability, when the correlation parameter is unknown in advance.

Then, we study distributed joint source-channel coding of two sources over wireless erasure channels whose erasure probability is unknown. In this chapter, we show that rateless LT codes are more suitable for these applications than the standard block codes. All papers on distributed source coding assume that the compressed data arrives at the decoder with no error (ideal wireless channel) or passes through a noisy channel with a known noise level.

As shown in previous chapters, distributed source coding can be modeled with a set of parallel channels. Instead of designing the best LT code for the equivalent channel, we propose to design a nonuniform LT code by considering the fact that different bits are subject to different sources of noise. We also propose an improved decoding algorithm for LT codes. The proposed decoding algorithm improves the performance of the LT codes, considerably. We show that distributed source coding using nonuniform LT codes and the proposed decoding algorithm performs close to the theoretical limit. The simulation results confirm that the gap from the theoretical limit remains almost the same for any erasure probability and every arbitrary rate on the Slepian-Wolf rate region.

6.2 *Distributed Source Coding with Unknown Correlation Parameter*

In this Section, we focus on distributed source coding of two correlated sources of length k with unknown correlation property at the time of code design [63]. Although the emphasize is on lossless distributed source coding at the asymmetric rate, the proposed method can be easily applied to lossy distributed source coding with the side information available at the decoder. We assume that source X is compressed conventionally and recovered perfectly at the decoder, and source Y is compresses using a systematic channel code. According to Chapter 3, if $P[Y \neq X|X] = p$, then a code of rate $\frac{1}{1+h(p)}$ is required to generate $kh(p)$ parity bits, where the generated parity bits construct the compressed Y. As stated earlier, the value of p is unknown in advance, but it will be available after the network deployment.

One approach to solve this problem is designing the channel code for the least correlation probability, p_w. In other words, a channel code of rate $R_w = \frac{1}{1+h(p_w)}$ is designed, which generates $kh(p_w)$ parity bits. Then, after obtaining the correlation probability p, the encoder punctures $k(h(p_w) - h(p))$ parity bits and send the remaining $kh(p)$ parity bits to the receiver. This method performs weakly. In [48], the authors proved the existence of LDPC code ensemble with no performance degradation after puncturing for asymptotic cases over erasure channels. However, for finite-length codes and non-erasure channels, performance loss was reported as we puncture more bits [25]. Therefore, we propose a method that would puncture fewer bits than the first method.

6.2.1 Proposed Method for Distributed Source Coding with Unknown Correlation Parameter

We propose to design a LDPC code of rate $\frac{1}{1+t}$, where the value of t is determined shortly. As stated earlier, after deployment the correlation value between X and Y is found to be equal to p. According to the Slepian-Wolf theorem, a total number of $k(1 + h(p))$ bits is required at the receiver. Based on the value of the parameter p, one of the following cases is possible:

1. $h(p) < t$: Hence, source X is compressed conventionally. For the source Y, we send $kh(p)$ parity bits to the receiver and puncture the remaining $k(t - h(p))$ parity bits. This procedure is demonstrated in Fig. 28

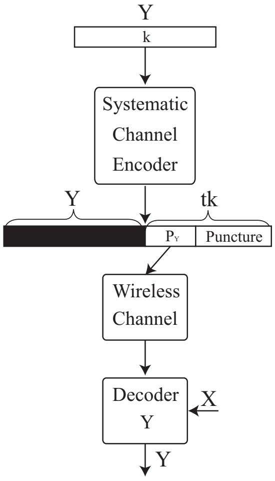

Figure 28: Distributed source coding with unknown correlation parameter-case 1.

2. $h(p) > t$: This case will result in distributed source coding at rates other than the corner point. Since the required number of the parity bits is not available, we compensate by sending a fraction of the information bits. In other words, a fraction of the information bits along with the required number of the parity bits are sent. The other source sends the complementary $(1-a)$ fraction of the information bits with the required parity bits. As shown in Chapter 4, $k(1 - a)h(p)$ and $kah(p)$ parity bits are required to recover both sources at the receiver, respectively. Therefore, the following two inequalities must be satisfied:

$$\begin{cases} (1-a)h(p) < t, \\ ah(p) < t. \end{cases}$$

Solving these inequalities results in $h(p) < 2t$. Since $\max(h(p)) = h(p_w)$, t must satisfy $t > \frac{h(p_w)}{2}$. The value of the parameter a can be chosen such that $(1-a)h(p) = t$.

Therefore, no puncturing is required for the first source. However, for the second source, $k(t - ah(p))$ parity bits are punctured and the remaining $kah(p)$ parity bits are transmitted. This procedure is shown in Fig. 29. As we can see, the proposed method results in compressing the sources at arbitrary rates in Chapter 4.

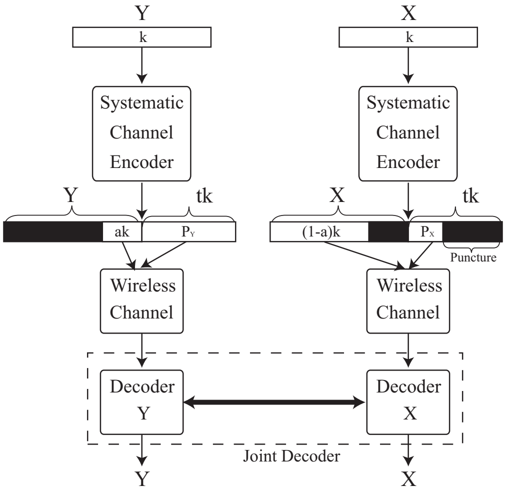

Figure 29: Distributed source coding with unknown correlation parameter-case 2.

In the following section, we compare the performance of the proposed method with the performance of the code optimized for a known value of correlation parameter. To design the channel code for this application, we use rate-compatible LDPC codes.

6.2.2 Simulation Results

We give experimental results for our proposed distributed source coding scheme, when correlation property is unknown at the time of code design and it becomes available at time of encoding. We assume that the least correlation between sources is $p_w = 0.2$. The first method, which is designed for the worst case, requires an LDPC code of rate $R_W = \frac{1}{1.72} = 0.58$. Our proposed scheme requires an LDPC code of rate $\frac{1}{1.36} = 0.735$. Suppose the actual correlation between sources is $p = 0.11$. The simulation results show that the first method and the proposed method are 0.24 and 0.195 bits away from the Slepian-Wolf limit. As we can see, our method performs better than the LDPC code designed for the worst case scenario. We also give the simulation results of the code optimized for the correlation parameter, i.e., the correlation parameter is known at the time of code design.

The following table summarizes these results.

Table 4: Gap from the Slepian-Wolf Limit using LDPC code of length 1000 with different methods for unknown/known correlation parameter.

	First Method	Proposed Method	DSC with Known p
Gap from the Slepian-Wolf Limit	0.24	0.195	0.18 ([61])

6.3 LT codes

Rateless codes are a new class of codes that have been invented recently. LT codes [41], raptor codes [68], and Online codes [44] are examples of such codes. The idea behind the rateless codes is that every receiver continues collecting the encoded data until decoding can be finished successfully. Unlike the traditional codes, rateless codes on lossy channels do not assume any knowledge about the channel. Therefore, rateless codes are very suitable candidates in the applications that the channel erasure probability is unknown, nonuniform, or time-varying. A rateless code produces a potentially limitless stream of output symbols for a given set of k input symbols. The receiver collects the output bits of the encoder from the channel. If the channel is erasure, then decoder collects any set of n output symbols, where $n = k(1+\gamma)$ and γ is a small number representing the coding overhead. However, if the channel is of any other types of symmetric channels, the receiver collects bits until the sum of the information of the individual bits is $k(1+\gamma)$ [17]. As stated in Chapter 3, distributed source coding can be modeled with a set of parallel channels consisting of binary symmetric channels with cross over probabilities varying from 0 to 1/2 and an erasure wireless channel with erasure probability ranging from 0 to 1/2. In this case, we have $n \propto \frac{k}{C_{eq}}$ where C_{eq} denotes the overall capacity for the set of parallel channels. It was shown that rateless codes have very simple encoding and decoding algorithms. Asymptotically good degree distributions for them were also developed in [41, 44, 68].

We briefly review the traditional construction of LT codes invented by [41]. Suppose we want to transmit a information sequence $M = (m_1, \ldots, m_k)$. Each output symbol is generated independently and randomly by first sampling from the distribution $\Omega(x) = \sum_{i=1}^{k} \Omega_i x^i$ to obtain a weight ω between 1 and k. Then, a vector $(v_1, \cdots, v_k)$ of weight ω is chosen uniformly at random with probability $\frac{\Omega_\omega}{\binom{k}{\omega}}$. The value of the output symbol is then calculated as $\sum_i v_i m_i$ (rateless code is defined by parameters $(k, \Omega(x))$. This procedure can also be demonstrated by matrix multiplication. The generator matrix $G_{n\times k}$ is constructed from row vectors v, where a row vector v is obtained by sampling independently from the distribution $\Omega(x)$ and the output symbols are calculated from $mG^T = y$.

6.3.1 Nonuniform LT Codes

To design the LT code for distributed source coding , we introduce the degree distribution $\Upsilon(x)$. Suppose the length of the codeword to be n. Each codeword is transmitted over a set of channels such that n_i bits from every codeword is passed through the i^{th} channel. We define $\Upsilon = \{\Omega^{(i)}(x) : i = 1, \ldots, s\}$, where $\Omega^{(i)}(x)$ is the output degree distribution for output node of type i. In other words, we propose to use the prior knowledge about which bits are transmitting over each channel in the design procedure. We use Gaussian approximation to find the degree distribution $\Upsilon(x)$. Let $m_{i,o}^{(l)}$ denotes the message send from an input node to its incident output node in the l^{th} iteration of the message passing algorithm. Also, let $m_{o,i}^{(l),(j)}$ denotes the message that is sent from an output node of type j to its incident input node. The edge degree distribution of the decoding graph for the output nodes of type j and the input nodes is given by $\omega^{(j)}(x) = \sum_d \omega_d^{(j)} x^{d-1}$ and $\nu(x) = \sum_d \nu_d x^{d-1}$, respectively. Note that we can approximate $\nu(x)$ by $\mathrm{e}^{\alpha(x-1)}$, where α is the average degree of the input nodes [17]. Assume that $E[\tanh(\frac{Z_o^{(j)}}{2})] = s^{(j)}$, where $Z_o^{(j)}$ denotes log-likelihood ratios (LLR) corresponding to the output bit of type j. Following the steps of [17], it can be shown that the mean of a message sent from an output node of type j at iteration $l+1$ is given by

$$E[m_{o,i}^{(l+1),(j)}] = \sum_b \omega_b^{(j)} \varphi^{-1}\Bigg(1 - s^{(j)}\Big(1 - \sum_d \nu_d \varphi\big((d-1)E[m_{o,i}^{(l)}]\big)\Big)^{b-1}\Bigg),$$

and

$$E[m_{o,i}^{(l+1)}] = \sum_j q^{(j)} E[m_{o,i}^{(l+1),(j)}],$$

where $q^{(j)} = \frac{n_j}{n}$ and $\varphi(x)$ is defined as

$$\varphi(x) = \begin{cases} 1 - \frac{1}{2\sqrt{\pi x}} \int_{-\infty}^{\infty} \tanh(\frac{u}{2}) \mathrm{e}^{-\frac{(u-x)^2}{4x}} \mathrm{d}u, & \text{if } x > 0, \\ 1, & \text{if } x = 0. \end{cases}$$

Using the convexity of $\varphi(x)$, one can write the following expression for the update rule of $E[m_{o,i}]$:

$$\begin{aligned} E[m_{o,i}^{(l+1)}] &\geq \sum_j q^{(j)} \varphi^{-1}\Big(1 - s^{(j)} \sum_b \omega_b^{(j)} \Big(1 - \sum_d \nu_d \varphi\big((d-1)E[m_{o,i}^{(l)}]\big)\Big)^{b-1}\Big) \\ &\geq \sum_j q^{(j)} \varphi^{-1}\Big(1 - s^{(j)} \sum_b \omega_b^{(j)} \Big(1 - \sum_d \varphi\big(\alpha E[m_{o,i}^{(l)}]\big)\Big)^{b-1}\Big) \\ &= \sum_j q^{(j)} \varphi^{-1}\Big(1 - s^{(j)} \omega^{(j)} \Big(1 - \varphi\big(\alpha E[m_{o,i}^{(l)}]\big)\Big)\Big). \end{aligned}$$

Let $\mu_l = E[m_{o,i}^{(l+1)}]$. Since $\varphi(x)$ is monotically decreasing, we have

$$\varphi(\mu_{l+1}) \leq 1 - \sum_j q^{(j)} s^{(j)} \omega^{(j)} (1 - \varphi(\alpha \mu_l)).$$

Since the means can take any value between 0 and ∞, we obtain the inequality

$$\varphi(x) \leq 1 - \sum_j q^{(j)} s^{(j)} \omega^{(j)} (1 - \varphi(\alpha x)),$$

for $x \in (0, \infty)$. This also means that the derivative of the left hand side is maximized by the derivative of the right hand side at $x = 0$. Since $\varphi'(0) \neq 0$, we have

$$\alpha \sum_j q^{(j)} s^{(j)} \omega'^{(j)}(0) \geq 1.$$

Note that $\omega'^{(j)}(0) = \omega_2^{(j)} = 2\Omega^{(j)}/\beta^{(j)}$, where $\beta^{(j)}$ is the average degree of the output nodes of type j. Therefore, we obtain

$$\sum_j q^{(j)} s^{(j)} \Omega_2^{(j)} \frac{\alpha}{\beta^{(j)}} \geq 1/2.$$

The minimum value for $\frac{\alpha}{\beta^{(j)}}$ is $\frac{\sum_i q^{(i)} \beta^{(i)}}{C_{eq} \beta^{(j)}}$, where C_{eq} denotes the overall capacity of the parallel subchannels. Therefore, to have a capacity-achieving LT code, we need to have

$$\sum_j q^{(j)} s^{(i)} \Omega_2^{(j)} \frac{\sum_i q^{(i)} \beta^{(i)}}{C_{eq} \beta^{(j)}} \geq 1/2. \tag{43}$$

Using (43), we can optimize the degree distribution of the LT code for the given channels.

