Low-density Parity-check Codes for Wireless Relay Networks

by

Xinsheng Zhou

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In wireless networks, it has always been a challenge to satisfy high traffic throughput demands, due to limited spectrum resources. In past decades, various techniques, including cooperative communications, have been developed to achieve higher communication rates.

This thesis addresses the challenges imposed by cooperative wireless networks, in particular focusing on practical code constructions and designs for wireless relay networks. The thesis is divided into the following four topics: 1) constructing and designing low-density parity-check (LDPC) codes for half-duplex three-phase two-way relay channels, 2) extending LDPC code constructions to half-duplex three-way relay channels, 3) proposing maximum-rate relay selection algorithms and LDPC code constructions for the broadcast problem in wireless relay networks, and 4) proposing an iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels.

Under the first topic, we construct codes for half-duplex three-phase two-way relay channels where two terminal nodes exchange information with the help of a relay node. Constructing codes for such channels is challenging, especially when messages are encoded into multiple streams and a destination node receives signals from multiple nodes. We first prove an achievable rate region by random coding. Next, a systematic LDPC code is constructed at the relay node where relay bits are generated from two source codewords. At the terminal nodes, messages are decoded from signals of the source node and the relay node. To analyze the performance of the codes, discretized density evolution is derived. Based on the discretized density evolution, degree distributions are optimized by iterative linear programming in three steps. The optimized codes obtained are 26% longer than the theoretic ones.

For the second topic, we extend LDPC code constructions from half-duplex three-phase two-way relay channels to half-duplex three-way relay channels. An achievable rate region of half-duplex three-way relay channels is first proved. Next, LDPC codes for each subregion of the achievable rate region are constructed, where relay bits can be generated only from a received codeword or from both the source codeword and received codewords.

Under the third topic, we study relay selection and code constructions for the broadcast problem in wireless relay networks. We start with the system model, followed by a theorem stating that a node can decode a message by jointly decoding multiple blocks of received signals. Next, the maximum rate is given when a message is decoded hop-by-hop or decoded by a set of nodes in a transmission phase. Furthermore, optimal relay selection algorithms are proposed for the two relay schemes. Finally, LDPC codes are constructed for the broadcast problem in wireless relay networks.

For the fourth topic, an iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels is proposed. The decoder is based on log-likelihood ratios (LLRs). Interference is estimated, quantized and subtracted from channel outputs. To analyze the codes, density evolution is derived. We show that the required signal-to-noise ratio (SNR) for the proposed low-complexity decoder is 0.2 dB higher than that for an existing sub-optimal belief propagation decoder at code rate $\frac{1}{3}$.

Acknowledgements

First and foremost, I would like to express my deepest and sincerest gratitude to my co-supervisors, Professor Liang-Liang Xie and Professor Xuemin (Sherman) Shen. Without their continuous guidance, high standard on research quality and strong commitment, this thesis would not have been possible. Their deep insights help me to understand the topic in a high level. Their inspiration encourages me to the end.

I would also like to thank my thesis examining committee members: Professor Jun Chen, Professor Xinzhi Liu, Professor Ajit Singh and Professor Zhou Wang. Their valuable suggestions help me to improve the quality of the thesis. I highly appreciate their efforts and their precious time.

Furthermore, the knowledge I learned from various courses, both in the department of electrical and computer engineering and the faculty of mathematics, has been the foundation of the research work in this thesis. I would like to thank all the professors for their professional teaching.

Special thanks are due to my colleagues: Xiugang Wu, Xiaoxia Zhang, Jian Qiao, Shamsul Alam and Mingxi Nan. They read my writings and corrected errors that I did not notice. In addition, I am fortunate to work with members of the Broadband Communications Research (BBCR) group and the cooperative networks subgroup. The group meetings and discussions have expanded my knowledge and kept me on up-to-date topics.

With great appreciation, I acknowledge the financial support from Ontario Graduate Scholarship (OGS) and the bursary provided by University of Waterloo during my parental leave.

Finally, I am indebted to my parents for their sacrifice and silent support. I am grateful to my wife, Haiyan Wang, for her endless support for the whole family. Special thanks go to my children, Yiran and Maggie, who bring me the greatest joy in life.

Dedication

To my wife Haiyan, children Yiran and Maggie.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AWGN	additive white Gaussian noise
BER	bit error rate
BIAWGN	binary input additive white Gaussian noise
BPSK	binary phase-shift keying
LDGM	low-density generator matrix
LDPC	low-density parity-check
LHS	left-hand side
LLR	log-likelihood ratio
LT	Luby transform
MAP	maximum a posterior
MIMO	multiple-input and multiple-output
RHS	right-hand side
SNR	signal-to-noise ratio

List of Notations

$A_\epsilon^{(n)}$	set of joint typical sequences $\{(x^n, y^n)\}$
$H(X)$	entropy of a discrete random variable X
$H(X, Y)$	joint entropy of a pair of discrete random variables (X, Y)
$I(X; Y)$	mutual information of X and Y
λ	variable node degree distribution
$\mathcal{O}$	asymptotic upper bound
ψ_c	check node function
ψ_v	variable node function
ρ	check node degree distribution
$\mathbf{H}$	parity-check matrix
d_c	degree of a check node
d_v	degree of a variable node

Chapter 1

Introduction

1.1 Research motivations and objectives

Satisfying high traffic throughput demands in wireless networks is an on-going challenge. The throughput is capped by limited spectrum resources, due to shared medium. Cooperative communications and various other techniques have been developed in recent decades to achieve higher communication rates.

When signals are broadcast from a source node to a destination node, only a small fraction of the energy is received at the destination node directly. If received signals at other nodes can be utilized, less energy or a higher transmission rate can be achieved. In addition, more reliable communications can be achieved when messages are received through multiple paths. Cooperative communications are similar to the multiple-input and multiple-output (MIMO) system where multiple data streams are received by multiple antennas, but the distance between transmitters or the distance between receivers is much longer.

This thesis focuses on practical coding schemes for wireless relay networks. Coding in such networks is challenging since messages must generally be encoded into multiple streams and jointly decoded from multiple blocks of signals. The thesis is divided into the following four topics.

The first topic explores practical coding schemes for two-way relay channels. In such channels, two source nodes exchange information with the help of a relay node. Military and disaster response applications are among potential applications, e.g., establishing a wireless link with the help of a relay node to enhance reliability or increase throughput. In half-duplex two-phase two-way relay channels, signals from the two source nodes are superimposed at the relay node. Without considering the noise, such superimposed signals can be thought as physical-layer network coding, where channel gains are considered as coefficients of a linear combination of the two source codewords. Different strategies, such as amplify-and-forward, decode-and-forward and denoise-and-forward [1], have been proposed to process the superimposed signals. In the amplify-and-forward strategy, superimposed signals are amplified and broadcast to the two destination nodes. Since no decoding is performed, noises are kept and amplified. In the decode-and-forward strategy, the two source codewords are decoded before a relay codeword is generated. However, when two signals $x_1, x_2 \in \{-1, +1\}$ are superimposed, the outcome signal is $x_1 + x_2 \in \{-2, 0, +2\}$. Even without the noise, x_1 and x_2 cannot be determined when $x_1 + x_2 = 0$. Thus, the decode-and-forward strategy is over-decoding. Based on this observation, in the denoise-and-forward strategy, only the superimposed codeword $\mathbf{x}_1 + \mathbf{x}_2$ is decoded, with no need to decode both $\mathbf{x}_1$ and $\mathbf{x}_2$.

In half-duplex two-phase two-way relay channels, due to the half-duplex constraint, the destination node cannot utilize signals from the source node. In addition, the complexity of processing superimposed signals is high. These two disadvantages have motivated some recent studies on half-duplex three-phase two-way relay channels. The relay codeword can be generated by simply adding the two source codewords in GF(2). However, this strategy is not optimal if links between the source node and the relay node are asymmetric, since equal amounts of information from the two codewords are included in the relay codeword. Hence, the objective of the first topic is to construct and design practical codes for the half-duplex three-phase two-way relay channels. This work proposes systematic low-density parity-check (LDPC) codes at the relay node. With this code construction, unequal amounts of information can be included in the relay bits.

LDPC code constructions can be further extended from half-duplex two-way relay channels to half-duplex three-way relay channels. In the latter channels, each node broadcasts its own messages to the other two nodes. Meanwhile, the node would help to relay messages when necessary. In the second topic, we are mainly interested in the following two questions: What is the achievable rate region of half-duplex three-way relay channels? How can one construct LDPC codes for such channels?

In the third topic, we extend the study of 3-node networks to wireless relay networks. In particular, we consider the broadcast problem and ask the following questions when cooperative communications are considered: What is the maximum rate for the broadcast problem in a given wireless relay network? How can one efficiently find the optimal relay route? How can one construct LDPC codes for the broadcast problem in wireless relay networks?

In the fourth topic, we study decoding algorithms for LDPC codes in 2-user multiple-access channels. These channels are in the first phase of the two-phase two-way relay channels where signals of two source codewords are superimposed. Since the complexity of an existing sub-optimal belief propagation joint decoder is high, we propose a simplified iterative hard interference cancellation decoder in the fourth topic of this thesis.

1.2 Research contributions

The main objective of this thesis is to construct and design LDPC codes for wireless relay networks. We next list the main contributions of this work.

For the first topic, we study half-duplex three-phase two-way relay channels where two terminal nodes exchange information with the help of a relay node. Designing practical coding schemes for such channels is challenging, especially when messages are encoded into multiple streams and a destination node receives signals from multiple nodes. The main contributions under this topic are as follows.

- An achievable rate region of half-duplex three-phase two-way relay channels is proved.
- Inspired by random binning, systematic LDPC codes are proposed for the relay node. Source codeword pairs generating the same relay codeword are in the same bin. By this code construction, we can effectively control the amounts of information included in the relay codeword from the two source codewords.
- A joint decoder for terminal nodes is designed, where signals received from the source node, signals received from the relay node and the codeword of the destination node are used.
- To analyze the performance of the codes, discretized density evolution is derived for the decoder of the terminal nodes.
- Based on the discretized density evolution, degree distributions are optimized by iterative linear programming in three steps.
- The length of the obtained optimized codes is 26% longer than the theoretic one.

In the second topic, we extend the study from half-duplex three-phase two-way relay channels to half-duplex three-way relay channels. Our main contributions are as follows.

- An achievable rate region of half-duplex three-way relay channels is proved.
- LDPC codes for each sub-region of the achievable rate region are constructed.

In the third topic, we study relay selection and LDPC code constructions for the broadcast problem in wireless relay networks when cooperative communications are considered. The main contributions under this topic are the following.

- We prove a theorem stating that a node can decode a message by jointly decoding multiple blocks of received signals even if it cannot decode the message from any single block of received signals.

- The maximum rates are given when the message is decoded hop-by-hop or decoded by a set of nodes in a transmission phase.

- In contrast to the complexity of $(N-1)!$ in a full search, we propose an optimal relay selection algorithm for the hop-by-hop relay with the complexity of $\mathcal{O}(N^2)$.

- Two theorems that can prune the search space are proved. Based on the two theorems, an optimal relay selection algorithm is proposed for the level-by-level relay.

- LDPC codes for the broadcast problem in wireless relay networks are constructed.

In the fourth topic, an iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels is proposed. In such channels, two users send their codewords to a destination node simultaneously. The main contributions in this topic are as follows.

- A graph of the codes in 2-user multiple-access channels is given, and includes variable nodes, check nodes and multiple-access nodes. By this representation, decoding algorithms can easily be described as message-passing algorithms.

- An iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels is proposed. The decoder is based on log-likelihood ratios (LLRs), and interference is estimated, quantized and subtracted from channel outputs.

- To analyze the performance of the codes, density evolution is derived.

- It is shown that the required signal-to-noise ratio (SNR) for the proposed low-complexity decoder is 0.2 dB higher than that for an existing sub-optimal belief propagation decoder at code rate $\frac{1}{3}$.

1.3 Outline of the thesis

This thesis is focused on constructing and designing LDPC codes for wireless relay networks. The remainder of the thesis is organized as follows. It begins with definitions and an introduction to LDPC codes in Chapter 2. Chapter 3 covers the construction and design of LDPC codes for half-duplex three-phase two-way relay channels where two nodes communicate with each other with the help of a relay node. The code constructions are extended to half-duplex three-way relay channels in Chapter 4. Chapter 5 explores relay selection and LDPC code constructions for the broadcast problem in wireless relay networks. An iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels is proposed in Chapter 6. Finally, Chapter 7 concludes this thesis and lists possible future work.

Chapter 2

Backgrounds

2.1 Definitions and a theorem

Definition 1. *(In [2] Chapter 2) The entropy $H(X)$ of a discrete random variable X is defined as*

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{2.1}$$

Definition 2. *(In [2] Chapter 2) The joint entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined as*

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y). \tag{2.2}$$

Definition 3. *(In [2] Chapter 2) The mutual information $I(X; Y)$ is defined as*

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \tag{2.3}$$

Definition 4. *(In [2] Chapter 7) The set $A_\epsilon^{(n)}(X, Y)$ of jointly typical sequence $\{(x^n, y^n)\}$ with respect to the distribution $p(x, y)$ is the set of n-sequences with empirical entropies*

ϵ-close to the true entropies:

$$
\begin{aligned}
A_\epsilon^{(n)} = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n :\\
\left| -\frac{1}{n} \log p(x^n) - H(X) \right| < \epsilon,\\
\left| -\frac{1}{n} \log p(y^n) - H(Y) \right| < \epsilon,\\
\left| -\frac{1}{n} \log p(x^n, y^n) - H(X, Y) \right| < \epsilon\},
\end{aligned}
$$

where

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i). \tag{2.4}$$

Theorem 1. *(Theorem 7.6.1 in [2]) Let (X^n, Y^n) be sequences of length n drawn i.i.d. according to $p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i)$. If $(\tilde{X}_n, \tilde{Y}_n) \sim p(x^n)p(y^n)$, then $Pr((\tilde{X}^n, \tilde{Y}^n) \in A_\epsilon^{(n)}) \leq 2^{-n(I(X;Y)-3\epsilon)}$.*

2.2 Low-density parity-check codes

2.2.1 Introduction

In 1948, Claude Shannon published his paper *A mathematical theory of communication* [3] where a fundamental question of communication systems was answered. He showed that, for any given channel bandwidth and signal-to-noise ratio (SNR), there exists a rate below which messages can be reliably decoded. After five decades of research, Turbo codes [4] became the first capacity-approaching codes with affordable complexity. Shortly after, low-density parity-check (LDPC) codes [5] were found to be another class of capacity-approaching codes [6]. LDPC codes were first introduced by Gallager in 1961. However, they were largely forgotten for almost three decades because the proposed decoding algorithm was simply too complicated to be implemented at that time. Unlike the Viterbi algorithm [7] which is a maximum-likelihood decoding algorithm on symbol sequences, the LDPC decoding algorithm is a maximum a posterior (MAP) algorithm on each individual symbol. In [8], an LDPC code with a threshold within 0.0045 dB of the Shannon limit was reported.

In LDPC codes, the fraction of ones in the parity check matrix is small. Due to this sparseness property, the decoding complexity is on the order of the codeword length. In addition, soft decoding is used, in which bits are represented by probability values. The algorithms also decode messages with multiple iteration, e.g., outputs from a decoding iteration are used as inputs in the next decoding iteration. During the entire iterative decoding process, belief on bits is gradually strengthened and the bit error rate (BER) is gradually decreased.

2.2.2 Message-passing algorithms

In this section, message-passing algorithms for LDPC codes are briefly reviewed.

An LDPC code can be represented by a parity-check matrix **H** which can be mapped to a corresponding Tanner graph [9]. In such a graph, each codeword bit corresponds to a variable node and each parity check constraint corresponds to a check node. An edge connecting a variable node and a check node corresponds to a 1 in the **H**. Hence, the Tanner graph is a bipartite graph. With the help of the Tanner graph, the message-passing algorithms can be easily described. Variable nodes and check nodes are associated with decoding functions. Messages flow between variable nodes and check nodes via edges, serving as inputs and outputs of the functions. As an example, the Tanner graph of the following parity check matrix

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

is shown in Figure 2.1, where a circle is a variable node and a square is a check node.

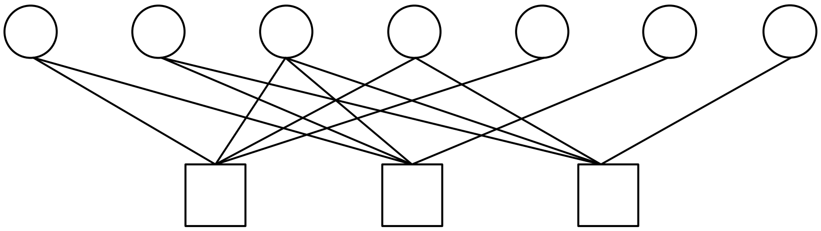

Figure 2.1: Corresponding Tanner graph of a parity check matrix

Assume the channel is $Y = X + Z$ where $p(X = 1) = p(X = -1) = 0.5$ and Z is a Gaussian random variable with mean zero and variance σ^2. A binary sequence $\mathbf{b} = [b_1, b_2 \ldots, b_n]$ is mapped to $\mathbf{x} = [x_1, x_2 \ldots, x_n]$ from $b_i \in \{0, 1\}$ to $x_i \in \{-1, 1\}$ for $i = 1, 2, \cdots, n$. At the receiver, the bit b_i can be represented by a probability value pair $(p(x_i = 1|y_i), p(x_i = -1|y_i))$, or a log-likelihood ratio (LLR)

$$l_i(y_i) = \ln \frac{p(x_i = 1|y_i)}{p(x_i = -1|y_i)} \tag{2.5}$$

$$= \ln \frac{e^{-\frac{1}{2}(\frac{y_i-1}{\sigma})^2}}{e^{-\frac{1}{2}(\frac{y_i+1}{\sigma})^2}} \tag{2.6}$$

$$= \frac{2y_i}{\sigma^2}. \tag{2.7}$$

The message-passing algorithm based on the LLRs can be briefly summarized as follows. Note that step 2 to step 4 are repeated for iterative decoding until the maximum allowed iteration number is reached.

Step 1: Calculate the LLR of the channel output $l_i = \frac{2y_i}{\sigma^2}$ for $i = 1, 2, \cdots, n$.

Step 2: Calculate the check node function. Assume three bits b_1, b_2, b_3 satisfy the parity check constraint $b_1 \oplus b_2 \oplus b_3 = 0$. When b_2 and b_3 are represented by LLRs $v_2 = \ln\left(\frac{p(x_2 = 1)}{p(x_2 = -1)}\right)$ and $v_3 = \ln\left(\frac{p(x_3 = 1)}{p(x_3 = -1)}\right)$ respectively, the LLR of b_1 can be calculated in the check node as

$$\begin{aligned}
c_1 = \ln\left(\frac{p(x_1=1)}{p(x_1=-1)}\right) &= \ln\left(\frac{p(x_2=1)p(x_3=1)+p(x_2=-1)p(x_3=-1)}{p(x_2=1)p(x_3=-1)+p(x_2=-1)p(x_3=1)}\right) && (2.8)\\
&= \ln\left(\frac{\frac{1}{2}(e^{\frac{v_2+v_3}{2}}+e^{-\frac{v_2+v_3}{2}})}{\frac{1}{2}(e^{\frac{v_2-v_3}{2}}+e^{-\frac{v_2-v_3}{2}})}\right) && (2.9)\\
&= \ln\left(\frac{\cosh(\frac{v_2+v_3}{2})}{\cosh(\frac{v_2-v_3}{2})}\right) && (2.10)\\
&= \ln\left(\frac{\cosh(\frac{v_2}{2})\cosh(\frac{v_3}{2})+\sinh(\frac{v_2}{2})\sinh(\frac{v_3}{2})}{\cosh(\frac{v_2}{2})\cosh(-\frac{v_3}{2})+\sinh(\frac{v_2}{2})\sinh(-\frac{v_3}{2})}\right) && (2.11)\\
&= \ln\left(\frac{1+\dfrac{\sinh(\frac{v_2}{2})\sinh(\frac{v_3}{3})}{\cosh(\frac{v_2}{2})\cosh(\frac{v_3}{2})}}{1-\dfrac{\sinh(\frac{v_2}{2})\sinh(\frac{v_3}{3})}{\cosh(\frac{v_2}{2})\cosh(\frac{v_3}{2})}}\right) && (2.12)\\
&= \ln\left(\frac{1+\tanh(\frac{v_2}{2})\tanh(\frac{v_3}{2})}{1-\tanh(\frac{v_2}{2})\tanh(\frac{v_3}{2})}\right) && (2.13)\\
&= 2\tanh^{-1}\left(\tanh\left(\frac{v_2}{2}\right)\tanh\left(\frac{v_3}{2}\right)\right) && (2.14)
\end{aligned}$$

where $\cosh(x) = \frac{1}{2}(e^x + e^{-x})$, $\cosh(x \pm y) = \cosh(x)\cosh(y) \pm \sinh(x)\sinh(y)$, $\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$ and $\tanh^{-1}(x) = \frac{1}{2}\ln\left(\frac{1+x}{1-x}\right), |x| < 1$. The variable (check) node degree is defined as the number of edges connected to the variable (check) node. A degree-i node is a node with i edges. A degree-i edge is an edge that is connected to a degree-i node. Assume the degree of a check node is m. Thus, (2.14) can be extended to

$$c_i^k = 2 \times \tanh^{-1}\left(\prod_{j/i}\tanh\left(\frac{v_j^k}{2}\right)\right) \tag{2.15}$$

where v_j^k is the LLR received from the j-th connected variable node, c_i^k is the LLR sent to the i-th connected variable node, k is the decoding iteration index, $i, j = 1, 2, \cdots, m$ and $j/i = 1, 2, \cdots, i-1, i+1, \cdots, m$.

Step 3: Calculate the variable node function. Assume a variable node receives two LLRs $c_1 = \ln\left(\dfrac{p_1(x_1=1)}{p_1(x_1=-1)}\right)$ and $c_2 = \ln\left(\dfrac{p_2(x_1=1)}{p_2(x_1=-1)}\right)$ from check nodes or channel outputs. The LLR of the bit given c_1 and c_2 is

$$\ln\left(\frac{p(x_1=1)}{p(x_1=-1)}\right) = \ln\left(\frac{p_1(x_1=1)p_2(x_1=1)}{p_1(x_1=-1)p_2(x_1=-1)}\right) \tag{2.16}$$

$$= \ln\left(\frac{p_1(x_1=1)}{p_1(x_1=-1)}\right) + \ln\left(\frac{p_2(x_1=1)}{p_2(x_1=-1)}\right) \tag{2.17}$$

$$= c_1 + c_2. \tag{2.18}$$

Assume the degree of a variable node is m. Thus, (2.18) can be extended to

$$v_i^{k+1} = l + \sum_{j/i} c_j^k \tag{2.19}$$

where c_j^k is the LLR from the j-th connected check node, v_i^{k+1} is the LLR sent to the i-th connected check node, l is the channel output LLR, k is the decoding iteration index, $i, j = 1, 2, \cdots, m$ and $j/i = 1, 2, \cdots, i-1, i+1, \cdots, m$.

Step 4: Calculate the posteriori probability. Assume the degree of the variable node is m. The posteriori probability of the bit is

$$l + \sum_{j} c_j^k \tag{2.20}$$

where c_j^k is the LLR from the j-th connected check node, l is the channel output LLR, k is the decoding iteration index and $j = 1, 2, \cdots, m$. If the result is greater than or equal to 0, $b_i = 0$. Otherwise, $b_i = 1$. Finally, the decoded codeword is verified by testing $\mathbf{H} \times \mathbf{b}^{\mathrm{T}} = \mathbf{0}$. If $\mathbf{b}$ is a valid codeword, the iterative decoding process is stopped.

2.2.3 Density evolution

In this section, density evolution [10, 11, 12, 13] for LDPC codes is reviewed. Density evolution is an analytic tool to track the probability density function (or simply called density) of the LLR message (or simply called message) during iterative decoding.

Let us first review regular LDPC codes. In Gallager's original work [5], a (d_v,d_c) regular LDPC code is a binary linear code whose parity check matrix $\mathbf{H}$ has d_v ones in each column and d_c ones in each row. The total number of ones in the $\mathbf{H}$ is linear with the block length n. A cycle is defined as a closed path in the bipartite graph. The start vertex and the end vertex of a cycle are the same vertex. The length of the cycle is defined as the number of edges in the cycle. To analyze the codes, Gallager put codes without short cycles in a code ensemble. To be specific, the codes in the ensemble do not contain cycles with the length less or equal to

$$\frac{2(\ln n - \ln \frac{d_v d_c - d_v - d_c}{2d_c})}{\ln\left[(d_c-1)(d_v-1)\right]}. \tag{2.21}$$

Unfortunately, Gallager codes are difficult to construct. In [14], Luby *et al.* introduced a new code ensemble based on degree distributions. With this approach, codes in the ensemble are allowed to have short cycles. Furthermore, sampling a code is almost trivial.

The ensemble of (d_v, d_c) regular LDPC codes is defined as follows [10]. In a bipartite graph, assign d_v (d_c) sockets to every variable (check) node and label these sockets as positive integers from 1 to nd_v. Pick a permutation π on check node socket labels with uniform distribution. Each code in the ensemble corresponds to a permutation. An edge of the bipartite graph is a mapping of the socket pair (i,$\pi(i)$) where i is a variable node socket and $\pi(i)$ is a check node socket.

Irregular LDPC codes were first introduced in [15] and further studied in [14, 13, 10, 11]. Unlike regular LDPC codes, the degree of variable (check) nodes could be different. The degree of each node is chosen according to the degree distribution.

Denote λ as the variable node degree distribution. λ_i is the fraction of edges which are connected to degree-i variable nodes. $\sum_i \lambda_i = 1$. Denote ρ as the check node degree distribution. ρ_i is the fraction of edges which are connected to degree-i check nodes. $\sum_i \rho_i = 1$. The rate of irregular LDPC codes for a given degree distribution (λ, ρ) is

$$R = 1 - \frac{\sum_j \frac{\rho_j}{j}}{\sum_i \frac{\lambda_i}{i}}. \tag{2.22}$$

For a given degree distribution (λ, ρ) and codeword length n, an ensemble of irregular

LDPC codes is defined by a permutation on all edges. All codes in the ensemble are equiprobable.

It is proved in [10] that if the following three symmetric conditions on the channel and the decoding algorithm are met, the probability of error is independent of the codeword. Hence, to simplify the analysis, it is assumed that only the codeword $\mathbf{b} = [0, 0, \cdots, 0]$ is transmitted in density evolution.

Channel symmetry:

$$p(y_i = q|x_i = 1) = p(y_i = -q|x_i = -1) \tag{2.23}$$

Check node symmetry:

$$\psi_c(x_1 m_1, x_2 m_2, \cdots, x_{d_c-1} m_{d_c-1}) = -\psi_c(m_1, m_2, \cdots, m_{d_c-1}) \prod_{i=1}^{d_c-1} x_i \tag{2.24}$$

where ψ_c is a check node function and $\mathbf{x} = [x_1, x_2, \cdots, x_{d_c-1}]$ is a ±1 sequence.

Variable node symmetry:

$$\psi_v(-m_0, \cdots, -m_{d_v-1}) = -\psi_v(m_0, \cdots, m_{d_v-1}) \tag{2.25}$$

where ψ_v is a variable node function.

Density evolution tracks the probability density function of LLR messages. Messages on each edge can be represented by a random variable. The output messages of a check node function (2.15) and a variable node function (2.19) can be represented by functions of random variables. If threshold decoding (A bit is decoded as 0 if the message is greater than or equal to zero and decoded as 1 if the message is less than 0) is used, the probability of error is simply the integral of the probability density function from $-\infty$ to 0.

Generalized density evolution for regular LDPC codes and irregular LDPC codes on binary-input memoryless channels has been derived in [10] and [11]. Discretized density evolution [8] and Gaussian approximation [12] are the other two approaches for additive white Gaussian noise (AWGN) channels. We include them here as an introduction.

Let us first consider variable node functions. In message-passing algorithms, the output of a variable node function (2.19) is a sum of multiple messages. Denote the probability density functions of l and c_j^k as f_l and $f_{c_j^k}$. Assume these messages are independent. From probability theory, the probability density function of the sum of two independent random variables is the convolution of the probability density function of the two independent random variables. Hence the probability density function of v_i^{k+1} is

$$f_l \otimes f_{c_1^k} \otimes f_{c_2^k} \cdots \otimes f_{c_{d_v-1}^k} \tag{2.26}$$

where $\otimes$ is the convolution

$$f_{X+Y} = f_X \otimes f_Y = \int_{-\infty}^{\infty} f_X(a-y) f_Y(y) dy. \tag{2.27}$$

(2.26) can be further calculated by

$$\mathcal{F}(f_X \otimes f_Y) = \mathcal{F}(f_X)\mathcal{F}(f_Y) \tag{2.28}$$

as

$$\mathcal{F}^{-1}(\mathcal{F}(f_l)\mathcal{F}(f_{c_1^k})\mathcal{F}(f_{c_2^k}) \cdots \mathcal{F}(f_{c_{d_v-1}^k})) \tag{2.29}$$

where $\mathcal{F}$ is the Fourier transform

$$\mathcal{F}(f(x)) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i f x} dx \tag{2.30}$$

and $\mathcal{F}^{-1}$ is the inverse Fourier transform

$$\mathcal{F}^{-1}(F(f)) = \int_{-\infty}^{\infty} F(f) e^{2\pi i x f} df. \tag{2.31}$$

Since the above analysis is based on continuous random variables, it is not convenient for computer aided analysis. In discretized density evolution [12], the continuous probability density function f is approximated to the probability mass function

$$p_{X_d}(X_d = x) = f_{X_c}(X_c = x)\Delta x \tag{2.32}$$

where X_c and f are the continuous random variable and the corresponding probability density function, X_d and p are the discrete random variable and the corresponding probability mass function, and Δx is the sample interval.

