

**LOW-COMPLEXITY ENERGY OPTIMIZATION OF WIRELESS
SENSOR NETWORKS**

NARIMAN FARSAAD

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO
NOVEMBER 2009

**LOW-COMPLEXITY ENERGY
OPTIMIZATION OF WIRELESS SENSOR
NETWORKS**

by **Nariman Farsad**

a thesis submitted to the Faculty of Graduate Studies of
York University in partial fulfilment of the requirements
for the degree of

MASTER OF SCIENCE

© 2009

Permission has been granted to: a) YORK UNIVERSITY LIBRARIES to lend or sell copies of this dissertation in paper, microform or electronic formats, and b) LIBRARY AND ARCHIVES CANADA to reproduce, lend, distribute, or sell copies of this thesis anywhere in the world in microform, paper or electronic formats *and* to authorise or procure the reproduction, loan, distribution or sale of copies of this thesis anywhere in the world in microform, paper or electronic formats.

The author reserves other publication rights, and neither the thesis nor extensive extracts for it may be printed or otherwise reproduced without the author's written permission.

LOW-COMPLEXITY ENERGY OPTIMIZATION OF WIRELESS SENSOR NETWORKS

by **Nariman Farsad**

By virtue of submitting this document electronically, the author certifies that this is a true electronic equivalent of the copy of the thesis approved by York University for the award of the degree. No alteration of the content has occurred and if there are any minor variations in formatting, they are as a result of the conversion to Adobe Acrobat format (or similar software application).

Examination Committee Members:

1. Andrew Eckford
2. Mokhtar Aboelaze
3. Amir Asif
4. Spiros Pagiatakis

Abstract

Wireless sensor networks (WSNs), which consist of numerous devices that take measurements of a physical phenomenon, are becoming a popular area of research. Since the sensor nodes are typically battery powered, energy optimization and efficiency is extremely important in WSNs. However, optimizing power proves to be a non-trivial task since decreasing the transmission power will result in degraded signal and unsuccessful transmission. In this thesis, we look at the physical layer and propose two schemes based on channel codes that could be employed for optimization of transmission power of WSNs. First we consider the fact that the phenomena being observed by the sensor nodes are commonly correlated in space. Therefore, we devise a low-complexity coding scheme for correlated sources based on Slepian-Wolf compression, and analyze its performance in terms of diversity order. The main idea of this scheme is to use the correlated measurements as a substitute for relay links. Although we show that the asymptotic diversity order is limited by the constant correlation factor, we give experimental results that

show excellent performance over practical ranges of SNR. In the second part of the thesis, we consider the fact that there may be many potential relays within radio range of a source; similarly, there may be many potential sources seeking to use relays. Allocating these resources is a non-trivial optimization problem. We consider fractional cooperation, where each potential relay only allocates a fraction of its resources to relaying. It is shown that linear programming can be used to optimally allocate resources in multi-source, multi-relay networks, where the relays use the demodulate-and-forward (DemF) or the decode-and-forward (DF) strategy, and where the transmissions are protected by low-density parity-check (LDPC) codes. Compared with existing optimization schemes, this method is particularly suitable for very large networks with numerous sources and relays. Simulation results are presented to demonstrate the performance of this scheme.

To My parents. Thank you for everything.

Acknowledgements

This thesis could not have been written without the encouragement, collaboration, and support of a tremendous number of people.

To my supervisor, Professor Andrew Eckford, thank you for your patience in thesis supervision, encouragement, advice and guidance. Step by step, you have shown me how to discover the problems and solve them, which has resulted in this valuable research.

To my committee members, Professors Mokhtar Aboelaze, Amir Asif, and Spiros Pagiatakis, thank you for taking the time to read this thesis and giving me valuable comments.

I am grateful to the administrative assistants and system administrators of the department for assisting me in many different ways. I would also like to thank the technical staff specially Ulya Yigit.

Finally, I like to thank my fellow graduate students, Damon Sotoudeh and Foroohar Foroozan, for their help and support.

Table of Contents

Abstract	iv
Acknowledgements	vii
Table of Contents	viii
List of Tables	xii
List of Figures	xiii
1 Introduction	1
1.1 Motivations	4
1.2 Contributions	5
1.2.1 Using Correlated Sources	6
1.2.2 Analytic Energy Optimization	8
1.3 Thesis Outline	10

2	Literature Survey	11
2.1	Cooperative Diversity	11
2.1.1	Cooperation Overview	12
2.1.2	Other Related Works	22
2.2	Slepian-Wolf Compression	29
2.2.1	Slepian-Wolf Compression Overview	31
2.2.2	Other Related Works	37
2.3	Thesis Overview	41
3	System Model	44
3.1	Relay Model	45
3.1.1	Single correlated source and relay	45
3.1.2	Multi-relay and multi-correlated-source	48
3.1.3	Multi-Source, Multi-Relay Fractional Cooperation	51
3.2	Coding Methods	55
3.2.1	Repeat-Accumulate Codes	56
3.2.2	Low-Density Parity-Check Codes	65
3.3	Summary	81
4	Low-Complexity Cooperation Using Correlated Sources	82
4.1	System Setup	84