In Gaussian approximation, it is assumed that all messages passed at every iteration is Gaussian. A more refined Gaussian approximation is studied in [17], which assumes that only messages passed from input nodes to their incident output nodes are Gaussian. The expectation of the message sent from output nodes to the input node is obtained from

$$E[m_{o,i}] = \sum_j \sum_d q^{(j)} \omega_d^{(j)} \underbrace{2E\Big[\tanh^{-1}\big(\frac{Z_o^{(j)}}{2}\big)\tanh\big(\frac{X}{2}\big)^{d-1}\Big]}_{f_d^{(j)}(\mu)}, \tag{44}$$

where X is a random variable with a Gaussian distribution $N(\mu, 2\mu)$. The update rule for the $E[m_{i,o}]$ is computed from

$$E[m_{i,o}] = \alpha E[m_{o,i}] = \alpha\big(\sum_j \sum_d q^{(j)} \omega_d^{(j)} f_d^{(j)}(\mu)\big). \tag{45}$$

The maximum value of $\frac{\sum q^{(j)}\beta^{(j)}}{\alpha}$ is C_{eq}. The goal is to maximize

$$\big(\sum_j \frac{q^{(j)}}{\sum_d \frac{\omega_d^{(j)}}{d}}\big)/(C_{eq}\alpha) \tag{46}$$

subject to the following constraints:

$$\begin{cases} \alpha\Big(\sum_j \sum_d q^{(j)} \omega_d^{(j)} f_d^{(j)}(\mu_i)\Big) < \mu_i, & \\ \sum_d \omega_d^{(j)} = 1, & \forall j \in \{1, \cdots, s\}, \\ \omega_d^{(j)} > 0, & \forall j \in \{1, \cdots, s\},\ \forall d > 0. \end{cases}$$

The above linear programming gives a degree distribution for an LT code that performs well for the set of parallel channels.

6.3.2 Systematic LT code

As stated in Chapter 3, distributed joint source-channel coding using channel codes requires a systematic code. In this section, we study the systematic LT codes. Let us define R as a $k \times k$ matrix consisting of the invertible section of the generator matrix G. The encoding procedure of input symbols $(m_1, \ldots, m_k)$ consists of two steps. First, the inverse of the matrix R is used to transform the information symbols into intermediate symbols

$(z_1, \dots, z_k)$, i.e, $(z_1, \dots, z_k)^{\mathrm{T}} = R^{-1}(m_1, \dots, m_k)^{\mathrm{T}}$. Then, a LT code with parameters $(k, \Omega(x))$ is applied to the intermediate symbols $(z_1, \dots, z_k)$. The output of the encoder is $(y_1 = m_1, \dots, y_k = m_k, y_{k+1}, \dots, y_n)$ [17]. The following lemma provides a lower bound on the probability that the matrix G has an invertible submatrix R.

Lemma 14. *The probability that the matrix G is full rank is at least*

$$1 - \sum_{r=1}^{k-1} \binom{k}{r} \Big(\sum_d \Omega_d \sum_{i=\max(0, \frac{\lceil d-k+r \rceil}{2})}^{\min(\lfloor \frac{d}{2} \rfloor, \lfloor \frac{r}{2} \rfloor)} \frac{\binom{r}{2i}\binom{k-r}{d-2i}}{\binom{k}{d}} \Big)^n.$$

Proof. Let us define hyperplane(μ) as the set of binary vectors X such that $\sum_{i=1}^{k} a_i X_i = 0$, where $a = (a_1, a_2, \dots, a_k)$ is a binary vector of weight μ. The probability that all rows of the matrix G belong to the hyperplane(μ) for some value of μ is equal to

$$\Big(\sum_d \Omega_d \sum_{i=\max(0, \frac{\lceil d-k+r \rceil}{2})}^{\min(\lfloor \frac{d}{2} \rfloor, \lfloor \frac{r}{2} \rfloor)} \frac{\binom{r}{2i}\binom{k-r}{d-2i}}{\binom{k}{d}} \Big)^n.$$

There are $\binom{k}{r}$ of these hyperplanes. Therefore, the probability that the matrix is not full rank is at most

$$\sum_{r=1}^{k} \binom{k}{r} \Big(\sum_d \Omega_d \sum_{i=\max(0, \frac{\lceil d-k+r \rceil}{2})}^{\min(\lfloor \frac{d}{2} \rfloor, \lfloor \frac{r}{2} \rfloor)} \frac{\binom{r}{2i}\binom{k-r}{d-2i}}{\binom{k}{d}} \Big)^n.$$

□

As we can see, increasing n (i.e., increasing the overhead) increases the probability that G is full rank. This lemma gives the number of symbols needed to be generated to guarantee the existence of a full rank matrix R.

6.3.3 Improved LT Decoding

As stated before, in many practical applications including distributed source coding in sensor networks, we should use finite-length codes. However, one of the problems in LT coding is the analysis and design of finite-length codes. Although LT codes have good asymptotic performance, they do not perform very well for finite lengths. To compensate for this problem, we introduce an improved decoding algorithm that decreases the error probability while keeping the decoding fast and efficient. First, we briefly review the conventional message passing algorithm that is used for decoding LT codes over symmetric channels. The conventional message passing algorithm is applied to the bipartite graph which has n vertices (output nodes) on one side and k vertices (input nodes) on the other side. To apply the algorithm, the LLR values of all bits are required. The LLR values of the output nodes are initialized based on the observation of the channel output while the LLR values of the input nodes are initialized to zero. The algorithm proceeds in several rounds. At each round, the LLR values of the input bits are updated. The LLR value of each bit represents the reliability of that bit. The decoding process continues till all output bits are satisfied or maximum number of iterations is reached. If the message passing algorithm decodes the received word completely, we are done. If the decoding fails, our proposed algorithm attempts to decode the received word.

The proposed algorithm is based on guessing the value of a few input nodes (γ nodes). Since guessing the values of some nodes are more critical than others for successful decoding, we use the following method to choose the input nodes whose values must be guessed. At the end of the unsuccessful standard message passing decoding, γ intermediate bits with the smallest reliability are marked. These bits are guessed prior to restarting the message passing decoding. Standard decoding is repeated with a modification on the LLR initialization of the guessed input nodes. The LLR values of the guessed bits that have value of one and zero are set to $+\infty$ and $-\infty$, respectively. The LLR values of the remaining input nodes are set to zero. This procedure is repeated till either decoding finishes successfully or all 2^{γ} choices are examined.

In our experiments, we have two values for number of guesses. First we guess γ input nodes and if the decoding fails, we guess ρ ($\rho > \gamma$) input nodes. The reason for having two numbers of guesses is to reduce the complexity of decoding. With high probability, the small number of guesses γ is sufficient for successful decoding. However, for few cases a greater number of guesses ρ is required. This is also confirmed by simulations. Choosing one large value for the number of guesses improves the decoding, but it also increases the decoding complexity. Therefore, we need to keep the number of guesses in a reasonable range. We believe that having two number of guesses help to have more successful decoding with low decoding complexity. Our experiments show that we can improve bit error rate by using the proposed decoding algorithm, while the increment in the decoding complexity is kept low.

In the following section, we study distributed joint source-channel coding of two correlated sources using systematic LT codes with the proposed decoding algorithm.

6.4 Distributed Joint Source-Channel Coding of Two Sources at the Corner Point

Similar to Chapters 3, 4, we consider a communication system consisting of two sources X_1 and X_2 that are independent, identically distributed binary sequences of length k. The bits of these signals are assumed to be zero and one, with equal probability. Signals X_1 and X_2 are statistically dependent to each other and the dependency between them can be fully described by the conditional probability mass function $P[X_1|X_2]$. In this chapter, the wireless channel is assumed to be an erasure channel.

First, we study distributed joint source-channel coding at the asymmetric rate, in which X_1 is compressed conventionally and sent at full rate $R_{X_1} \geq H(X_1)$ and is recovered perfectly at the decoder, and X_2 is fed into a rate R systematic channel encoder. At the output of the encoder, we only send the corresponding parity bits, P_2, of the codewords. Source coding of X_2 with side information X_1 at the decoder can be modeled with a set of parallel channels whose overall capacity is given by

$$C_{eq} = R(1 - H(p) + (1 - R)(1 - \delta), \qquad (47)$$

where $(1 - H(p))$ and $(1 - \delta)$ represent the capacity of the corerlation channel and erasure wireless channel with unknown erasure probability δ, respectively. The following lemma gives the required number of parity bits for distributed source coding of X_2.

Lemma 15. *For distributed joint source-channel coding of X_2 with side information X_1 available at the decoder with the method explained above, P_2 needs to be at least $\frac{kH(p)}{1-\delta}$.*

Proof. The highest code rate for this channel is $R \leq C_{eq}$, which results in a coding rate of $R = \frac{k}{k+P_2} \leq \frac{1}{1+H(p)/(1-\delta)}$. This implies $P_2 \geq \frac{kH(p)}{1-\delta}$. □

As stated earlier, we are dealing with wireless channels whose erasure probability is unknown. Therefore, the total number of the required parity bits is unknown a priori. One solution is to design different channel codes for each distinct value of δ. However, this method is highly complex and suboptimal. An alternative solution is to design a single channel code for the worst case which puts unnecessary burdens on the network. We propose to use LT codes for this scheme, which is optimal in the number of required parity bits.

As stated in the lemma, $\frac{kH(p)}{1-\delta}$ parity bits is required to be generated by the encoder. After passing through the wireless channel with erasure probability of δ, on the average $kH(p)$ of the parity bits are available at the decoder. Hence, LT codes in this system, the encoder generates limitless stream of parity bits. The receiver collects parity bits till it receives $kH(p)$ bits. Then, the joint source-channel decoder recovers source X_2 by using the information bits in X_1 and the collected parity bits. In then, notify the encoder to stop generating parity bits. In the following section, we extend our method to distributed source coding at any arbitrary rate.

6.5 Distributed Joint Source-Channel Coding of Two Sources at Arbitrary Rates

As stated in Chapter 4, for some applications, we are interested in distributed source coding at rates other than the corner point. The method of distributed source coding at arbitrary rate introduced in Chapter 4 needs to be modified, because of the non-ideal wireless channel whose erasure probability is unknown.

We encode source X_1 as follows: X_1 is fed into a systematic channel encoder. At the output of the encoder, only the first a fraction of the information bits in X_1 along with the corresponding parity bits P_1 are sent. The compression process of the source X_2 is similar to that of source X_1, with the modification that for the second source we send the remaining $(1-a)$ fraction of the information bits along with the parity bits P_2. These bits are passed through the wireless channel whose erasure probability is unknown. As shown later, the required number of parity bits depends on the number of the information bits in X_1 and X_2 that are not erased during the transmission and are available at the decoder.

At the receiver, the decoders need to communicate to each other for recovering both X_1 and X_2. The decoder of X_1 receives $a_1 = a(1-\delta)$ fraction of the information bits in X_1 perfectly ($a\delta$ fraction of the bits in X_1 are dropped during the transmission). To construct the whole sequence, the decoder tries to recover the remaining bits from the $a_2 = (1-a)(1-\delta)$ fraction of the information bits of X_2 which is available at the decoder and P_1 parity bits. The distributed source coding can be demonstrated by a set of parallel channels whose overall capacity is given by

$$C_{eq} = aR(1-\delta) + R(1-a)(1-\delta)c(p) + (1-R)(1-\delta), \tag{48}$$

where δ represents the erasure probability of the wireless channel. The overall capacity depends on the parameters a and δ. Since we have no prior knowledge about $delta$, we are dealing with a channel whose capacity is unknown in advance With the same argument as the one in Section 6.4, we can conclude that LT code is a suitable choice for this application.

For distributed joint source-channel coding of X_1, we send the first a fraction of the information bits in X_1 and P_1 parity bits, which are generated using the LT code. Source X_2 is coded similarly with the exception that the remaining $a_2 = (1-a)$ fraction of the information bits in X_2 along with P_2 parity bits are sent. Assume a_1 fraction of X_1 and a_2 fraction of X_2 are present at the decoder. Then, P_1 must be at least $\frac{1-(a_1+a_2)+a_2H(p)}{1-\delta}k$ to recover source X_1 with arbitrary small error rates at the decoder.