When $\mathbf{b} = [0, 0, \cdots, 0]$ is sent, the received signal Y is a Gaussian distributed random variable with mean one and variance σ^2. The corresponding LLR of the Y is $L = \frac{2Y}{\sigma^2}$, which is a Gaussian distributed random variable with mean $\frac{2}{\sigma^2}$ and variance $\frac{4}{\sigma^2}$. Hence the probability mass function is

$$p_{X_d}(X_d = z\Delta x) = \frac{\sigma}{2\sqrt{2\pi}} e^{-8\sigma^2(z\Delta x - \frac{2}{\sigma^2})^2} \Delta x, z \in \mathcal{Z}, \tag{2.33}$$

where Δx is the sample interval and z is an integer. Truncate p_{X_d} into a vector with length N where N is an odd number and $z \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$. Make N large enough to ensure that $\sum\limits_{z=-\frac{N-1}{2}}^{\frac{N-1}{2}} p_{X_d}(X_d = z\Delta x)$ approaches 1.

When two probability mass functions p_{X_1} and p_{X_2} are represented by vectors with length N_1 and N_2, the probability mass function of $X = X_1 + X_2$ is

$$\begin{aligned} p_X(X = z\Delta x) = & \sum_{z_1=\max(-\frac{N_1-1}{2}, z-\frac{N_2-1}{2})}^{\min(z, \frac{N_1-1}{2})} p_{X_1}(X_1 = z_1\Delta x) p_{X_2}(X_2 = (z - z_1)\Delta x), \\ & z = -\frac{N_1+N_2-2}{2}, -\frac{N_1+N_2}{2}, \cdots, \frac{N_1+N_2-2}{2}, \\ z_1 = -\frac{N_1-1}{2}, -\frac{N_1+1}{2}, \cdots, \frac{N_1-1}{2}, & z_2 = -\frac{N_2-1}{2}, -\frac{N_2+1}{2}, \cdots, \frac{N_2-1}{2} \end{aligned} \tag{2.34}$$

where N_1 and N_2 are odd numbers.

The probability mass function (2.34) can be calculated by circular discrete convolution. Circular discrete convolution on two vectors f and g with length N is defined as

$$z_n = \sum_{m=1}^{n} f_m g_{n-m+1} + \sum_{m=n+1}^{N} f_m g_{N+n-m+1}, n = 1, 2, \cdots, N. \tag{2.35}$$

To calculate p_X by circular discrete convolution, first append $N_2 - 1$ and $N_1 - 1$ zeros at the end of the two truncated vectors $\tilde{p}_{X_1}$ and $\tilde{p}_{X_2}$. The length of both zero-padded vectors is $N_1 + N_2 - 1$. After appending, apply circular discrete convolution on the two padded vectors.

Furthermore, in order to speed up the calculation, circular discrete convolution can be calculated by $\mathbf{DFT}^{-1}(\mathbf{DFT}(x_1(n)) \times \mathbf{DFT}(x_2(n)))$ where $\mathbf{DFT}$ is the discrete Fourier transform

$$X(k) = \sum_{n=1}^{N} x(n) e^{-\frac{2\pi i(k-1)(n-1)}{N}}, 1 \leq k \leq N \tag{2.36}$$

and $\mathbf{DFT}^{-1}$ is the inverse discrete Fourier transform

$$x(n) = \frac{1}{N} \sum_{k=1}^{N} X(k) e^{-\frac{2\pi i(k-1)(n-1)}{N}}, 1 \leq n \leq N. \tag{2.37}$$

Note that here $x_1(n)$ and $x_2(n)$ would be the padded vectors $\tilde{p}_{X_1}$ and $\tilde{p}_{X_2}$. The probability mass function of $X_1 + X_2 + \cdots + X_N$ can be calculated based on $(((X_1 + X_2) + X_3) +, \cdots, +X_N)$ by calculating the sum of two random variables $N-1$ times where $X_1, \cdots, X_N$ are independent random variables.

Now let us consider check node functions (2.15). In [10], probability density functions are calculated by changing measures to the logarithm domain. However, this approach requires finer quantization due to numerical problems. On the contrary, discretized density evolution does not change the measures. In general, if discrete random variables X_1 and X_2 are independent, the probability mass function of $Z = p(X_1, X_2)$ is

$$P(Z = z) = \sum_{z = p(x_1, x_2)} P(X_1 = x_1) P(X_2 = x_2) \tag{2.38}$$

with the summation on all (x_1, x_2) pairs that satisfy $z = p(x_1, x_2)$. For the function $Z = 2\tanh^{-1}\left(\tanh\frac{X_1}{2} \tanh\frac{X_2}{2}\right)$ (from 2.15), the probability mass function of Z is

$$P(Z = z) = \sum_{z = 2\tanh^{-1}\left(\tanh\frac{x_1}{2} \tanh\frac{x_2}{2}\right)} P(X_1 = x_1) P(X_2 = x_2). \tag{2.39}$$

If the function is in the form of $Z = 2\tanh^{-1}\left(\prod_i \tanh\frac{X_i}{2}\right)$, we can calculate P_Z by recursively calculating the probability mass function of the function of two input random variables with (2.39).

For irregular LDPC codes, it is hard to track message densities for a specific code because in general the densities on each edges are different. However, if a random variable represents incoming messages of a socket over all codes in the ensemble, densities of input messages at all variable (check) node sockets would be the same by symmetry. Thus, the densities of output messages of variable (check) nodes would be the same if the node degrees are the same. The densities of input messages would be simply a linear combination of the densities of output messages.

Denote P_i as the probability mass function of messages from degree-i variable nodes and denote λ_i as the fraction of degree-i variable node edges. The probability mass function of input messages of check nodes is a linear combination of P_i

$$\sum_i \lambda_i P_i \tag{2.40}$$

Similarly, denote Q_i as the probability mass function of messages from degree-i check nodes and denote ρ_i as the fraction of degree-i check node edges. The probability mass function of input messages of variable nodes is a linear combination of Q_i

$$\sum_i \rho_i Q_i \tag{2.41}$$

Now we introduce a special density called fixed point. If the density at the decoder input is the same as the density at the decoder output, the density is a fixed point. For example, $p_X = \Delta_\infty$ is a fixed point, where Δ is the Dirac delta function, the corresponding probability mass function is $p(X = \infty) = 1$, and the corresponding probability of error is 0. If the density becomes a fixed point during iterative decoding, the density will not be changed any more. In [11], It is shown that density evolution for massage-passing algorithms always converges to a fixed point for irregular LDPC codes. If a non-zero error fixed point is reached, zero-error decoding cannot be realized.

A stability condition is the condition under which the density would converge to a fixed point. In [11], a general stability condition of the fixed point Δ_∞ is given for general binary-input memoryless output-symmetric channels. It is shown that if the density is close to Δ_∞ enough, it would converge to Δ_∞.

Now we briefly introduce Gaussian approximation [12]. From (2.19), if all incoming c messages are independent and identically distributed Gaussian random variables, the sum v would be Gaussian distributed. Even if c is not Gaussian distributed, the sum would look like Gaussian if many of them are added due to the central limit theorem. In fact, the c from (2.15) looks less like Gaussian. However, it is generally safe to assume that c and v are both Gaussian distributed [12]. Furthermore, it is proved that the variance of the messages is always twice the mean [10]. Hence, only the mean of the Gaussian random variable is tracked. With this approximation, the complexity of density evolution reduces to one dimensional space.

The evolution of the mean of the messages for (d_v, d_c) regular LDPC codes is

$$m_{u^l} = \phi^{-1}\left(1 - \left[1 - \phi(m_{u_0} + (d_v - 1)m_{u^{l-1}})\right]^{d_c - 1}\right), \tag{2.42}$$

where m_{u_0} is the mean of messages from channel outputs, m_{u^l} is the mean of messages from check nodes in the l-th decoding iteration, d_c is the check node degree, d_v is the variable node degree and ϕ is defined as

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{\Re} \tanh \frac{u}{2} e^{-\frac{(u-x)^2}{4x}} \mathrm{d}u & \text{if } x > 0 \\ 1 & \text{if } x{=}0. \end{cases} \tag{2.43}$$

The function ϕ and the inverse function ϕ^{-1} are both monotonically decreasing functions. Furthermore, $\phi(x)$ can be approximated as

$$\phi(x) = \begin{cases} e^{-0.4527x^{0.86}+0.0218} & \text{if } x \leq 10 \\ \sqrt{\frac{\pi}{x}} e^{-\frac{x}{4}}\left(1 + \frac{1}{14x} - \frac{3}{2x}\right) & \text{if } x > 10. \end{cases} \tag{2.44}$$

In density evolution, it is assumed that incoming messages of variable nodes or check nodes are independent. This assumption implies that the bipartite graph has no cycles. However, cycles almost always exist. Hence, it is natural to ask whether density evolution is still viable when cycles exist? In [10], it is proved that the number of incorrect messages among all variable-to-check edges converges to the expectation exponentially fast towards codeword length n for regular LDPC codes. Furthermore, if n tends to infinity, the expectation converges to the case where no short cycle exists. However, it is much harder to prove the convergence for irregular LDPC codes. Without this proof, one might expect that codes with short length would deviate significantly from density evolution results. In [11], it is shown that the performance of finite length irregular LDPC codes also converges fast by simulation.

2.2.4 Code optimization

In this section, three code optimization solvers for regular and irregular LDPC codes are reviewed. In general, the code optimization problem is to find the optimal degree distribution in a large degree distribution space, e.g., maximizing the code rate for a given σ or maximizing the σ for a given code rate.

First, we review the code optimization for regular LDPC codes. The optimization objective is to maximize σ for a given code rate. For a fixed (d_v, d_c), there exists a threshold σ_t. Below the threshold, the probability of error is at most ϵ. Above the threshold, the probability of error is bigger than a constant. σ_t can be determined by sweeping the σ from an upper bound to a lower bound. For each σ, density evolution can be employed to track the message density and tell whether the (d_v, d_c) code can be decoded. When thresholds for all (d_v, d_c) pairs are known, the optimal (d_v, d_c) pair is the one with the maximum σ_t.

Next, we review two optimization solvers for irregular LDPC codes.

Solver 1: In this solver, the optimization objective is to maximize the σ for a given code rate.

The problem can be solved by employing a local optimization solver and a global optimization solver together [11]. The local optimization solver can use the hill climbing algorithm to find local maxima. In the algorithm, a current best degree distribution (λ^*, ρ^*) and the corresponding threshold σ_t^* are maintained. Each time, the current best degree distribution is slightly changed. If the new degree distribution has a better σ_t, it becomes the current best degree distribution. Here, the local maximum is a σ_t which cannot be further increased no matter how the degree distribution is slightly changed.

The complexity of the hill climbing algorithm could be exponentially increased with a linearly increased variable node degree and check node degree. In [11], it is shown that ρ can be concentrated. Furthermore, variable node degrees can be limited to 2, 3, a pre-defined maximum node degree and a few other node degrees in the between.

To further reduce the optimization complexity, the message density at critical points can be memorized. The critical point is a point where the output message density is almost the same as the input message density. When a small change on the current best degree distribution is made, the memorized message density is used as the input message density. For each small degree distribution change, the decrease of the probability of error in one decoding iteration can be measured. The degree distribution with the most decreased probability of error is chosen as the next best degree distribution candidate. Since it is not guaranteed that the candidate degree distribution can be decoded, the degree distribution should be verified by density evolution.

To get the global maximum, general global optimization solvers such as multiple-start algorithms or genetic algorithms [11] can be used. Since these solvers can be found in many global optimization literatures, the details are omitted here.

Solver 2: In this solver, the optimization objective is to maximize the code rate for a given σ.

The problem can be solved by a global optimization solver called iterative linear programming [14, 16]. The optimization objective

$$\max_{\lambda,\rho} R = 1 - \frac{\sum_{j\geq 2} \rho_j/j}{\sum_{i\geq 2} \lambda_i/i} \tag{2.45}$$

is non-linear when λ and ρ are jointly optimized. The original optimization problem can be converted into a sequence of sub-optimization problems with different ρ values. For a fixed ρ, (2.45) is equivalent to maximize a linear objective $\sum_{i\geq 2} \lambda_i/i$.

The sub-optimization problem can now be formulated as

$$\max_{\lambda_i, i\geq 2} \quad \sum_{i\geq 2} \quad \lambda_i/i \tag{2.46}$$

$$\text{s.t.} \quad \sum_{i\geq 2} \quad \lambda_i e(p_i^{l+1}) < e(p^l), l = 1, \cdots, L \tag{2.47}$$

$$\sum_{i\geq 2} \quad \lambda_i = 1 \tag{2.48}$$

where $e(.)$ is a function calculating the probability of error of a message density (For threshold decoding, the function is an integral of the message density from $-\infty$ to 0.), p^l is a mixture density of input messages of check nodes at the l-th decoding iteration, p_i^{l+1} is the density of output messages of degree i variable nodes at the $(l+1)$-th decoding iteration, and L is the total number of decoding iterations. Note that $p^l = \sum_i \lambda_i p_i^l$. (2.47) is a sequence of constraints on the decoding rule, that is, the probability of error is monotonically decreased during the iterative decoding. Note that (2.47) is non-linear since the message density is dependent on the degree distribution (λ, ρ).

In iterative linear programming, the probabilities of error $e(p_i^{l+1})$ and $e(p^l)$ in the non-linear (2.47) are treated as constants. By this treatment, the non-linear sub-optimization problem becomes linear and can be solved by solvers such as simplex method. However, optimized codes from the simplex method might not be decodable. In this case, they can be verified by density evolution. In the algorithm, a current best degree distribution and the corresponding code rate are maintained. If optimized codes can be decoded, their degree distribution and the code rate become the current best degree distribution and the current maximum code rate. During density evolution, a new set of $e(p^l)$ and $e(p_i^{l+1})$ is calculated, which is applied in the next optimization iteration as constants in (2.47).

If codes cannot be decoded by density evolution, their code rate might be too high. To reduce the code rate, the feasible region, which is a space of all variable node degree distributions that satisfy (2.47) and (2.48), can be shrunk by reducing the value of the right-hand side (RHS) of (2.47). After the feasible region is shrunk, the total number of degree distribution candidates is decreased. Hence, the code rate of the optimized code is decreased. The feasible region is a polytope, which is a geometric object with flat sides. Inside the feasible region, some codes can be decoded, while some cannot. It is also possible that higher rate codes outside the feasible region can be decoded.

Chapter 3

Low-density Parity-check Codes for Half-duplex Three-phase Two-way Relay Channels

3.1 Introduction

In recent decades, cooperative communications [17] have been developed to achieve higher communication rates. A typical example of cooperative communications is the communication through relay channels [18] where a source node transmits information to a destination node with the help of a relay node. Although the exact capacity of the relay channel is still unknown, two different relay schemes, known as decode-and-forward and compress-and-forward [18], have been developed. In general, when the source-relay link is reliable, the decode-and-forward scheme is a better choice since noise can be fully eliminated.

A natural extension of the one-way relay channel is the two-way case where two terminal nodes exchange information with the help of a relay node. Some fundamental bounds [19, 20, 21] for two-way relay channels have been proposed by several research groups. In [19], an achievable rate region of the decode-and-forward scheme based on block Markov superposition coding and an achievable rate region of the compress-and-forward scheme based on Wyner-Ziv coding for full-duplex two-way relay channels were proposed. In [20], another achievable rate region for full-duplex two-way relay channels was proved by using random binning. The authors of [21] determined the capacity region of the broadcast phase of the two-way relay channels when destination nodes use side information for decoding.

In addition to the research on fundamental bounds, various practical coding schemes have also been proposed for relay channels. Constructing codes for such channels is challenging, especially when messages are encoded into multiple streams and a destination node receives signals from multiple nodes. Low-density parity-check (LDPC) codes were proposed for one-way relay channels in [22, 23, 16]. In [22], codes within 1.2 dB of the theoretical limit were found. Furthermore, LDPC codes with random puncturing were applied to fading relay channels in [23]. In order to improve the performance of one-way relay channels, bilayer LDPC codes were designed based on the bilayer density evolution in [16].

Two-way relay channels can be generally modeled as two-phase or three-phase. In the two-phase case, the relay node receives superimposed signals from the two source nodes. Considering this unique property, various coding schemes, such as physical-layer network coding [1], repeat-accumulate codes [24], lattice codes [25] and LDPC codes [26, 27], have been proposed recently.

In this chapter, we focus on half-duplex three-phase two-way relay channels. Half-duplex is a practical assumption since it is generally difficult for a node to detect weak received signals when they are mingled with its own strong transmitting signals. Compared with two-phase two-way relay channels, signals from the source node can be utilized for decoding at the destination node. In addition, decoding at the relay node is simpler since no superimposed signals are involved. However, to the best of our knowledge, only a few practical coding schemes [28, 29] have been proposed for three-phase two-way relay channels.

In this chapter, we propose LDPC codes for half-duplex three-phase two-way relay channels [30, 31]. LDPC codes are good candidates since they can approach the capacities of point-to-point channels. In addition, they have a comprehensive set of design tools along with their flexible code constructions.

The main contributions of this chapter are four-fold. First, an achievable rate region for half-duplex three-phase two-way relay channels is proved. Second, inspired by random coding, a code construction is proposed which is composed of two irregular LDPC codes at terminal nodes and a systematic LDPC code at the relay node. Note that the relay codeword can be generated by simply adding two source codewords in GF(2). However, simple addition is not optimal if links between the source node and the relay node are asymmetric since equal amounts of information from the two codewords are included in the relay codeword. Encoding by systematic LDPC codes at the relay node can be thought of as parity forwarding or random binning on multiple sources. This code construction is similar to that of non-systematic low-density generator matrix (LDGM) codes [32], or Luby transform (LT) codes (a class of rateless code) [33], which were originally proposed for point-to-point channels. Third, in order to analyze the performance of the codes, discretized density evolution is derived. Last, based on the discretized density evolution, a three-step degree distribution optimization is proposed based on iterative linear programming. It is shown that the length of the obtained optimized codes is 26% longer than the theoretic one.

This chapter is organized as follows. It begins with an introduction of the system model and a proof of an achievable rate region for half-duplex three-phase two-way relay channels in section 3.2. Section 3.3 presents LDPC code constructions and the corresponding graphs. In addition, a message-passing algorithm is proposed in section 3.4. In section 3.5, discretized density evolution is derived as a tool to analyze the codes. An iterative linear programming algorithm for code optimization is introduced in section 3.6. An optimized degree distribution is reported and decoding simulation results are included in section 3.7. Finally, section 3.8 concludes this chapter.

3.2 System model and an achievable rate region

3.2.1 System model

In two-way relay channels, two terminal nodes communicate with each other with the help of a relay node. We consider the case when signals from the source node can be utilized for decoding at the destination node. In this case, the transmission over half-duplex two-way relay channels can be modeled as a three-phase transmission. We label the two terminal nodes as node 1 and node 2, respectively, and label the relay node as node 3. In phase 1, node 1 encodes its message and broadcasts the codeword. Both node 2 and node 3 can receive signals. In phase 2, node 2 encodes its message and broadcasts the codeword. Both node 1 and node 3 can receive signals. The relay node can decode the message of node 1 and node 2 at the end of phase 1 and phase 2, respectively. In phase 3, node 3 encodes the two source codewords to a relay codeword and broadcasts the relay codeword. Both node 1 and node 2 can receive signals. At the end of phase 3, node 1 can jointly decode the message of node 2 from signals received from node 2 in phase 2, signals received from node 3 in phase 3 and its own codeword. Similarly, node 2 can jointly decode the message of node 1. The three-phase model is shown in Figure 3.1.

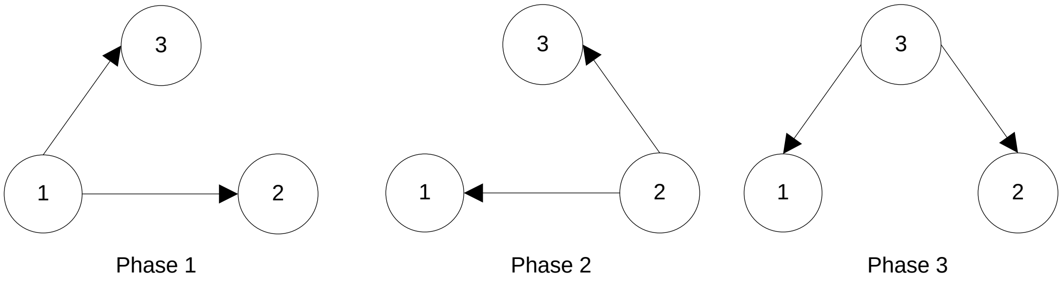

Figure 3.1: Three phases in half-duplex two-way relay channels

3.2.2 An achievable rate region of half-duplex three-phase two-way relay channels

In this section, we prove an achievable rate region of half-duplex three-phase two-way relay channels. Note that an achievable rate region for full-duplex two-way relay channels was given in [20].

The two-way relay channel consists of source input alphabet sets $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, channel output alphabet sets $\mathcal{Y}_1$, $\mathcal{Y}_2$, $\mathcal{Y}_3$ and a set of distributions $p(y_1, y_2, y_3|x_1, x_2, x_3)$. Considering time division, the distributions during phase 1, phase 2 and phase 3 are $p(y_2, y_3|x_1)$, $p(y_1, y_3|x_2)$ and $p(y_1, y_2|x_3)$, respectively.

Assume the lengths of codewords in the three phases are n_1, n_2 and n_3, respectively, and $n = \sum_{i=1}^{3} n_i$. Set $\alpha = \frac{n_1}{n}$, $\beta = \frac{n_2}{n}$ and $\gamma = \frac{n_3}{n}$.

A $((2^{nR_1}, 2^{nR_2}), n_1, n_2)$ code for the half duplex three-phase two-way relay channel consists of two sets of integers $\mathcal{W}_1 = \{1, 2, \cdots, 2^{nR_1}\}$ and $\mathcal{W}_2 = \{1, 2, \cdots, 2^{nR_2}\}$, three encoding functions $X_1 : \mathcal{W}_1 \to \mathcal{X}_1^{n_1}$, $X_2 : \mathcal{W}_2 \to \mathcal{X}_2^{n_2}$ and $X_3 : \mathcal{W}_1 \times \mathcal{W}_2 \to \mathcal{X}_3^{n_3}$, and four decoding functions $\mathcal{Y}_3^{n_1} \to \mathcal{W}_1$, $\mathcal{Y}_3^{n_2} \to \mathcal{W}_2$, $\mathcal{Y}_2^{n_1} \times \mathcal{Y}_2^{n_3} \to \mathcal{W}_1$, and $\mathcal{Y}_1^{n_2} \times \mathcal{Y}_1^{n_3} \to \mathcal{W}_2$.

Theorem 2. *For discrete memoryless half-duplex three-phase two-way relay channels, all rate pairs (R_1, R_2) satisfying*

$$R_1 < \min \{\alpha I(X_1; Y_3), \gamma I(X_3; Y_2) + \alpha I(X_1; Y_2)\} \tag{3.1}$$

and

$$R_2 < \min \{\beta I(X_2; Y_3), \gamma I(X_3; Y_1) + \beta I(X_2; Y_1)\} \tag{3.2}$$

are achievable for some $p(x_1)p(x_2)p(x_3)$ where $\alpha + \beta + \gamma = 1$.

Proof. Codebook generation: Generate 2^{nR_1} codewords $\mathbf{x}_1 = x_1^{n_1}$ according to $\prod_{i=1}^{n_1} p(x_1)$ and index them as $\mathbf{x}_1(w_1)$, $w_1 \in \{1, 2, \cdots, 2^{nR_1}\}$. Generate 2^{nR_2} codewords $\mathbf{x}_2 = x_2^{n_2}$ according to $\prod_{i=1}^{n_2} p(x_2)$ and index them as $\mathbf{x}_2(w_2)$, $w_2 \in \{1, 2, \cdots, 2^{nR_2}\}$. Generate

$2^{n(R_1+R_2)}$ codewords $\mathbf{x}_3 = x_3^{n_3}$ according to $\prod_{i=1}^{n_3} p(x_3)$ and index them as $\mathbf{x}_3(w_1, w_2)$, $w_1 \in \{1, 2, \cdots, 2^{nR_1}\}$, $w_2 \in \{1, 2, \cdots, 2^{nR_2}\}$.

Encoding: In phase 1, to send index w_1, node 1 sends $\mathbf{x}_1(w_1)$. In phase 2, to send w_2, node 2 sends $\mathbf{x}_2(w_2)$. In phase 3, node 3 sends $\mathbf{x}_3(\hat{w}_1, \hat{w}_2)$ after decoding w_1 and w_2 (See the decoding part).

Decoding: Denote $\mathbf{y}_{i,j}$ as the channel output at node i in phase j. At the end of phase 1, node 3 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies the joint typicality check $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{3,1}) \in A_\epsilon^{(n_1)}(X_1, Y_3)$ where $A_\epsilon^{(n_1)}(X_1, Y_3)$ is the set of jointly typical sequences of X_1 and Y_3. If there is no such or more than one such $\hat{w}_1$, an error is declared. Similarly, at the end of phase 2, node 3 decodes w_2 by finding the unique $\hat{w}_2$ that satisfies $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{3,2}) \in A_\epsilon^{(n_2)}(X_2, Y_3)$. If there is no such or more than one such $\hat{w}_2$, an error is declared. At the end of phase 3, node 1 decodes w_2 by finding the unique $\hat{w}_2$ that satisfies $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{1,2}) \in A_\epsilon^{(n_2)}(X_2, Y_1)$ and $(\mathbf{x}_3(w_1, \hat{w}_2), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3, Y_1)$. Node 2 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{2,1}) \in A_\epsilon^{(n_1)}(X_1, Y_2)$ and $(\mathbf{x}_3(\hat{w}_1, w_2), \mathbf{y}_{2,3}) \in A_\epsilon^{(n_3)}(X_3, Y_2)$.

Analysis of the probability of error: When node 1 sends $\mathbf{x}_1(w_1)$, the probability that independent $\mathbf{x}_1$ and $\mathbf{y}_{3,1}$ are jointly typical is upper bounded by $2^{-n_1(I(X_1;Y_3)-3\epsilon)}$. There are in total $2^{nR_1} - 1$ such $\mathbf{x}_1$. With the union bound, the probability of error at node 3 is upper bounded by $(2^{nR_1} - 1)2^{-n_1(I(X_1;Y_3)-3\epsilon)}$, which approaches zero when $n_1 \to \infty$ and $R_1 < \alpha I(X_1; Y_3)$ (from $nR_1 - n_1 I(X_1; Y_3) < 0$ and $\alpha = \frac{n_1}{n}$). Similarly, we need $R_2 < \beta I(X_2; Y_3)$ for node 3 to decode $\mathbf{x}_2$.

When node 2 sends $\mathbf{x}_2(w_2)$, the probability that independent $\mathbf{x}_2$ and $\mathbf{y}_{1,2}$ are jointly typical is upper bounded by $2^{-n_2(I(X_2;Y_1)-3\epsilon)}$. When node 3 sends $\mathbf{x}_3(\hat{w}_1, \hat{w}_2)$, the probability that independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical is upper bounded by $2^{-n_3(I(X_3;Y_1)-3\epsilon)}$. There are totally $2^{nR_2} - 1$ such w_2 when node 1 knows w_1. With the union bound, the probability of the event that any independent $\mathbf{x}_2$ and $\mathbf{y}_{1,2}$ are jointly typical and any independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical at node 1 is upper bounded by $(2^{nR_2} - 1)2^{-n_2(I(X_2;Y_1)-3\epsilon)}2^{-n_3(I(X_3;Y_1)-3\epsilon)}$, which approaches zero when $n_2 \to \infty$, $n_3 \to \infty$ and $R_2 < \beta I(X_2; Y_1) + \gamma I(X_3; Y_1)$ (from $nR_2 - n_2 I(X_2; Y_1) - n_3 I(X_3; Y_1) < 0$, $\beta = \frac{n_2}{n}$ and

$\gamma = \frac{n_3}{n}$). Similarly, $R_1 < \alpha I(X_1; Y_2) + \gamma I(X_3; Y_2)$ is required for node 2 to decode w_1. $\square$

α, β, γ can be optimized by maximizing $R_1 + R_2$ in two-way relay channels by the following linear programming:

$$\max_{\alpha,\beta\gamma,R_1,R_2} R_1 + R_2 \tag{3.3}$$

$$\text{s.t. } \alpha + \beta + \gamma = 1 \tag{3.4}$$

$$0 \leq \alpha, \beta, \gamma \leq 1 \tag{3.5}$$

$$R_1, R_2 \geq 0 \tag{3.6}$$

$$R_1 \leq \alpha I(X_1; Y_3) \tag{3.7}$$

$$R_1 \leq \gamma I(X_3; Y_2) + \alpha I(X_1; Y_2) \tag{3.8}$$

$$R_2 \leq \beta I(X_2; Y_3) \tag{3.9}$$

$$R_2 \leq \gamma I(X_3; Y_1) + \beta I(X_2; Y_1). \tag{3.10}$$

Gaussian half-duplex three-phase two-way relay channels can be modeled as follows. In phase 1, $Y_{3,1} = X_1 + Z_{3,1}$ and $Y_{2,1} = X_1 + Z_{2,1}$. In phase 2, $Y_{3,2} = X_2 + Z_{3,2}$ and $Y_{1,2} = X_2 + Z_{1,2}$. In phase 3, $Y_{1,3} = X_3 + Z_{1,3}$ and $Y_{2,3} = X_3 + Z_{2,3}$. $Z_{i,j}$ is a Gaussian distributed random variable with mean zero and variance $\sigma_{i,j}^2$, where i is the source node and j is the destination node. When binary phase-shift keying (BPSK) is considered, the codeword bit is mapped from $\{0, 1\}$ to $\{1, -1\}$. The signal $\mathbf{X}_1 = (X_{1,1}, \cdots, X_{1,n_1})$ has a power constraint $\frac{1}{n_1}\sum_{i=1}^{n_1} X_{1,i}^2 \leq P_1$. Similarly, $\mathbf{X}_2$ and $\mathbf{X}_3$ have power constraints $\frac{1}{n_2}\sum_{i=1}^{n_2} X_{2,i}^2 \leq P_2$ and $\frac{1}{n_3}\sum_{i=1}^{n_3} X_{3,i}^2 \leq P_3$.