4.1.1	An Example	85
4.1.2	Problem Setup	90
4.2	Theoretical Result	91
4.2.1	Key Assumptions and Definitions	92
4.2.2	Main Theoretical Results	94
4.3	Experimental Results	96
4.3.1	System Model Implementation	96
4.3.2	Simulation Results	99
4.4	Summary	111
5	Analytic Optimization	113
5.1	System Setup	114
5.1.1	An Example	115
5.2	Linear Programming Model	122
5.2.1	Key Assumptions and Definitions	123
5.2.2	Successful Transmission Requirements	126
5.2.3	Energy Minimization	130
5.3	Experimental Results	134
5.3.1	Simple Example	134
5.3.2	Realistic Example	136

5.4 Summary	141
6 Conclusion and Future Work	145
Bibliography	148

List of Tables

5.1	Relay selections and demodulations	117
5.2	Relay selections and retransmissions	120
5.3	Optimized and non-optimized ϵ	143

List of Figures

1.1	A typical sensor node.	2
1.2	A typical Sensor Network.	3
1.3	Wireless Sensor Network used for volcano monitoring.	3
2.1	Basic Relay Channel Cooperation	14
2.2	Cooperation where each user is both a source and a relay.	15
2.3	Decode-and-Forward (DF)	16
2.4	Amplify-and-Forward (AF)	17
2.5	Coded Cooperation.	19
2.6	Relay Channel Coded Cooperation.	20
2.7	Demodulate-and-Forward	21
2.8	Two correlated sources that communicate with each other.	30
2.9	Two correlated sources that do not communicate with each other.	33
2.10	Achievable rate region defined by the Slepian-Wolf bounds.	34

3.1	Single relay, single correlated source model.	46
3.2	Multi-relay, multi-correlated-source model.	49
3.3	Multi-Source, Multi-relay fractional cooperation model.	52
3.4	The factor graph representation of RA codes.	59
3.5	The factor graph representation of PSRA codes.	64
3.6	Parity-check matrix of an LDPC code.	67
3.7	The factor graph representation of an LDPC codes.	68
3.8	The factor graph representation of an LDPC codes.	72
3.9	The corner point of Slepian-Wolf rate region (x compressed fully). .	74
3.10	The factor graph used in LDPC compression and decompression al- gorithms.	75
3.11	(3,6) regular LDPC code tree.	78
3.12	EXIT chart of (3,6) regular LDPC code.	80
4.1	Example of compression by a (2,4) regular LDPC code.	86
4.2	Example of PSRA code.	87
4.3	Error rate of LDPC compression.	97
4.4	FER and BER of systems with no relays and one correlated source in Rayleigh fading.	100
4.5	FER and BER of systems with no relays and two correlated sources in Rayleigh fading.	101

4.6	FER and BER of systems with no relays and three correlated sources in Rayleigh fading.	103
4.7	FER and BER of systems with one relay and one correlated source in Rayleigh fading.	105
4.8	FER and BER of systems with one relay and two correlated sources in Rayleigh fading.	107
4.9	FER and BER of systems with one relay and three correlated sources in Rayleigh fading.	108
4.10	Using relays for diversity vs using correlated sources.	109
4.11	Frame error rate of a system with a relay and three correlated sources in AWGN channel.	110
5.1	DF frame error rate and bit error rate.	135
5.2	DemF frame error rate and bit error rate.	136
5.3	Source 1 optimal and non-optimal error rates.	137
5.4	Source 2 optimal and non-optimal error rates.	137
5.5	Source 3 optimal and non-optimal error rates.	138
5.6	Source 4 optimal and non-optimal error rates.	138
5.7	Source 5 optimal and non-optimal error rates.	139
5.8	Overall system's average optimal and non-optimal error rates. . . .	139

1 Introduction

Wireless Sensor Networks [1](WSNs), are a special class of wireless networks where distributed sensors, embedded in nodes, take local measurements of a phenomenon, and form a wireless network to share their information amongst themselves, or transmit it to some central authority, known as the data sink. For many applications, wireless sensor networks (WSNs) are required to be unobtrusive, with numerous nodes that are dependent on a battery power source. These nodes are typically very simple, small, and inexpensive modules that are equipped with a sensor to measure a phenomenon. A simple transceiver is used to transmit and receive the measured observations to and from neighboring nodes. Figure 1.1 depicts a typical sensor node with these simple components. Ultimately, the sensor nodes cooperate in transmitting their observations to a data sink where they can be processed as shown in Figure 1.2. Since nodes must be as small, inexpensive, and as efficient as possible, there are stringent constraints on their computational and energy resources. On the other hand, the data sink is assumed to have access