The new source coding of X_1 is done as follows: source X_1 is fed into a systematic LT code. The encoder sends the first a fraction of the information bits and limitless sequence of parity bits. The source X_2 is coded similarly with the modification that the remaining $(1-a)$ fraction of the information bits in X_2 are sent. At the receiver, the decoders communicate and discover the fraction of the information bits they have received perfectly (i.e. they know the values of a_1 and a_2). The decoder of source X_1 waits till it collects $(1-(a_1+a_2)+a_2H(p))k$ parity bits. This results in compression rate of

$$R_{X_1} = \frac{ka+P_1}{k} = a + \frac{\delta + a_2H(p)}{1-\delta}.$$

It can be verified that the required number of parity bits for source X_2 is $P_2 = \frac{(1-a_2)H(p)}{1-\delta}$.

Thus, the compression rate of the second source is equal to $R_{X_2} = (1-a) + P_2 = (1-a) + \frac{(1-a_2)H(p)}{1-\delta}$.

As we can see $R_{X_1} + R_{X_2} = \frac{1+H(p)}{1-\delta} = \frac{H(X_1,X_2)}{C_{WC}}$, where C_{WC} denotes the capacity of the wireless channel. Therefore, by changing the value of the parameter a, any point on the Slepian-Wolf rate region is achievable using a single LT code regardless of the value of the erasure probability.

In the following section, the simulation results for distributed source coding using LT codes is provided.

6.6 Simulation Results

We provide experimental results for distributed source coding using our proposed improved decoding and nonuniform LT codes. We assume that the correlation between sources is $p = 0.11$, which results in $R_{X_1} + R_{X_2} = H(X_1, X_2) = 1.5$. We also assume the length of the information bits to be $k = 666$. First, we give the simulation results for the improved decoding algorithm.

6.6.1 Improved Decoding Algorithm

First, we provide the results for distributed source coding at the corner point. In this case, the decoder needs to determine a length-k sequence of X_2 from its available parity bits P_2 and side information X_1. LLR values of the input nodes are assigned with the same method described in Section 6.3.3. LLR values of the fraction of the output bits that correspond to parity bits are infinity and LLR values of the fraction that correspond to the information bits are $\ln(\frac{1-p}{p})$. Then, by knowing the LLRs of all bits, the improved decoding algorithm can be executed to decode the source message X_2. In this experiment γ and ρ are equal to five and nine, respectively. The simulation result shows that in 75% of the cases guessing is not needed, in 23% of the cases five guesses are needed and in the remaining 2% of the cases nine guesses are needed for bit error rates (BERs) lower than 10^{-5}. This shows that in the worst case the time that improved decoding algorithm needs to decode a received word can be 18 times the average running time for a standard algorithm.

The following degree distribution is used [68]:

$$\Omega(x) = 0.008x + 0.494x^2 + 0.166x^3 + 0.073x^4 + 0.083x^5 +$$

$$0.056x^8 + 0.037x^9 + 0.056x^{19} + 0.025x^{65} + 0.003x^{66}.$$

First, we assume the wireless channel to be ideal. We allow the number of guesses to vary from zero to maximum of 12. Figure 30 shows the average number of guesses required to achieve BER lower than 10^{-4} for different values of p. As we can see, in more than 90% of cases, five guesses is sufficient while nine guesses is sufficient to achieve BER lower than 10^{-4} in more than 95% cases. Therefore, γ and ρ are chosen to be equal to five and nine, respectively.

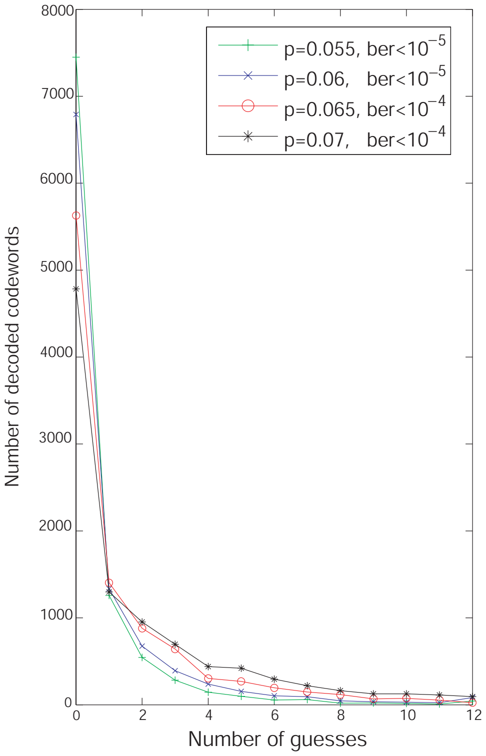

Figure 30: Required number of guesses to achieve BER lower than 10^{-4}.

Figure 31 demonstrates the results of distributed source coding using LT code at the corner point with the standard and improved decoding along with the results of [59] and [65]. As we can see, the LT code with improved decoding algorithm performs considerably better than the LT code with the standard decoding algorithm. In other words, more than one

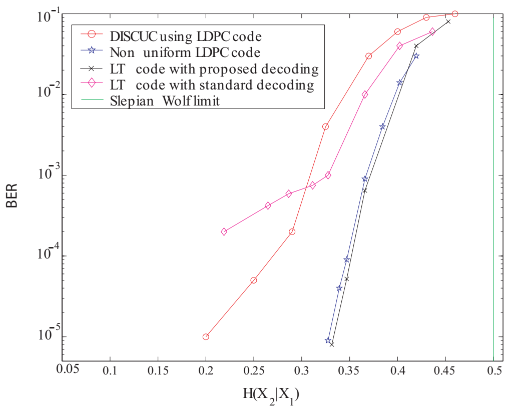

Figure 31: Comparison of distributed source coding using LT code of length 1000 with the proposed decoding algorithm and the standard algorithm at the asymmetric rate along with the results of [65] and Chapter 3.

order of magnitude improvement is achieved by using the proposed decoding algorithm compared to the conventional decoding algorithm. To check on performance variation over noisy wireless channel, we vary δ from 0 to 0.5. The simulation result shows that the gap from the theoretical limit, $\frac{H(p)}{(1-\delta)}$, remains the same. Although, distributed source coding using LDPC codes performs the same as distributed source coding using LT codes, the LT code-based distributed source coding has the additional advantage that we do not need to know the erasure probability δ.

Then, we study performance of distributed source coding algorithm for different rates on the Slepian-Wolf rate region. Compressing at the following rates are considered

$$\begin{aligned} R_{X_1} &= 2/3,\ R_{X_2} = 5/6\ \text{(point } a \text{ in Fig. 32)} \\ R_{X_1} &= 3/4,\ R_{X_2} = 3/4\ \text{(point } b \text{ in Fig. 32)} \\ R_{X_1} &= 5/6,\ R_{X_2} = 2/3\ \text{(point } c \text{ in Fig. 32).} \end{aligned} \tag{49}$$

The above rates are taken when the parameter a having the values $1/3, 1/2$, and $2/3$, respectively. Figure 32 shows the results of the bit error rate averaged over the two sources X_1 and X_2 as a function of the joint entropy for rates in (49). In this experiment, BER lower than 10^{-5} was desired. In addition to the rates in (49), the result of the asymmetric rate is also presented. As we can see, there is a small performance degradation for points a, b and c in compare with the asymmetric rate. For length 1000, convergence of the LT codes for asymmetric rate, point a, point b, and point c are achieved at $H(X_1, X_2) = 1.3353$, $H(X_1, X_2) = 1.3195$, $H(X_1, X_2) = 1.3227$, and $H(X_1, X_2) = 1.3195$, respectively.

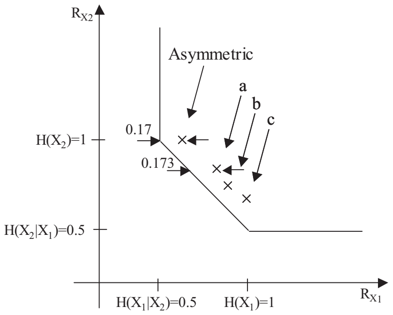

Figure 32: The gap between Slepian-Wolf limit and the convergence of the LT code of length 1000 for different rates of distributed source coding.

6.6.2 Nonuniform LT Code

Here, we provide the results for distributed source coding using nonuniform LT codes. As shown in Section 6.4, the process of distributed source coding consists of bits passing through two independent channels: information bits passed through a BSC with crossover probability of $p_1 = 0.11$ and the received parity bits passed through ideal wireless channel. Suppose we want to use an LT code of length 1000 in which 666 bits are information bits and the receiver needs to collect 334 parity bits, $q^{(1)} = 2/3$ and $q^{(2)} = 1/3$.

To reduce the complexity of the linear programming in (46), we make a few assumptions. First, we assume that the degrees are chosen from the set $D = \{1, 2, 3, 4, 10, 15, 20, 30\} \bigcup, \{50, 51, \dots, 70\}$ Since the second channel is ideal, we assume all degree-one bits are passing through this channel, i.e., $\Omega_1^{(1)} = 0$. Based on this assumption we run the linear programming and find the following degree distribution $\Upsilon(x) = \{\Omega^{(1)}(x), \Omega^{(2)}(x)\}$.

$$\Omega^{(1)}(x) = 0.07x + 0.4x^2 + 0.21x^3 + 0.16x^4 + 0.04x^{10} + 0.05x^{20} + 0.043x^{30} + 0.035x^{69},$$

$$\Omega^{(2)}(x) = 0.5x^2 + 0.17x^3 + 0.15x^4 + 0.07x^{10} + 0.05x^{15} + 0.06x^{58}.$$

Figure 33 compares the performance of the distributed source coding using the nonuniform LT code of length 1000 with the results of distributed source coding using LT code designed for the equivalent channel (a BSC with total capacity of $C_{eq} = 2/3$). As we can

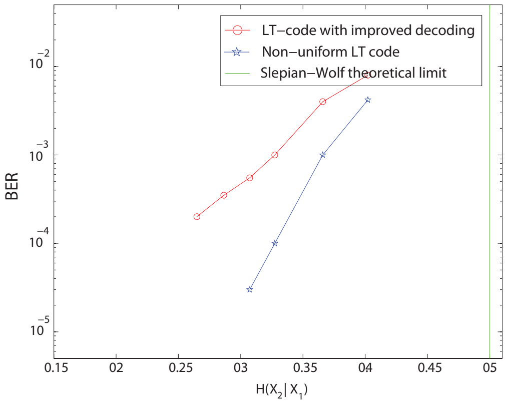

Figure 33: Comparison of distributed source coding using the nonuniform LT code and the LT code designed for the BSC with $c_{eq} = 2/3$ and the code length of 1000.

see the nonuniform LT code performs considerably better than the LT code designed for the equivalent channel.

6.7 Conclusion

First, we proposed a scheme for lossy distributed source coding with side information available at the decoder based on sending parity bits using LDPC codes. The design procedure for the LDPC code that guarantees a performance close to the Wyner-Ziv limit for long-length LDPC codes is provided. Simulation results for short-length sequences are obtained. We expect that if longer LDPC codes were used, the performance would approach the theoretical limit. Then, we studied the problem of distributed source coding of correlated sources when there is no prior knowledge about the correlation parameter. The proposed method results in compressing the sources at rates other than the corner-point rate of the Slepian-Wolf rate region. The simulation results confirmed that our compression technique is superior to the design of the code for the lowest possible correlation probability when the correlation parameter is unknown in advance. Finally, we proposed a scheme for distributed joint source-channel coding of correlated sources at every arbitrary rate on the Slepian-Wolf rate region when the erasure probability of the wireless channel is unknown. The proposed scheme requires systematic LT codes. Therefore, we studied systematic LT codes and found a bound on the number of generated codeword bits. We also introduced nonuniform LT codes, since the distributed source coding can be modeled with a set a parallel channels.

We proposed a decoding algorithm that improves the performance of LT codes, while keeps the decoding fast and efficient.

CHAPTER VII

APPLICATION OF DISTRIBUTED SOURCE CODING TO MULTICAST OVER LOSSY WIRELESS NETWORKS

7.1 Introduction

In this chapter, we study the application of distributed source coding to unicast and multicast over noisy wireless channels. In wireless networks, multicast is an elementary service that is used by many applications. Examples of multicast applications include notification to a set of nodes, one-to-many delivery in peer-to-peer networks, cellular phone-based teleconference / game among a group of people. In this work, we specifically consider ad hoc networks with stationary nodes such as distributed sensor networks. Although, it is focused on non-mobile nodes, we expect that the work can be extended to the mobile ad hoc networks. In a large and dense wireless network, multicast should be done efficiently given the resource constraints. In this work, the following constraints are considered:

- Energy efficiency: Minimizing total number of transmissions. This would save energy in wireless devices that are equipped with limited power supplies, e.g., batteries.
- Reliability: Providing robustness against lossy links in the network.
- Rate optimality: Maximizing the throughput of the network.