For Gaussian half-duplex three-phase two-way relay channels, all rate pairs (R_1, R_2) satisfying

$$\begin{aligned} R_1 < \min\left(\frac{1}{2}\alpha \log_2\left(1 + \frac{P_1}{N_{3,1}}\right),\right. \\ \left.\frac{1}{2}\alpha \log_2\left(1 + \frac{P_1}{N_{2,1}}\right) + \frac{1}{2}\gamma \log_2\left(1 + \frac{P_3}{N_{2,3}}\right)\right) \end{aligned} \tag{3.11}$$

and

$$R_2 < \min\left(\frac{1}{2}\beta\log_2\left(1+\frac{P_2}{N_{3,2}}\right),\right.$$
$$\left.\frac{1}{2}\beta\log_2\left(1+\frac{P_2}{N_{1,2}}\right)+\frac{1}{2}\gamma\log_2\left(1+\frac{P_3}{N_{1,3}}\right)\right) \tag{3.12}$$

are achievable where $N_{i,j} = \sigma_{i,j}^2$ and $\alpha + \beta + \gamma = 1$. Note that (3.11) and (3.12) can be easily derived from (3.1) and (3.2).

In Figure 3.2, the achievable rate R_1 is plotted when $\alpha = \beta = \gamma = \frac{1}{3}$ and $\frac{P_1}{N_{3,1}} = 1$. The x-axis and y-axis are signal-to-noise ratios (SNRs) $\frac{P_1}{N_{2,1}}$ and $\frac{P_3}{N_{2,3}}$, respectively. The z-axis is the achievable rate R_1. The flat area is the area where R_1 is limited by the SNR of the source-relay link, while the slope area is the area where R_1 is limited by SNRs of the source-destination link and the relay-destination link.

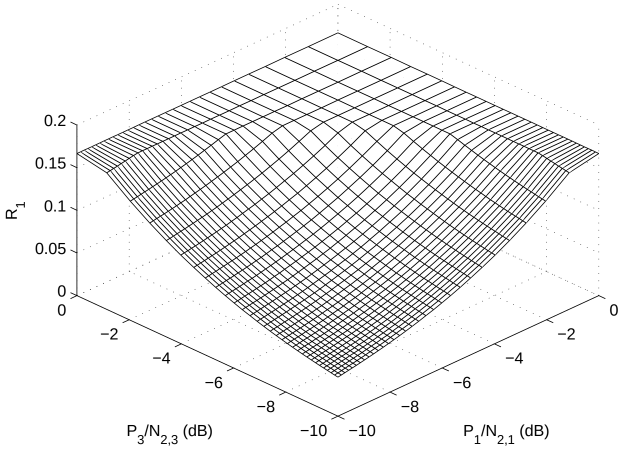

Figure 3.2: Achievable rates of R_1 for $\left(\frac{P_1}{N_{2,1}}, \frac{P_3}{N_{2,3}}\right)$ pairs

In Figure 3.3, the achievable rate R_1 is plotted when $\alpha = \beta = \gamma = \frac{1}{3}$ and $\frac{P_1}{N_{2,1}} = \frac{P_3}{N_{2,3}}$. The x-axis and y-axis are SNRs $\frac{P_1}{N_{3,1}}$ and $\frac{P_1}{N_{2,1}}$ $\left(\frac{P_3}{N_{2,3}}\right)$. The z-axis is the achievable rate R_1. The left slope area is the area where R_1 is limited by the SNR of the source-relay link, while the right slope area is the area where R_1 is limited by SNRs of the source-destination link and the relay-destination link.

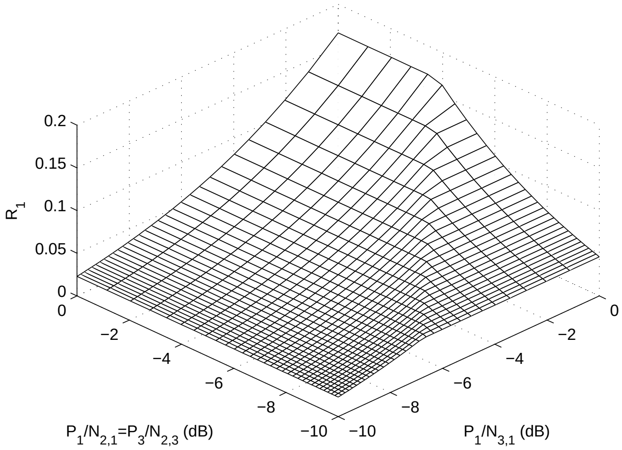

Figure 3.3: Achievable rates of R_1 for $\left(\frac{P_1}{N_{3,1}}, \frac{P_1}{N_{2,1}}\left(\frac{P_3}{N_{2,3}}\right)\right)$ pairs

Figure 3.4 shows the sum-rate difference between three-phase two-way relay channels and two-phase two-way relay channels. Node 1 and node 2 are located at (-0.5,0) and (0.5,0), respectively. The relay node 3 can be located at an arbitrary location. Assume the link rate between any two nodes is

$$r_{i,j} = \frac{1}{2}\log_2\left(1 + \frac{\frac{P_i}{d_{i,j}^{\alpha}}}{\sigma_{i,j}^2}\right) \tag{3.13}$$

where P_i is the power of node i, $\sigma_{i,j}^2$ is the noise power at node j when node i sends, $d_{i,j}$ is the distance between node i and node j, and α is the path loss exponent. The sum rate of three-phase two-way relay channels is hence

$$r_{s,3} = \min\left(\frac{1}{3}r_{1,3}, \frac{1}{3}r_{1,2} + \frac{1}{3}r_{3,2}\right) + \min\left(\frac{1}{3}r_{2,3}, \frac{1}{3}r_{2,1} + \frac{1}{3}r_{3,1}\right) \tag{3.14}$$

when $\alpha = \beta = \gamma = \frac{1}{3}$. The sum rate of two-phase two-way relay channels is

$$r_{s,2} = \min\left(\min\left(r_{3,1}, r_{3,2}\right), \frac{1}{4}\log_2\left(1 + \frac{\frac{P}{d_{1,3}^{\alpha}} + \frac{P}{d_{2,3}^{\alpha}}}{\sigma^2}\right)\right) \tag{3.15}$$

where σ_3^2 is the noise power at node 3, and time durations of the two phases are the same. When the rate difference $r_{s,3} - r_{s,2}$ is positive, the three-phase case outperforms. When the rate difference is negative, the two-phase case outperforms. As we can see, at most locations between the two terminal nodes, the three-phase case outperforms.

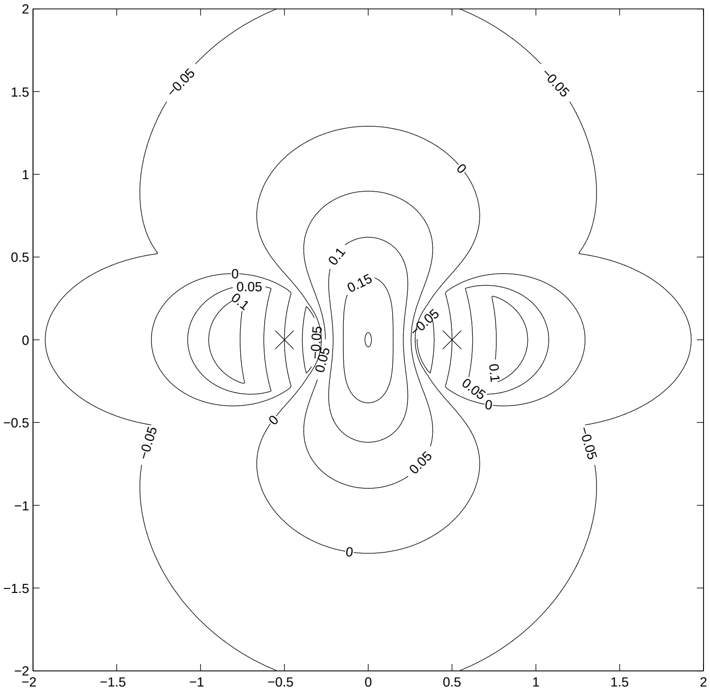

Figure 3.4: Sum-rate difference between three-phase two-way relay channels and two-phase two-way relay channels

3.3 Low-density parity-check code constructions and graph representations

In this section, we propose LDPC code constructions for half-duplex three-phase two-way relay channels and show the corresponding graph representations.

At terminal node i for $i = 1, 2$, the node encodes a k_i-bit message into an n_i-bit codeword. The codeword is broadcast to the relay node and the other terminal node. Under the decode-and-forward scheme, the relay node can decode the message, whereas the other terminal node cannot decode without the help of the relay node. Intuitively, with additional bits from the relay node, the effective code rate is reduced.

Next, two code constructions at the relay node are given. In code construction 1, two source codewords are disjointly encoded. In code construction 2, two source codewords are jointly encoded.

Code construction 1: At the relay node, the relay codewords (also called relay bits) $\mathbf{r}_1$ and $\mathbf{r}_2$ are generated by two systematic LDPC codes, where $\mathbf{r}_1 = \mathbf{c}_1\mathbf{G}_1$ and $\mathbf{r}_2 = \mathbf{c}_2\mathbf{G}_2$. The lengths of the source codewords $\mathbf{c}_1$, $\mathbf{c}_2$ and relay codewords $\mathbf{r}_1$, $\mathbf{r}_2$ are n_1, n_2, n_3 and n_4, respectively. $\mathbf{r}_1$ and $\mathbf{r}_2$ are broadcast to both terminal nodes. $\mathbf{c}_1$ and $\mathbf{c}_2$ are not sent. The sizes of the generator matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are $n_1 \times n_3$ and $n_2 \times n_4$, respectively. Since in general n_3 is less than n_1, and n_4 is less than n_2, multiple source codeword pairs are mapped to a relay codeword. This scheme is often called binning or parity forwarding [16]. A relay codeword can be considered as a bin index. All source codeword pairs that satisfy parity check constraints with the relay codeword are in the same bin.

This code construction is similar to those of non-systematic LDGM codes [32] and LT codes [33]. LDGM codes were initially proposed as an alternative to LDPC codes. In these codes, check bits $\mathbf{c}$ are generated from source bits $\mathbf{s}$ by $\mathbf{c} = \mathbf{s}\mathbf{G}$. For systematic LDGM codes, both source bits and check bits are sent to a destination node. For non-systematic LDGM codes, only check bits are sent. LDGM codes were proposed for channels with known channel parameters and their code rates are fixed. LT codes are the first practical rateless codes, and originated from Fountain codes [34]. They can be considered as non-systematic LDGM codes, though check bits are continuously generated until a receiver can recover the source message.

In general, any linear code can be represented by a Tanner graph [9]. Variable nodes and check nodes are two types of nodes in the graph. The variable node corresponds to a bit in a codeword, or a column in a parity check matrix. The check node corresponds to a parity check constraint, or a row in a parity check matrix. The graph of code construction 1 at the relay node is shown in Figure 3.5. Circles are variable nodes and squares are check nodes. The corresponding graph for decoding at the destination node is shown in Figure 3.6.

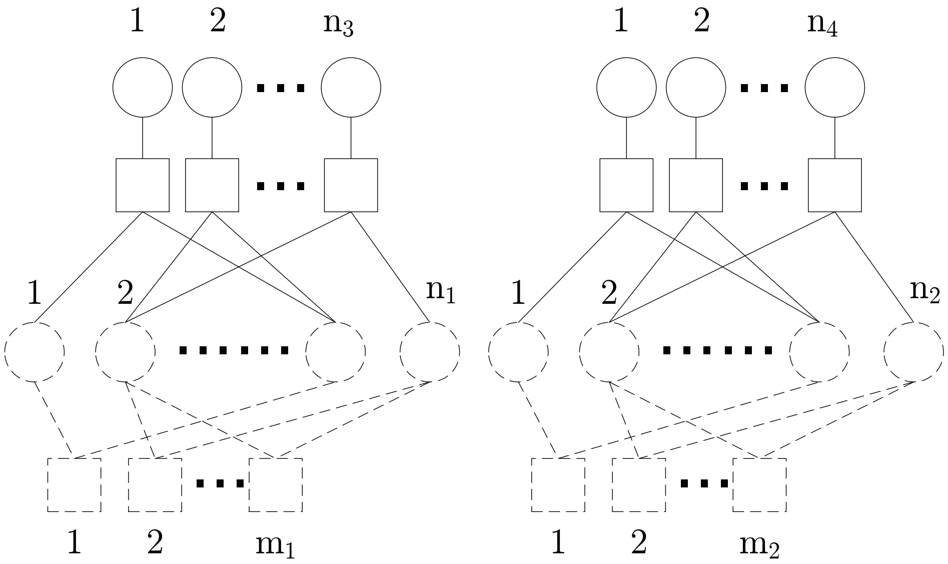

Figure 3.5: Graph of code construction 1 at the relay node

Code construction 2: At the relay node, two codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ from terminal nodes are concatenated as a source message $\mathbf{c} = [\mathbf{c}_1\mathbf{c}_2]$. The lengths of $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}$ are n_1, n_2 and $n_1 + n_2$ respectively. An n_3-bit relay codeword $\mathbf{r}$ is generated by a systematic LDPC code where $\mathbf{r} = \mathbf{cG}$ and $\mathbf{G}$ is a generator matrix with the size $(n_1+n_2)\times n_3$. $\mathbf{r}$ is broadcast to both terminal nodes. $\mathbf{c}$ is not sent. Here, $2^{n_1+n_2}$ codeword pairs are mapped to 2^{n_3}

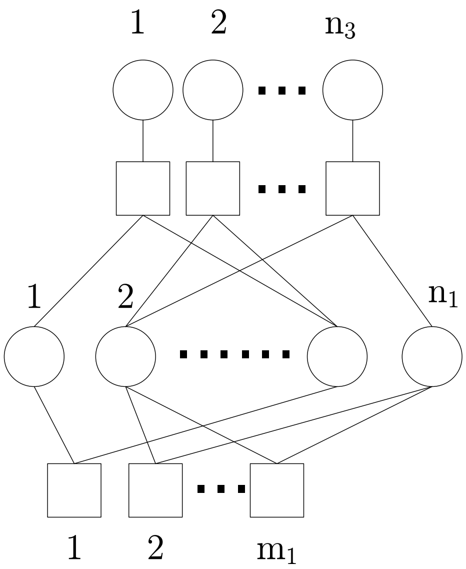

Figure 3.6: Graph for decoding with code construction 1 at terminal nodes

codewords. Since n_3 is in general less than $n_1 + n_2$, multiple codeword pairs are mapped to a relay codeword.

The graph of code construction 2 at the relay node is shown in Figure 3.7. The n_3 upper layer variable nodes (in black) represent the relay bits. The lower layer n_1 variable nodes (in white) and n_2 variable nodes (in grey) represent two source codewords from terminal nodes.

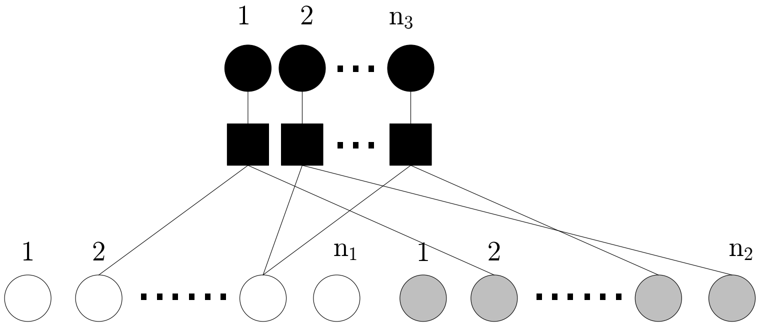

Figure 3.7: Graph of code construction 2 at the relay node

Note that the above graph is a bipartite graph. The n_3 upper layer variable nodes can be moved to the lower layer, as shown in Figure 3.8. The total number of variable nodes is $n_1 + n_2 + n_3$. The first two groups of variable nodes represent codewords of two terminal nodes, called Group 1 variable nodes and Group 2 variable nodes respectively. The n_3 right-most variable nodes represent relay bits, called Group 3 variable nodes.

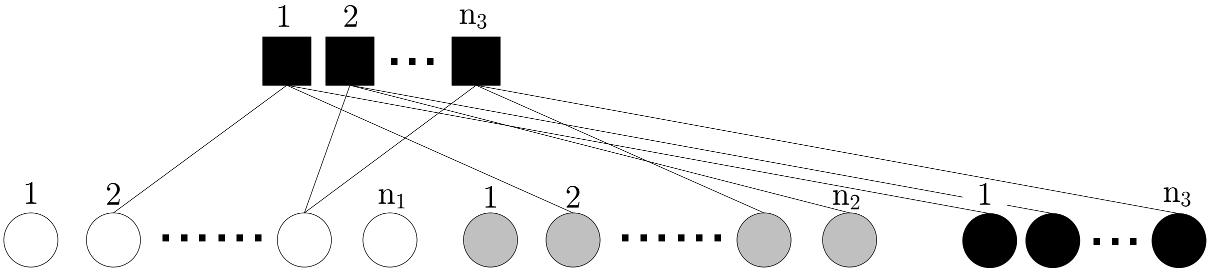

Figure 3.8: Equivalent graph of code construction 2 at the relay node

At the terminal node, messages are decoded based on three pieces of information: signals received from the source terminal node, signals received from the relay node, and the codeword of the destination node. The graph for decoding is shown in Figure 3.9. Compared to Figure 3.8, two groups of check nodes are added to the lower layer. These check nodes represent parity check constraints of LDPC codes at the terminal node, which are called Group 1 check nodes and Group 2 check nodes, respectively. The upper layer check nodes are called Group 3 check nodes. Group 3 check nodes in Figure 3.9 can also be moved to the lower layer. In this sense, the decoding algorithm at the terminal node is similar to those for point-to-point channels. In the following sections, we propose message-passing algorithms, discrete density evolution and iterative linear programming based on code construction 2.

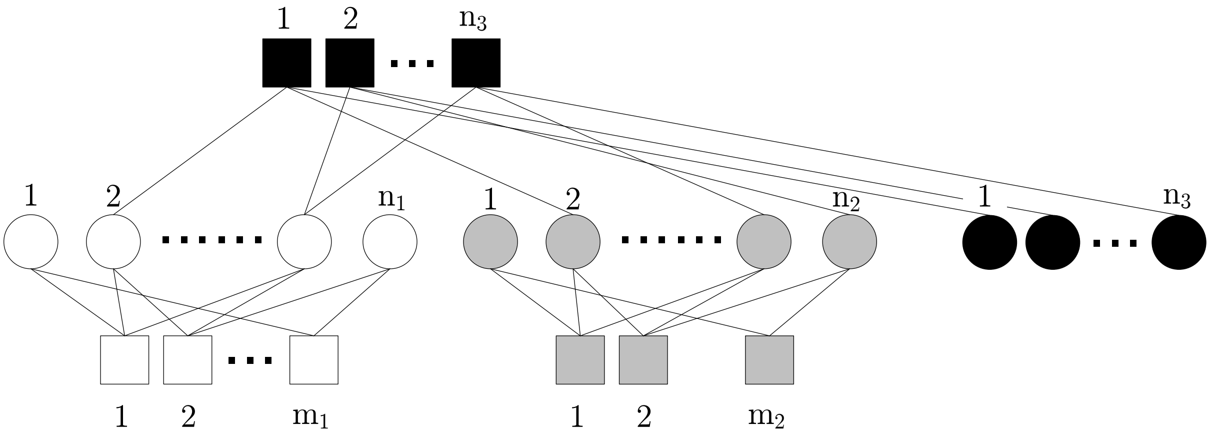

Figure 3.9: Graph for decoding with code construction 2 at terminal nodes

3.4 Message-passing algorithms

In this section, we propose message-passing algorithms for decoders in half-duplex three-phase two-way relay channels. In such channels, decoding happens at all three nodes. Since any existing decoding algorithms for point-to-point channels can be used at the relay node, the details are omitted here. In the following, we focus only on the message-passing algorithm at destination nodes.

Binary input additive white Gaussian noise (BIAWGN) half-duplex three-phase two-way relay channels are Gaussian half-duplex three-phase two-way relay channels with the constraint $x_i \in \{1, -1\}$ for $i = 1, 2, 3$. The received bit can be represented by a probability pair $(p(x_j = 1|y_{i,j}), p(x_j = -1|y_{i,j}))$, or in a log-likelihood ratio (LLR) form

$$\log \frac{p(x_j = 1|y_{i,j})}{p(x_j = -1|y_{i,j})} = \frac{2y_{i,j}}{\sigma^2_{i,j}} \tag{3.16}$$

for $i, j = 1, 2, 3$.

A half-duplex three-phase two-way relay channel is said to be memoryless if

$$p(\mathbf{y}_{2,1}, \mathbf{y}_{3,1}|\mathbf{x}_1) = \prod_t p(y_{2,1,t}, y_{3,1,t}|x_{1,t}), \tag{3.17}$$

$$p(\mathbf{y}_{1,2}, \mathbf{y}_{3,2}|\mathbf{x}_2) = \prod_t p(y_{1,2,t}, y_{3,2,t}|x_{2,t}), \tag{3.18}$$

$$p(\mathbf{y}_{1,3}, \mathbf{y}_{2,3}|\mathbf{x}_3) = \prod_t p(y_{1,3,t}, y_{2,3,t}|x_{3,t}). \tag{3.19}$$

In addition, it is said to be symmetric if

$$p(y_{2,1,t}, y_{3,1,t}|x_{1,t} = 1) = p(-y_{2,1,t}, -y_{3,1,t}|x_{1,t} = -1), \tag{3.20}$$

$$p(y_{1,2,t}, y_{3,2,t}|x_{2,t} = 1) = p(-y_{1,2,t}, -y_{3,2,t}|x_{2,t} = -1), \tag{3.21}$$

$$p(y_{1,3,t}, y_{2,3,t}|x_{3,t} = 1) = p(-y_{1,3,t}, -y_{2,3,t}|x_{3,t} = -1). \tag{3.22}$$

In the following, only decoding functions at node 1 are derived since decoding functions at node 2 are similar.

A decoding algorithm can generally be considered as functions on multiple input variables. At node 1, decoding is based on channel outputs $\mathbf{y}_{1,2}$, $\mathbf{y}_{1,3}$ and side information $\mathbf{x}_1$. For a bit-wise maximum a posterior (MAP) decoder,

$$\begin{aligned}
\hat{x}_{2,i} &= \arg\max_{x_{2,i}} p(x_{2,i}|\mathbf{x}_1, \mathbf{y}_{1,2}, \mathbf{y}_{1,3}) && (3.23)\\
&= \arg\max_{x_{2,i}} \sum_{\sim x_{2,i}} p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{y}_{1,2}, \mathbf{y}_{1,3}) && (3.24)\\
&= \arg\max_{x_{2,i}} \sum_{\sim x_{2,i}} p(\mathbf{y}_{1,2}, \mathbf{y}_{1,3}|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) && (3.25)\\
&= \arg\max_{x_{2,i}} \sum_{\sim x_{2,i}} \prod_j p(y_{1,2,j}|x_{2,j}) \prod_k p(y_{1,3,k}|x_{3,k}) \mathbb{1}_{\mathbf{x}_1 \in \mathcal{C}_1, \mathbf{x}_2 \in \mathcal{C}_2, \mathbf{x}_3 = f(\mathbf{x}_1, \mathbf{x}_2)}, && (3.26)
\end{aligned}$$

where i, j, k are the time indices, $\mathcal{C}_1$ and $\mathcal{C}_2$ are code books at node 1 and node 2, respectively, and $\mathbb{1}$ is the indicator function. Equations (3.23) to (3.24) are based on the law of total probability. $\sum_{\sim x_{2,i}}$ is a summation over all bits of $\mathbf{x}_2$ except the bit $x_{2,i}$. (3.24) to

(3.25) is based on the Bayes' theorem. The codeword $\mathbf{x}_3$ is added since $\mathbf{x}_3 = f(\mathbf{x}_1, \mathbf{x}_2)$. Equations (3.25) to (3.26) are based on the memoryless channel assumption and the uniform distributed codeword pair $(\mathbf{x}_1, \mathbf{x}_2)$ assumption. We also assume that each bit in the $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ is independently generated. Note that, in fact, bits among a codeword are dependent due to parity check constraints.

With the help of the graph in Figure 3.9, the message-passing algorithm can be easily described. Variable nodes and check nodes are associated with decoding functions. Messages flow between variable nodes and check nodes via edges, serving as inputs or outputs of the functions. The algorithm adopts an iterative decoding method by passing messages multiple times between variable nodes and check nodes. In variable nodes, functions are in the form of summation (See (2.19)). In check nodes, functions are in the form of $2\tanh^{-1}(\prod \tanh)$ (See (2.15)). In general, a message passing schedule is required during the iterative decoding. In this work, a flooding schedule is used. In this schedule, all messages from variable nodes are passed to check nodes along all edges, and all output messages from check nodes are passed back to variables nodes thereafter to complete one decoding iteration.

Let v_i^l be a message from a Group i variable node to a Group i check node in the l-th decoding iteration for $i = 1, 2$. Let u_i^l be a message from a Group i check node to a Group i variable node in the l-th decoding iteration for $i = 1, 2$. Let $v_{i,3}^l$ be a message from a Group i variable node to a Group 3 check node in the l-th decoding iteration for $i = 1, 2, 3$. Let $u_{3,i}^l$ be a message from a Group 3 check node to a Group i variable node in the l-th decoding iteration for $i = 1, 2, 3$. For a variable node, a lower/upper variable node degree is defined as the total number of edges connected to a lower/upper layer check node. An upper/lower-degree-i variable node is a variable node with i upper/lower edges. An upper/lower-degree-i variable node edge is an edge connected to an upper/lower-degree-i variable node. Let d_i be a lower degree of a Group i variable node for $i = 1, 2$. Let $d_{3,i}$ be an upper degree of a Group i variable node for $i = 1, 2, 3$. Let g_i be a degree of a Group i check node for $i = 1, 2$. A Group 3 check node has three degrees. Let $g_{3,i}$ be a degree of a Group 3 check node, a degree that is the total number of edges connecting to a Group i

variable node for $i = 1, 2, 3$. Let $u_{0,i}$ be a channel output LLR associated with a Group i variable node for $i = 1, 2, 3$.

The functions used for decoding messages of terminal node 2 at terminal node 1 are

$$
\begin{aligned}
v_2^l &= \sum_{i=1}^{d_2-1} u_{2,i}^{l-1} + \sum_{j=1}^{d_{3,2}} u_{3,2,j}^{l-1} + u_{0,2}, && (3.27)\\
v_{1,3}^l &= u_{0,1}, && (3.28)\\
v_{2,3}^l &= \sum_{i=1}^{d_2} u_{2,i}^{l-1} + \sum_{j=1}^{d_{3,2}-1} u_{3,2,j}^{l-1} + u_{0,2}, && (3.29)\\
v_{3,3}^l &= u_{0,3}, && (3.30)\\
u_{3,2}^l &= 2\tanh^{-1}\left[\prod_{i=1}^{g_{3,1}} \tanh\left(\frac{v_{1,3,i}^l}{2}\right)\right. \\
&\qquad \left.\prod_{j=1}^{g_{3,2}-1} \tanh\left(\frac{v_{2,3,j}^l}{2}\right) \tanh\left(\frac{v_{3,3}^l}{2}\right)\right], && (3.31)\\
u_2^l &= 2\tanh^{-1}\left[\prod_{i=1}^{g_2-1} \tanh\left(\frac{v_{2,i}^l}{2}\right)\right]. && (3.32)
\end{aligned}
$$

The function in Group 1 variable nodes is shown in (3.28). Since terminal node 1 knows its own codeword, the intrinsic values of Group 1 variable nodes are $+\infty$ or $-\infty$. Hence, no matter what messages are received from check nodes, Group 1 variable nodes always send $u_{0,1}$ ($+\infty$ or $-\infty$) to upper check nodes. The function in Group 3 variable nodes is shown in (3.30). Group 3 variable nodes send only $u_{0,3}$, since the degree of Group 3 variable nodes is 1. Functions in Group 2 variable nodes are shown in (3.27) and (3.29). Group 2 variable nodes receive messages $u_{3,2,j}^{l-1}$ and $u_{2,i}^{l-1}$ from upper layer check nodes and lower layer check nodes respectively. These messages are added together with the channel output LLR $u_{0,2} = \frac{2y_{1,2}}{\sigma_{1,2}^2}$. The output $v_{2,3}^l$ is sent to a Group 3 check node in the upper layer. The output v_2^l is sent to a Group 2 check node in the lower layer. The function in Group 3 check nodes at the upper layer is shown in (3.31). Upper layer check nodes send only the output message $u_{3,2}^l$ to a Group 2 variable node. The function of Group 2 check nodes at the lower layer is shown in (3.32). The output u_2^l is sent to a Group 2 variable node.

3.5 Discretized density evolution

In this section, density evolution [10] is used as a tool to analyze codes in message-passing algorithms for half-duplex three-phase two-way relay channels.

First, we formally define an ensemble of codes via graphs in half-duplex three-phase two-way relay channels. The ensemble is a sequence of codes with the same variable node degree distributions and check node degree distributions.

For systematic LDPC codes at the relay node, we define one variable node degree distribution and two check node degree distributions. These degree distributions are defined from a node perspective. Since the relay node forwards only partial information of the source codeword, we allow degree 0 as an upper degree of a variable node. The upper-degree-0 variable node does not connect to any upper layer check nodes. Denote λ_3 as the variable node degree distribution. $\lambda_{3,i,j}$ is the fraction of the total number of upper-degree-j variable nodes in Group i variable nodes to the total number of all three groups of variable nodes. $\sum_{i,j} \lambda_{3,i,j} = 1$. Denote $\rho_{3,1}$ and $\rho_{3,2}$ as the two check node degree distributions. $\rho_{3,i,j}$ is the fraction of the total number of upper layer check nodes with degree $d_{3,i} = j$ to the total number of all upper layer check nodes. $\sum_j \rho_{3,i,j} = 1$ for $i = 1, 2$.