Our method uses rateless error correcting codes to provide reliability and rate optimality, and distributed source coding to ensure the energy efficiency. First, we develop a unicast algorithm for a randomly deployed lossy wireless network with a single source and a single receiver. Then, we modify our proposed method for multicast over lossy wireless networks with local information. For these networks, we also provide a multicast subgraph algorithm with the goal of delivering information to destinations by maximizing the use of common links. We show that distributed source coding coupled with the multicast subgraph algorithm generate a reliable, rate-optimal, and energy-efficient multicast algorithm. We compare our scheme with energy-efficient network coding. Our simulations reveal that our schemes perform very close to network coding, while having lower complexity and higher adaptability.

7.1.1 Related Work

Multicast problem has two different solutions in the following two scenarios. In the first scenario, nodes only have relaying capability. In this case, reliable and energy-efficient multicast for a lossy wireless network becomes the problem of finding minimum-cost multicast tree that is a NP-hard problem. Moreover, for networks involve only relaying nodes, it is not possible to achieve the min-cut capacity of the network [14]. In the second scenario, in addition to relaying, each node has the capability of local processing and coding. Network coding was the first method that allows intermediate nodes to encode. Network coding provides a rate-optimal, reliable, and energy-efficient multicast technique using the full knowledge of the network topology [3]. An algebraic framework for network coding was developed by Koetter and Médard [30], who translated the network code design to an algebraic problem which depends on the structure of the underlying graph. In network coding, the intermediate nodes are allowed to encode the information they receive. Therefore, the information capacity between a sender and a receiver can come close to the network min-cut capacity, which makes network coding a rate-optimal technique. Network coding uses linear programming to find a subgraph on which coding is performed. The objective of the linear programming is to minimize the number of transmissions making network coding energy efficient. The coding consists of random linear combinations of received packets. The coefficients of the linear combination is selected from $\mathbb{F}_q$. The work of Li, Yeung, and Cai [33] shows that the network multicast is possible, if linear coding is performed over a sufficiently large finite field. However, the larger the size of the field, the more complicated the operations of encoding and decoding. Furthermore, the complexity of decoding is cubic in the length of the packet. Another drawback of the network coding is its scalability. The deterministic network code methods proposed so far may need to be completely redesigned to accommodate addition of any single node.

The problem of reliable and energy-efficient data dissemination in multi-hop wireless network has been studied in [54, 55] in the context of broadcast (which is a special case of multicast). These schemes employ rateless error correcting codes to provide reliability and reduce the number of transmissions. In the case of broadcasting, it is necessary that every node other than the source receive all the necessary data to retrieve the original data. Therefore, each node is able to generate new packets based on decoding and re-encoding on the original data. However, this is not the case in optimal multicast. Since in multicast non-destination nodes might receive a portion of data, decoding and re-encoding would not be possible at all nodes. Therefore, these schemes cannot be applied for multicast.

The closest work to our approach is the work on reliable multicast in lossy networks using rateless codes [11, 45, 68]. In single-hop networks, this method is rate optimal and reliable. In this method, no prior knowledge of the channel noise is needed and coding is performed only at the source. However, in wireless networks, we are more interested in multi-hop routes because of the power-saving in multi-hop transmissions. Recently, Pakzad et. al. [47] proposed a coding scheme for unicast on line networks. They allow the intermediate nodes to perform decoding and re-encoding, greedy random coding, or encoding. Based on the application and the constraint, the operation of the intermediate nodes are chosen. Our work extends this to multicast on wireless networks by combing the rateless coding with distributed source coding. As shown later, unicast or multicast in wireless network using only rateless codes is not energy efficient because of the redundant transmissions at intermediate nodes.

7.1.2 Contribution

First, we show that multicast using only rateless coding is not energy efficient, because of redundant packet transmissions. We address the energy efficiency problem in wireless networks by exploring cooperation among nodes in the network. This method is based on the philosophy that a single node may not be able to achieve significant energy savings, but collaborating with other nodes enables it to achieve much larger energy savings. One should note this collaboration is done without nodes communicating with each other.

For a single source / single destination network, we propose a Distributed Source Coding-based Unicast algorithm (DSCU) that is energy efficient, rate optimal (for a given directed subgraph), and reliable over multiple disjoint paths. Our proposed algorithm consists of rateless coding and distributed source coding (nodes collaboration). Rateless coding is used to provide reliability and rate optimality, while distributed source coding is used to ensure energy efficiency. The DSCU can be modified for multicast applications. For Distributed Source Coding-based Multicast (DSCM), first we develop a simple multicast subgraph using local information available for each node. This algorithm is based on finding the path that uses the minimum number of transmissions. The proposed multicast subgraph algorithm can be applied to wireless networks that has property $\mathcal{P}$ defined as follows:

1. Nodes are aware of their neighbors.

2. Nodes are aware of destination node locations.

For DSCM, then we show how modified DSCU can be applied on this subgraph to generate a multicast algorithm that would satisfy all the required multicast constraints.

In the presentation of this chapter, we first address the following question, given a subgraph, how to provide a mechanism for energy-efficient, rate-optimal, and reliable unicast / multicast. Next, we will address the problem of designing such subgraphs for Unicast / multicast. In designing such subgraphs, our main goal is the simplicity of the unicast / multicast subgraph, rather than optimality of such constructions.

7.2 Wireless Erasure Network

The wireless erasure network is modeled by a directed graph $g = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. The set $\mathcal{V}$ can be partitioned into S, $\mathcal{D}$, and $\mathcal{I}$ disjoint subsets. S, $\mathcal{D}$, and $\mathcal{I}$ contain the source node, destination nodes, and intermediate nodes, respectively. In other words, $\mathcal{I}$ is the collection of nodes in $\mathcal{V}$ that are neither source nor destination. Each link $(i, j) \in \mathcal{E}$ represents a memoryless erasure channel from node i to node j with erasure probability $\epsilon_{i,j}$. The message transmitted from source s is denoted by $w^{(s)}$. The input of all channels originating from node i in $\mathcal{I}$ is denoted by

X_i and X_i^n represents X_i consisting of n packets. We define $Y_{i,j}$ as the packets received from edge (i,j). We denote the packets received at node i from all incoming channels by Y_i. The wireless nature of the network constrains each node to transmit the same packet on all its outgoing edges, i.e., all neighbors of the node can potentially receive the packet. We also assume that each node receives the packets from all its incoming edges without interference. In other words, we assume that the medium access control (MAC) layer would handle the medium access issues. The packet losses due to channel noise and congestion are all embedded in the link erasure probability $\epsilon_{i,j}$. In our model, we consider only the energy spent for radio frequency (RF) transmission as in [73]. Therefore, the energy consumption is proportional to the number of packet transmissions in the network.

7.2.1 Capacity of Wireless Network

Consider a single source / single destination wireless erasure network described by the directed graph $g(\mathcal{V},\mathcal{E})$ and assumptions in Section 7.2. Let s and d denote the source and the destination, respectively. Similar to [14], we define $x-y$ cut, for $x,y \in \mathcal{V}$, as a partition of $\mathcal{V}$ into two subsets $\mathcal{V}_x$ and $\mathcal{V}_y$ such that $x \in \mathcal{V}_x$ and $y \in \mathcal{V}_y$. The x-set $\mathcal{V}_x$ (or y-set $\mathcal{V}_y$) determines the cut uniquely. For the $x-y$ cut given by $\mathcal{V}_x$, the cut-set $[\mathcal{V}_x,\mathcal{V}_y]$ is the set of edges from the x-set to y-set, i.e.,

$$[\mathcal{V}_x,\mathcal{V}_y] = \{(i,j)|(i,j) \in \mathcal{E}, i \in \mathcal{V}_x, j \in \mathcal{V}_y\}.$$

The capacity of such a network is given by [14]

$$\mathcal{C}_d = \min_{\mathcal{V}_s:\mathcal{V}_s \text{ a } s\text{-}d \text{ cut}} \mathcal{C}(\mathcal{V}_s), \tag{50}$$

where $\mathcal{C}(\mathcal{V}_s)$ corresponds to cut-capacity of a s-d cut and is computed by

$$\mathcal{C}(\mathcal{V}_s) = \sum_{i\in\mathcal{V}_s}(1 - \prod_{j:(i,j)\in[\mathcal{V}_s,\mathcal{V}_d]} \epsilon_{ij}). \tag{51}$$

The capacity in (51) is achievable, if the side information about the location of the erased packets is available at the decoder. This overhead is negligible when the number of packets is large enough.

For a lossy wireless network with single source and multiple destinations d_i, $d_i \in \mathcal{D}$, the capacity of the network is given by the minimum value of the capacity between the source and any of the destinations, i.e.,

$$\mathcal{C} = \min_{d_i \in \mathcal{D}} \quad \mathcal{C}_{d_i} \tag{52}$$

One should note that the capacity in (52) can be reached, if the intermediate nodes are allowed to encode.

Example 1. *Consider the directed graphs in Fig. 34. We assume all edges have the same erasure probability ϵ. In both graphs the goal is to send the information from source s to destination d. In Fig. 34(a), the message is transmitted to the destination through two paths while in Fig. 34(b), the message is transmitted through one path. According to (52), the capacity of these two networks are $\mathcal{C}_a = 1 - \epsilon^2$ and $\mathcal{C}_b = 1 - \epsilon$. As we can see, the network in Fig. 34(a) has a higher min-cut capacity than that of Fig. 34(b). Therefore, we can conclude that the greater the number of the paths, the higher the capacity. One should note that although the packets in Fig. 34(a) flow through more links than that of Fig. 34(b), this does not necessarily increase the energy consumption. As shown later, the total number of packets sent after hop 1 is the same in both cases. In situations, where the number of hops is the same, energy consumption is equal.*

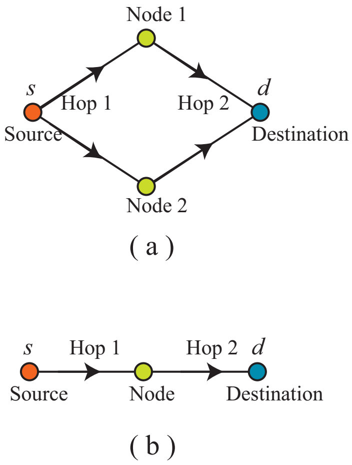

Figure 34: Single source / single destination wireless network: Source information is sent to the destination through (a) two paths, (b) single path.

From now on, we are focusing on unicast / multicast over subgraphs with more than one path from the source to the destination. First, we will assume that these subgraphs with certain properties are given, but later, we will provide a simple algorithm to construct them.

7.3 Unicast Using Rateless Coding

In this section, we show that unicast using only rateless coding can be rate optimal and reliable, but it is not energy efficient. For a single source / single destination network, we assume two disjoint shortest paths from source s to destination d are given. These paths construct graph $g_s = (\mathcal{V}_s, \mathcal{E}_s)$. The neighbors of the source node are called hop-1 nodes. We assume all links of g_s have the same erasure probability of ϵ. We also assume that the source has information traffic consisting of k packets to send to the destination node. According to (52), the capacity of the network is $\mathcal{C} = 1 - \epsilon^2$.

We propose to use rateless coding for unicast. The operation of nodes are as follows:

- Source node s: Source s encodes the information packets using a channel code with a rate close to the capacity of the network. Since we are using rateless codes, source s generates $n = \frac{k}{1-\epsilon^2}$ encoded packets.
- Intermediate nodes $\mathcal{I}$: Node $i \in \mathcal{I}$ receives on average $\frac{k}{1+\epsilon}$ packets. Node i encodes the received packets using a channel code with a rate close to the capacity of its outgoing link. In other words, node i generates $\frac{k/(1+\epsilon)}{1-\epsilon} = n$ encoded packets using rateless codes. The intermediate node are allowed to encode but they are not required to decode.
- Destination node d: The receiver recovers the information at hop-1 nodes by sequentially decoding on intermediate nodes starting from neighbors of destination node going toward the source node. By comparing Y_1 and Y_2, $(1-\epsilon^2)n = k$ encoded packets would be available at the destination node. Therefore, out of n encoded packets by the source, k packets are received by at least one of the immediate-intermediate nodes. According to rateless code properties, for large k, any k encoded symbols is sufficient for recover k information packets. Thus, the destination node recovers the information packets.

Example 2. *Consider the single source / single destination network shown in Fig. 34(a). As mentioned in Example 1, the capacity of the network is* $(1-\epsilon^2)$*. Therefore, the information traffic consisting of* k *packets is mapped to* $w^{(s)}$ *consisting of* $n = \frac{k}{1-\epsilon^2}$ *packets. Each of these encoded packets is reached by node 1 and node 2 via a single transmission with probability* $1-\epsilon$*. During the transmission of* $w^{(s)}$ *over lossy links of* $(s,1)$ *and* $(s,2)$*, some of the packets are erased. Nodes 1 and 2 receive* Y_1 *and* Y_2 *consisting of the average number of* $n(1-\epsilon)$ *packets. Nodes 1 and 2 encode* Y_1 *and* Y_2 *to* X_1^n *and* X_2^n*, respectively using rateless coding. The destination node receives information from both edges without interference and attempts to recover the information traffic using the side information about the erasure locations (the sequence number of lost packets).*

7.3.1 Analysis of Unicast Using Rateless Coding

Now, we evaluate the performance of the proposed unicast algorithm using rateless codes in terms of reliability, rate optimality, and energy efficiency.

Reliability: As explained above, the destination can recover the information packets perfectly. This shows that the proposed method is reliable.

Rate optimality: Since the ratio of number of the packets in information traffic to the number of encoded packets is $1-\epsilon^2$, the same as min-cut capacity, this method is rate optimal. It is worth noting that this is achievable because we let the intermediate nodes to encode.

Energy efficiency: This method is not energy efficient. As an example, consider the network in Fig. 35. For this network we assume that information traffic consists of three packets. Source node generates $w^{(s)}$ using rateless codes. Receiving any three of the packets in $w^{(s)}$, would be sufficient for recovering the original information traffic. Nodes 1 and 2 receive some of the packets. They apply rateless coding on Y_1 and Y_2 to generate X_1 and X_2 consisting of four encoded packets. The destination node receives packets from both paths (s,1,3) and (s,2,3). Applying decoding on the received packets, Y_1 and Y_2 are recovered. Some of these packets are redundant. As an example, consider the packets at nodes 1 and 2 in Fig. 35. The packets in Y_1 and Y_2 can be categorized as follows:

1. Received at both nodes,

2. Received at only one of the nodes,

3. Erased at both nodes.

Clearly, packets in the first category is redundant and should not be transmitted by both nodes 1 and 2. Therefore, unicast via rateless coding does not meet all the requirements. In the next section, we propose a method that is not only rate optimal and reliable, but also energy efficient. Our proposed method eliminates redundant transmissions, because correlated information at nodes are eliminated by appropriate distributed source coding algorithm. In the following section, we introduce the Distributed Source Coding-based Unicast (DSCU).

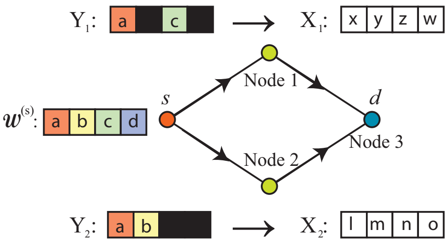

Figure 35: Single source / single destination wireless network of Fig. 34a, when $\epsilon = .5$.

7.4 The DSCU Scheme with Equal Erasure Probabilities on All Edges

For a single source / single destination network, we perform DSCU on graph $g_s = (\mathcal{V}_s, \mathcal{E}_s)$. As shown later in Section 7.6, g_s consists of two disjoint shortest paths from the source node to the destination node. According to [14], the capacity of the g_s under the assumption that erasure locations on all links of the network are provided to the destination is $\mathcal{C}_s = 1 - \epsilon^2$, where ϵ is the erasure probability of all links. We assume that the information traffic consists of k packets, where k is large. The nodes are divided into four disjoint sets based on their operations:

- Broadcasting (S) nodes: Consists of source node s.

- Distributed Source Coding ($\mathcal{DSC}$) nodes: Consists of nodes $i \in \mathcal{DSC}$ such that $(j, i) \in \mathcal{E}_s$ and $j \in S$.
- Destinations ($\mathcal{D}$) nodes: Consists of destination node d.
- Intermediate ($\mathcal{I}$) nodes: Consists of nodes that do not belong to the previous three sets.

Operation of each set of the nodes is as follows:

1. S-Set: Source node s generates $\frac{k}{\mathcal{C}_s}$ encoded packets using rateless codes. These packets are sent out on all outgoing edges of s.

2. $\mathcal{I}$-Set: Node $i \in \mathcal{I}$ applies rateless encoding on the received packets such that $\frac{|Y_i|}{1-\epsilon}$ encoded packets are generated, where $|Y_i|$ denotes the number of the packets received by node i. We are assuming that $\mathcal{I}$ nodes only perform encoding. All the operations introduced for line networks introduced in [47] can also be performed by intermediate nodes. As an example if node i does decoding and re-encoding, the system can be adaptive to cases where the erasure is unknown. However, for simplicity, we assume that intermediate nodes apply only encoding.

3. $\mathcal{DSC}$-Set: First, node $i \in \mathcal{DSC}$ operates distributed source coding explained in Section 7.4.1. Then, on the resulting k_i packets applies rateless coding to compensate the lossy link. In other words, node i sends out $\frac{k_i}{1-\epsilon}$ packets on its outgoing links.

4. $\mathcal{D}$-Set: Destination node d gathers information from all its neighbors. Using the channel coding property and the side information about the erasure locations, d recovers the source information.

7.4.1 Distributed Source Coding

To describe distributed source coding algorithm, consider a single source / single destination network whose source node is s and $\mathcal{DSC}$ nodes are 1 and 2. As stated above, the source node s sends $n = \frac{k}{1-\epsilon^2}$ packets over both links $(s, 1)$ and $(s, 2)$. On the average, $n\epsilon$ packets are erased in Y_1 and Y_2 during the transmission from which $n\epsilon^2$ encoded packets are lost in both of them, these packets are shaded with solid black in Fig. 36. The dotted sections in Fig. 36 show $n(\epsilon - \epsilon^2)$ packets that are lost in only one of the Y_1 or Y_2. As we can see, there are packets that both Y_1 and Y_2 received. Applying the encoding on Y_1 and Y_2 results in sending redundant packets. The goal is to compress Y_1 and Y_2 such that all redundant packets are eliminated. One should note that this compression must be done without any communication between nodes 1 and 2. This is because the intercommunication between nodes consume energy which is in contrast with our goal. Since the sources are distributed and are being compressed independently, this system requires distributed source coding. According to Slepian-Wolf theorem [69], the output of two correlated sources that do not communicate can be compressed with the same rate as if they were communicating. This is true when the decoder has access to both compressed outputs. Because of the specific correlation between Y_1 and Y_2, the distributed source coding methods mentioned in Chapters 3, 4 cannot be applied here.

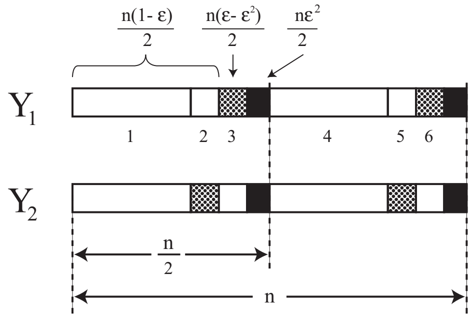

Figure 36: Illustration of correlation between Y_1 and Y_2.

If Y_1 and Y_2 communicate with each other, then the total number of $n(1-\epsilon^2)$ packets are sufficient to be sent from both nodes 1 and 2. According to Slepian-Wolf theorem, although nodes 1 and 2 are compressing their information without communicating with each other, it is sufficient for them to send total of $n(1-\epsilon^2)$ packets.

We focus on distributed source coding at the middle point, i.e, Y_1 and Y_2 are getting compressed at the same rate. In other words, each node sends $n(1-\epsilon^2)/2$ encoded packets. Distributed source coding requires a systematic channel code [61]. Y_1 is compressed as follows: Y_1 is fed into systematic rateless encoder. At the output of the encoder, the packets received in the first half of the Y_1 along with parity P_{Y_1} are sent. The number of packets in P_{Y_1} is $n(\epsilon - \epsilon^2)/2$. It is important to note that P_{Y_1} is generated using all the received packets in Y_1. The same procedure is applied at node 2, with the modification that the packets in second half of Y_2 with generated P_{Y_2} are sent. The correlation between Y_1 and Y_2 is eliminated and the generated signals X_1 and X_2 are independent. Figure 37 clarifies our description.

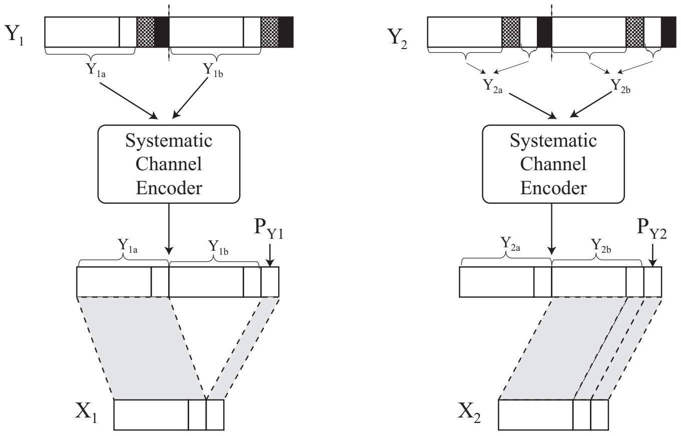

Figure 37: Distributed source coding of correlated signals Y_1 and Y_2.

7.4.2 DSCU over $l > 2$ Disjoint Paths

In this section, we assume that g_s consists of $l > 2$ disjoint shortest paths. The min-cut capacity of this network is

$$\mathcal{C}_s = 1 - \epsilon^l. \tag{53}$$

In this case, the method described for $l = 2$ can be applied with a minor modification to the operation of the $\mathcal{DSC}$ nodes. Set A is defined such that it contains nodes i_h, for all $h \in \{1, 2, \ldots, l\}$, where $i_h \in \mathcal{DSC}$-set and $(s, i) \in \mathcal{E}_s$. A systematic LT-code is applied to the information of these nodes and then fraction $1/l$ of their information packets along with the corresponding parity packets are sent. The fractions of the information packets in each node is chosen such that the bit positions are complimentary of each other (i.e., each bit position is sent from only one node). This process generates $X_{i_h}^{(\frac{n(1+\epsilon)}{l})}$. With the same reasoning as before, this method provides a reliable, rate-optimal, and energy-efficient multicast method.

From (53), one can see that the greater the number of the paths, the higher the capacity. We would like to emphasize that increasing the number of paths does not necessarily increase the energy usage. In fact, in this case the total number of the packets after Hop i, where i is a number smaller than the length of the paths in g_s, is equal to $n(1+\epsilon)$ which is the same as that of the network with $l = 2$. In other words, our proposed method uniformly distribute the packets over the disjoint paths. Therefore, the total number of the packets remain the same as the case with smaller number of paths and so is the energy consumption. We assume there are always two edge-disjoint paths from source to each destination. Studying the impact of having l outgoing edges from source is part of the ongoing research.

7.5 Extension of DSCU

In this section, we study DSCU on a lossy single source / single destination network, where erasure probabilities are different on different links. For a single source / single destination network, DSCU is done on graph $g_s = (\mathcal{V}_s, \mathcal{E}_s)$. As shown later in Section 7.6, g_s consists of two edge-disjoint minimum-cost paths from the source node to the destination node, where the cost is defined as the total number of packet transmission by the entire network. Neighbors of the source nodes are labeled as 1 and 2. The capacity of g_s, $\mathcal{C}_s$ is computed by (52), where ϵ_{ij} denotes the erasure probability of the link (i, j). The information traffic consists of k packets. The nodes are divided into the four disjoint sets similar to the previous section. The operations of nodes belonging to S-set, $\mathcal{D}$-set, and $\mathcal{I}$-set remain the same as before. In other words, source encodes k packets to $n = \frac{k}{\mathcal{C}_s}$ encoded packets, intermediate nodes apply rateless coding to compensate the lossy edges, and destination nodes gather all the packets from their neighbors and using decoding algorithm they find the original packets. However, the operation of $\mathcal{DSC}$ nodes, for the following reason, need to be modified.

According to Section 7.4, it can be shown that node 1 needs to generate

$$\frac{\frac{n}{2}(1-\epsilon_{s1}) + \frac{n}{2}(\epsilon_{s2} - \epsilon_{s1}\epsilon_{s2})}{1-\epsilon_{13}} \tag{54}$$

packets, where the numerator corresponds to the total number of the information packets and parity packets needed by the Slepian-Wolf theorem, and the denominator is required for compensation of the packet loss on the link $(1, 3)$. Similarly, node 2 generates

$$\frac{\frac{n}{2}(1-\epsilon_{s2}) + \frac{n}{2}(\epsilon_{s1} - \epsilon_{s1}\epsilon_{s2})}{1-\epsilon_{24}} \tag{55}$$

packets. The total number of the packets in (54) and (55) might be greater than the available bandwidth, depending on the value of the erasure probabilities (we assume that in each second n packets are sent, which this makes the network to have bandwidth of n packets/sec). In that case, compression with higher rates is required. Assume $\epsilon_{\max}$ denotes the largest erasure probability of the network. Also, assume node 1 requires to send $\frac{k_1}{1-\epsilon_{13}}$ packets and node 2 required to send $\frac{k_2}{1-\epsilon_{24}}$ packets. To find the value of k_1 and k_2, the following equations must be solved for λ and a:

$$\begin{cases} \lambda a(1-\epsilon_{s1}) + (1-\lambda)a(\epsilon_{s2} - \epsilon_{s1}\epsilon_{s2}) = k_1 \\ (1-\lambda)a(1-\epsilon_{s2}) + \lambda a(\epsilon_{s1} - \epsilon_{s1}\epsilon_{s2}) = k_2 \\ k_1 + k_2 = k \\ \frac{k_1}{1-\epsilon_{\max}} < n \quad \text{and} \quad \frac{k_2}{1-\epsilon_{\max}} < n \end{cases}$$

- The first two equations show the total number of the packets each node has to send. In both Y_1 and Y_2, the first $(n-a)$ packets are punctured and the compression is applied on the last a packets. From these a packets, $a(1-\epsilon_{s1})$ packets are received correctly at node 1, from which λ fraction is sent along with $(1-\lambda)a(\epsilon_{13} - \epsilon_{12}\epsilon_{13})$ parity packets, according to Slepian-Wolf theorem. For node 2, the received packets in the remaining $(1-\lambda)$ packets along with $\lambda a(\epsilon_{s1} - \epsilon_{s1}\epsilon_{s2})$ parity packets are sent on link $(2,4)$.
- The third equation shows the total number of the packets send out from nodes 1 and 2, before considering the extra parity packets needed for the noisy links $(1,3)$ and $(2,4)$. According to the rateless code properties, any k encoded packets are sufficient for recovery of the source message consisting of k packets.
- The fourth equation guarantees that the number of the packets injected to the links at any node is less than n.