The ensemble of codes is defined based on four permutations: π_i is a permutation for codes at terminal node i for $i = 1, 2$; π_3 and π_4 are permutations for the code at the relay node. The definitions of π_1 and π_2 are the same as those in point-to-point channels. Here, we define only π_3 and π_4. Assign some sockets to every Group i variable node according to the degree distribution $\lambda_{3,i}$ for $i = 1, 2$. The sockets on Group i variable nodes are called Group i variable node sockets. Assign two groups of sockets to every upper layer check node according to degree distributions $\rho_{3,1}$ and $\rho_{3,2}$. We call them Group i check node sockets for $i = 1, 2$. Edges connecting to Group i check node sockets are connected to

Group i variable nodes. Two groups of variable node sockets and two groups of check node sockets are labeled separately with positive integers starting from 1. Group 1 and Group 2 check node socket labels are permuted by π_3 and π_4. Edges are identified by pairs of sockets, and are denoted as $(i,\, \pi_3(i))$ and $(j,\, \pi_4(j))$, where i or j is a Group 1 or Group 2 variable node socket, $\pi_3(i)$ or $\pi_4(j)$ is a check node socket in the two groups of check node sockets respectively. A code is an element in the permutation space $\pi_1 \times \pi_2 \times \pi_3 \times \pi_4$. All codes in the permutation space are equiprobable.

Recall that in point-to-point channels, if the three symmetric conditions (2.23-2.25) are met, the probability of error is independent of the codeword. Since the three conditions are also met in half-duplex three-phase two-way relay channels, the probability of error is independent of the codeword. Hence, we assume that only codewords $\mathbf{b} = [0, 0, \cdots, 0]$ are transmitted at both terminal nodes and the relay node. By this assumption, the channel output $Y_{i,j}$ is a Gaussian distributed random variable with the parameter $\mathcal{N}(1, \sigma^2_{i,j})$. A received codeword bit is represented by the LLR $L_{i,j} = \frac{2Y_{i,j}}{\sigma^2_{i,j}}$, which is a Gaussian distributed random variable with the parameter $\mathcal{N}(\frac{2}{\sigma^2_{i,j}}, \frac{4}{\sigma^2_{i,j}})$.

In discretized density evolution, probability density functions are approximated by probability mass functions. Recall that the function of the variable node is a sum of independent random variables, e.g. (3.27) and (3.29). The probability mass function of the sum of two independent discrete random variables can be calculated by convolving the probability mass functions of the two random variables by (2.34) or by circular discrete convolution (2.35). Furthermore, in order to speed up the calculation, the circular discrete convolution can be calculated by discrete Fourier transform (2.36) and inverse discrete Fourier transform (2.37).

For a variable node with an upper degree i and a lower degree j, denote the probability mass function of input messages on the upper edge and the lower edge as P_i and P_j, respectively. The probability mass function of the output messages on upper edges is

$$P_v^l = P_0 * \{\otimes_{i-1} P_i^{l-1}\} * \{\otimes_j P_j^{l-1}\} \tag{3.33}$$

where l is the decoding iteration number, $*$ is discrete convolution, $\otimes_i$ is discrete convolution on i random variables, and P_0 is the probability mass function of the channel output LLR message. Similarly, the probability mass function of the output messages on lower edges is

$$P_v^l = P_0 * \{\otimes_i P_i^{l-1}\} * \{\otimes_{j-1} P_j^{l-1}\}. \tag{3.34}$$

In general, if discrete random variables X_1 and X_2 are independent, the probability mass function of $Z = p(X_1, X_2)$ is

$$P(Z = z) = \sum_{z=p(x_1,x_2)} P(X_1 = x_1)P(X_2 = x_2). \tag{3.35}$$

The probability mass function of check node output messages in (3.31) and (3.32) can be calculated by this way. For the function $Z = 2\tanh^{-1}\left(\tanh\frac{X_1}{2}\tanh\frac{X_2}{2}\right)$, the probability mass function of Z is

$$P(Z = z) = \sum_{z=2\tanh^{-1}\left(\tanh\frac{x_1}{2}\tanh\frac{x_2}{2}\right)} P(X_1 = x_1)P(X_2 = x_2). \tag{3.36}$$

If the function is in the form of $Z = 2\tanh^{-1}\left(\prod_i \tanh\frac{X_i}{2}\right)$, we can calculate P_Z by recursively calculating the probability mass function of the function of two input random variables with (3.36).

Probability mass functions of random variables X_1, X_2 and Z can be represented by vectors $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{Z}$, respectively. Assume that the length of the vectors is N. $x_{i,j} = p(X_i = (j - 1 - \frac{N-1}{2})\Delta)$ where $i = 1, 2$, $j = 1, 2, \cdots, N$, $\sum_j x_{i,j} = 1$ and Δ is an interval. To calculate the probability mass function of $Z = p(X_1, X_2)$, we first calculate a matrix $\mathbf{M}$, where

$$m_{i,j} = \text{round}\left(2\tanh^{-1}\left(\tanh\left(\frac{x_{1,i}}{2}\right)\tanh\left(\frac{x_{2,j}}{2}\right)\right)\right), \tag{3.37}$$

$i, j = 1, 2, \cdots, N$, and the function round(x) is to round the x to the nearest number in the set $\{i\Delta\}$ for $i \in \mathcal{Z}$. Then

$$z_k = p\left(Z = \left(k - 1 - \frac{N-1}{2}\right)\Delta\right) = \sum_{m_{i,j}=\left(k-1-\frac{N-1}{2}\right)\Delta} p(X_1 = x_{1,i})p(X_2 = x_{2,j}), \tag{3.38}$$

for $k = 1, 2, \cdots, N$.

Denote $P_{2,i}$ as the probability mass function of output messages from upper-degree-i Group 2 variable nodes. The probability mass function of input messages at Group 2 check node sockets is

$$\sum_i \lambda_{3,2,i} P_{2,i}. \tag{3.39}$$

Denote $Q_{2,i}$ as the probability mass function of output messages from upper layer check nodes with degree $d_{3,2} = i$. The probability mass function of input messages at Group 2 variable node sockets is

$$\sum_i \rho_{3,2,i} Q_{2,i}. \tag{3.40}$$

3.6 Code optimization

In this section, we propose a three-step code optimization to find good codes for half-duplex three-phase two-way relay channels.

In the first two code optimization steps, two irregular LDPC codes for the two source-relay links are designed. Since the underlying channels are point-to-point channels, any existing optimization methods [10, 11, 12, 8] for such channels can be used.

In this work, iterative linear programming [16] is used as the optimization solver. In this solver, the code rate is maximized when $\sigma^2_{3,i}$ is given for $i = 1, 2$. A feasible region is a space on λ_i and ρ_i where λ_i is a variable node degree distribution of irregular LDPC codes at terminal node i and ρ_i is a check node degree distribution of irregular LDPC codes at terminal node i. When the degree of ρ_i is concentrated [12], the optimization problem becomes a sequence of sub-problems: finding an optimal λ_i with a fixed ρ_i. The details of iterative linear programming for point-to-point channels can be found in Appendix II of [16].

In the third code optimization step, systematic LDPC codes at the relay node are optimized. The optimized irregular LDPC codes obtained in the first two steps are used in the third step. The optimization objective is to find the optimal degree distributions that minimize the ratio of the length of the relay codeword to the sum of the lengths of two source codewords

$$\frac{\lambda_{3,3,1}}{\sum_i \lambda_{3,1,i} + \sum_j \lambda_{3,2,j}}. \tag{3.41}$$

In this optimization problem, the feasible region is a space on $\lambda_{3,1}$, $\lambda_{3,2}$, $\lambda_{3,3}$, $\rho_{3,1}$ and $\rho_{3,2}$. To simplify this task, the original problem is divided into a sequence of optimization problems on $\lambda_{31}, \lambda_{32}, \lambda_{33}$ with fixed ρ_{31}, ρ_{32}.

The global optimization problem in the third optimization step is

$$\min_{\lambda_{3,1},\lambda_{3,2},\lambda_{3,3}} \lambda_{3,3,1} \tag{3.42}$$

$$\text{s.t.} \sum_i \lambda_{3,1,i} + \sum_j \lambda_{3,2,j} + \lambda_{3,3,1} = 1 \tag{3.43}$$

$$0 \leq \lambda_{3,1,i}, \lambda_{3,2,j}, \lambda_{3,3,1} \leq 1 \tag{3.44}$$

$$\sum_i i\lambda_{3,1,i} - \left(\sum_j j\rho_{3,1,j}\right)\lambda_{3,3,1} = 0 \tag{3.45}$$

$$\sum_i i\lambda_{3,2,i} - \left(\sum_j j\rho_{3,2,j}\right)\lambda_{3,3,1} = 0 \tag{3.46}$$

$$\sum_j j(e^{l+1}_{3,1,j} - e^l_{3,1})\lambda_{3,1,j} < 0, l = 1, \cdots, L_1 \tag{3.47}$$

$$\sum_j j(e^{l+1}_{3,2,j} - e^l_{3,2})\lambda_{3,2,j} < 0, l = 1, \cdots, L_2, \tag{3.48}$$

where L_1 and L_2 are the total numbers of decoding iterations, $e^l_{3,i,j}$ is the probability of error on upper-degree-j edges of Group i variable nodes in the l-th decoding iteration, and $e^l_{3,i}$ is the probability-of-error mixture on Group i check node sockets in the l-th decoding iteration. Note that the probability of error is calculated during discretized density evolution by $\sum_{a\leq 0} P(a)$, where P is the probability mass function of messages.

The probability-of-error mixture $e^l_{3,i}$ is calculated from the probability mass function of the message mixture at inputs of check nodes.

With constraint (3.43), (3.41) becomes (3.42). Constraint (3.43) is the condition that the sum of the probabilities is 1. Constraint (3.44) is the condition that a probability is upper bounded by 1 and lower bounded by 0.

Constraint (3.45) comes from

$$\sum_i ni\lambda_{3,1,i} = \sum_j n_3 j\rho_{3,1,j}. \tag{3.49}$$

The left-hand side (LHS) of (3.49) is the total number of upper edges connected to Group 1 variable nodes, where n is the total number of all three groups of variable nodes. In addition, from the upper layer check node perspective, the total number of upper edges connected to Group 1 variable nodes is the right-hand side (RHS) of (3.49), where n_3 is the total number of upper layer check nodes. The LHS and the RHS should be equal, and this condition becomes constraint (3.45) due to $\lambda_{3,3,1} = \frac{n_3}{n}$. Constraint (3.46) is similar to (3.45), but it applies to Group 2 variable nodes.

Constraint (3.47) comes from

$$\sum_i e^{l+1}_{3,1,i} \frac{ni\lambda_{3,1,i}}{\sum_j nj\lambda_{3,1,j}} < e^l_{3,1}. \tag{3.50}$$

The LHS and the RHS of (3.50) are mixtures of probabilities of error of input messages at upper layer check nodes in the $(l+1)$-th and l-th decoding iteration, respectively. (3.47) is a sequence of constraints on the decoding rule; that is, the probability of error is monotonically decreased during iterative decoding. Constraint (3.48) is similar to (3.47) , but it applies to Group 2 variable nodes.

Since the probability of error is a non-linear function of the degree distribution, constraints (3.47) and (3.48) are non-linear. However, if the probabilities of error are treated as constants, the non-linear optimization problem becomes a linear optimization problem.

Since the probabilities of error are treated as constants, codes from linear programming might not be decodable. In this case, discretized density evolution can be used to verify whether codes can be decoded. If they can, their degree distribution becomes the current best degree distribution. During the discretized density evolution, the probability of error in each decoding iteration can be calculated and then used in the next optimization iteration. If codes cannot be decoded, the feasible region needs to be shrunk by reducing the value of the RHS of (3.47) and (3.48), denoted as μ, towards $-\infty$. When the feasible region is shrunk, the ratio (3.41) becomes larger. Hence decoding should be easier. As the value μ is reduced, the problem could be infeasible at some point. In other words, no degree distribution satisfies all constraints from (3.43) to (3.48). In such a case, we modify the iterative linear programming algorithm proposed in [16] by reducing L_1 and L_2. Since fewer constraints are applied, the feasible region is enlarged. Note that the probabilities of error in (3.47) and (3.48) come from the preceding optimization iteration, and are provided as hints on the boundary of the feasible region in the next optimization iteration. The optimal degree distribution could be inside or outside of the feasible region.

The iterative linear programming algorithm is shown in Algorithm 1. In the algorithm, we adjust the μ between a and b, where $a < b \leq 0$. When μ is close to b, codes cannot be decoded since the feasible region is large and $\lambda_{3,3,1}$ is small. When μ is close to a, the problem is infeasible, and so L_1 and L_2 are reduced. Otherwise, the optimized degree distribution is verified by discretized density evolution. If codes can be decoded, μ moves towards a higher value u, and $\lambda_{3,3,1}$ reduces. If codes cannot be decoded, μ moves towards a lower value l so that $\lambda_{3,3,1}$ becomes higher.

Algorithm 1 Iterative linear programming

```
$l \leftarrow a$, $u \leftarrow b$, $\mu \leftarrow \frac{l+u}{2}$
while $i \leq I$ do
    $i \leftarrow i + 1$
    $\lambda \leftarrow$ optimization($\mu$)
    if $\lambda$ is INFEASIBLE then
        if $L_1 = 0$ and $L_2 = 0$ then
            $l \leftarrow \mu$, $\mu \leftarrow \frac{\mu+u}{2}$
        else
            $L_1 \leftarrow L_1 - 1$ or $L_2 \leftarrow L_2 - 1$
        end if
    else
        if densityEvolution($\lambda$) = DECODED then
            $l \leftarrow \mu$, $\mu \leftarrow \frac{\mu+u}{2}$
        else
            $u \leftarrow \mu$, $\mu \leftarrow \frac{\mu+l}{2}$
        end if
    end if
end while
```

To initialize the above optimization problem, the following linear programming

$$\max \lambda_{3,3,1} \tag{3.51}$$

$$\text{s.t.} \sum_i \lambda_{3,1,i} + \sum_j \lambda_{3,2,j} + \lambda_{3,3,1} = 1 \tag{3.52}$$

$$0 \leq \lambda_{3,1,i}, \lambda_{3,2,j}, \lambda_{3,3,1} \leq 1 \tag{3.53}$$

$$\sum_i i\lambda_{3,1,i} - \left(\sum_j j\rho_{3,1,j}\right) \lambda_{3,3,1} = 0 \tag{3.54}$$

$$\sum_i i\lambda_{3,2,i} - \left(\sum_j j\rho_{3,2,j}\right) \lambda_{3,3,1} = 0 \tag{3.55}$$

is used to find the initial degree distribution and the corresponding probabilities of error. Since $\lambda_{3,3,1}$ is maximized, the code rate is small, which ensures that codes can be decoded.

3.7 Simulation results

In this section, two optimized degree distributions for irregular LDPC codes at terminal nodes and an optimized degree distribution for systematic LDPC codes at the relay node are obtained when a half-duplex three-phase two-way relay channel is given. Codes sampled from the optimized degree distributions are simulated. We show that good codes can be found with our proposed three-step optimization. The obtained optimized codes are 26% longer than the theoretic one. In addition, during simulation, the required SNR for a finite-length code converges fast to that for cycle-free infinite-length codes.

The first step optimizes irregular LDPC codes for terminal node 1, where a k_1-bit source message is encoded into an n_1-bit codeword. The code rate is $\frac{k_1}{n_1}$. The code optimization problem is to maximize the code rate when channel parameter $\sigma_{3,1}$ is given. Codes with rate 0.3277 are found when $\sigma_{3,1}$ is 1.295 and the check node degree is 8. Note that the capacity rate is $\frac{1}{3}$, which can be determined by the equation of the capacity of BIAWGN channels

$$C_{BIAWGN}(\sigma) = -\int \phi_\sigma(y)\log_2 \phi_\sigma(y)\mathrm{d}y - \frac{1}{2}\log_2(2\pi e\sigma^2) \tag{3.56}$$

where

$$\phi_\sigma(y) = \frac{1}{\sqrt{8\pi\sigma^2}}\left(e^{-\frac{(y+1)^2}{2\sigma^2}} + e^{-\frac{(y-1)^2}{2\sigma^2}}\right) \tag{3.57}$$

is the probability density function of received signal Y. The optimized variable node degree distribution is shown in Table 3.1.

Table 3.1: Variable node degree distribution for codes with rate 0.3277

i	$\lambda_{1,i}$	i	$\lambda_{1,i}$
2	0.5277	3	0.2903
6	0.0022	7	0.1392
21	0.0199	22	0.0003
100	0.0204		

In the second optimization step, for the link between terminal node 2 and the relay node, codes with rate 0.4852 are found when $\sigma_{3,2}$ is 0.979 and the check node degree is 8. The corresponding capacity rate is $\frac{1}{2}$. The optimized variable node degree distribution is shown in Table 3.2.

Two codes, Code A and Code B with the length of 10^5 bits, are randomly sampled from the above two degree distributions. Code A (B) is sampled from the degree distribution for irregular LDPC codes at node 1 (node 2). Simulation results are shown in Figure 3.10 and Figure 3.11, labeled as *Code A, no relay,* $n = 10^5$ and *Code B, no relay,* $n = 10^5$, respectively. The waterfall curve given in the figure can be considered as the case when no relay node exists. Each code is simulated with multiple SNRs. The maximum number of decoding iterations is 200. The corresponding bit error rate (BER) is presented in the logarithmic Y-axis. The equivalent SNRs of $\sigma_{3,1} = 1.295$ and $\sigma_{3,2} = 0.979$ are represented by the vertical lines.

Table 3.2: Variable node degree distribution for codes with rate 0.4852

i	$\lambda_{2,i}$	i	$\lambda_{2,i}$
2	0.4928	3	0.2889
5	0.0011	6	0.0517
7	0.1050	8	0.0010
9	0.0007	10	0.0091
11	0.0005	12	0.0004
13	0.0003	14	0.0002
15	0.0001	16	0.0001
22	0.0183	23	0.0275
24	0.0001	25	0.0021

In the third optimization step, in order to optimize the degree distribution of systematic LDPC codes at the relay node, channel parameters $\sigma_{1,2}$, $\sigma_{1,3}$, $\sigma_{2,1}$, $\sigma_{2,3}$ are given. We consider the case when $\sigma_{2,1} = \sigma_{2,3} = 1.9483$ and $\sigma_{1,2} = \sigma_{1,3} = 1.5490$. The degree of upper layer check nodes is $g_{3,1} = g_{3,2} = 3$. By iterative linear programming, the optimized $\lambda_{3,3,1}$ is 0.3867, and the optimized $\lambda_{3,1}$, $\lambda_{3,2}$ are shown in Table 3.3 and 3.4. Note that for the given $\sigma_{1,2}$, $\sigma_{1,3}$, $\sigma_{2,1}$, $\sigma_{2,3}$, the lower bound of $\lambda_{3,3,1}$ is $\frac{1}{3}$.

Table 3.3: Variable node degree distribution $\lambda_{3,1}$

i	$\lambda_{3,1,i}$	i	$\lambda_{3,1,i}$
0	0.2238	14	0.0829

Irregular LDPC codes with the lengths of 10^3, 10^4 and 10^5 bits and systematic LDPC codes with the relay codeword lengths of 1.26×10^3, 1.26×10^4 and 1.26×10^5 bits are randomly sampled from the above degree distributions. Simulation results for decoding Code A and Code B at destination nodes are shown in Figure 3.10 and Figure 3.11, respectively. The maximum number of decoding iterations is 500. In the figure, the SNRs are defined as $\frac{1}{\sigma_{1,3}^2}$ and $\frac{1}{\sigma_{2,3}^2}$ for the two decoders at two destination nodes. The BER is

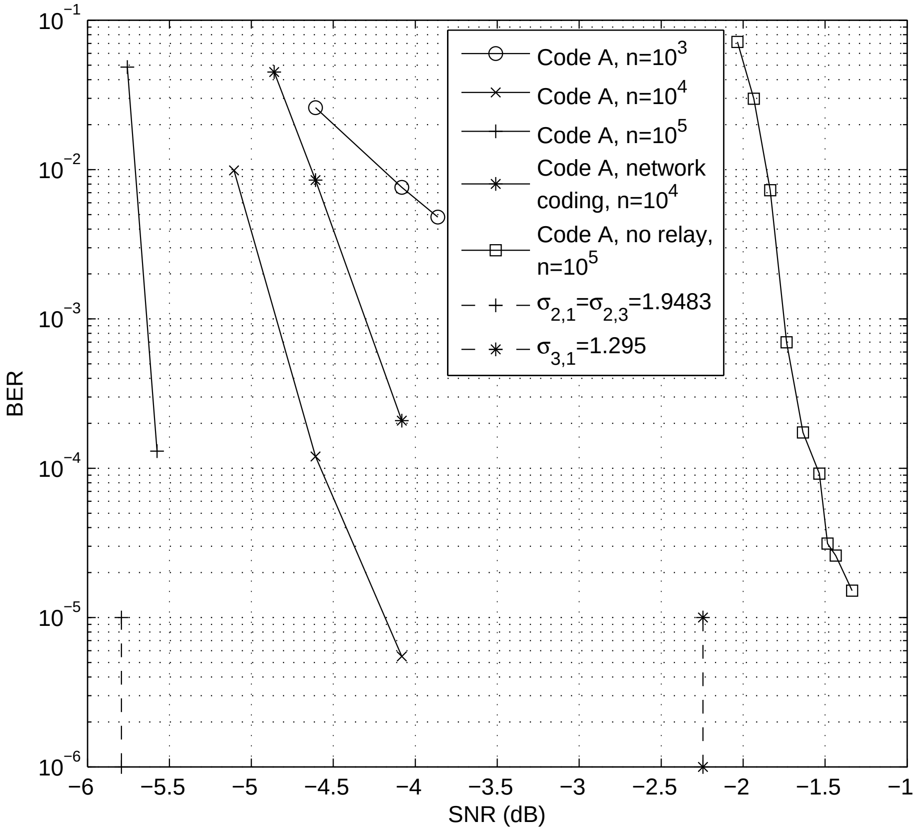

Figure 3.10: Simulation results when decoding code A at terminal node 2

defined as the ratio of the total number of erroneous bits to the total number of bits in Group i variable nodes for $i = 1, 2$. In this sense, BERs are evaluated at two destination nodes separately. The equivalent SNRs of $\sigma_{2,1} = \sigma_{2,3} = 1.9483$ and $\sigma_{1,2} = \sigma_{1,3} = 1.5490$ are represented by vertical dashed lines. As we can see, with the help of the relay node, the required SNRs are reduced from 0.2 dB to -3.8 dB and -2.2 dB to -5.7 dB, respectively. In Figure 3.10 and Figure 3.11, we also simulate the network coding case where two source codewords are added in GF(2) at the relay node. At the BERs of 10^{-4}, the required SNR of the network coding case is around 1.5 dB higher than that of our proposed LDPC code

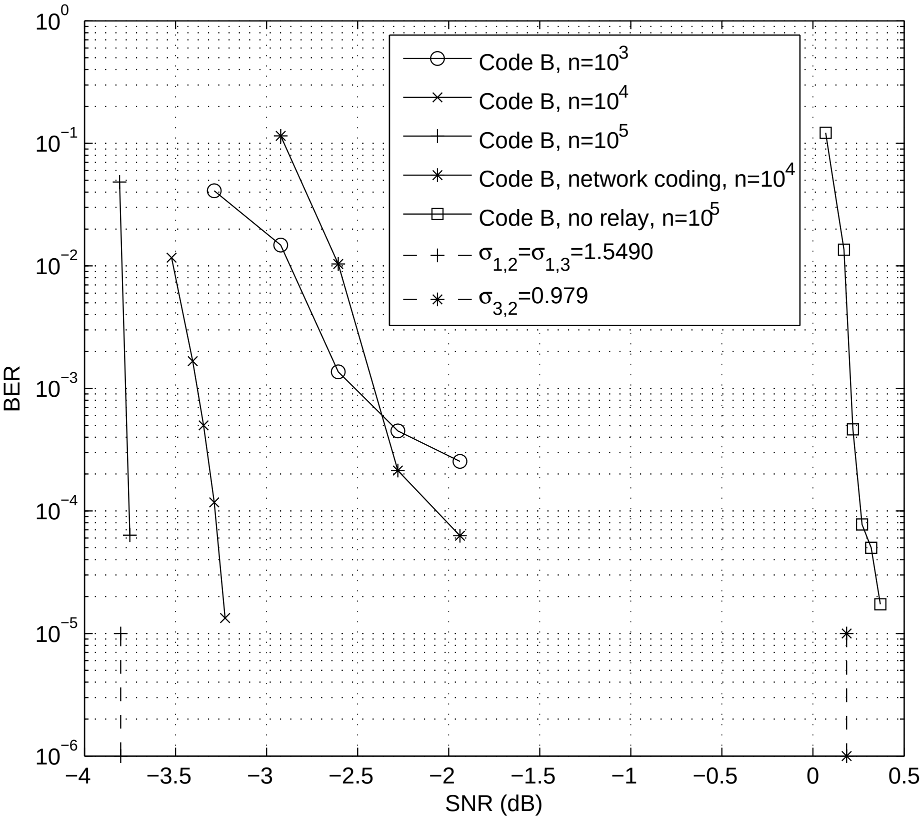

Figure 3.11: Simulation results when decoding code B at terminal node 1

construction.

In density evolution, it is assumed that incoming messages of variable nodes and check nodes are independent. This assumption implies that the bipartite graph has no cycles. However, cycles almost always exist. A natural question is whether the actual density is close to the density in density evolution, especially when deviation is accumulated during the iterative decoding. This question can be empirically answered by Figure 3.10 and Figure 3.11. As we can see, when the length of codewords grows from 10^3 to 10^5, the waterfall curve moves closer to the vertical dashed lines, which shows that the required

Table 3.4: Variable node degree distribution $\lambda_{3,2}$

i	$\lambda_{3,2,i}$	i	$\lambda_{3,2,i}$
0	0.1482	1	0.0148
2	0.0131	3	0.0121
4	0.0113	5	0.0109
6	0.0107	7	0.0106
8	0.0107	9	0.0108
10	0.0106	11	0.0094
12	0.0059	13	0.0036
14	0.0239		

SNR for a finite-length code converges fast to that for cycle-free infinite-length codes.

For the given optimized degree distributions, the evolution of the BER under discretized density evolution is shown in Figure 3.12. $P_1 = P_2 = P_3 = 1$, $\sigma_{1,2} = \sigma_{1,3} = 1.5490$ and $\sigma_{2,1} = \sigma_{2,3} = 1.9483$ are used. The two BER curves are monotonically decreasing during iterative decoding. The required decoding iterations at two destination nodes are close to 100 and 300, respectively. The decrease of the BER as a function of the current BER is shown in Figure 3.13. The critical point [11] is where the decrease of the BER is a local minimum. As shown in Figure 3.13, the critical points of two codes at two destination nodes are close to 0.12 and 0.19, respectively.

3.8 Conclusion

In half-duplex three-phase two-way relay channels, codewords are broadcast and signals are received from the source node and the relay node. In this work, we constructed systematic LDPC codes at the relay node to encode two source codewords. At the destination node, signals from the source node and the relay node are used for joint decoding. We designed the codes with discretized density evolution and iterative linear programing, and demonstrated

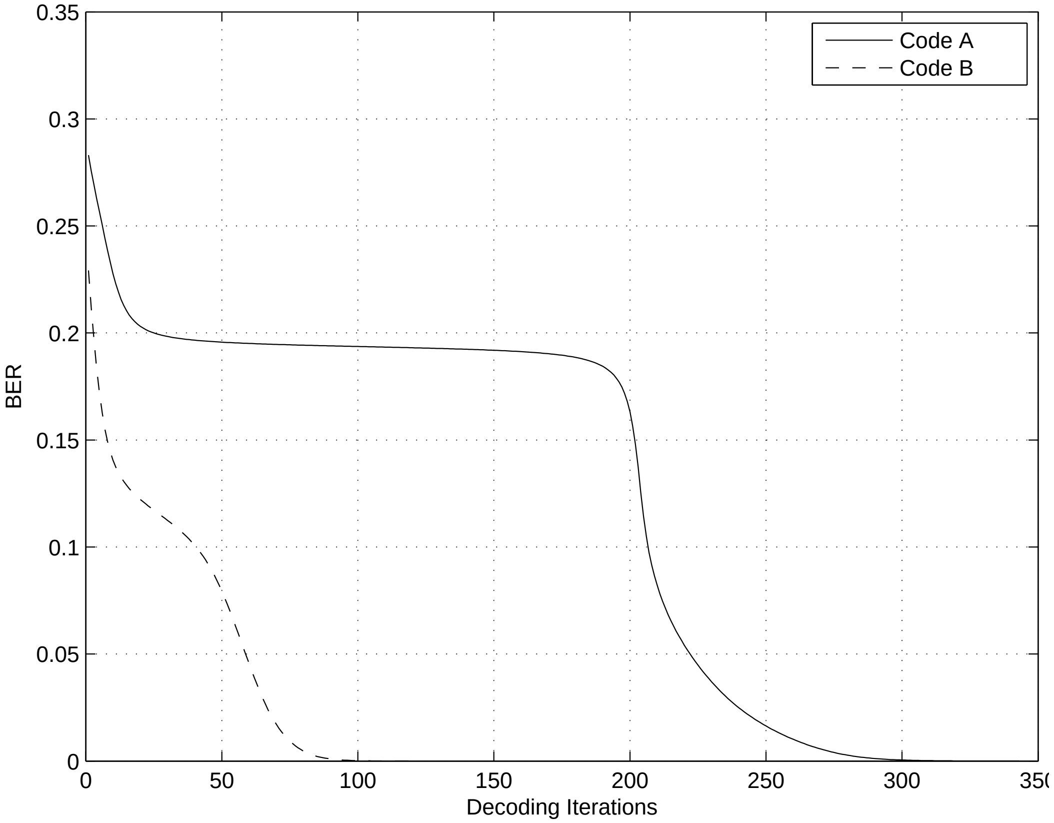

Figure 3.12: Evolution of the bit error rate during iterative decoding

that good codes can be found within our framework. For future work, structured codes, short codes, codes for high spectral density and codes on GF(q) can be studied. In addition, this work can be extended to fading channels and wireless relay networks.