Solving these equation, we have

$$a = \frac{k}{1-\epsilon_{s1}\epsilon_{s2}}, \tag{56}$$

and the value of λ is bounded by the fourth equation. Figure 38 shows Y_1 and Y_2 after puncturing. One should note that extra parity packets are generated to compensate the

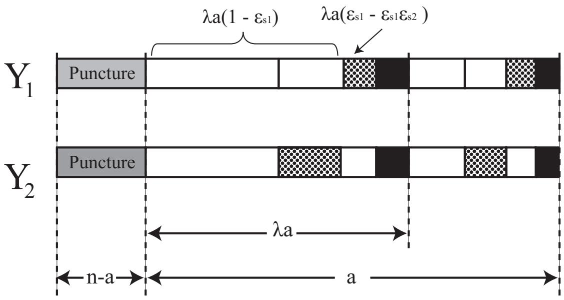

Figure 38: Signals Y_1 and Y_2 after puncturing.

lossy links $(1,3)$ and $(2,4)$. In summary $X_1^{\frac{k_1}{1-\epsilon_{13}}}$ and $X_2^{\frac{k_2}{1-\epsilon_{24}}}$ are sent out on links $(1,3)$ and $(2,4)$, respectively.

Now, we evaluate the performance of the proposed unicast algorithm using rateless codes in terms of reliability, rate optimality, and energy efficiency. The proposed algorithm is reliable and rate optimal with the same reasoning as the one presented in 7.3.1. DSCU is also energy efficient, since all correlated packets at are eliminated by distributed source coding algorithm.

7.6 Multicast Subgraph for Grid Networks

In this section, we introduce a suboptimal algorithm that finds a suitable multicast subgraph on the network graph $g(\mathcal{V},\mathcal{E})$ with property $\mathcal{P}$. We specifically, consider $g(\mathcal{V},\mathcal{E})$ to be a gird network. However, we note that our algorithm can also be applied to every network with property $\mathcal{P}$. In other words, our proposed multicast protocol applies to a large class of networks and is not confined to the grid networks. The grid network is restricted to a square of $2r-1$ rows and columns with neighbors spaced equally away from each other. We define N_i as a set of neighbors of i. After running the multicast subgraph algorithm, node i finds node $\pi[i] \in N_i$ to forward the message to.

The multicast subgraph algorithm is a modified version of Bellman-Ford algorithm and is based on finding two edge-disjoint minimum-cost paths from source to each destination nodes. Cost of node i, $\xi[i]$, is defined as the total number of packets required to be sent from node i and its descendant to destination for full recovery of the message. The cost considers rateless coding at all intermediate nodes based on the erasure on the links. Given a lossy, directed graph $g(\mathcal{V}, \mathcal{E})$ with source s, and destination nodes $\mathcal{D}$, the multicast subgraph algorithm finds two minimum-cost paths for each destination node. The paths to each destination node are edge-disjoint, but the paths to multiple destinations can have edges in common.

Algorithm 1 *Multicast subgraph Algorithm*

```
Require: $g(\mathcal{V}, \mathcal{E})$ with destination node set $\mathcal{D}$.
  for $d \in \mathcal{D}$ do
    $R(g(\mathcal{V}, \mathcal{E}), d)$.
  end for
```

Algorithm 2 $R(g(\mathcal{V}, \mathcal{E}), d)$

```
Require: $g(\mathcal{V}, \mathcal{E})$ with $d \in \mathcal{D}$
  for $i \in \mathcal{V}$ do
    $\xi[i] \leftarrow \infty$
    $\pi[i] \leftarrow$ Unknown
  end for
  $\xi[d] \leftarrow 0$
  for $i \in \mathcal{V}$ do
    for each neighbor $j$ do
      $RELAX(g, i, j)$
    end for
  end for
```

Algorithm 3 $RELAX(g, i, j)$

```
Require: $g(\mathcal{V}, \mathcal{E})$ with $(i, j) \in \mathcal{E}$
  if $\xi[j] > \xi[i] + \frac{\xi[i]}{1-\epsilon_{ij}}$ then
    $\xi[j] \leftarrow \xi[i] + \frac{\xi[i]}{1-\epsilon_{ij}}$
    $\pi[j] \leftarrow i$
  end if
  if $\xi[j] = \xi[i] + \frac{\xi[i]}{1-\epsilon_{ij}}$ and $j \neq s$ then
    if $i$ is closer to more destination nodes than $j$ then
      $\pi[j] \leftarrow i$
    end if
  end if
  if $\xi[j] = \xi[i] + \frac{\xi[i]}{1-\epsilon_{ij}}$ and $j = s$ then
    $\pi[j] \leftarrow \pi[j] \bigcup i$
  end if
```

The proposed algorithm finds one minimum-cost path for each destination. For each destination, we have at most $|N_s|$ paths, from which we choose the one that has the largest overlap with paths to other other destination nodes. As stated in Section 7.4.2, we want to have two edge-disjoint paths from source to each destination. After determining the first path for each destination, we delete the edges on these paths while keeping the vertices. The new graph is indicated with $g_p(\mathcal{V}, \mathcal{E})$. Then, we execute the algorithm again. The first and second sets of paths construct the graph g_s. The following examples show how the execution of the proposed algorithm works on a single source / single destination grid network and single source / two destination grid network.

Example 3. *Figure 39 shows a single source / single destination network with equal erasure probability on all edges. The proposed routing algorithm finds two paths from which one is randomly chosen. Figure 40 finds the second path. The edges on the path found in Fig. 39 are deleted and the routing algorithm is executed to find the second path on the network of Fig. 40(a). This process is shown in Fig. 40.*

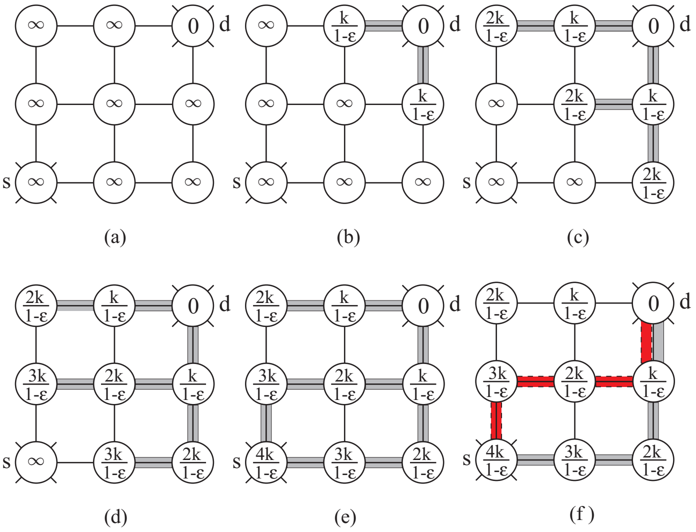

Figure 39: The execution of the proposed algorithm. The node values and shaded edges indicate the cost values and the values of π with the iteration of algorithm, respectively. (a) Before the first pass over the edges. (b)-(e) After each successive pass over the edges. (f) shows two paths from source to destination d. The path with lighter shade is chosen.

Example 4. *The network in Fig. 41 consists of one source and two destination nodes. Figures 41(c) and 42(c) show paths from the source to destination nodes d_1 and d_2.*

The proposed algorithm only uses the local information available for each node and has low complexity.

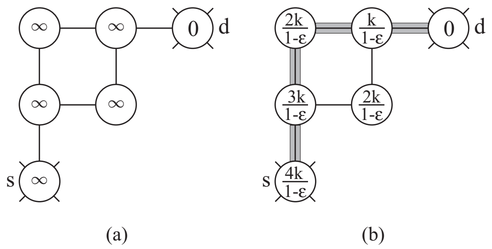

Figure 40: The execution of the proposed algorithm to find the second path for network of Fig. 39(a). (a) The network on which the routing algorithm is executed. (b) The second path from source to the destination.

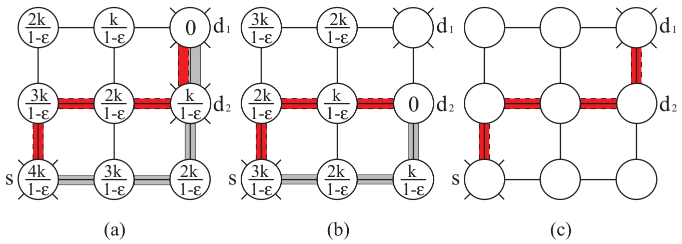

Figure 41: (a) The result of the execution of the proposed algorithm for destination d_1. (b) The result of the execution of the proposed algorithm for destination d_2. (c) The first path from source to both destinations.

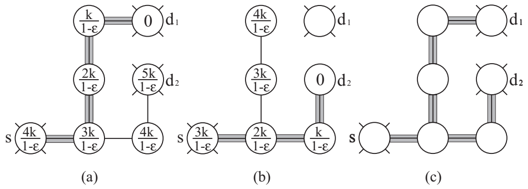

Figure 42: The proposed algorithm finds the second path for network of Fig. 41(a) after deleting the edges of the first path. (a) and (b) The paths from source to destination node d_1 and d_2, respectively. (c) The second path from source to destinations.

7.6.1 Multicast on Grid Network

In this Section, we study the DSCM on the subgraph g_s generated based on the method proposed in Section 7.6. The nodes on the generated subgraph find the class they belong to and their operation is defined based on their class. One should note destination nodes might belong to more than one class. In other words, a destination node might perform as an intermediate node for other destination nodes. For nodes belonging to the set $\mathcal{DSC}$, we imply the following restriction: node $i \in \mathcal{DSC}$, sends out the information packets from the first half of the packets, and the generated parity packets, if edge $(j, i) \in g_s$ is horizontal and sends out the information packets from the second half of the information packets along with the parity packets otherwise. As an example, we explain the proposed multicast algorithm for the network of Fig. 43.

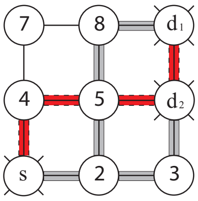

Figure 43: A single source / two destination network.

The nodes s, 2, 3, 4, 5, 8, d_1, and d_2 and the shaded edges in Fig. 43 construct graph g_s. The nodes are divided as follows: $S = \{s\}$, $\mathcal{D} = \{d_1, d_2\}$, $\mathcal{DSC} = \{2, 4\}$, and $\mathcal{I} = \{3, 5, d_2, 8\}$. The min-cut capacity of g_s is equal to $1 - \epsilon^2$. Therefore, the source node s, encodes the information traffic consisting of k packets to $n = \frac{k}{1-\epsilon^2}$ encoded packets. Nodes 2 and 4, on average, receive $n(1-\epsilon)$ packets. Node 2, first performs distributed source coding to generate Z_2. Z_2 consists of the information packets from the first half of the packets in Y_2, and generated parity packets. The required number of parity packets is $n(\epsilon - \epsilon^2)/2$. Then, rateless coding is done on Z_2 to generate $X_2^{\frac{k/2}{1-\epsilon}}$. Node 4 uses the information packets from the second half of the packets in Y_4, and generated parity packets to construct Z_4. Then, using rateless coding, X_4 consisting of $\frac{k/2}{1-\epsilon^2}$ packets is generated. Nodes 3, d_1, and 8 apply rateless coding on their received packets such that each send out $\frac{k/2}{1-\epsilon}$ packets. Since there are two edges entering node 5, this node generates $\frac{k/2}{1-\epsilon}$ encoded packets from the packets received from each link. Therefore, this node sends out $\frac{k}{1-\epsilon}$ encoded packets. Node d_2 receives total of k independent encoded packets from links $(3, d_2)$ and $(5, d_2)$. According to the rateless code properties, the information packets can be recovered at the destination d_2. Similarly, the destination node d_2 recovers the original message. Node d_1 receives packets from links (d_1, d_2) and $(8, d_1)$ and recovers the original message from its received packets.