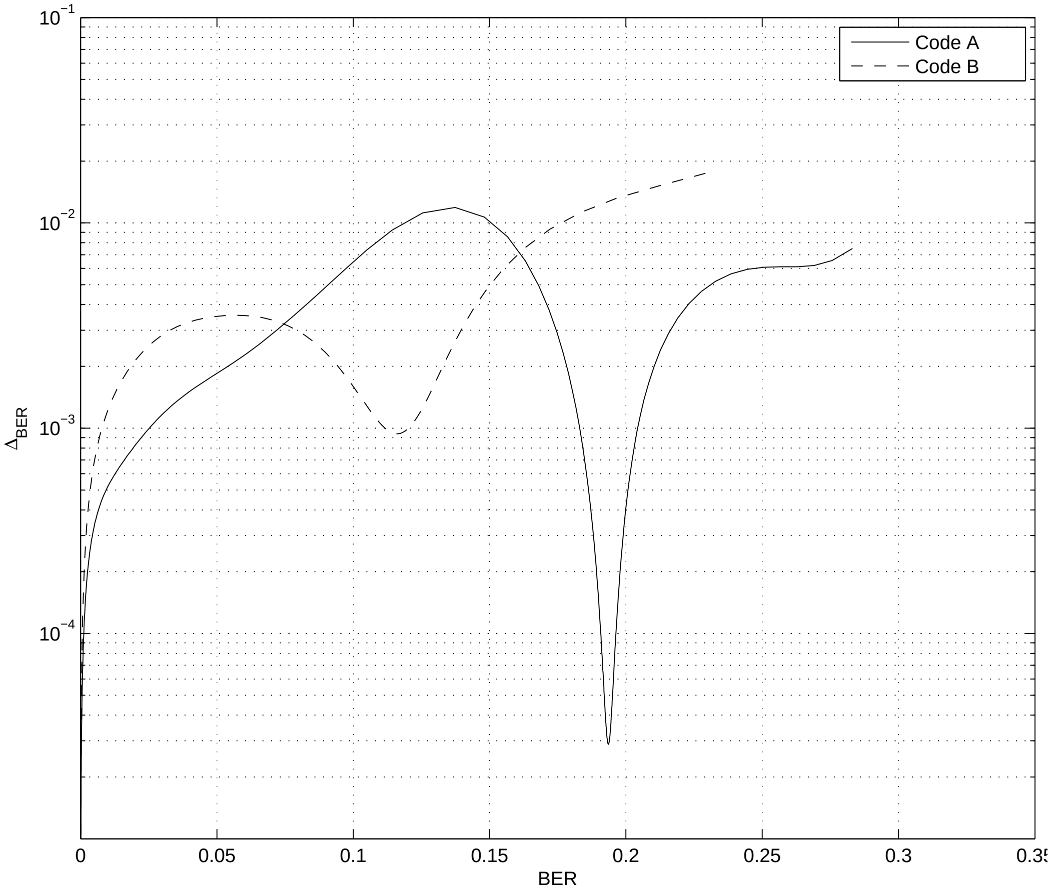

Figure 3.13: Decrease of the bit error rate as a function of the current bit error rate

Chapter 4

Low-density Parity-check Codes for Half-duplex Three-way Relay Channels

4.1 Introduction

In this chapter, we study practical coding schemes for wireless relay networks with one step forward. To be specific, the study of low-density parity-check (LDPC) codes is extended from half-duplex three-phase two-way relay channels to half-duplex three-way relay channels. In three-way relay channels, each of the three nodes broadcasts its messages to the other two nodes. In section 4.2, an achievable rate region of half-duplex three-way relay channels is proved. In the proof, the region is divided into seven sub-regions. For each sub-region, a random code is constructed. A node may relay one or two messages, or may not relay any messages. Inspired by random coding, LDPC codes are constructed for each sub-region in section 4.3. The code constructions can be generalized into two categories. In the first category, relay bits are generated only from received codewords. In the second category, bits are generated from both a source codeword and received codewords. Finally, section 4.4 concludes the chapter.

4.2 An achievable rate region of half-duplex three-way relay channels

In this section, an achievable rate region of half-duplex three-way relay channels is proved. Note that an achievable rate region for full-duplex three-way relay channels was given in [20].

Without loss of generality, it is assumed that node i broadcasts signals in phase i for $i = 1, 2, 3$. In each phase, a node sends a message and helps to relay messages when necessary.

A three-way relay channel consists of source input alphabet sets $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, channel output alphabet sets $\mathcal{Y}_1$, $\mathcal{Y}_2$, $\mathcal{Y}_3$ and a distribution $p(y_1, y_2, y_3|x_1, x_2, x_3)$. Considering time division, distributions during phase 1, phase 2 and phase 3 are $p(y_2, y_3|x_1)$, $p(y_1, y_3|x_2)$ and $p(y_1, y_2|x_3)$, respectively. Assume the lengths of codewords in three phases are n_1, n_2 and n_3 respectively, and $n = \sum_{i=1}^{3} n_i$. Set $\alpha = \frac{n_1}{n}$, $\beta = \frac{n_2}{n}$ and $\gamma = \frac{n_3}{n}$. A $((2^{nR_1}, 2^{nR_2}, 2^{nR_3}), n_1, n_2, n_3)$ code for the half-duplex three-way relay channel consists of three sets of integers $\mathcal{W}_1 = \{1, 2, \cdots, 2^{nR_1}\}$, $\mathcal{W}_2 = \{1, 2, \cdots, 2^{nR_2}\}$ and $\mathcal{W}_3 = \{1, 2, \cdots, 2^{nR_3}\}$.

Theorem 3. *For discrete memoryless half-duplex three-way relay channels, all rate tuples* (R_1, R_2, R_3) *satisfying*

$$R_1 < \max\left(\alpha I(X_1; Y_2), \alpha I(X_1; Y_3)\right), \tag{4.1}$$

$$R_2 < \max\left(\beta I(X_2; Y_3), \beta I(X_2; Y_1)\right), \tag{4.2}$$

$$R_3 < \max\left(\gamma I(X_3; Y_1), \gamma I(X_3; Y_2)\right) \tag{4.3}$$

and

$$R_1 + R_2 < \alpha I(X_1;Y_3) + \beta I(X_2;Y_3), \tag{4.4}$$

$$R_2 + R_3 < \beta I(X_2;Y_1) + \gamma I(X_3;Y_1), \tag{4.5}$$

$$R_3 + R_1 < \gamma I(X_3;Y_2) + \alpha I(X_1;Y_2) \tag{4.6}$$

for some $p(x_1)p(x_2)p(x_3)$ are achievable.

Proof. In order to prove the theorem, the achievable rate region is divided into smaller sub-regions. For each sub-region, the rate tuple is achievable by employing a coding construction. By symmetry, only seven cases need to be considered.

Case 1: (R_1, R_2, R_3) satisfies

$$R_1 < \min\left(\alpha I(X_1;Y_2), \alpha I(X_1;Y_3)\right), \tag{4.7}$$

$$R_2 < \min\left(\beta I(X_2;Y_3), \beta I(X_2;Y_1)\right), \tag{4.8}$$

$$R_3 < \min\left(\gamma I(X_3;Y_1), \gamma I(X_3;Y_2)\right). \tag{4.9}$$

In this case, each node i simply sends its own message w_i. Node j can decode w_i for $j \neq i$.

Codebook generation: Generate 2^{nR_1} codewords $\mathbf{x}_1 = x_1^{n_1}$ according to $\prod_{i=1}^{n_1} p(x_1)$ and index them as $\mathbf{x}_1(w_1), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}$. Generate 2^{nR_2} codewords $\mathbf{x}_2 = x_2^{n_2}$ according to $\prod_{i=1}^{n_2} p(x_2)$ and index them as $\mathbf{x}_2(w_2), w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$. Generate 2^{nR_3} codewords $\mathbf{x}_3 = x_3^{n_3}$ according to $\prod_{i=1}^{n_3} p(x_3)$ and index them as $\mathbf{x}_3(w_3), w_3 \in \{1, 2, \ldots, 2^{nR_3}\}$.

Encoding: In phase 1, to send index w_1, node 1 sends $\mathbf{x}_1(w_1)$. In phase 2, to send w_2, node 2 sends $\mathbf{x}_2(w_2)$. In phase 3, to send w_3, node 3 sends $\mathbf{x}_3(w_3)$.

Decoding: At the end of phase 1, node 2 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies the jointly typicality check $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{2,1}) \in A_\epsilon^{(n_1)}(X_1, Y_2)$ where $A_\epsilon^{(n_1)}(X_1, Y_2)$ is the set of jointly typical sequences of X_1 and Y_2. If there is no such or more than one such $\hat{w}_1$, an error is declared. Node 3 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies

$(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{3,1}) \in A_\epsilon^{(n_1)}(X_1, Y_3)$. If there is no such or more than one such w_1, an error is declared. Similarly, the joint typicality decoding is used at the end of phase 2 and phase 3.

Analysis of the probability of error: When node 1 sends, the probability that independent $\mathbf{x}_1$ and $\mathbf{y}_{2,1}$ are jointly typical is upper bounded by $2^{-n(I(X_1;Y_2)-3\epsilon)}$. There are totally $2^{nR_1}-1$ such $\mathbf{x}_1$. With the union bound, the probability of error at node 2 is upper bounded by $(2^{nR_1}-1)2^{-n_1(I(X_1;Y_2)-3\epsilon)}$, which approaches zero when $n_1 \to \infty$ and $R_1 < \alpha I(X_1;Y_2)$ (from $nR_1 - n_1 I(X_1;Y_2) < 0$ and $\alpha = \frac{n_1}{n}$). Similarly, we need $R_1 < \alpha I(X_1;Y_3)$ for node 3 to decode $\mathbf{x}_1$, $R_2 < \beta I(X_2;Y_1)$ for node 1 to decode $\mathbf{x}_2$, $R_2 < \beta I(X_2;Y_3)$ for node 3 to decode $\mathbf{x}_2$, $R_3 < \gamma I(X_3;Y_1)$ for node 1 to decode $\mathbf{x}_3$ and $R_3 < \gamma I(X_3;Y_2)$ for node 2 to decode $\mathbf{x}_3$.

Case 2: (R_1, R_2, R_3) satisfies

$$\alpha I(X_1;Y_2) \leq R_1 < \alpha I(X_1;Y_3), \tag{4.10}$$

$$R_2 < \min\left(\beta I(X_2;Y_3), \beta I(X_2;Y_1)\right), \tag{4.11}$$

$$R_3 < \min\left(\gamma I(X_3;Y_1), \gamma I(X_3;Y_2)\right). \tag{4.12}$$

In this case, node 3 relays messages of node 1, and node 1 and node 2 do not relay messages.

Codebook generation: Generate 2^{nR_1} codewords $\mathbf{x}_1 = x_1^{n_1}$ according to $\prod_{i=1}^{n_1} p(x_1)$ and index them as $\mathbf{x}_1(w_1), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}$. Generates 2^{nR_2} codewords $\mathbf{x}_2 = x_2^{n_2}$ according to $\prod_{i=1}^{n_2} p(x_2)$ and index them as $\mathbf{x}_2(w_2), w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$. Generate $2^{n(R_3+R_1)}$ codewords $\mathbf{x}_3 = x_3^{n_3}$ according to $\prod_{i=1}^{n_3} p(x_3)$ and index them as $\mathbf{x}_3(w_3, w_1)$, $w_3 \in \{1, 2, \ldots, 2^{nR_3}\}$, $w_1 \in \{1, 2, \ldots, 2^{nR_1}\}$.

Encoding: In phase 1, to send w_1, node 1 sends $\mathbf{x}_1(w_1)$. In phase 2, to send index w_2, node 2 sends $\mathbf{x}_2(w_2)$. In phase 3, node 3 sends $\mathbf{x}_3(w_3, \hat{w}_1)$ after decoding w_1.

Decoding: At the end of phase 1, node 3 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{3,1}) \in A_\epsilon^{(n_1)}(X_1;Y_3)$. If there is no such or more than one such $\hat{w}_1$, an error is declared. At the end of phase 2, node 1 decodes w_2 by finding the unique $\hat{w}_2$ that satisfies $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{1,2}) \in A_\epsilon^{(n_2)}(X_2; Y_1)$. If there is no such or more than one such $\hat{w}_2$, an error is declared. Node 3 decodes w_2 by finding the unique $\hat{w}_2$ that satisfies $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{3,2}) \in A_\epsilon^{(n_2)}(X_2; Y_3)$. If there is no such or more than one such $\hat{w}_2$, an error is declared. At the end of phase 3, node 1 decodes w_3 by finding the unique $\hat{w}_3$ that satisfies $(\mathbf{x}_3(\hat{w}_3, w_1), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3; Y_1)$. If there is no such or more than one such $\hat{w}_3$, an error is declared. Node 2 decodes w_3 and w_1 by finding the unique $(\hat{w}_3, \hat{w}_1)$ pair that satisfies $(\mathbf{x}_3(\hat{w}_3, \hat{w}_1), \mathbf{y}_{2,3}) \in A_\epsilon^{(n_3)}(X_3; Y_2)$ and $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{2,1}) \in A_\epsilon^{(n_1)}(X_1; Y_2)$. If there is no such or more than one such $(\hat{w}_3, \hat{w}_1)$ pair, an error is declared.

Analysis of the probability of error: First, $R_2 < \beta I(X_2; Y_1)$ is needed for node 1 to decode $\mathbf{x}_2$, $R_2 < \beta I(X_2; Y_3)$ is needed for node 3 to decode $\mathbf{x}_2$ and $R_1 < \alpha I(X_1; Y_3)$ is needed for node 3 to decode $\mathbf{x}_1$. When node 3 sends $\mathbf{x}_3(w_3, \hat{w}_1)$, the probability that independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical is upper bounded by $2^{-n_3(I(X_3;Y_1)-3\epsilon)}$. There are totally $2^{nR_3} - 1$ such $\mathbf{x}_3$ when node 1 knows w_1. With the union bound, the probability of error at node 1 is upper bounded by $(2^{nR_3} - 1)2^{-n_3(I(X_3;Y_1)-3\epsilon)}$, which approaches zero when $n_3 \to \infty$ and $R_3 < \gamma I(X_3; Y_1)$ (from $nR_3 - n_3 I(X_3; Y_1) < 0$ and $\gamma = \frac{n_3}{n}$). When node 3 sends, the probability that independent $\mathbf{x}_3$ and $\mathbf{y}_{2,3}$ are jointly typical is upper bounded by $2^{-n_3(I(X_3;Y_2)-3\epsilon)}$ and the probability that independent $\mathbf{x}_1$ and $\mathbf{y}_{2,1}$ are jointly typical is upper bounded by $2^{-n_1(I(X_1;Y_2)-3\epsilon)}$. There are totally $(2^{nR_3} - 1)(2^{nR_1} - 1)$ such $(\hat{w}_3, \hat{w}_1)$ pairs that $\hat{w}_3 \neq w_3$ and $\hat{w}_1 \neq w_1$. With the union bound, the probability of the event that any independent $\mathbf{x}_1$ and $\mathbf{y}_{2,1}$ are jointly typical and any independent $\mathbf{x}_3$ and $\mathbf{y}_{2,3}$ are jointly typical is upper bounded by $(2^{nR_3}-1)(2^{nR_1}-1)2^{-n_3(I(X_3;Y_2)-3\epsilon)}2^{-n_1(I(X_1;Y_2)-3\epsilon)}$, which approaches zero when $n_3 \to \infty$, $n_1 \to \infty$ and $R_3 + R_1 < \gamma I(X_3; Y_2) + \alpha I(X_1; Y_2)$ (from $nR_3 + nR_1 - n_3 I(X_3; Y_2) - n_1 I(X_1; Y_2) < 0$, $\gamma = \frac{n_3}{n}$ and $\alpha = \frac{n_1}{n}$). When $\hat{w}_1$ is correct, there are totally $2^{nR_3} - 1$ independent $\mathbf{x}_3(\hat{w}_3, w_1)$. The probability of the event that any independent $\mathbf{x}_3$ and $\mathbf{y}_{2,3}$ are jointly typical at node 2 is upper bounded by $(2^{nR_3} - 1)2^{-n_3(I(X_3;Y_2)-3\epsilon)}$, which approaches zero when $n_3 \to \infty$ and $R_3 < \gamma I(X_3; Y_2)$ (from $nR_3 - n_3 I(X_3; Y_2) < 0$ and $\gamma = \frac{n_3}{n}$).

Case 3: (R_1, R_2, R_3) satisfies

$$\alpha I(X_1;Y_3) \leq R_1 < \alpha I(X_1;Y_2), \tag{4.13}$$

$$\beta I(X_2;Y_1) \leq R_2 < \beta I(X_2;Y_3), \tag{4.14}$$

$$R_3 < \min\left(\gamma I(X_3;Y_1), \gamma I(X_3;Y_2)\right). \tag{4.15}$$

In case 3, node 2 relays messages of node 1, node 3 relays messages of node 2, and node 1 does not relay messages.

Codebook generation: Generate 2^{nR_1} codewords $\mathbf{x}_1 = x_1^{n_1}$ according to $\prod_{i=1}^{n_1} p(x_1)$ and index them as $\mathbf{x}_1(w_1), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}$. Generate $2^{n(R_1+R_2)}$ codewords $\mathbf{x}_2 = x_2^{n_2}$ according to $\prod_{i=1}^{n_2} p(x_2)$ and index them as $\mathbf{x}_2(w_1, w_2), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}, w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$. Generate $2^{n(R_2+R_3)}$ codewords $\mathbf{x}_3 = x_3^{n_3}$ according to $\prod_{i=1}^{n_3} p(x_3)$ and index them as $\mathbf{x}_3(w_2, w_3)$, $w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$, $w_3 \in \{1, 2, \ldots, 2^{nR_3}\}$.

Encoding: In phase 1, to send index w_1, node 1 sends $\mathbf{x}_1(w_1)$. In phase 2, node 2 sends $\mathbf{x}_2(\hat{w}_1, w_2)$ after decoding w_1. In phase 3, node 3 sends $\mathbf{x}_3(\hat{w}_2, w_3)$ after decoding w_2.

Decoding: At the end of phase 1, node 2 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{2,1}) \in A_\epsilon^{(n_1)}(X_1;Y_2)$. If there is no such or more than one such $\hat{w}_1$, an error is declared. At the end of phase 2, node 3 decodes w_1 and w_2 by finding the unique $(\hat{w}_1, \hat{w}_2)$ pair that satisfies $(\mathbf{x}_2(\hat{w}_1, \hat{w}_2), \mathbf{y}_{3,2}) \in A_\epsilon^{(n_2)}(X_2;Y_3)$ and $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{3,1}) \in A_\epsilon^{(n_1)}(X_1;Y_3)$. If there is no such or more than one such $(\hat{w}_1,\hat{w}_2)$ pair, an error is declared. At the end of phase 3, node 2 decodes w_3 by finding the unique $\hat{w}_3$ that satisfies $(\mathbf{x}_3(w_2, \hat{w}_3), \mathbf{y}_{2,3}) \in A_\epsilon^{(n_3)}(X_3;Y_2)$. If there is no such or more than one such $\hat{w}_3$, an error is declared. Node 1 decodes w_2 and w_3 by finding the unique $(\hat{w}_2, \hat{w}_3)$ pair that satisfies $(\mathbf{x}_2(w_1, \hat{w}_2), \mathbf{y}_{1,2}) \in A_\epsilon^{(n_2)}(X_2;Y_1)$ and $(\mathbf{x}_3(\hat{w}_2, \hat{w}_3), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3;Y_1)$. If there is no such or more than one such $(\hat{w}_2,\hat{w}_3)$ pair, an error is declared.

Analysis of the probability of error: First, $R_1 < \alpha I(X_1;Y_2)$ is needed for node 2 to decode $\mathbf{x}_1$. When node 2 sends, the probability of error for node 3 to decode the $(\hat{w}_1, \hat{w}_2)$ pair approaches zero when $n_1 \to \infty$, $n_2 \to \infty$, $R_1 + R_2 < \alpha I(X_1;Y_3) + \beta I(X_2;Y_3)$ and

$R_2 < \beta I(X_2; Y_3)$. When node 3 sends $\mathbf{x}_3(\hat{w}_2, w_3)$, the probability of error for node 2 to decode w_3 approaches zero when $n_3 \to \infty$ and $R_3 < \gamma I(X_3; Y_2)$. In addition, the probability that independent $\mathbf{x}_2$ and $\mathbf{y}_{1,2}$ are jointly typical is upper bounded by $2^{-n_2(I(X_2;Y_1)-3\epsilon)}$, and the probability that independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical is upper bounded by $2^{-n_3(I(X_3;Y_1)-3\epsilon)}$. There are totally $(2^{nR_2}-1)(2^{nR_3}-1)$ such $(\hat{w}_2, \hat{w}_3)$ pairs. With the union bound, the probability of the event that any independent $\mathbf{x}_2$ and $\mathbf{y}_{1,2}$ are jointly typical and independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical is upper bounded by $(2^{nR_2}-1)(2^{nR_3}-1)2^{-n_2(I(X_2;Y_1)-3\epsilon)}2^{-n_3(I(X_3;Y_1)-3\epsilon)}$, which approaches zero when $n_2 \to \infty$, $n_3 \to \infty$ and $R_2 + R_3 < \beta I(X_2; Y_1) + \gamma I(X_3; Y_1)$ (from $nR_2 + nR_3 - n_2 I(X_2; Y_1) - n_3 I(X_3; Y_1) < 0$, $\beta = \frac{n_2}{n}$ and $\gamma = \frac{n_3}{n}$). When $\hat{w}_2$ is correct, there are totally $2^{nR_3} - 1$ independent w_3. The probability of the event that any independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical at node 1 is $(2^{nR_3}-1)2^{-n_3(I(X_3;Y_1)-3\epsilon)}$, which approaches zero when $n_3 \to \infty$ and $R_3 < \gamma I(X_3; Y_1)$ (from $nR_3 - n_3 I(X_3; Y_1) < 0$ and $\gamma = \frac{n_3}{n}$).

Case 4: (R_1, R_2, R_3) satisfies

$$\alpha I(X_1; Y_2) \leq R_1 < \alpha I(X_1; Y_3), \tag{4.16}$$

$$\beta I(X_2; Y_3) > R_2 \geq \beta I(X_2; Y_1), \tag{4.17}$$

$$R_3 < \min\left(\gamma I(X_3; Y_1), \gamma I(X_3; Y_2)\right). \tag{4.18}$$

In this case, node 3 relays messages of both node 1 and node 2, and node 1 and node 2 do not relay messages.

Codebook generation: Generate 2^{nR_1} codewords $\mathbf{x}_1 = x_1^{n_1}$ according to $\prod_{i=1}^{n_1} p(x_1)$ and index them as $\mathbf{x}_1(w_1), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}$. Generate 2^{nR_2} codewords $\mathbf{x}_2 = x_2^{n_2}$ according to $\prod_{i=1}^{n_2} p(x_2)$ and index them as $\mathbf{x}_2(w_2), w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$. Generate $2^{n(R_1+R_2+R_3)}$ codewords $\mathbf{x}_3 = x_3^{n_3}$ according to $\prod_{i=1}^{n_3} p(x_3)$ and index them as $\mathbf{x}_3(w_3, w_1, w_2), w_3 \in \{1, 2, \ldots, 2^{nR_3}\}, w_1 \in \{1, 2, \ldots, 2^{nR_1}\}$ and $w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$.

Encoding: In phase 1, to send index w_1, node 1 sends $\mathbf{x}_1(w_1)$. In phase 2, to send index w_2, node 2 sends $\mathbf{x}_2(w_2)$. In phase 3, node 3 sends $\mathbf{x}_3(w_3, \hat{w}_1, \hat{w}_2)$ after decoding w_1 and w_2.

Decoding: At the end of phase 1, node 3 decodes w_1 by finding the unique $\hat{w}_1$ that satisfies $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{3,1}) \in A_\epsilon^{(n_1)}(X_1, Y_3)$. If there is no such or more than one such $\hat{w}_1$, an error is declared. At the end of phase 2, node 3 decodes w_2 by finding the unique $\hat{w}_2$ that satisfies $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{3,2}) \in A_\epsilon^{(n_2)}(X_2, Y_3)$. If there is no such or more than one such $\hat{w}_2$, an error is declared. At the end of phase 3, node 1 decodes w_3 and w_2 by finding the unique pair $(\hat{w}_3, \hat{w}_2)$ that satisfies $(\mathbf{x}_3(\hat{w}_3, w_1, \hat{w}_2), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3, Y_1)$ and $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{1,2}) \in A_\epsilon^{(n_2)}(X_2, Y_1)$. If there is no such or more than one such pair $(\hat{w}_3, \hat{w}_2)$, an error is declared. At the end of phase 3, node 2 decodes w_3 and w_1 by finding the unique pair $(\hat{w}_3, \hat{w}_1)$ that satisfies $(\mathbf{x}_3(\hat{w}_3, \hat{w}_1, w_2), \mathbf{y}_{2,3}) \in A_\epsilon^{(n_3)}(X_3, Y_2)$ and $(\mathbf{x}_1(\hat{w}_1), \mathbf{y}_{2,1}) \in A_\epsilon^{(n_1)}(X_1, Y_2)$. If there is no such or more than one such pair $(\hat{w}_3, \hat{w}_1)$, an error is declared.

Analysis of the probability of error: First, $R_1 < \alpha I(X_1; Y_3)$ is needed for node 3 to decode $\mathbf{x}_1$ and $R_2 < \beta I(X_2; Y_3)$ for node 3 to decode $\mathbf{x}_2$. When node 3 sends $\mathbf{x}_3(w_3, \hat{w}_1, \hat{w}_2)$, the probability that $(\mathbf{x}_3(\hat{w}_3, w_1, \hat{w}_2), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3, Y_1)$ when $\hat{w}_3 \neq w_3$ and $\hat{w}_2 \neq w_2$ at node 1 is upper bounded by $2^{-n_3(I(X_3;Y_1)-3\epsilon)}$, and the probability that $(\mathbf{x}_2(\hat{w}_2), \mathbf{y}_{1,2}) \in A_\epsilon^{(n_2)}(X_2, Y_1)$ is upper bounded by $2^{-n_2(I(X_2;Y_1)-3\epsilon)}$. There are totally $(2^{nR_2} - 1)(2^{nR_3} - 1)$ such $(\hat{w}_3, \hat{w}_2)$ pairs. With the union bound, the probability of the event that any independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical and any independent $\mathbf{x}_2$ and $\mathbf{y}_{1,2}$ are jointly typical at node 1 is $(2^{nR_2} - 1)(2^{nR_3} - 1)2^{-n_3(I(X_3;Y_1)-3\epsilon)}2^{-n_2(I(X_2;Y_1)-3\epsilon)}$, which approaches zero when $n_2 \to \infty$, $n_3 \to \infty$ and $R_2 + R_3 < \gamma I(X_3; Y_1) + \beta I(X_2; Y_1)$ (from $nR_2 + nR_3 - n_3 I(X_3; Y_1) - n_2 I(X_2; Y_1) < 0$, $\gamma = \frac{n_3}{n}$ and $\beta = \frac{n_2}{n}$). When $\hat{w}_2$ is correct and $\hat{w}_3$ is not correct, there are totally $2^{nR_3} - 1$ independent $\hat{w}_3$. The probability that $(\mathbf{x}_3(\hat{w}_3, w_1, w_2), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3, Y_1)$ when $\hat{w}_3 \neq w_3$ and $\hat{w}_2 = w_2$ at node 1 is upper bounded by $2^{-n_3(I(X_3;Y_1)-3\epsilon)}$. With the union bound, the probability of the event that any independent $\mathbf{x}_3$ and $\mathbf{y}_{1,3}$ are jointly typical at node 1 is $(2^{nR_3} - 1)2^{-n_3(I(X_3;Y_1)-3\epsilon)}$, which approaches zero when $n_3 \to \infty$ and $R_3 < \gamma I(X_3; Y_1)$ (from $nR_3 - n_3 I(X_3; Y_1) < 0$ and $\gamma = \frac{n_3}{n}$). By symmetry, when $R_1 + R_3 < \gamma I(X_3; Y_2) + \alpha I(X_1; Y_2)$ and $R_3 < \gamma I(X_3; Y_2)$, node 2 can decode the (w_3, w_1) pair with the probability of error approaching zero.

Case 5: (R_1, R_2, R_3) satisfies

$$\alpha I(X_1; Y_2) > R_1 \geq \alpha I(X_1; Y_3), \tag{4.19}$$

$$\beta I(X_2; Y_3) \leq R_2 < \beta I(X_2; Y_1), \tag{4.20}$$

$$R_3 < \min\left(\gamma I(X_3; Y_1), \gamma I(X_3; Y_2)\right). \tag{4.21}$$

This case is impossible because $R_1 + R_2 \geq \alpha I(X_1; Y_3) + \beta I(X_2; Y_3)$, which is conflict to (4.4).

Case 6: (R_1, R_2, R_3) satisfies

$$\alpha I(X_1; Y_3) \leq R_1 < \alpha I(X_1; Y_2), \tag{4.22}$$

$$\beta I(X_2; Y_1) \leq R_2 < \beta I(X_2; Y_3), \tag{4.23}$$

$$\gamma I(X_3; Y_2) \leq R_3 < \gamma I(X_3; Y_1). \tag{4.24}$$

In this case, node 1 relays messages of node 3, node 2 relays messages of node 1 and node 3 relays messages of node 2.

Codebook generation: Generate $2^{n(R_1+R_3)}$ codewords $\mathbf{x}_1 = x_1^{n_1}$ according to $\prod_{i=1}^{n_1} p(x_1)$ and index them as $\mathbf{x}_1(w_1, w_3), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}, w_3 \in \{1, 2, \ldots, 2^{nR_3}\}$. Generate $2^{n(R_1+R_2)}$ codewords $\mathbf{x}_2 = x_2^{n_2}$ according to $\prod_{i=1}^{n_2} p(x_2)$ and index them as $\mathbf{x}_2(w_1, w_2), w_1 \in \{1, 2, \ldots, 2^{nR_1}\}, w_2 \in \{1, 2, \ldots, 2^{nR_2}\}$. Generate $2^{n(R_2+R_3)}$ codewords $\mathbf{x}_3 = x_3^{n_3}$ according to $\prod_{i=1}^{n_3} p(x_3)$ and index them as $\mathbf{x}_3(w_2, w_3), w_2 \in \{1, 2, \ldots, 2^{nR_2}\}, w_3 \in \{1, 2, \ldots, 2^{nR_3}\}$.

Encoding: In phase 1, node 1 sends $\mathbf{x}_1(w_1, \hat{w}_3')$ where $\hat{w}_3'$ is the last decoded w_3. In phase 2, node 2 sends $\mathbf{x}_2(\hat{w}_1, w_2)$. In phase 3, node 3 sends $\mathbf{x}_3(\hat{w}_2, w_3)$.

Decoding: At the end of phase 1, node 2 decodes w_1 and w_3' by finding the unique $(\hat{w}_1, \hat{w}_3')$ pair that satisfies $(\mathbf{x}_1(\hat{w}_1, \hat{w}_3'), \mathbf{y}_{2,1}) \in A_\epsilon^{(n_1)}(X_1, Y_2)$ and $(\mathbf{x}_3(w_2', \hat{w}_3'), \mathbf{y}_{2,3}') \in A_\epsilon^{(n_3)}$ (X_3, Y_2) where w_2' is the last w_2 and $\mathbf{y}_{2,3}'$ is the last received $\mathbf{y}_{2,3}$. If there is no such or more than one such pair $(\hat{w}_1, \hat{w}_3')$, an error is declared. At the end of phase 2, node 3 decodes w_1 and w_2 by finding the unique $(\hat{w}_1, \hat{w}_2)$ pair that satisfies $(\mathbf{x}_1(\hat{w}_1, w_3'), \mathbf{y}_{3,1}) \in$

$A_\epsilon^{(n_1)}(X_1, Y_3)$ and $(\mathbf{x}_2(\hat{w}_1, \hat{w}_2), \mathbf{y}_{3,2}) \in A_\epsilon^{(n_2)}(X_2, Y_3)$. If there is no such or more than one such pair $(\hat{w}_1, \hat{w}_2)$, an error is declared. At the end of phase 3, node 1 decodes w_2 and w_3 by finding the unique $(\hat{w}_2, \hat{w}_3)$ pair that satisfies $(\mathbf{x}_2(w_1, \hat{w}_2), \mathbf{y}_{1,2}) \in A_\epsilon^{(n_2)}(X_2, Y_1)$ and $(\mathbf{x}_3(\hat{w}_2, \hat{w}_3), \mathbf{y}_{1,3}) \in A_\epsilon^{(n_3)}(X_3, Y_1)$. If there is no such or more than one such pair $(\hat{w}_2, \hat{w}_3)$, an error is declared.

Analysis of the probability of error: When node 1 sends, node 2 can decode w_1 and w_3' if $R_1 + R_3 < \alpha I(X_1; Y_2) + \gamma I(X_3; Y_2)$ and $R_1 < \alpha I(X_1; Y_2)$. When node 2 sends, node 3 can decode w_1 and w_2 if $R_1 + R_2 < \alpha I(X_1; Y_3) + \beta I(X_2; Y_3)$ and $R_2 < \beta I(X_2; Y_3)$. When node 3 sends, node 1 can decode w_2 and w_3 if $R_2 + R_3 < \beta I(X_2; Y_1) + \gamma I(X_3; Y_1)$ and $R_3 < \gamma I(X_3; Y_1)$.

Case 7: (R_1, R_2, R_3) satisfies

$$\alpha I(X_1; Y_2) \leq R_1 < \alpha I(X_1; Y_3), \tag{4.25}$$

$$\beta I(X_2; Y_3) \leq R_2 < \beta I(X_2; Y_1), \tag{4.26}$$

$$\gamma I(X_3; Y_1) > R_3 \geq \gamma I(X_3; Y_2). \tag{4.27}$$

This case is impossible because $R_1 + R_3 \geq \alpha I(X_1; Y_2) + \gamma I(X_3; Y_2)$, which is conflict to (4.6). □

Gaussian half-duplex three-way relay channels can be modeled as follows. In phase 1, node 1 broadcasts messages to node 2 and node 3. The received signals at node 2 and node 3 are $Y_{2,1} = X_1 + Z_{2,1}$ and $Y_{3,1} = X_1 + Z_{3,1}$ respectively where $Y_{i,j}$ is the received signal at node i in phase j, $Z_{i,j}$ is a Gaussian distributed random variable with mean zero and variance $\sigma_{i,j}^2$, and $\sigma_{i,j}^2$ is the noise power for the link from node j to node i. In phase 2, node 2 broadcasts messages to node 1 and node 3. The received signals at node 1 and node 3 are $Y_{1,2} = X_2 + Z_{1,2}$ and $Y_{3,2} = X_2 + Z_{3,2}$, respectively. In phase 3, node 3 broadcasts message to node 1 and node 2. The received signals at node 1 and node 2 are $Y_{1,3} = X_3 + Z_{1,3}$ and $Y_{2,3} = X_3 + Z_{2,3}$, respectively. The power constraint of signal $\mathbf{X}_1 = (X_{1,1}, \cdots, X_{1,n_1})$ is $\frac{1}{n_1}\sum_{i=1}^{n_1} X_{1,i}^2 \leq P_1$. Similarly, the power constraints of $\mathbf{X}_2$ and $\mathbf{X}_3$ are $\frac{1}{n_2}\sum_{i=1}^{n_2} X_{2,i}^2 \leq P_2$ and $\frac{1}{n_3}\sum_{i=1}^{n_3} X_{3,i}^2 \leq P_3$.

Theorem 4. *For Gaussian half-duplex three-way relay channels, all rate tuples* (R_1, R_2, R_3) *satisfying*

$$R_1 < \max\left(\frac{1}{2}\alpha\log\left(1+\frac{P_1}{N_{2,1}}\right), \frac{1}{2}\alpha\log\left(1+\frac{P_1}{N_{3,1}}\right)\right), \tag{4.28}$$

$$R_2 < \max\left(\frac{1}{2}\beta\log\left(1+\frac{P_2}{N_{3,2}}\right), \frac{1}{2}\beta\log\left(1+\frac{P_2}{N_{1,2}}\right)\right), \tag{4.29}$$

$$R_3 < \max\left(\frac{1}{2}\gamma\log\left(1+\frac{P_3}{N_{1,3}}\right), \frac{1}{2}\gamma\log\left(1+\frac{P_3}{N_{2,3}}\right)\right) \tag{4.30}$$

and

$$R_1 + R_2 < \frac{1}{2}\alpha\log\left(1+\frac{P_1}{N_{3,1}}\right) + \frac{1}{2}\beta\log\left(1+\frac{P_2}{N_{3,2}}\right), \tag{4.31}$$

$$R_2 + R_3 < \frac{1}{2}\beta\log\left(1+\frac{P_2}{N_{1,2}}\right) + \frac{1}{2}\gamma\log\left(1+\frac{P_3}{N_{1,3}}\right), \tag{4.32}$$

$$R_3 + R_1 < \frac{1}{2}\gamma\log\left(1+\frac{P_3}{N_{2,3}}\right) + \frac{1}{2}\alpha\log\left(1+\frac{P_1}{N_{2,1}}\right) \tag{4.33}$$

are achievable where $N_{i,j} = \sigma_{i,j}^2$ *is the noise power at the receiver of node* i *in phase* j *and* $\alpha + \beta + \gamma = 1$*.*

4.3 Low-density parity-check code constructions for half-duplex three-way relay channels

In Theorem 3, the achievable rate region of half-duplex three-way relay channels is divided into multiple sub-regions. In each sub-region, a random code is constructed since a terminal node may relay one message or two messages, or may not relay any messages. Inspired by Theorem 3, LDPC codes can be constructed for each sub-region. The constructions can be generalized into two categories. In both categories, the source codeword is first generated from the source message by non-systematic LDPC codes. Next, in category 1, relay bits are generated only from received codewords by systematic LDPC codes, and in category 2, bits are generated from both the source codeword and received codewords by systematic LDPC codes.

Category 1: In this category, relay bits are generated only from one or two received codewords.

First, consider the case when a terminal node encodes a k_1-bit source message into an n_1-bit codeword and helps to relay an n_2-bit codeword. m_3 relay bits are generated from the n_2-bit codeword by a systematic LDPC code. The terminal node sends the n_1-bit codeword and the m_3 relay bits. The n_2-bit codeword is not sent. The encoding graph is shown in Figure 4.1.

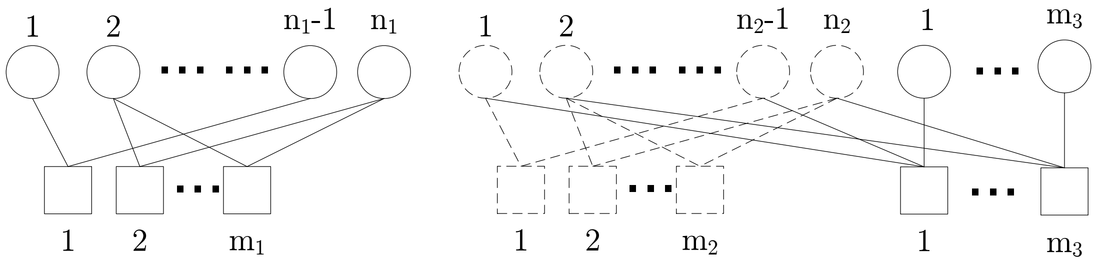

Figure 4.1: Graph for disjointly encoding a source codeword and a received codeword at a node

Next, consider the case when a terminal node helps to relay two codewords. The encoding graph is shown in Figure 4.2. The m_4 relay bits and the m_5 relay bits are generated from the n_2-bit codeword and the n_3-bit codeword by systematic LDPC codes. The terminal node sends the n_1-bit codeword, the m_4 relay bits and the m_5 relay bits. The n_2-bit codeword and the n_3-bit codeword are not sent.

When a destination node decodes the n_2-bit codeword with the help of the m_3 relay bits, the graph for joint decoding is shown in Figure 4.3. The n_2 bits are received from the source node. The m_3 bits are received from the relay node and appended to the source codeword for joint decoding.

Note that code constructions in category 1 can be extended to a general relay network

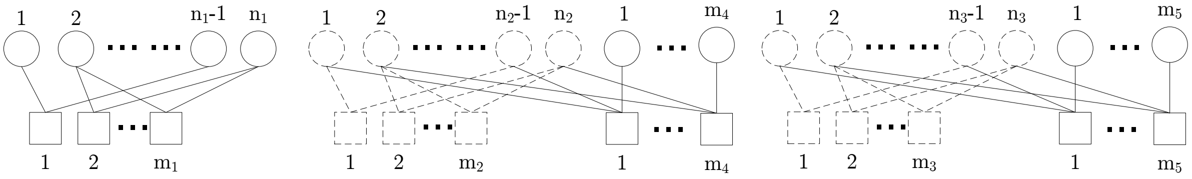

Figure 4.2: Graph for disjointly encoding a source codeword and two received codewords at a node

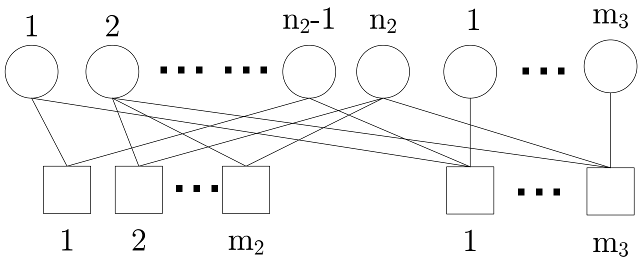

Figure 4.3: Graph for decoding when disjointly encoding a source codeword and a received codeword at a node

where a terminal node can help to relay multiple messages by generating a sequence of relay bits for each received codeword. In addition, if a codeword is helped by multiple relay nodes, multiple sequences of relay bits can be simply appended to the source codeword for joint decoding.

Category 2: In this category, relay bits are generated from both the source codeword and received codewords. In the following, LDPC codes are constructed for each sub-region in Theorem 3.

Case 1: In this case, the destination node decodes messages based on signals only from the source node. Hence, any existing code constructions for point-to-point channels can be used.

Case 2: In this case, node 3 relays messages of node 1, and node 1 and node 2 do not relay messages.

The encoding graph at node 3 is shown in Figure 4.4. Node 3 first encodes its own k_3-bit message into an n_3-bit codeword. Next, additional m_4 relay bits are generated from the n_3-bit codeword and the received n_1-bit codeword of node 1. Node 3 sends the n_3-bit source codeword and the m_4 relay bits. The n_1-bit codeword is not sent.

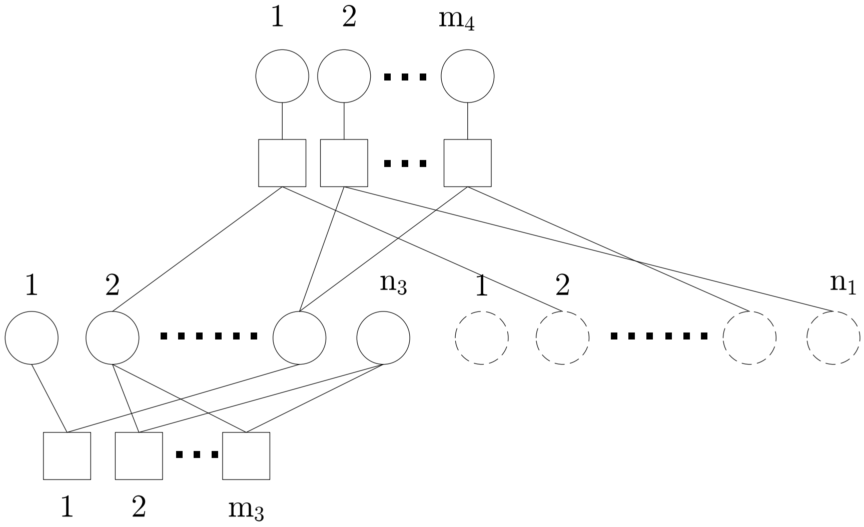

Figure 4.4: Graph for jointly encoding a source codeword and a received codeword at a node

When node 1 decodes messages of node 2, and node 3 decodes messages of node 1 and node 2, signals only from the source node are used.

Node 2 jointly decodes messages of node 1 and node 3. The decoding graph is shown in Figure 4.5. Node 2 receives the n_3-bit codeword and the m_4 relay bits from node 3, and receives the n_1-bit codeword from node 1.

Node 1 decodes the message of node 3 with the same graph in Figure 4.5. Node 1 receives the n_3-bit codeword and the m_4 relay bits from node 3. In addition, node 1 knows its own n_1-bit codeword, which is used for decoding as side information.

Case 3: In this case, node 2 relays messages of node 1, node 3 relays messages of node 2 and node 1 does not relay messages.

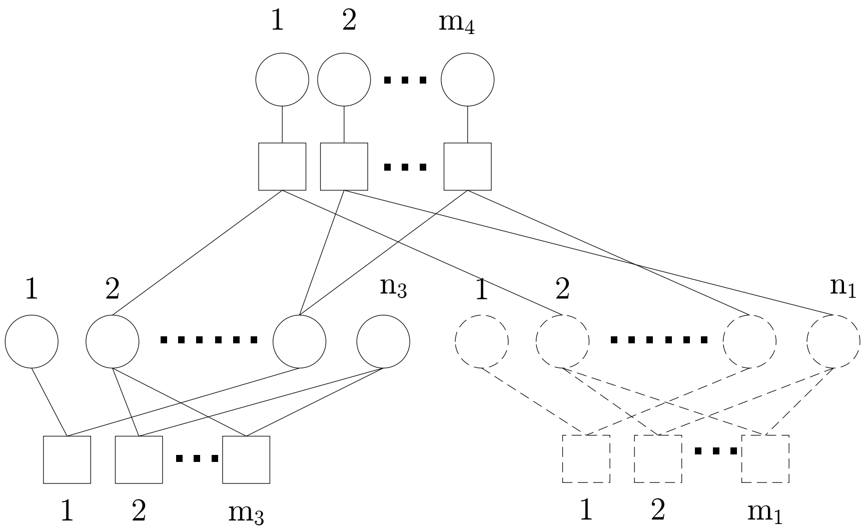

Figure 4.5: Graph for decoding when jointly encoding a source codeword and a received codeword at a node

The encoding graph at node 2 and node 3 is shown in Figure 4.6. In the left part of the figure, node 2 first encodes a k_2-bit source message into an n_2-bit codeword. Next, additional m_4 relay bits are generated from the n_2-bit source codeword and the received n_1-bit codeword of node 1. Node 2 sends the n_2-bit source codeword and the m_4 relay bits. The n_1-bit codeword is not sent. Similarly, in the right part of the figure, node 3 first encodes a k_3-bit source message into an n_3-bit codeword. Next, additional m_5 relay bits are generated from the n_3-bit source codeword and the received n_2-bit codeword of node 2. Node 3 sends the n_3-bit source codeword and the m_5 relay bits. The n_2-bit codeword is not sent.

Node 2 decodes messages of node 1 based on signals only from the source node. When node 3 jointly decodes messages of node 1 and node 2 from signals of node 1 and node 2, or when node 2 decodes messages of node 3 from signals of node 3 and its own codeword, the decoding graph is similar to the graph in Figure 4.5.

Node 1 decodes messages of node 2 and node 3 after receiving signals from node 2 and

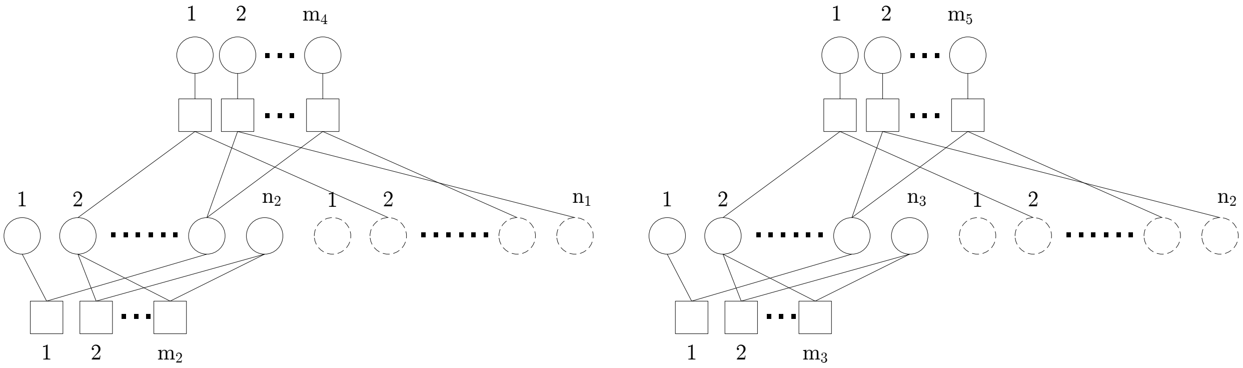

Figure 4.6: Graph for jointly encoding a source codeword and a received codeword at two nodes

node 3. The decoding graph is shown in Figure 4.7. Node 1 knows its own n_1-bit codeword, which is used for decoding as side information. The two sequences of n_2 bits in the graph are the same codeword. Thus they are combined in the decoding graph.

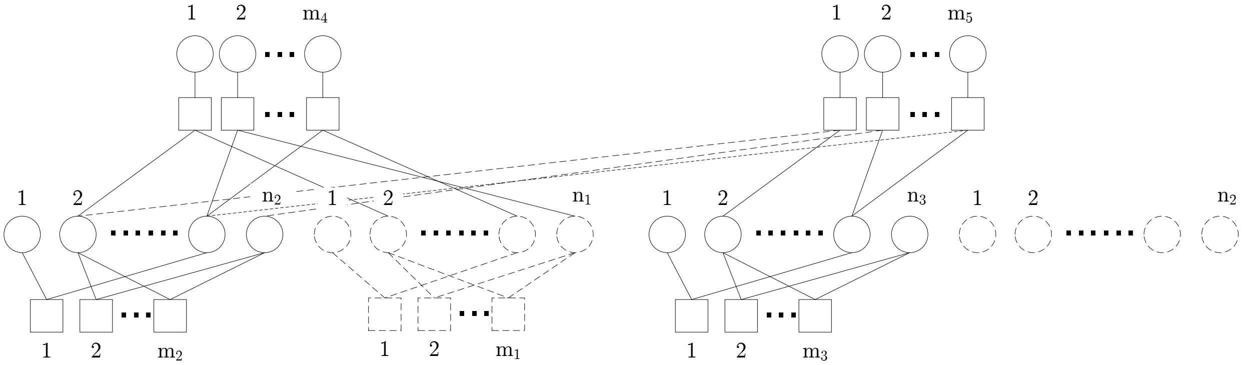

Figure 4.7: Graph for decoding when jointly encoding a source codeword and a received codeword at two nodes

Case 4: In this case, node 3 relays messages of both node 1 and node 2, and node 1 and node 2 do not relay messages.

The encoding graph at node 3 is shown in Figure 4.8. Node 3 first encodes a k_3-bit message into an n_3-bit codeword. Next, additional m_4 relay bits are generated based on the n_3-bit source codeword, the n_1-bit codeword of node 1 and the n_2-bit codeword of node 2. Node 3 sends the n_3-bit codeword and the m_4 relay bits. The n_1-bit codeword and the n_2-bit codeword are not sent.

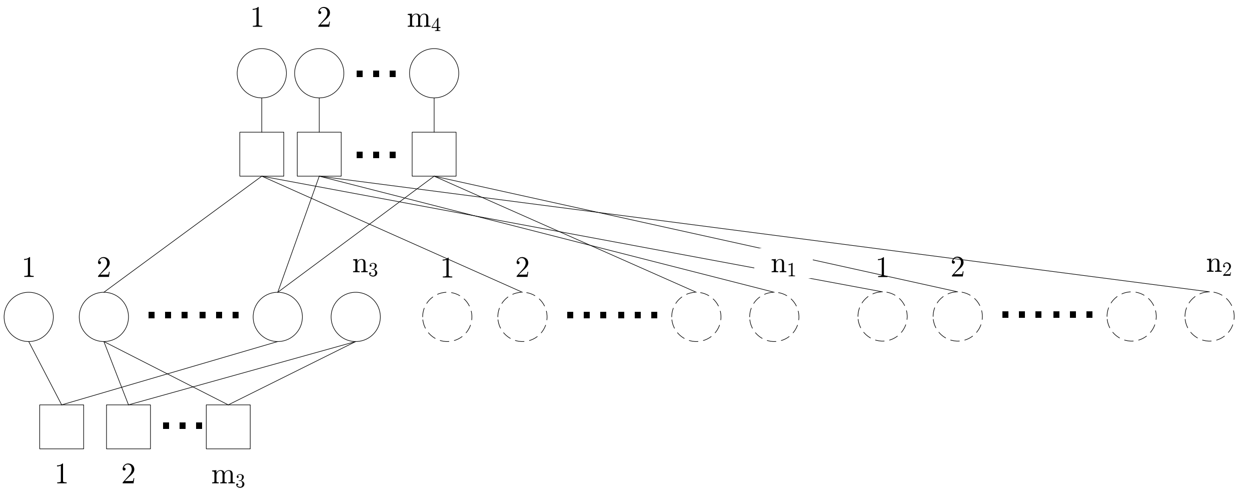

Figure 4.8: Graph for jointly encoding a source codeword and two received codewords at a node

Node 3 decodes messages of node 1 and node 2 based on signals only from the source.

When node 2 jointly decodes messages of node 1 and node 3, the decoding graph is shown in Figure 4.9. Node 2 receives the n_3-bit codeword and m_4 relay bits from node 3 and receives the n_1-bit codeword from node 1. In addition, node 2 knows its own n_2-bit codeword, which is used for decoding as side information.

When node 1 jointly decodes messages of node 2 and node 3, the same graph in Figure 4.9 is used. The only difference is that the n_2-bit codeword is received from node 2 and its own n_1-bit codeword is used for decoding as side information.

Case 5: This case is impossible.

Case 6: In this case, node 1 relays messages of node 3, node 2 relays messages of node 1 and node 3 relays messages of node 2. The graphs for encoding and decoding at all three nodes are similar to the graphs in Figure 4.6 and Figure 4.7, respectively.

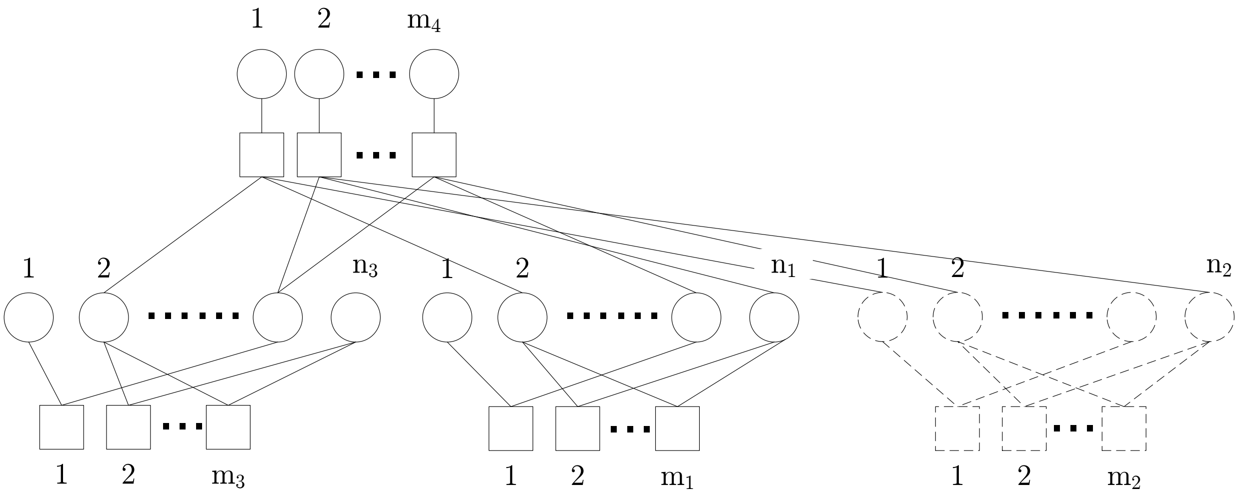

Figure 4.9: Graph for decoding when jointly encoding a source codeword and two received codewords at a node

Case 7: This case is impossible.

4.4 Conclusion

In this chapter, the study of LDPC codes was extended from half-duplex three-phase two-way relay channels to half-duplex three-way relay channels. An achievable rate region of half-duplex three-way relay channels was first proved. Furthermore, LDPC codes were constructed for each sub-region of the achievable rate region for half-duplex three-way relay channels.

Chapter 5

Relay Selection and Low-density Parity-check Codes for the Broadcast Problem in Wireless Relay Networks

5.1 Introduction

In this chapter, the single source broadcast problem in wireless relay networks is studied. In these networks, signals are broadcast and all nodes in the network can receive the signals. In [35], the optimal relay selection problem was studied to achieve the minimum energy. However, this problem is NP-Complete [36] due to the wireless broadcast property. To be specific, when a message is broadcast with higher power, the total energy could be reduced since more nodes can be reached. The minimum-energy problem was extended to the case where energies are accumulated over multiple transmission phases [37] and to the case where multiple nodes cooperatively transmit the message by beamforming [38]. The dual problem of [37] was recently studied in [39] as a minimum-delay problem with constraints on the energy and the bandwidth.

In this chapter, the maximum-rate broadcast problem for a given discrete memoryless wireless relay network is studied. In this problem, signals are broadcast and messages are decoded based on received signals from multiple transmission phases. The objective is to find a relay route that can achieve the maximum rate. This problem starts with the system model, followed by a theorem stating that a node can decode a message from multiple blocks of received signals even if the message cannot be decoded from any single block of signals in section 5.2. Section 5.3 shows the maximum rate for the hop-by-hop relay scheme. In addition, an optimal relay selection algorithm with the complexity of $\mathcal{O}(N^2)$ is proposed, compared with the complexity of $(N-1)!$ in a full search. In section 5.4, the maximum rate for the level-by-level relay scheme is shown. Two theorems that can prune the search space are proved. Furthermore, an optimal relay selection algorithm is proposed. Finally, low-density parity-check (LDPC) codes are constructed for the single source broadcast problem in section 5.5, and this chapter is concluded in section 5.6.

5.2 System model

A discrete memoryless wireless relay network $\mathcal{N}$ with N nodes consists of source input alphabet sets $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_N$, channel output alphabet sets $\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_N$ and a given distribution $p(y_1, y_2, \cdots, y_N | x_1, x_2, \cdots, x_N)$. Without loss of generality, node 1 sends messages to all other nodes $2, 3, \cdots, N$. A reliable node set contains nodes that have successfully decoded the message. The set is initialized to $\{1\}$. After node 1 transmits signals, a set of nodes that can decode the message is added to the reliable node set. In the following transmission phase, a node in the reliable node set is chosen to relay the message. The message is relayed until all nodes are added to the reliable node set.

A message w is chosen from an index set $\{1, 2, \cdots, M = 2^{nR}\}$. An encoding function $X_i^n : \{1, 2, \cdots, M\} \rightarrow \mathcal{X}_i^n$ generates codewords $x_i^n(1), x_i^n(2), \cdots, x_i^n(M)$ for the i-th node for $i = 1, 2, \cdots, N$. When node i transmits w, it sends $x_i^n(w)$. Assume the total number of transmission phases is K. Thus, the network rate R_n is defined as

$$R_n = \frac{\log M}{Kn} = \frac{1}{K} R. \tag{5.1}$$

When node 1 transmits signals, node i cannot decode the message if $R > I(X_1; Y_i)$. However, due to the broadcast property, node i can still receive the signals. When the message is relayed in multiple transmission phases, a node can accumulate multiple blocks of received signals. We now show a theorem stating that a node can decode the message from multiple blocks of received signals even if the message cannot be decoded from any single block of signals.

Theorem 5. *Assume a wireless relay network consists of source input alphabet sets $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_K$, channel output alphabet set $\mathcal{Y}$ and a given distribution $p(y|x_1, x_2, \cdots, x_K)$. In each transmission phase i for $i = 1, 2, \cdots, K$, node i sends the same message $w \in \{1, 2, \cdots, 2^{nR}\}$. A node Y overhears the signals in all K phases. Thus, all rates satisfying $R < \sum_i I(X_i; Y)$ for some $p(x_1)p(x_2) \cdots p(x_K)$ are achievable.*

Proof. Codebook generation: Generate 2^{nR} codewords $\mathbf{x}_i = x_i^n$ according to $\prod_{j=1}^n p(x_i)$ for node i and index them as $\mathbf{x}_i(w)$.