7.7 Simulation Results

In this section, we demonstrate the performance of the multicast method using distributed source coding with the proposed multicast subgraph. We performed simulations to compare our method with network coding scheme for grid wireless network. For network coding, we used the scheme proposed in [42] to find the data rates to be sent on each link. We assume that the direction of the edges is either to the right or up. We present the total number of transmissions per packet for lossy grid networks of different sizes, different source location, and different destination locations. The location of the source node and destinations are reported in the first and second columns of the table, respectively. The node labeling is shown in Fig. 44. We assume that $k = 1000$ packets are sent from source node to the destination nodes.

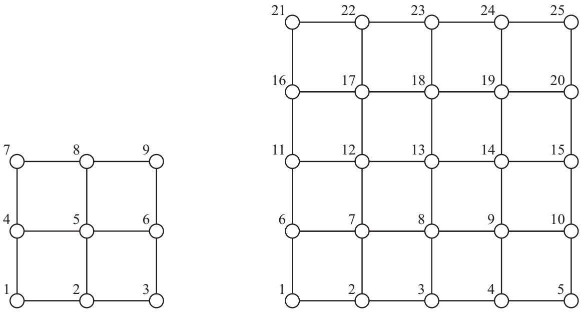

Figure 44: The grid networks used for simulation results.

Tables 5, 6, and 7 show the total number of transmissions per packet by the entire network, when all edges have the same erasure probability of $\epsilon = 0.1$.

Table 5: Total number of transmissions per packet for network grid 3 × 3, where all edges have the same erasure probability of $\epsilon = 0.1$.

S	$\mathcal{D}$	DSCM	Network Coding
1	9	4.14	4.14
1	5,9	4.24	4.24
1	5,6,8	4.3	3.23
2	7,9	4.3	4.14

As we can see, the simulation results for our method perform very close to the results of network coding for the majority of the cases. One should note that although network coding performs better than our method in some cases, network coding assumes full knowledge of

Table 6: Total number of transmissions per packet for network grid 5×5, where all edges have the same erasure probability of $\epsilon = 0.1$ and source is located at point 1.

$\mathcal{D}$	DSCM	Network Coding
19	6.5	6.28
7,18,25	8.7	8.40
7,15,17,25	9.34	9.14

Table 7: Total number of transmissions per packet for network grid 7×7, where all edges have the same erasure probability of $\epsilon = 0.1$ and source is located at point 1.

$\mathcal{D}$	DSCM	Network Coding
9,25,34,49	13.1	12.54
9,24,34,46	12	10.75

the network topology to perform a computationally intensive optimization. Network coding uses linear programming to find a subgraph on which coding is performed. Therefore, when complete knowledge of the network topology is unavailable, network coding is impossible while one can still use our method. Moreover, the decoding complexity of network coding is higher than that of DSCM. To retrieve k packets, decoding complexity is in order of $O(k^3)$ and $O(k)$ for network coding and DSCM, respectively.

We also provide simulation results for network grid of size 3×3, where erasure probabilities are different. The erasure values are shown in Fig. 45. Total number of packet transmissions per packet for this network is shown in Table 8.

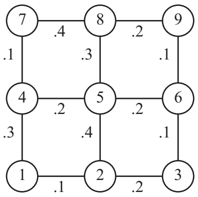

Figure 45: The grid networks whose links have different erasure probabilities.

First, we need to solve the set of equations in Section 7.5 to find the number of the packets each nodes 2 and 4 have to send.

$$\begin{cases} k_2 = \frac{1}{0.97}(0.63\lambda + 0.27), \\ k_4 = \frac{1}{0.97}(-0.63\lambda + 0.7), \end{cases}$$

where $0.1587 < \lambda < 0.5238$. We choose $\lambda = 0.2$, which results in $k_2 = 0.4k$ and $k_4 = 0.6k$. The following table summarizes the simulation results for this network.

Table 8: Total number of transmissions per packet by the entire network in Fig. 45.

S	$\mathcal{D}$	DSCM	Network Coding
1	9	4.63	4.33
1	5,9	5.27	4.67
1	5,6,8	5.07	4.1

7.8 Conclusion

In this chapter, we study the application of distributed source coding to unicast and multicast over lossy wireless networks. First, we showed that unicast / multicast using only rateless error correcting codes on lossy wireless networks is not energy efficient. Then, we proposed a unicast scheme that uses rateless error correcting codes and distributed source coding. This algorithm is reliable, rate optimal, and energy efficient. We modified the unicast algorithm for multicast application on lossy wireless networks, where every node only has the local information of its neighbors and their positions and the destination node positions. We also provided a subgraph multicast algorithm that is based on finding a subgraph with minimum number of packet transmission. This suboptimal algorithm provides two edge-disjoint paths from source to each destination. The paths to different destinations can have edges in common. Finally, we compared the energy efficiency of our multicast method with network coding that assumes global knowledge. We found that our algorithm performs very close to network coding with lower decoding complexity.

CHAPTER VIII

CONCLUSIONS

The goal of this dissertation was to study the theoretical and practical aspects of modern error-correcting codes. Wavelet codes, LDPC codes, and rateless codes are considered as modern error control codes.

8.1 Two-Dimensional Wavelet Codes

We extended the idea of one-dimensional wavelet coding to two-dimensional case and introduced half-rate two-dimensional wavelet codes (TDWCs). We investigated properties of these codes. We showed that these linear codes are lattice cyclic. This property simplifies the encoding and the erasure decoding of TDWCs. We also introduced a methodology to design TDWCs over binary erasure channels. We showed that the half-rate TDWCs of dimensions $N_1 \times N_2$ satisfy the Reiger bound with equality for burst erasures of dimensions to $N_1 \times N_2/2$ and $N_1/2 \times N_2$. These codes can recover burst erasures with a simple and efficient ML decoding. We also provided examples that recover any rectangular burst erasure of area $N_1 N_2/2$.

8.2 Distributed Source Coding

We studied the problem of distributed source coding for two and three correlated signals using LDPC code or rateless codes. Our method is based on sending a fraction of information bits and a fraction of parity bits. Since our method is based on parity bits, it can be adapted to the applications whose wireless channels are noisy. We proposed to model the distributed source coding problem with a set of parallel channel that simplifies the distributed source coding to designing non-uniform channel codes. This design criterion improves the performance of the source coding considerably. We studied the following problems related to distributed source coding.

8.2.1 Lossless Distributed Source Coding, when the Correlation Parameter is Known

We studied the distributed source coding at the corner point. We showed that this problem is simplified to non-uniform LDPC code and semi-random punctured LDPC codes for a system of two and three correlated sources, respectively. Then, we extended our method to distributed source coding at any arbitrary rate on the Slepian-Wolf rate region. We showed that the distributed source coding problem is simplified to designing a rate-compatible LDPC code that has unequal error protection property. We proved that there is no rate loss across the Slepian-Wolf rate region for every arbitrary compression rate. This was also confirmed by simulation results. We also provided a decoding algorithm that lets the sources be decoded independently, thus avoids any error propagation. The simulation results confirmed that our proposed method for both corner point and arbitrary rate on the Slepian-Wold rate region performs considerably better than the previous works on this field.

8.2.2 Lossless Distributed Source Coding, when the Correlation Parameter is Unknown in Advance

We undertook the problem of distributed source coding, when there is no information about the correlation value at the time of code design. In this problem, we assumed the nodes become aware of their correlation parameters after deployment. We showed that the solution to this problem is similar to distributed source coding at arbitrary rates on the Slepian-Wolf limit. The simulation results confirmed that our compression technique is superior to the design of the code for the lowest possible correlation probability, when the correlation parameter is unknown in advance.

8.2.3 Lossy Distributed Source Coding

We extended the idea of lossless distributed source coding to the problem of lossy distributed source coding, when the side information is available at the decoder. Our proposed method is based on sending parity bits generated by LDPC codes. We provided a design procedure for the LDPC code that guarantees performance close to the Wyner-Ziv limit for long LDPC codes.

8.2.4 Lossless Distributed Joint Source-Channel Coding, when the Channel Status is Unknown

We studied the distributed source coding problem for applications whose wireless channel is an erasure channel with unknown erasure probability. We showed that for these application, rateless codes are better candidates than standard error correcting codes. As stated above, we modeled the distributed source coding problem with a set of parallel channels, which requires non-uniform channel coding. We introduced non-uniform LT codes and proposed a decoding algorithm that improves the performance of LT codes, while keeps the decoding fast and efficient. The simulation results showed that distributed source coding using non-uniform rateless code with the proposed improved decoding algorithm performs almost the same as LDPC code, while they are a better fit for the applications with erasure wireless channels.

8.2.5 Multicast

We studied, for the first time, the application of distributed source coding to unicast / multicast over lossy wireless channels. We showed that unicast / multicast via only rateless error-correcting codes is not energy efficient. We proposed a unicast scheme that uses rateless error-correcting codes and distributed source coding. Rateless coding is used to provide reliability and rate optimality, while distributed source coding is used to ensure energy efficiency. Therefore, this algorithm is reliable, rate optimal, and energy efficient. We modified the unicast algorithm to arrive at a multicast scheme on lossy wireless networks, where nodes only have the local information of their neighbors and the destination node positions. We also provided an algorithm that finds a multicast subgraph with minimum number of packet transmission. This suboptimal algorithm provides two edge-disjoint paths from source to each destination. The paths to different destinations can have edges in common. The proposed algorithm is low complex. We compared the energy efficiency of our multicast method with network coding that assumes global knowledge. With, respect to energy efficiency, we found that our algorithm performs very close to the network coding, however, with lower decoding complexity.

8.3 Suggestions for Future Work

This dissertation opened up many theoretical and practical research possibilities in error-control coding and related areas in signal processing and communications. In the following, only some of the several interesting and potentially rich problems are listed.

- Problems related to distributed source coding
  - Studying finite-length non-uniform LDPC codes
  - Investigating finite-length rate-compatible LDPC codes
  - Extending distributed source coding to more than three sources
  - Applying coding methods to applications such as
    1. Watermarking and data hiding
    2. Multimedia data transmission
- Problems related to multicasting over lossy wireless networks
  - Providing an optimal routing using only local information about the network
  - Investigating the optimal number of paths from source to destinations

APPENDIX A

PROOFS FOR CHAPTER 2

A.1 Proof of Lemma 3

First, we show that $< g_{eq}, g'_{eq} >= 0$, if the 2-D code generated using g_{eq} is self dual. The array $g_{eq}(\mathbf{n})$ is a valid 2-D codeword and so is $g'_{eq}(\mathbf{n})$, because of the lattice-cyclic property of the code. Since the code is self dual, the inner product of any two codewords is zero, i.e., $< g_{eq}, g'_{eq} >= 0$.

Next, we need to show that if $< g_{eq}, g'_{eq} >= 0$, then the code is self dual. Let $c(\mathbf{n})$ and $c'(\mathbf{n})$ be codewords corresponding to the message arrays $m(\mathbf{n})$ and $m'(\mathbf{n})$, respectively. The codeword array $c'(\mathbf{n})$ can also be generated from $c'(\mathbf{n}) = (m_{\uparrow \mathbf{M}} \circledast g'_{eq})(\mathbf{n})$. The inner product of codewords c and c' can be written as

$$< c, c' >= \sum_{i,j,k,l} m_{\uparrow \mathbf{M}}(i,j) m_{\uparrow \mathbf{M}}(k,l) < g_{eq}[i,j], g'_{eq}[k,l] >, \tag{57}$$

where $g_{eq}[i,j]$ is a cyclic shift of g_{eq} in the direction of lattice point (i,j). One should note that (i,j) and (k,l) are lattice points, since the $m_{\uparrow \mathbf{M}}$ has value of zero at non-lattice points. Since we assumed the inner product of the $g_{eq}(\mathbf{n})$ and any cyclic shift of that is zero, (57) is equal to zero. In other words, the code is self dual.

A.2 Proof of Lemma 8

It is assumed that the element at position $(1,1)$ is a lattice point. We know that $|\det \mathbf{M}| = 2$. Thus half of the elements are non-lattice points. We note that at least one of the elements at positions $(1,2)$ or $(2,1)$ is a non-lattice point. We may assume the element at $(1,2)$ is a non-lattice point. As mentioned earlier, the traveling vector is $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$. Therefore, to prove that $(N_1/2+1, N_2/2+1)$ is the coordinate for a non-lattice point, we need to show that there exists an integer k such that

$$(((1+2k))_{N_1}, ((2+2k))_{N_2}) = (N_1/2+1, N_2/2+1). \tag{58}$$

has an answer. This equation can be rewritten as the following:

$$\begin{cases} 1+2k \equiv \frac{N_1}{2}+1 & \mod N_1 \\ 2+2k \equiv \frac{N_2}{2}+1 & \mod N_2 \end{cases}$$

N_1 must be a multiple of four to have an answer for the equation. Thus N_2 must be an even number such that $GCD(N_1, N_2) = 2$ for the second equation to have an answer.

REFERENCES

[1] Aaron, A. and Girod, B., "Compression with side information using turbo codes," *Proc. IEEE Data Compression Conference*, pp. 252–261, April 2002.

[2] Abdel-Ghaffar, K. A. S., McEliece, R. J., and van Tilborg, H. C. A., "Two-dimensional burst identification and their use in burst correction," *IEEE Transactions on Information Theory*, vol. 34, pp. 494–504, May 1988.