Encoding: In phase i, to send index w, node i sends $\mathbf{x}_i(w)$.

Decoding: Denote $\mathbf{y}_i$ as the channel output in phase i. At the end of phase K, node Y decodes w by finding the unique $\hat{w}$ that satisfies the joint typicality checks $(\mathbf{x}_i(\hat{w}), \mathbf{y}_i) \in A_\epsilon^{(n)}(X_i, Y)$ for all i. If there is no such or more than one such $\hat{w}$, an error is declared.

Analysis of the probability of error: The probability that independent $\mathbf{x}_i$ and $\mathbf{y}_i$ are jointly typical is upper bounded by $2^{-n(I(X_i;Y)-3\epsilon)}$ for all i. There are totally $2^{nR} - 1$ such w. With the union bound, the probability of error is upper bounded by $(2^{nR} - 1) \prod_i 2^{-n(I(X_i;Y)-3\epsilon)}$, which approaches zero when $n \to \infty$ and $R < \sum_i I(X_i; Y)$. □

5.3 Hop-by-hop relay

In this section, the maximum-rate broadcast problem for a given discrete memoryless wireless network is studied. In this problem, node 1 broadcasts messages to all other nodes

$2, 3, \cdots, N$. Signals are broadcast and messages are decoded based on received signals in multiple transmission phases. In each phase, one of the node not in the reliable node set can decode the message. The total number of phases in this hop-by-hop relay scheme is $N-1$.

Theorem 6. *For the broadcast problem in an N-node wireless relay network with the hop-by-hop relay scheme, the maximum rate of R_n over all relay routes is*

$$\max_{\boldsymbol{\pi}} \frac{1}{N-1} \min_{t=1,2,\cdots,N-1} \left[I(X_1; Y_{\pi_t}) + \sum_{i=1}^{t-1} I(X_{\pi_i}; Y_{\pi_t}) \right] \tag{5.2}$$

where node 1 is the source node, $\boldsymbol{\pi}$ is a set of all permutations on the node set $\{2, 3, \cdots, N\}$ and π_i is the i-th element in the permutation π.

Proof. In Theorem 6, a relay route is represented by a permutation π of the node set $\{2, 3, \cdots, N\}$. For a given π, from Theorem 5, all rates $R < \left[I(X_1; Y_{\pi_t}) + \sum_{i=1}^{t-1} I(X_{\pi_i}; Y_{\pi_t}) \right]$ are achievable for the node π_t. Considering all nodes in the wireless relay network, all rates

$$R < \min_{t=1,2,\cdots,N-1} \left[I(X_1; Y_{\pi_t}) + \sum_{i=1}^{t-1} I(X_{\pi_i}; Y_{\pi_t}) \right]$$

are achievable. Thus, the maximum rate of R_n over all relay routes is (5.2). □

To find the optimal relay route π^*, R_n can be calculated for all $(N-1)!$ permutations of the node set $\{2, 3, \cdots, N\}$. In the following, Algorithm 2 is proposed to find the optimal π^* with only $\sum_{i=1}^{N-1} i$ calculations.

First, assign node 1 to the dummy π_0^*. Next, initialize a set $\mathcal{S} = \{2, 3, \cdots, N\}$ which contains nodes waiting for their positions in the π. Initialize s_i to 0, where s_i is the accumulated rate for node i. Assume $\pi_1^*, \pi_2^*, \cdots, \pi_{j-1}^*$ have been determined. To determine the π_j^*, calculate $s_i = s_i + r_{\pi_{j-1}^*, i}$ where $r_{\pi_{j-1}^*, i}$ is the link capacity from node π_{j-1}^* to node i. Set π_j^* to n where n is the node with the maximum s_i for all $i \in \mathcal{S}$. Finally, remove node n from the set $\mathcal{S}$. This iterative process continues until π^* is fully determined.

Algorithm 2 The relay selection algorithm for the hop-by-hop relay scheme

```
$r_{i,j} \leftarrow I(X_i; Y_j), \forall i, j = 1, 2, \cdots, N, i \neq j$
$\mathcal{S} \leftarrow \{2, 3, \cdots, N\}$
$s_i \leftarrow 0, \forall i \in \mathcal{S}$
$j \leftarrow 1$
$\pi_0^* \leftarrow 1$
while $j \leq N - 1$ do
    $m \leftarrow 0$
    for $i \in \mathcal{S}$ do
        $s_i \leftarrow s_i + r_{\pi_{j-1}^*, i}$
        if $s_i > m$ then
            $m \leftarrow s_i$
            $n \leftarrow i$
        end if
    end for
    $\pi_j^* \leftarrow n$
    $\mathcal{S} \leftarrow \mathcal{S} - n$
    $j \leftarrow j + 1$
end while
```

Theorem 7. *The rate R_n^* of the route π^* given by Algorithm 2 is the maximum R_n over all possible relay routes.*

Proof. The node π_m^* is defined as the bottleneck node if the minimum of $\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_j^*}$ for all $j = 1, 2, \cdots, N-1$ is $\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{m-1}^*} r_{i,\pi_m^*}$.

Without loss of generality, we prove that by exchanging π_j^* with π_k^* for $j < k$, R_n would not exceed R_n^*.

If any node between π_1^* and π_{j-1}^* or between π_{k+1}^* and π_{N-1}^* is the bottleneck node π_m^*, R_n would not exceed R_n^* by exchanging π_j^* and π_k^* since $\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{m-1}^*} r_{i,\pi_m^*}$ does not change. If π_j^* is the bottleneck node, $R_n^* = \frac{1}{N-1}\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_j^*}$. After exchanging π_j^* and π_k^*, the R_n of the new route would be upper bounded by $\frac{1}{N-1}\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_k^*}$. Hence, R_n would not exceed R_n^* by exchanging π_j^* and π_k^* since $\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_j^*} \geq \sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_k^*}$. If π_k^* is the bottleneck node, $R_n^* = \frac{1}{N-1}\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{k-1}^*} r_{i,\pi_k^*}$. After exchanging π_j^* and π_k^*, the R_n of the new route would be upper bounded by $\frac{1}{N-1}$ $\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_k^*}$, which is less than R_n^*. Hence, R_n would not exceed R_n^* by exchanging π_j^* and π_k^*. Now we consider the case when a node between π_{j+1}^* and π_{k-1}^* is the bottleneck node. We prove this case by contradiction. Assume R_n would exceed R_n^* after exchanging π_j^* and π_k^*. By this assumption, $\sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{m-1}^*} r_{i,\pi_k^*} \geq \sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{j-1}^*} r_{i,\pi_k^*} > \sum_{i=1,\pi_1^*,\pi_2^*,\cdots,\pi_{m-1}^*} r_{i,\pi_m^*}$ and hence π_k^* would be ahead of π_m^*, which contradicts the fact that the bottleneck node is between π_{j+1}^* and π_{k-1}^*. Thus, R_n would not exceed R_n^* by exchanging π_j^* and π_k^*. □

5.4 Level-by-level relay

In an N-node wireless relay network $\mathcal{N}$ with the level-by-level relay scheme, a relay route is an ordered node subset $\mathcal{M} \subseteq \{2, 3, \cdots, N\}$. When a node $i \in \mathcal{M}$ transmits, all other nodes in the network can receive the signals. A set of nodes can decode the message based on all received signals in previous phases. In this section, an example is first given to show that the level-by-level relay scheme could achieve higher rate than the hop-by-hop relay scheme. Next, the maximum rate for the level-by-level relay scheme is shown. Finally, an optimal relay selection algorithm is proposed.

First, we show that the level-by-level relay scheme could achieve higher rates. Consider a 3-node wireless relay network where source node 1 broadcasts messages to node 2 and node 3, as shown in Figure 5.1. The network is constrained by $I(X_2;Y_3) < I(X_1;Y_3) < I(X_1;Y_2)$ and $\frac{1}{2}I(X_1;Y_2) < I(X_1;Y_3)$. If the hop-by-hop relay scheme is employed, the optimal relay route is $1 \to 2 \to 3$ and the achievable rate is $\min(\frac{1}{2}(I(X_1;Y_3) + I(X_2;Y_3)), \frac{1}{2}I(X_1;Y_2))$. However, with the level-by-level relay scheme, node 1 can broadcast at a higher rate $I(X_1;Y_3)$ and complete the transmission in one phase.

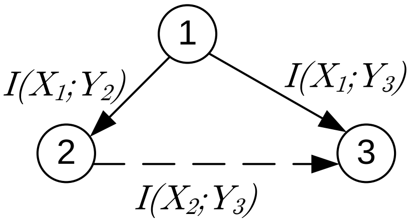

Figure 5.1: A three-node broadcast network

An example of the level-by-level relay scheme is shown in Figure 5.2. In this example, source node 1 broadcasts a message to 8 nodes in 4 phases. In phase 1, 4 nodes can decode the message. In phase 2, nodes 2 relays the message and two additional nodes can decode the message. In the last 2 phases, node 3 and node 4 relay the message. After 4 phases, all nodes can decode the message.

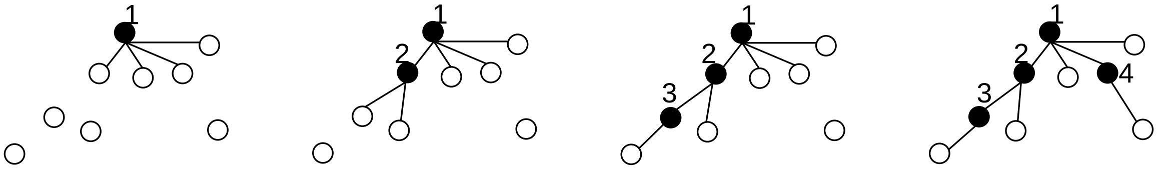

Figure 5.2: Level-by-level relay for the broadcast problem in wireless relay networks

Theorem 8. *For the level-by-level relay scheme, the maximum rate of R_n over all relay routes is*

$$\max_{\boldsymbol{\pi}} \frac{1}{|\pi|+1} \min\left(\min_{t=1,2,\cdots,|\pi|} \left(I(X_1;Y_{\pi_t}) + \sum_{i=1}^{t-1} I(X_{\pi_i};Y_{\pi_t}) \right), \min_{j \notin \pi} \left(I(X_1;Y_j) + \sum_{i=1}^{|\pi|} I(X_{\pi_i};Y_j) \right) \right) \tag{5.3}$$

where $\boldsymbol{\pi}$ is a set of all the ordered subset of $\{2,3,\cdots,N\}$, $|\pi|$ is the total number of elements in the π, and π_i is the i-th element in the π.

Proof. For a given π, from Theorem 5, all rates

$$R < I(X_1;Y_{\pi_t}) + \sum_{i=1}^{t-1} I(X_{\pi_i};Y_{\pi_t}) \tag{5.4}$$

are achievable for a relay node π_t. Similarly, all rates

$$R < I(X_1;Y_j) + \sum_{i=1}^{|\pi|} I(X_{\pi_i};Y_j) \tag{5.5}$$

are achievable for a non-relay node j. The achievable rate for the given π, denoted as $R_{n,\pi}$, is the minimum of (5.4) and (5.5) over nodes $2,3,\cdots,N$ divided by $|\pi|+1$. Hence, the maximum rate of R_n over all relay routes is (5.3). □

To find the optimal relay route π^*, an exhaustive searching algorithm can be used to search the entire space $\boldsymbol{\pi}$. For each π in the $\boldsymbol{\pi}$, (5.3) is calculated. The total number of relay routes with i relay nodes is $\frac{(N-1)!}{(N-1-i)!}$. For an N-node wireless relay network, the maximum number of relay nodes is $N-2$, excluding the source node and the last destination node. Hence the degree of the search space, or the total number of relay routes, is $\sum_{i=0}^{N-2} \frac{(N-1)!}{(N-1-i)!}$.

Theorem 9. *Assume there are totally N nodes in the wireless relay network. Assume node 1 is the source node. Denote $r_{1,i} = I(X_1;Y_i)$ for $i = 2,3,\cdots,N$. Denote $R_{max} = \max_i r_{1,i}$ and $R_{min} = \min_i r_{1,i}$. Then the degree of the optimal relay route π^* is less than or equal to $\left\lfloor \frac{R_{max}}{R_{min}} \right\rfloor - 1$ where $\lfloor x \rfloor$ is the largest integer not greater than x.*

Proof. We prove this theorem by contradiction. Assume the degree of the optimal relay route π^* is $\left\lfloor \frac{R_{max}}{R_{min}} \right\rfloor$ or higher. By this assumption, the maximum rate of R_n would be less than or equal to $\frac{R_{max}}{\left\lfloor \frac{R_{max}}{R_{min}} \right\rfloor + 1}$ which is less than R_{min}. However, this contradicts the fact that R_{min} is achievable since the source node can broadcast at the rate of R_{min} and complete the broadcast in one phase. □

With Theorem 9, the degree of the search space $\boldsymbol{\pi}$ can be reduced from $\sum_{i=0}^{N-2} \frac{(N-1)!}{(N-1-i)!}$ to $\sum_{i=0}^{\left\lfloor \frac{R_{max}}{R_{min}} \right\rfloor - 1} \frac{(N-1)!}{(N-1-i)!}$.

Theorem 10. *For the k-th node π_k in a π, denote $R_k = \sum_{i=1,\pi_1,\pi_2,\cdots,\pi_{k-1}} I(X_i; Y_{\pi_k})$. The achievable rate of a given route π, denoted as $R_{n,\pi}$, is less than or equal to $\frac{1}{|\pi|+1} R_k$.*

In the following, a branch and bound algorithm is proposed in Algorithm 3, which requires less computational complexity compared to that of the exhaustive search. The basic idea is to enumerate all possible π by breadth-first search. In addition, the search space is pruned by two bounds based on Theorem 9 and Theorem 10. Denote $R_{n,max}$ as the current maximum $R_{n,\pi}$ during the enumeration. From Theorem 10, if a node is added to a relay route π, the achievable rate would be upper bounded by $\frac{1}{|\pi|+2} R_k$. In addition, if $\frac{1}{|\pi|+2} R_k$ is less than $R_{n,max}$, adding relay nodes to the π will not increase the $R_{n,max}$. Initialize $R_{n,max} = \min_{i=2,\cdots,N} I(X_1; Y_i)$ and enqueue $\{\emptyset\}$ to a First-In-First-Out (FIFO) queue Q, where $\{\emptyset\}$ is a route only containing the source node. Dequeue a π' from the Q. For each node c that is not in the π', append c at the end of π' and get a new π. For each π, if $R_{n,\pi}$ is more than $R_{n,max}$, set $R_{n,max}$ to $R_{n,\pi}$. In addition, calculate R_k for $k = 1, 2, \cdots, |\pi|$. According to Theorem 9 and Theorem 10, if $\frac{1}{|\pi|+2} R_k \geq R_{n,max}$ for $k = 1, 2, \cdots, |\pi|$, and $|\pi| \leq \left\lfloor \frac{R_{max}}{R_{min}} \right\rfloor - 1$, enqueue the π to Q.

Algorithm 3 is simulated in the following settings. A wireless network of N nodes is generated where locations of the N nodes are uniformly distributed in a space $(0, 1) \times (0, 1)$. The transmission power of node i is P_i, and is uniformly distributed in $(1, 2)$. The link

Algorithm 3 The relay selection algorithm for the level-by-level relay scheme

```
$R_{n,max} \leftarrow \min_{i=2,3,\cdots,N} I(X_1;Y_i)$
Enqueue $(Q, \{\emptyset\})$
**while** $Q$ is not empty **do**
    $\pi' \leftarrow$ Dequeue$(Q)$
    **for** $c$ not in $\pi'$ **do**
        $\pi \leftarrow \pi' + c$
        $R_{n,\pi} \leftarrow \frac{1}{|\pi|+1} \min\left( \min_{t=1,\cdots,|\pi|} \left( I(X_1;Y_{\pi_t}) + \sum_{i=1}^{t-1} I(X_{\pi_i};Y_{\pi_t}) \right), \min_{j \notin \pi} \left( I(X_1;Y_j) + \sum_{i=1}^{|\pi|} I(X_{\pi_i};Y_j) \right) \right)$
        **if** $R_{n,\pi} > R_{n,max}$ **then**
            $R_{n,max} \leftarrow R_{n,\pi}$
            $\pi^* \leftarrow \pi$
        **end if**
        **for** $k = 1, 2, \cdots, |\pi|$ **do**
            $R_k = \sum_{i=1,\pi_1,\cdots,\pi_{k-1}} I(X_i;Y_{\pi_k})$
        **end for**
        **if** $\frac{1}{|\pi|+2} R_k \geq R_{n,max}$ for $k = 1, 2, \cdots, |\pi|$ and $|\pi| \leq \left\lfloor \frac{R_{max}}{R_{min}} \right\rfloor - 1$ **then**
            Enqueue$(Q, \pi)$
        **end if**
    **end for**
**end while**
```

rate from node i to node j is

$$r_{i,j} = \frac{1}{2} \log \left(1 + \frac{\frac{P_i}{d_{i,j}^{\alpha}}}{N_{i,j}} \right) \tag{5.6}$$

where $d_{i,j}$ is the geometric distance between node i and node j, and α is the path loss exponent. As an example, in Figure 5.3, a wireless network of 20 nodes is randomly generated. The noise power $N_{i,j} = 1$ and $\alpha = 2$. Assume the star is the source node. The size of a node corresponds to the power of the node. The optimal relay route is found and shown in the figure.

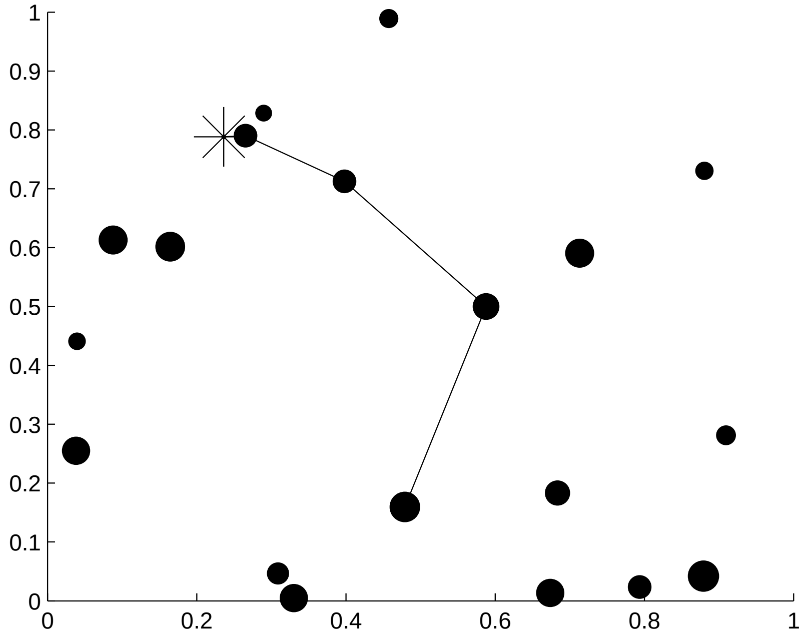

Figure 5.3: Optimal relay route of an example 20-node network

To see how many routes can be pruned by the two bounds, 100 wireless networks of 20 nodes are generated. The total number of searched routes for each network is recorded. The algorithm is stopped if more than 10^5 routes have been searched. The histogram of the number of searched routes is shown in Figure 5.4. As we can see, in most cases, the optimal relay route can be found quickly with the help of the two bounds.

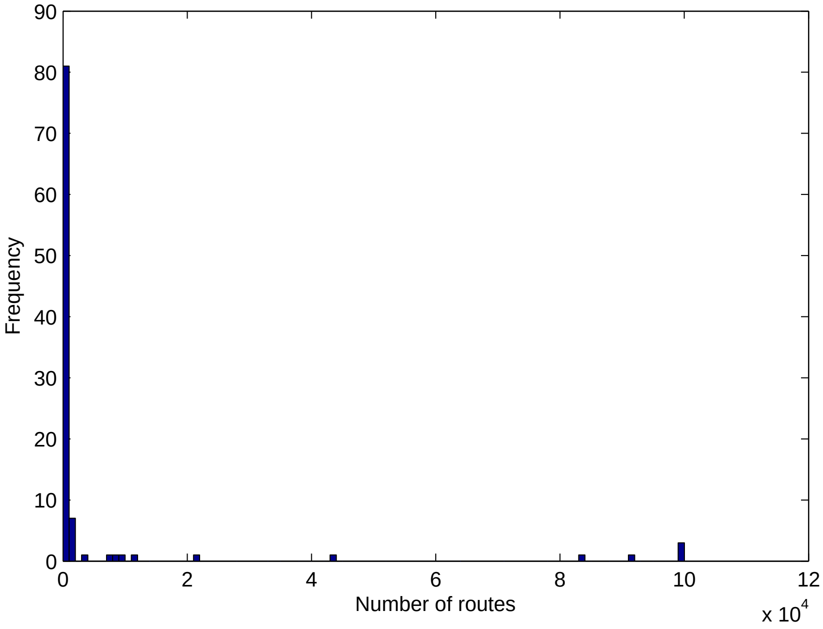

Figure 5.4: The histogram of the number of searched routes

5.5 Low-density parity-check code constructions for the broadcast problem in wireless relay networks

In wireless networks, signals are broadcast and messages could be decoded based on received signals in multiple transmission phases. In the preceding sections, relay selection was studied for the single source broadcast problem when considering properties of wireless networks. In this section, LDPC codes are constructed for the single source broadcast problem.

At the source node, a k_1-bit message is encoded into an n_1-bit codeword by a non-systematic LDPC code. The n_1-bit codeword is sent to all nodes. Assume a node has decoded the message and is chosen to relay the message. The encoding graph at the relay node is shown in Figure 5.5. The m_2 bits in the upper layer of the graph are generated from the n_1-bit codeword by a systematic LDPC code. The relay node sends the m_2 relay bits. The n_1-bit codeword is not sent. This code construction is similar to the constructions for disjoint encoding in two-way relay channels and three-way relay channels where relay bits are generated only from received codewords.

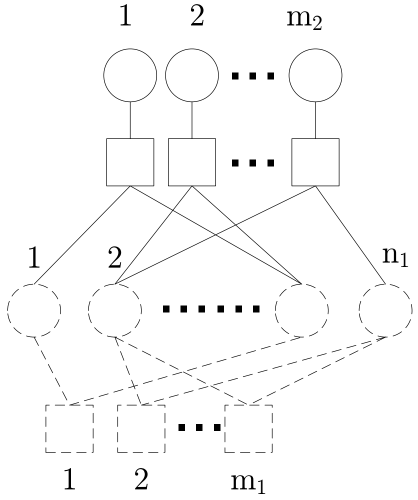

Figure 5.5: Graph for encoding at the relay node

The decoding graph is shown in Figure 5.6. Assume a node receives signals of the n_1-bit codeword from the source node, and signals of the $m_2, m_3, \cdots$ relay bits from multiple relay nodes. All relay bits are concatenated in the upper layer of the graph. The lower layer of the graph is the code at the source node.

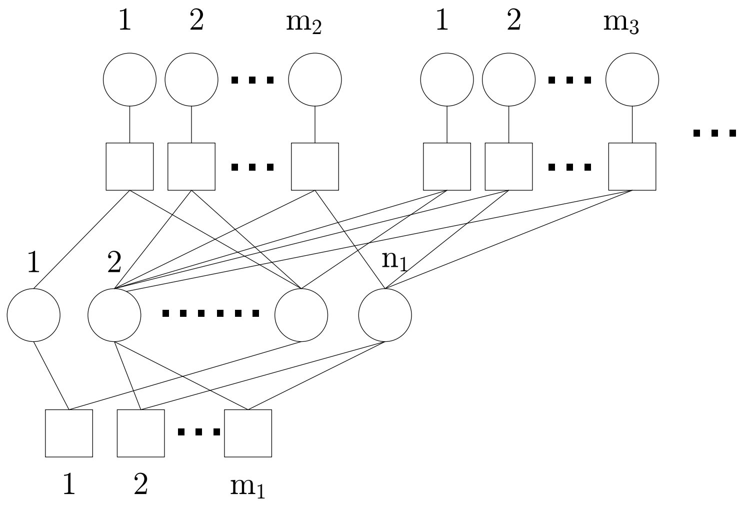

Figure 5.6: Graph for decoding at intermediate nodes and the destination node

5.6 Conclusion

In the chapter, the maximum-rate broadcast problem for a given discrete memoryless wireless relay network was studied where signals are broadcast and messages are decoded based on received signals in multiple transmission phases. The maximum rates for the hop-by-hop relay scheme and the level-by-level relay scheme were shown. Furthermore, optimal relay selection algorithms for the two schemes were proposed. Finally, LDPC codes were constructed for the single source broadcast problem in wireless relay networks.

Chapter 6

The Iterative Hard Interference Cancellation Decoder for Low-density Parity-check Codes in 2-user Multiple-access Channels

6.1 Introduction

The capacity region of multiple-access channels was first found by Ahlswede [40]. The simplest multiple-access channels are 2-user multiple-access channels. In these channels, signals from two source codewords are superimposed. These channels are also the channels for the multiple-access phase in two-phase two-way relay channels where two source nodes send their codewords to the relay node simultaneously. To achieve the capacity, techniques such as time sharing [2], rate splitting [41, 42] or joint decoding [43, 44] can be used.

In [45], a sub-optimal belief propagation decoder was proposed to decode low-density parity-check (LDPC) codes. However, the complexity of this decoder is high. In this chapter, a simplified iterative hard interference cancellation decoder is proposed. The graph of the codes includes variable nodes, check nodes and multiple-access nodes. By this representation, message-passing algorithms can be easily described. The decoder is based on log-likelihood ratios (LLRs). Interference is estimated, quantized and subtracted from channel outputs. To analyze the codes, density evolution is derived. It is shown that the required signal-to-noise ratio (SNR) for the proposed low-complexity decoder is only 0.2 dB higher than that for the sub-optimal belief propagation decoder at code rate $\frac{1}{3}$.

This chapter is organized as follows. In section 6.2, 2-user multiple-access channels are reviewed. Section 6.3 introduces the system model. In section 6.4, the sub-optimal belief propagation decoder based on probability values is reviewed. Section 6.5 proposes an iterative hard interference cancellation decoder for LDPC codes based on LLRs. To analyze the codes, density evolution is derived in section 6.6. In section 6.7, simulation results on the two decoders are given. Finally, section 6.8 concludes this chapter.

6.2 2-user multiple-access channels

2-user multiple-access channels contain with two inputs X_1, X_2 and one output Y. The capacity region of discrete memoryless multiple-access channels $(\mathcal{X}_1 \times \mathcal{X}_2, p(y|x_1, x_2), \mathcal{Y})$ is the closure of the convex hull of all (R_1, R_2) satisfying

$$R_1 < I(X_1; Y|X_2), \tag{6.1}$$

$$R_2 < I(X_2; Y|X_1), \tag{6.2}$$

$$R_1 + R_2 < I(X_1, X_2; Y) \tag{6.3}$$

for some product distribution $p_1(x_1)p_2(x_2)$ on $\mathcal{X}_1 \times \mathcal{X}_2$ [2].

Gaussian multiple-access channels are modeled as $Y = X_1 + X_2 + Z$, where Z is a Gaussian distributed random variable with mean zero and variance σ^2. Assume the power of X_1 and X_2 is P, that is $\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq P$. Thus, the capacity region of the Gaussian multiple-access channels is

$$R_1 < C\left(\frac{P}{N}\right), \tag{6.4}$$

$$R_2 < C\left(\frac{P}{N}\right), \tag{6.5}$$

$$R_1 + R_2 < C\left(\frac{2P}{N}\right) \tag{6.6}$$

where $C(\frac{P}{N}) = \frac{1}{2}\log(1+\frac{P}{N})$ [2].

The achievable rate region of the multiple-access channel for a fixed input distribution is shown in Figure 6.1. In this figure, point A is achievable when X_2 does not send any information. Point B is achievable when successive decoding and interference cancellation are used, i.e., treating X_1 as noise and decoding X_2 first, and then subtracting the decoded X_2 from the received Y and decoding X_1. For non-corner points, it can be achieved by time sharing [2], rate splitting [41, 42] or joint decoding [43, 44].

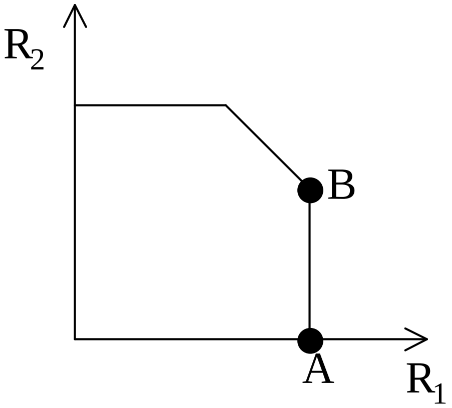

Figure 6.1: An achievable rate region of 2-user multiple-access channels for a fixed input distribution

6.3 System model

The system model for 2-user multiple-access channels is shown in Figure 6.2. For user i, a K-bit source message $\mathbf{s}_i = [s_{i,1}, s_{i,2}, \cdots, s_{i,K}]$ is encoded to an N-bit codeword $\mathbf{c}_i = [c_{i,1}, c_{i,2}, \cdots, c_{i,N}]$ by an LDPC code $\mathcal{C}$ where $s_{i,k} \in \{0,1\}, k = 1, 2, \cdots, K, c_{i,n} \in \{0,1\}, n = 1, 2, \cdots, N$ and $i = 1, 2$. The rate of each user is $\frac{K}{N}$. The codeword $\mathbf{c}_2$ is then interleaved to $\tilde{\mathbf{c}}_2$ according to π , that is, $c_{2,n} = \tilde{c}_{2,\pi_n}$, where π is a permutation of $[1, 2, \cdots, N]$ and π_n is the n-th element of π. $c_{i,n}$ is mapped to $x_{i,n}$ where $x_{i,n} \in \{-1, 1\}$ and the rule of the mapping is $0 \to 1$ and $1 \to -1$. The channel output is $Y = X_1 + X_2 + Z$ where $x_1, x_2 \in \{-1, 1\}$ and Z is a Gaussian distributed random variable with mean zero and variance σ^2.

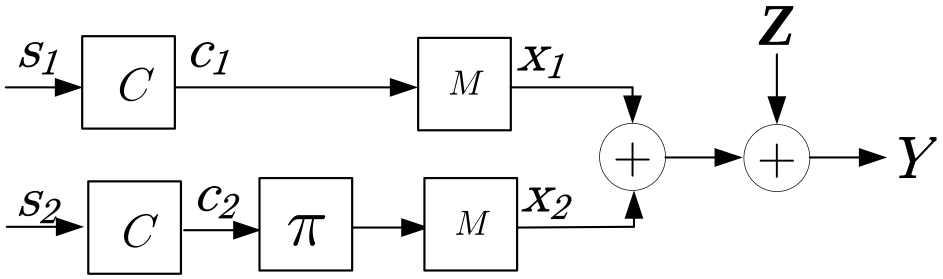

Figure 6.2: System model for 2-user multiple-access channels

6.4 The sub-optimal belief propagation decoder

In this section, the sub-optimal belief propagation decoder for 2-user multiple-access channels [45] is reviewed.

First, a graph is introduced which can easily describe the message-passing algorithms in section 6.4 and section 6.5. The graph is shown in Figure 6.3. In this graph, upper layer variable nodes (VN) and check nodes (CN) represent the code for user 1. Lower layer variable nodes and check nodes represent the code for user 2. In addition, this graph has a new type of nodes called multiple-access nodes (MAN). The multiple-access node is connected to a variable node of code 1 and a variable node of code 2.

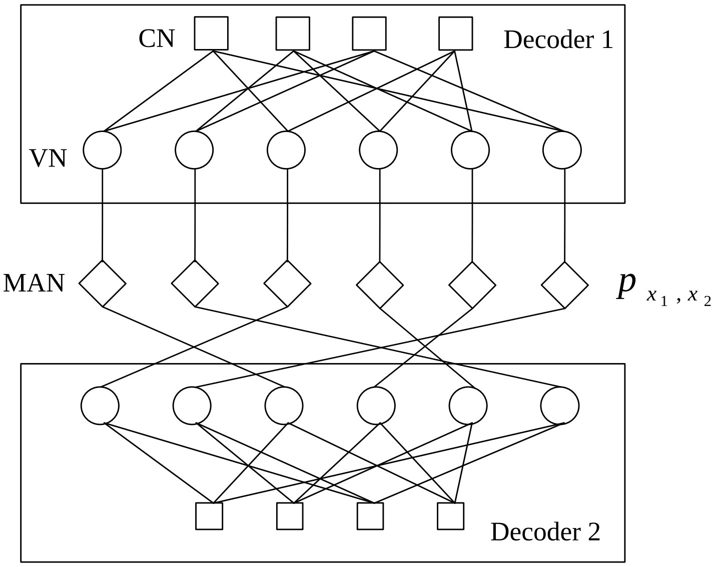

Figure 6.3: Graph of codes in 2-user multiple-access channels

Assume the source is equiprobable, that is, $p(X_1 = 1, X_2 = 1) = p(X_1 = 1, X_2 = -1) = p(X_1 = -1, X_2 = 1) = p(X_1 = -1, X_2 = -1) = \frac{1}{4}$. The intrinsic values of the multiple-access node are four posterior probabilities

$$p_{x_1,x_2} \propto p(y|X_1 = x_1, X_2 = x_2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-x_1-x_2)^2}{2\sigma^2}} \tag{6.7}$$

where $p_{x_1,x_2} = p(X_1 = x_1, X_2 = x_2|y)$, $x_1, x_2 \in \{1, -1\}$ and σ^2 is the noise power.

For a multiple-access node, its input message from a variable node of node 1 is $p_{\tilde{x}_1} = \prod_i p_{x_1,i}$ where $p_{x_1,i}$ is a message from a check node connecting to the variable node. The message flow through a variable node to a multiple-access node is shown in Figure 6.4. Similarly, the input message from a variable node of code 2 is $p_{\tilde{x}_2} = \prod_i p_{x_2,i}$ where $p_{x_2,i}$ is a message from a check node connecting to the variable node.

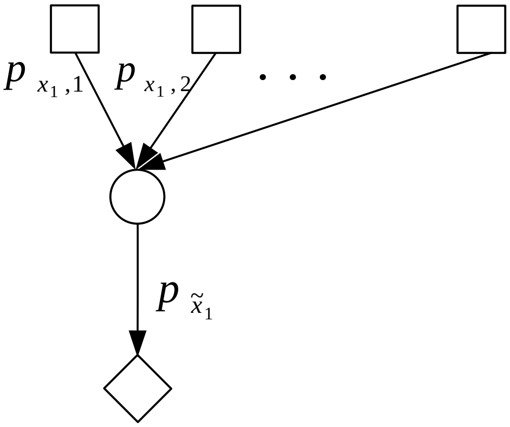

Figure 6.4: Message flow through a variable node to a multiple-access node in the belief propagation decoder

Functions of the multiple-access nodes are

$$p_{\hat{x}_1} \propto \sum_{x_2=\tilde{x}_2\in\{1,-1\}} p_{x_1,x_2} p_{\tilde{x}_2}, \tag{6.8}$$

$$p_{\hat{x}_2} \propto \sum_{x_1=\tilde{x}_1\in\{1,-1\}} p_{x_1,x_2} p_{\tilde{x}_1} \tag{6.9}$$

where $p_{\hat{x}_1}$ is the message passed to a variable node of code 1 and $p_{\hat{x}_2}$ is the message passed to a variable node of code 2. The message flow through a multiple-access node is shown in Figure 6.5.

Note that in the first decoding iteration, multiple-access nodes receive nothing from variable nodes. In this case, $p_{\tilde{x}_1=1}$, $p_{\tilde{x}_1=-1}$, $p_{\tilde{x}_2=1}$ and $p_{\tilde{x}_2=-1}$ are set to $\frac{1}{2}$. Hence

$$p_{\hat{x}_1} \propto \sum_{x_2} p_{x_1,x_2}, \tag{6.10}$$

$$p_{\hat{x}_2} \propto \sum_{x_1} p_{x_1,x_2}. \tag{6.11}$$

For a variable node, one of its edges is connected to a multiple-access node and the

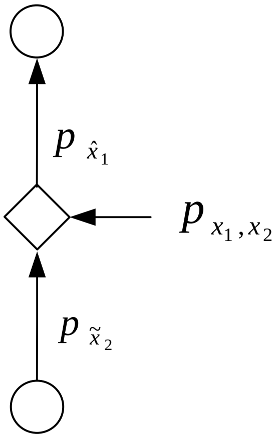

Figure 6.5: Message flow through a multiple-access node in the belief propagation decoder

remaining edges are connected to check nodes. Functions of the variable node are

$$p_{x_{1,j}} \propto \prod_{i \neq j} p_{x_{1,i}} p_{\hat{x}_1}, \tag{6.12}$$

$$p_{x_{2,j}} \propto \prod_{i \neq j} p_{x_{2,i}} p_{\hat{x}_2} \tag{6.13}$$

where i and j are check node indices. The message flow through a variable node to a check node is shown in Figure 6.6.

For a degree-M check node, its output message is based on $M-1$ input messages and can be recursively calculated by

$$\mathrm{chk}(m_1, m_2, ..., m_{M-1}) = \mathrm{chk}(m_1, \mathrm{chk}(m_2, ..., m_{M-1})) \tag{6.14}$$

where $m_i = (p_{i,1}, p_{i,-1})$ is the i-th input message. To calculate $\mathrm{chk}(m_1, m_2)$ where $m_1 = (p_{1,1}, p_{1,-1})$ and $m_2 = (p_{2,1}, p_{2,-1})$, the output (p_1, p_{-1}) is

$$\begin{aligned} (p_1, p_{-1}) &= \mathrm{chk}(m_1, m_2) && (6.15) \\ &= \mathrm{chk}((p_{1,1}, p_{1,-1}), (p_{2,1}, p_{2,-1})) && (6.16) \\ &= (p_{1,1}p_{2,1} + p_{1,-1}p_{2,-1}, p_{1,1}p_{2,-1} + p_{1,-1}p_{2,1}). && (6.17) \end{aligned}$$

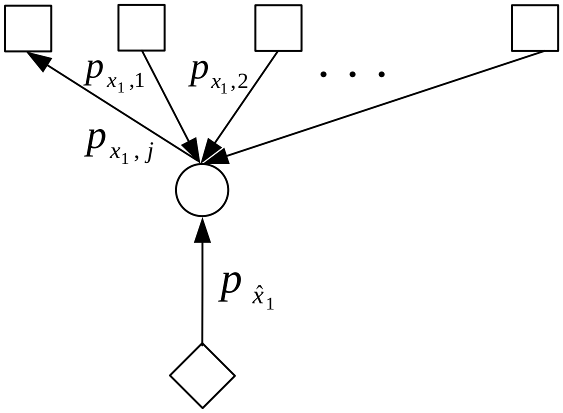

Figure 6.6: Message flow through a variable node to a check node in the belief propagation decoder

The sub-optimal belief propagation decoder adopts the following message passing scheduling. First, multiple-access nodes pass messages to variable nodes based on (6.10) and (6.11). Next, the two codewords are decoded in the upper and lower decoders simultaneously. The decoding outputs $p_{\tilde{x}_1}$ and $p_{\tilde{x}_2}$ are passed back to multiple-access nodes from variable nodes. Finally, new messages calculated from (6.8) and (6.9) are passed to the variable nodes for the next decoding iteration.

In the sub-optimal belief propagation decoder, a message is represented by a probability value pair, rather than an LLR due to the functions (6.8) and (6.9) in multiple-access nodes. Since the functions of variable nodes and check nodes are the same as those in point-to-point channels, LLRs can still be used in these nodes. However, before the LLR is passed to the multiple-access node, it is transformed into the probability value pair (p_1, p_{-1}) by

$$p_1 = \frac{e^m}{1+e^m}, \tag{6.18}$$

$$p_{-1} = \frac{1}{1+e^m} \tag{6.19}$$

where m is the LLR message. In addition, when the probability value pair (p_1, p_{-1}) is passed to the variable node, it is transformed into the LLR by $m = \log \frac{p_1}{p_{-1}}$.

6.5 The iterative hard interference cancellation decoder

In this section, an iterative hard interference cancellation decoder is proposed for LDPC codes in 2-user multiple-access channels.

In 2-user multiple-access channels, the channel output is $Y = X_1 + X_2 + Z$ where X_1 and X_2 are source signals from the two users and Z is the noise. At the destination node, X_2 is considered as interference when decoding X_1 and vice versa. If the interference signal is removed, X_1 and X_2 could be decoded from $X_1 + Z$ and $X_2 + Z$ by two individual point-to-point channel decoders. However, the exact interference is unknown before codewords are decoded. Only the soft estimate of the interference is known. In this section, we propose an iterative hard interference cancellation decoder, inspired by the observation that the estimates become more accurate during the iterative decoding. In each decoding iteration, interference is estimated, quantized and subtracted from the received signals.

In this decoder, functions of variable nodes and check nodes are based on LLRs. When LLRs $c_{1,i}$ from check nodes of code 1 are received, the variable node calculates the soft estimate $\tilde{x}_1 = \sum_i c_{1,i}$. The message flow through a variable node to a multiple-access node is shown in Figure 6.7. Similarly, $\tilde{x}_2 = \sum_i c_{2,i}$ where $c_{2,i}$ is a message from a check node of code 2. $\tilde{x}_1$ and $\tilde{x}_2$ are passed to the multiple-access node. In the multiple-access node, the hard estimates $\hat{x}_1$ and $\hat{x}_2$ are simply the sign of $\tilde{x}_1$ and $\tilde{x}_2$ where $\operatorname{sgn}(x) = 1$ when $x \geq 0$ and $\operatorname{sgn}(x) = -1$ when $x < 0$. Thus, the estimates of channel outputs without interference are $y_1 = y - \hat{x}_2$ and $y_2 = y - \hat{x}_1$, whose corresponding LLRs $m_1 = \frac{2y_1}{\sigma^2}$ and $m_2 = \frac{2y_2}{\sigma^2}$ are passed to variable nodes. Finally, the variable node of code 1 calculates

$$v_{1,j} = \sum_{i \neq j} c_{1,i} + m_1 \tag{6.20}$$

and the variable node of code 2 calculates

$$v_{2,j} = \sum_{i \neq j} c_{2,i} + m_2 \tag{6.21}$$

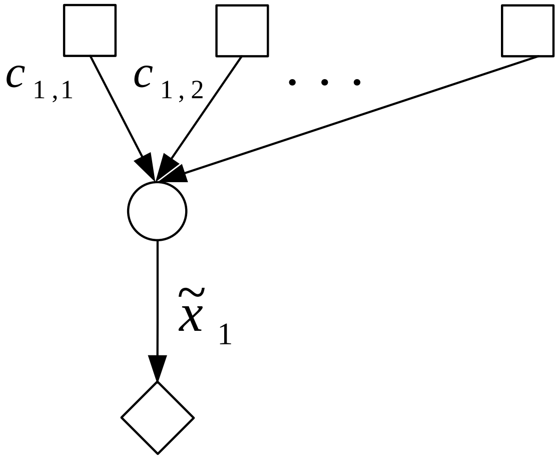

Figure 6.7: Message flow through a variable node to a multiple-access node in the iterative hard interference cancellation decoder

where $v_{1,j}$ is an output message of a variable node of code 1 and $v_{2,j}$ is an output message of a variable node of code 2. The messages are sent to check nodes for the next decoding iteration. The message flow through a variable node to a check node is shown in Figure 6.8.

6.6 Density evolution

To analyze codes for multiple-access channels, it seems at first that each codeword has to be analyzed since the probability of error is dependent on the codeword. For example, if $\mathbf{x}_1 = \mathbf{x}_2 = [1, 1, 1, \cdots]$, the probability of error would be 0 when no noise exists. However, if $\mathbf{x}_1 = [1, -1, 1, -1, \cdots]$ and $\mathbf{x}_2 = [-1, 1, -1, 1, \cdots]$, they cannot be decoded at all.

It is shown in [46] that an $(\mathbf{x}_1, \mathbf{x}_2)$ pair, whose probability of error is the same as the average probability of error on all $(\mathbf{x}_1, \mathbf{x}_2)$ pairs, can be found. Define type τ of an $\mathbf{x}_1$ or $\mathbf{x}_2$ as the fraction of ones in the $\mathbf{x}_1$ or $\mathbf{x}_2$. Define type φ of an $X_1 + X_2$ sequence as $(\varphi_1, \varphi_2, \varphi_3)$ where $\varphi_1, \varphi_2, \varphi_3$ are the fractions of -2, 0 and 2 in the sequence. By law of large numbers, the dominant type of the $X_1 + X_2$ sequence is $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$. The probabilities of other

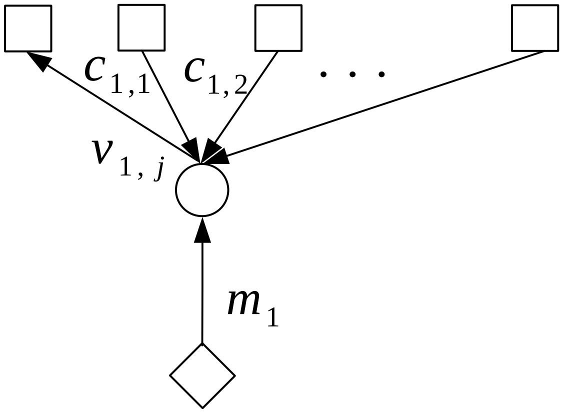

Figure 6.8: Message flow through a variable node to a check node in the iterative hard interference cancellation decoder

types are negligible. The probability of error for each $X_1 + X_2$ sequence with the same type φ would be the same if the length of the sequence is infinity. In addition, by channel symmetry, if an (x_1, x_2) pair (-1,-1) is inverted to (1,1), the probability of error does not change. By these properties, it is assumed that only $(\mathbf{x}_1, \mathbf{x}_2)$ pairs of $\mathbf{x}_1 = [1, 1, 1, \cdots]$ and $\mathbf{x}_2 = [1, -1, 1, -1, \cdots]$ are sent in density evolution. The type of the $X_1 + X_2$ sequence is $(0,\frac{1}{2},\frac{1}{2})$. The probability density function of Y is

$$P_Y = \frac{1}{2\sigma\sqrt{2\pi}} \left(e^{-\frac{(y-2)^2}{2\sigma^2}} + e^{-\frac{y^2}{2\sigma^2}} \right). \tag{6.22}$$

Thus, any existing density evolution methods [11, 12, 8] can be used to analyze the codes and optimize degree distributions. For a 2-user Gaussian multiple-access channel with the rate pair (R_1,R_2) = (0.5, 0.5), an optimized code ensemble which is only 0.18 dB away from the capacity is reported in [47].

In the following, density evolution equations for multiple-access nodes in the hard interference cancellation decoder are derived. Assume the probability density function of $\tilde{X}_1$ and $\tilde{X}_2$ is $P_{\tilde{X}_1}$ and $P_{\tilde{X}_2}$. Since $\hat{X}_1 = \mathrm{sgn}(\tilde{X}_1)$ and $\hat{X}_2 = \mathrm{sgn}(\tilde{X}_2)$,

$$P_{-\hat{X}_1} = \delta_{-1} \int_0^{\infty} P_{\tilde{X}_1} + \delta_1 \int_{-\infty}^{0} P_{\tilde{X}_1}, \tag{6.23}$$

$$P_{-\hat{X}_2} = \delta_{-1} \int_0^{\infty} P_{\tilde{X}_2} + \delta_1 \int_{-\infty}^{0} P_{\tilde{X}_2} \tag{6.24}$$

where δ is the Dirac delta function. Assume Y is independent with $\hat{X}_1$ and $\hat{X}_2$. By $Y_1 = Y - \hat{X}_2$ and $Y_2 = Y - \hat{X}_1$, the probability density functions of the output of the multiple-access nodes are

$$P_{Y_1} = P_Y \otimes P_{-\hat{X}_2}, \tag{6.25}$$

$$P_{Y_2} = P_Y \otimes P_{-\hat{X}_1} \tag{6.26}$$

where $\otimes$ is convolution. The probability density functions of LLRs $M_1 = \frac{2Y_1}{\sigma^2}$ and $M_2 = \frac{2Y_2}{\sigma^2}$ are

$$P_{M_1} = \frac{\sigma^2}{2} P_{Y_1} \left(\frac{\sigma^2}{2} m_1 \right), \tag{6.27}$$

$$P_{M_2} = \frac{\sigma^2}{2} P_{Y_2} \left(\frac{\sigma^2}{2} m_2 \right). \tag{6.28}$$

6.7 Simulation results

In this section, simulation results of an (1920.1280.3.303) [48] LDPC code for Gaussian multiple-access channels are reported in Figure 6.9. The length of the codeword is 1920. By the third inequality of the capacity region of multiple-access channels $R_1+R_2 < C(\frac{2P}{N})$, the minimum required SNR $\frac{P}{N}$ is -1.19 dB for error-free communications when $R_1 = R_2 = \frac{1}{3}$. As we know, there is a 2 to 3 dB gap between soft decoders and hard decoders in point-to-point channels. However, if the soft interference cancellation is replaced with a hard one in 2-user multiple-access channel decoder, and soft decoding is kept in variable nodes and check nodes, the required SNR for the iterative hard interference cancellation decoder is only 0.2 dB higher than that for the sub-optimal belief propagation decoder at the bit error rate (BER) of $10^{-3.7}$.

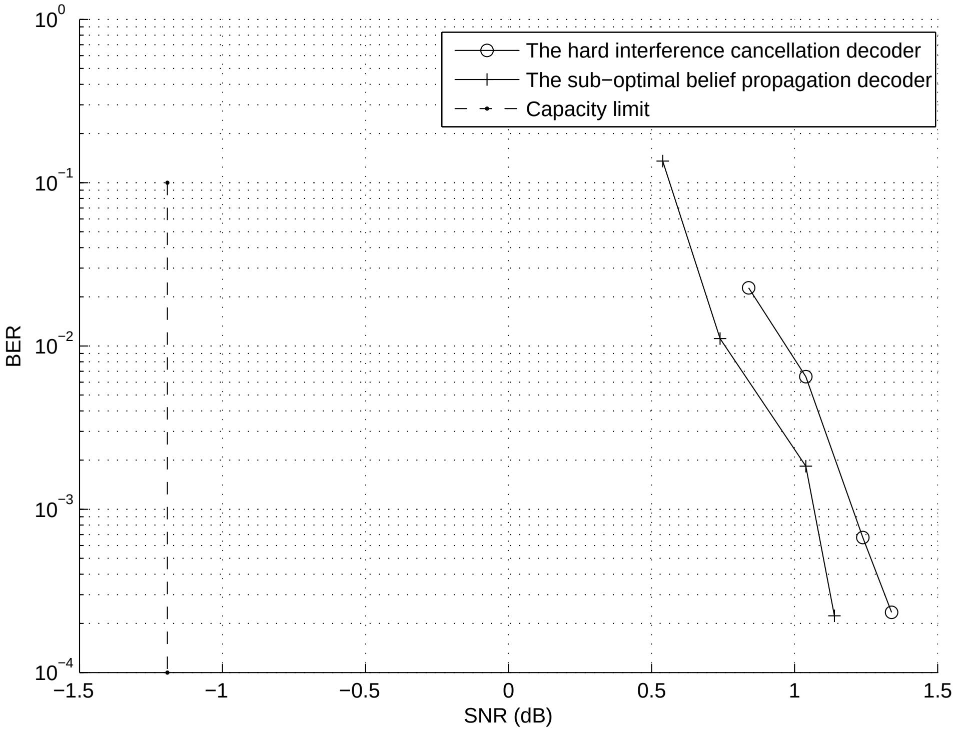

Figure 6.9: Simulation results of an (1920.1280.3.303) low-density parity-check code in Gaussian multiple-access channels

6.8 Conclusion

In the chapter, an iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels is proposed. The decoder is based on LLRs, and interference is estimated, quantized and subtracted from channel outputs. To analyze codes for multiple-access channels, density evolution is derived. It was shown that the required SNR for the proposed low-complexity decoder is only 0.2 dB higher than that for an existing sub-optimal belief propagation decoder at code rate $\frac{1}{3}$.

Chapter 7

Conclusions and Future Work

7.1 Major research contributions

In wireless networks, signals are broadcast and codewords can be jointly decoded based on multiple received signals. This thesis mainly studied low-density parity-check (LDPC) code constructions in such networks. The main contributions of the thesis are as follows.

- In Chapter 3, LDPC codes for half-duplex three-phase two-way relay channels were proposed. At the relay node, a systematic LDPC code was constructed to encode two source codewords. At the destination node, signals from the source node and the relay node were used for joint decoding. It was demonstrated that good codes can be found by discretized density evolution and iterative linear programing.

- Chapter 4 extended the study of LDPC codes to half-duplex three-way relay channels. An achievable rate region of half-duplex three-way relay channels was first proved. In addition, LDPC codes were constructed for each sub-region of the achievable rate region for half-duplex three-way relay channels.

- Chapter 5 studied the maximum-rate broadcast problem for a given discrete memoryless wireless relay network. The maximum rates for the hop-by-hop relay scheme and the level-by-level relay scheme were given. Furthermore, optimal relay selection algorithms were proposed for the two relay schemes. Finally, LDPC codes were constructed for the broadcast problem in wireless relay networks.

- Chapter 6 proposed an iterative hard interference cancellation decoder for LDPC codes in 2-user multiple-access channels. The iterative hard interference cancellation decoder is based on log-likelihood ratios (LLRs). Interference is estimated, quantized and subtracted from channel outputs. The required signal-to-noise ratio (SNR) for the proposed low-complexity decoder has been shown to be only 0.2 dB higher than that for an existing sub-optimal belief propagation decoder at code rate $\frac{1}{3}$.

7.2 Future work

7.2.1 Codes in fading channels and wireless networks

This thesis has considered only the case when Gaussian noises exist. Future work could extend this study to more realistic wireless channel models, such as fading channels. In addition, LDPC code constructions could be extended from two-way relay channels to wireless relay networks. In such cases, a terminal node can help to relay multiple messages and a codeword can be helped by multiple relay nodes. The performance of LDPC codes in such networks requires further investigation.

7.2.2 Relay selection for multicast and unicast in wireless relay networks

This thesis has examined the relay selection problem for broadcasting messages in wireless relay networks. The relay selection problem can also be extended to the multicast and unicast cases, even for multiple-source cases. Since the relay selection problem is in general not solvable in polynomial time, efficient heuristic algorithms should be developed.

References

[1] S. Zhang, S.-C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proc. 12th Annu. Int. Conf. on Mobile Computing and Networking*, Los Angeles, CA, Sep. 2006, pp. 358–365.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* John Wiley & Sons, Inc., 2006.

[3] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, Jul. and Oct. 1948.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Communications*, Geneva, May 1993, pp. 1064–1070.

[5] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[6] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEE Electron. Lett.*, vol. 32, pp. 1645–1655, Aug. 1996.

[7] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, pp. 260–269, Apr. 1967.

[8] S.-Y. Chung, G. Forney, T. Richardson, and R. Urbanke, "On the low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[9] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, pp. 533–547, Sep. 1981.

[10] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[11] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[12] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[13] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[14] M. Luby, M. Mitzenmacher, A. Shokrollah, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proc. 30th Annu. ACM Symp. Theory of Computing*, Dallas, TX, May 1998, pp. 249–258.

[15] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Annu. ACM Symp. Theory of Computing*, El Paso, TX, May 1997, pp. 150–159.

[16] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3723–3739, Oct. 2007.

[17] J. Laneman, D. Tse, and G. Wornell, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior,” *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[18] T. Cover and A. Gamal, “Capacity theorems for the relay channel,” *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 572–584, Sep. 1979.

[19] B. Rankov and A. Wittneben, “Achievable rate regions for the two-way relay channel,” in *Proc. IEEE Int. Symp. Information Theory*, Seattle, WA, Jul. 2006, pp. 1668–1672.

[20] L.-L. Xie, “Network coding and random binning for multi-user channels,” in *Proc. 10th Canadian Workshop Inf. Theory*, Edmonton, Alta., Jun. 2007, pp. 85–88.

[21] T. Oechtering, C. Schnurr, I. Bjelakovic, and H. Boche, “Broadcast capacity region of two-phase bidirectional relaying,” *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 454–458, Jan. 2008.

[22] A. Chakrabarti, A. D. Baynast, A. Sabharwal, and B. Aazhang, “Low density parity check codes for the relay channel,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 2, pp. 280–291, Feb. 2007.

[23] J. Hu and T. Duman, “Low density parity check codes over wireless relay channels,” *IEEE Trans. Commun.*, vol. 6, no. 9, pp. 3384–3394, Sep. 2007.

[24] S. Zhang and S.-C. Liew, “Channel coding and decoding in a relay system operated with physical-layer network coding,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 788–796, Jun. 2009.

[25] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[26] D. Wübben and Y. Lang, “Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems,” in *Proc. IEEE Globecom*, Miami, FL, Dec. 2010, pp. 1–5.

[27] J. Liu, M. Tao, and Y. Xu, "Pairwise check decoding for LDPC coded two-way relay block fading channels," *IEEE Trans. Commun.*, vol. 60, no. 8, pp. 2065–2076, Aug. 2012.

[28] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," in *Proc. IEEE Int. Conf. Communications*, vol. 4, Istanbul, Jun. 2006, pp. 1568–1573.

[29] T. Cui, F. Gao, T. Ho, and A. Nallanathan, "Distributed space time coding for two-way wireless relay networks," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 658–671, Feb. 2009.

[30] X. Zhou, L.-L. Xie, and X. Shen, "Low-density parity-check codes for two-way relay channels," in *Proc. IEEE 72nd Vehicular Technology Conf. Fall*, Ottawa, ON, Sep. 2010, pp. 1–5.

[31] ——, "Design of low-density parity-check codes for half-duplex three-phase two-way relay channels," *IEEE Trans. Wireless Commun.*, submitted for publication, in revision.

[32] J. Garcia-Frias and W. Zhong, "Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 266–268, Jun. 2003.

[33] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. on Foundations of Computer Science*, Vancouver, BC, Nov. 2002, pp. 271–280.

[34] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, vol. 28, no. 4, Vancouver, BC, Sep. 1998, pp. 56–67.

[35] J. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, vol. 2, Tel Aviv, Mar. 2000, pp. 585–594.

[36] M. Čagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proc. 8th Annu. Int. Conf. on Mobile Computing and Networking*, Atlanta, GA, Sep. 2002, pp. 172–182.

[37] I. Maric and R. Yates, "Cooperative multihop broadcast for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1080–1088, Aug. 2004.

[38] A. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in static wireless networks," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2185–2192, Nov. 2007.

[39] S. Draper, L. Liu, A. Molisch, and J. S. Yedidia, "Cooperative transmission for wireless networks using mutual-information accumulation," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5151–5162, Jul. 2011.

[40] R. Ahlswede, "Multi-way communication channels," in *Proc. IEEE Int. Symp. Information Theory*, Ashkelon, Jun. 1973, pp. 23–52.

[41] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 364–375, Mar. 1996.

[42] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting, "Rate-splitting multiple access for discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 873–890, Mar. 2001.

[43] N. Ibrahim and G. K. Kaleh, "Iterative decoding and soft interference cancellation for the Gaussian multiple access channel," *Wireless Personal Communications*, vol. 13, no. 1-2, pp. 15–25, May 2000.

[44] N. Chayat and S. Shamai, "Convergence properties of iterative soft onion peeling," in *Proc. of the IEEE Information Theory and Communications Workshop*, Kruger National Park, Jun. 1999.

[45] R. Palanki, A. Khandekar, and R. McEliece, "Graph-based codes for synchronous multiple access channels," in *Proc. 39th Annu. Allerton Conf. on Communication, Control, and Computing*, Monticello,IL, Oct. 2001.

[46] T. Richardson and R. L. Urbanke, *Modern coding theory.* Cambridge University Press, 2008.

[47] A. Amraoui, S. Dusad, and R. Urbanke, "Achieving general points in the 2-user Gaussian MAC without time-sharing or rate-splitting by means of iterative coding," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Jun. 2002, p. 334.

[48] D. J. MacKay. Encyclopedia of sparse graph codes. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html