[3] Ahlswede, R., Cai, N., Li, S.-Y. R., and Yeung, R. W., "Network information flow," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 1204–1216, July 2000.

[4] Bajcy, J. and Mitran, P., "Coding for the Slepian-Wolf problem with turbo codes," *Proc. IEEE Globecom*, pp. 1400–1404, November 2001.

[5] Berrou, C. and Glavieux, A., "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, pp. 1261–1271, October 1996.

[6] Blaum, M., "A family of efficient burst-correcting array codes," *IEEE Transactions on Information Theory*, vol. 36, pp. 671–675, May 1990.

[7] Blaum, M. and Farrell, P. T., "A family of cluster-correcting array codes," *IBM Res. Rep. RJ 9836*, June 1994.

[8] Blaum, M., Farrell, P. T., and van Tilborg, H. C. A., "Multiple burst-correcting array codes," *IEEE Transactions on Information Theory*, vol. 34, pp. 1061–1065, September 1988.

[9] Blaum, M. and Roth, R. M., "New array codes for multiple phased burst correction," *IEEE Transactions on Information Theory*, vol. 39, pp. 66–77, January 1993.

[10] Breitbach, M., Bossert, M., Zyzblov, V., and Sidorenko, V., "Array codes correcting a two-dimensional cluster of errors," *IEEE Transactions on Information Theory*, vol. 44, pp. 2025–2031, September 1998.

[11] Byers, J. W., Luby, M., and Miltzenmacher, M., "A digital fountain approach to asynchronous reliable multicast," *IEEE Journal on Selected Areas in Communication*, pp. 1528–1540, October 2002.

[12] Cabarcas, F. and Garcia-Frias, J., "Approaching the Slepian-Wolf boundary using practical channel codes," *Proc. IEEE International Symposium on Information Theory*, p. 330, June 2004.

[13] Cover, T. and Thomas, J., *Elements of Information Theory.* 1991.

[14] Dana, A. F., Gowaikar, R., Palanki, R., Hassibi, B., and Effros, M., "On the capacity of erasure wireless networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 789 – 804, March 2006.

[15] Delgosha, F. and Fekri, F., "Factorization of two-channel 2D paraunitary filter banks over fields of characteristic two," *Proc. IEEE International Symposium on Information Theory*, p. 565, June 2004.

[16] Elspas, B., "Notes on multidimensional burst-error correction," *Proc. IEEE International Symposium on Information Theory*, September 1967.

[17] Etesami, O., Molkaraie, M., and Shokrollahi, A., "Raptor codes on symmetric channels," *Proc. IEEE International Symposium on Information Theory*, p. 38, June 2004.

[18] Fekri, F., McLaughlin, S. W., Mersereau, R. M., and Schafer, R. W., *Double Circulant Self-Dual Codes Using Finite-Field Transforms.* Lecture Notes in Computer Science, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 1999.

[19] Fekri, F., Mersereau, R. M., and Schafer, R. W., "Two-band wavelets and filter banks over finite fields with connections to error control coding," *IEEE Transactions on Signal Processing*, vol. 51, pp. 3143–3151, December 2003.

[20] Gallager, R., "Low density parity check codes," *MIT Press*, 1963.

[21] Gallager, R. G., "Low-density parity-check codes," *IRE Trans, Information Theory*, vol. IT-8, pp. 21–28, January 1962.

[22] Garcia-Frias, J., "Joint source-channel decoding of correlated sources over noisy channels," *Proc. IEEE Data Compression Conference*, pp. 283–292, March 2001.

[23] Garcia-Frias, J. and Zhao, Y., "Compression of correlated binary sources using turbo codes," *IEEE Communication Letters*, vol. 5, pp. 417–419, October 2002.

[24] Gehrig, N. and Dragotti, P., "Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes," *IEEE Communication Letters*, vol. 9, pp. 61–63, January 2005.

[25] Ha, J., Kim, J., and McLaughlin, S. W., "Puncturing for finite length low-density parity-check codes," *Proc. IEEE International Symposium on Information Theory*, p. 151, July 2004.

[26] Imai, H., "Two-dimensional Fire codes," *IEEE Transactions on Information Theory*, vol. 19, pp. 796–806, November 1973.

[27] Karlsson, G. and Vetterli, M., "Theory of two-dimensional multirate filter banks," *IEEE Transactions on Acoustic, Speech, Signal Processing*, vol. 38, pp. 925–937, June 1990.

[28] Keren, O. and Litsyn, S., "Codes correcting phased burst erasures," *IEEE Transactions on Information Theory*, vol. 44, pp. 416 – 420, January 1998.

[29] Keren, O. and Litsyn, S. N., "A class of array codes correcting multiple column erasures," *IEEE Transactions on Information Theory*, vol. 43, pp. 1843–1851, November 1997.

[30] Koetter, R. and Médard, M., "Beyond routing: an algebraic approach to network coding," *Proc. IEEE INFOCOM*, pp. 122–130, June 2002.

[31] Kovacevic, J. and Vetterli, M., "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathcal{R}^n$," *IEEE Transactions on Information Theory*, vol. 38, pp. 533–555, March 1992.

[32] Krasikov, I. and Litsyn, S., "An improved upper bound on the minimum distance of doubly-even self-dual codes," *IEEE Transactions on Information Theory*, vol. 46, pp. 274–278, January 2000.

[33] Li, S.-Y. R., Yeung, R. W., and Cai, N., "Linear network coding," *IEEE Transactions on Information theory*, pp. 371–381, February 2003.

[34] Lin, S. and Jr., D. J. C., *Error control coding: fundamentals and applications.* Prentice-Hall, 1983.

[35] Liveris, A. D., Lan, C., Narayanan, K. R., Xiong, Z., and Georghiades, C. N., "Slepian-Wolf coding of three binary sources using LDPC codes," *Proc. International Symposium on Turbo codes*, September 2003.

[36] Liveris, A. D., Xiong, Z., and Georghiades, C. N., "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communication Letters*, vol. 6, pp. 440–442, October 2002.

[37] Liveris, A. D., Xiong, Z., and Georghiades, C. N., "Compression of binary sources with side information at the decoder using low-density parity-check codes," *Proc. IEEE Globecom*, pp. 1300–1304, November 2002.

[38] Liveris, A. D., Xiong, Z., and Georghiades, C. N., "A distributed source coding technique for correlated image using turbo codes," *IEEE Communication Letters*, vol. 6, pp. 379–381, September 2002.

[39] Liveris, A. D., Xiong, Z., and Georghiades, C. N., "Joint source-channel coding of binary sources with side information at the decoder using IRA codes," *Proc. IEEE Workshop on Multimedia Signal Processing*, pp. 53–56, December 2002.

[40] Liveris, A. D., Xiong, Z., and Georghiades, C. N., "Nested convolutional/turbo codes for the binary Wyner-Ziv problem," *Proc. International Conference on Image Processing*, pp. 601–604, 2003.

[41] Luby, M. G., "LT codes," *Proc. IEEE Symposium on the Foundations of Computer Science*, pp. 271–280, November 2002.

[42] Lun, D. S., Médard, M., and Koetter, R., "Efficient operation of wireless packet networks using network coding," *Proc. International Workshop on Convergent Technologies*, June 2005.

[43] MacKay, D., "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 399–431, March 1999.

[44] Maymounkov, P., "Online codes," *NYU Technical Report TR2003-883*, 2002.

[45] Miltzenmacher, M., "Digital fountain: A survey and look forward," *Proc of Information Theory Workshop*, pp. 271–276, October 2004.

[46] Mitran, P. and Bajcy, J., "Turbo source coding: A noise-robust approach to data compression," *Proc. IEEE Data Compression Conference*, p. 465, April 2002.

[47] Pakzad, P., Fragouli, C., and Shokrollahi, A., "Coding schemes for line networks," *Proc. IEEE International Symposium on Information Theory*, pp. 1853 – 1857, September 2005.

[48] Pishro-Nik, H. and Fekri, F., "Results on punctured low-density parity-check codes and improved iterative decoding techniques," *Submitted to IEEE Transactions on Information Theory.*

[49] Pishro-Nik, H. and Fekri, F., "Results on punctured LDPC codes," *Proc. IEEE Information Theory Workshop*, pp. 215–219, October 2004.

[50] Pishro-Nik, H., Rahnavard, N., and Fekri, F., "Non-uniform error correction using low-density parity-check," *IEEE Transactions on Information Theory*, vol. 51, pp. 2702 – 2714, July 2005.

[51] Pradhan, S. S. and Ramchandran, K., "Distributed source coding using syndromes (DISCUS): design and construction," *Proc. IEEE Data Compression Conference*, pp. 158–167, March 1999.

[52] Pradhan, S. S. and Ramchandran, K., "Distributed source coding: symmetric rates and applications to sensor networks," *Proc. IEEE Data Compression Conference*, pp. 363–372, March 2000.

[53] Rahnavard, N. and Fekri, F., "Unequal error protection using low-density parity-check codes," *Proc. IEEE International Symposium on Information Theory*, p. 449, June 2004.

[54] Rahnavard, N. and Fekri, F., "CRBcast: A collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks," *Proc. ACM/IEEE International Conference on Information Processing in Sensor Networks*, April 2006.

[55] Rahnavard, N., R., B. N. V., and Fekri, F., "Efficient broadcasting via rateless coding in multihop wireless networks with local information," *submitted to the IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2006.

[56] Reiger, S., "Codes for the correction of clustered errors," *IEEE Transactions on Information Theory*, vol. 6, pp. 16–21, March 1960.

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

[58] Sankarasubramaniam, Y., Akyildiz, I. F., and McLaughlin, S. W., "Energy efficiency based packet size optimization in wireless sensor networks," *Proceedings IEEE International Workshop on Sensor Network Protocols and Applications*, April 2003.

[59] Sartipi, M. and Fekri, F., "Source and channel coding in wireless sensor networks using LDPC codes," *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 309–316, October 2004.

[60] Sartipi, M. and Fekri, F., "Distributed source coding in wireless sensor networks using LDPC coding: a non-uniform framework," *Proc. IEEE Data Compression Conference*, pp. 477 – 477, March 2005.

[61] Sartipi, M. and Fekri, F., "Distributed source coding in wireless sensor networks using LDPC coding: The entire Slepian-Wolf rate region," *Proc. IEEE Wireless Communications and Networking Conference*, pp. 1939–1944, March 2005.

[62] Sartipi, M. and Fekri, F., "Distributed source coding in wireless sensor networks using rate-compatible LDPC codes: The entire slepian-wolf rate region," *Submitted to IEEE Transactions on Communications*, December 2005.

[63] Sartipi, M. and Fekri, F., "Distributed source coding using LDPC codes: lossy and lossless cases with unknown correlation parameter," *Proc. Allerton Conference on Communication, Control and Computing*, October 2005. Invited paper.

[64] Sartipi, M. and Fekri, F., "Two-dimensional error correcting codes using finite-field wavelets," *Proc. IEEE Information Theory Workshop*, pp. 22–27, October 2004.

[65] Schonberg, D., Ramchandran, K., and Pradhan, S. S., "LDPC codes can approach the Slepian-Wolf bound for general binary sources," *Proc. Allerton Conference on Communication, Control and Computing*, October 2002.

[66] Schonberg, D., Ramchandran, K., and S. S, P., "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," *Proc. IEEE Data Compression Conference*, March 2004.

[67] Shamai, S., Verdu, S., and Zamir, R., "Systematic lossy source/channel coding," *IEEE Transactions on Information Theory*, vol. 44, pp. 564 – 579, March 1998.

[68] Shokrollahi, A., "Raptor codes," *Proc. IEEE International Symposium on Information Theory*, p. 36, June 2004.

[69] Slepian, D. and Wolf, J. K., "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471–480, July 1973.

[70] Stankovic, V., Liveris, A. D., Xiong, Z., and Georghiades, C. N., "Design of Slepian-Wolf codes by channel code partitioning," *Proc. IEEE Data Compression Conference*, pp. 302–311, March 2004.

[71] Vaidyanathan, P. P., "A tutorial on multirate digital filter banks," *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 2241–2248, June 1988.

[72] Vaidyanathan, P. P., *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[73] Wieselthier, E., Nguyen, G. D., and Ephremides, A., "Energy-efficient broadcast and multicast trees in wireless networks," *ACM/Kluwer Journal on Mobile Networks and Applications*, pp. 481–492, June 2002.

[74] Wolf, J. K., "Data reduction for multiple correlated sources," *Proc. 5th Colloquium Microwave Communication*, pp. 287–295, June 1973.

[75] Wyner, A., "Recent results in the Shannon theory," *IEEE Transactions on Information Theory*, vol. 20, pp. 2–10, January 1974.

[76] Wyner, A. and Ziv, J., "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, pp. 1–10, January 1976.

VITA

Mina Sartipi received her B.S. Degree in Electrical Engineering from the Sharif University of Technology, Tehran, Iran in 2001 and her M.S. Degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA in 2003. She received her Ph.D. Degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA in 2006. Her graduate research focused on the general area of communications; in particular, wavelets and filter banks, error control coding and communication networks.