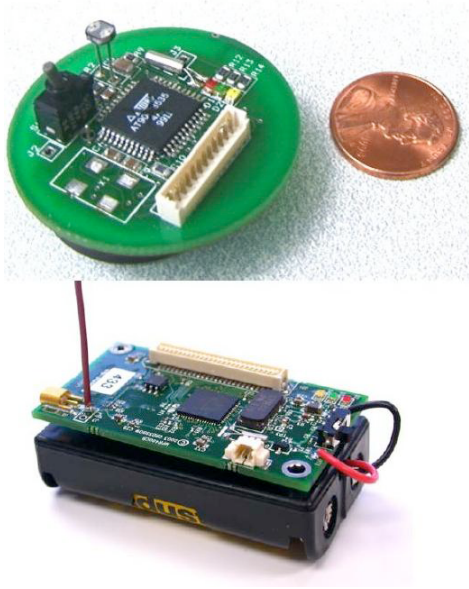


Figure 1.1: A typical sensor node.

to substantial energy and computational resources, within the limits of reasonable expense and contemporary technology. Therefore one of the main challenges of designing a successful WSN is in minimizing the probability of error in transmitting data, subject to constraints in available power and computational resources.

WSNs have a wide variety of potential applications, from habitat monitoring [2] to load monitoring in structures [3] to industrial monitoring and control, security and military sensing, and health monitoring. In [2] for example, Zhang et al. present their experiences developing hard-ware and low-level software for ZebraNet, a wireless network of sensor nodes used for monitoring Zebra movements in the wild. They develop a fully-functional, highly-mobile, energy-efficient sensing

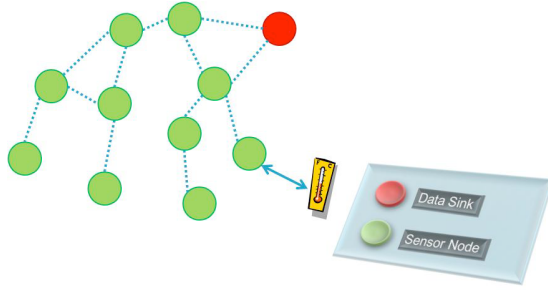


Figure 1.2: A typical Sensor Network.

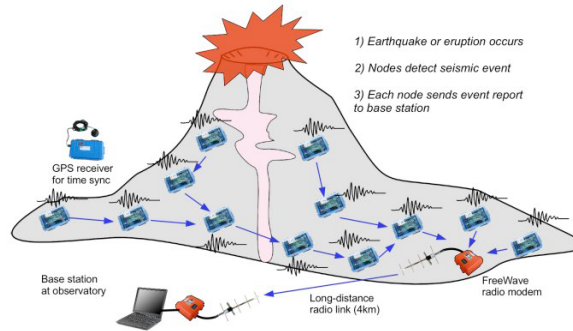


Figure 1.3: Wireless Sensor Network used for volcano monitoring.

system that determines accurate positional data and can propagate it through the network. As another example, in [3], the authors describe the design of a wireless structural data acquisition system called Wisden. The system mimics wired data acquisition systems, and incorporates novel reliable transport, time synchronization, and compression algorithms. As a final example, in [4], Werner-Allen et al. study the feasibility of using wireless sensors for volcanic studies. They use an array

of microphones along with seismometers connected wirelessly to monitor volcanic activity. Figure 1.3 shows the setup proposed by the authors¹.

1.1 Motivations

Driven by ever increasing applications WSNs are becoming a popular area of research. There are numerous research challenges associated with WSNs. Some of these challenges include real-world protocols, real time data collection and transmission, power management, and security and privacy [5]. The most active area of research is however power management.

The low-cost deployment of WSNs is one of its acclaimed advantages. To keep the cost down typically sensor nodes are small and have limited computation power and energy. The limited processor power and small memory are two constraints in sensor networks, which will disappear with the development of new fabrication techniques. However, the energy constraint is unlikely to be solved soon due to slow progress in development of new battery technologies. Moreover, for many applications, the untended nature of sensor nodes and hazardous sensing environments preclude battery replacement as a feasible solution. On the other hand, the surveillance nature of many sensor network applications requires a long lifetime. Therefore, it is a very important research issue to provide a form of energy-efficient

¹Picture taken from the project website at: <http://fiji.eecs.harvard.edu/Volcano>

surveillance service.

There are different techniques in literature for minimizing the energy consumption of WSNs that are not based on modification of hardware. They can be divided into different groups. In one major group efficient protocols such as energy efficient MAC layer protocols [6–10] are used to reduce the energy. Other methods based on efficient data sensing and gathering are also proposed in [11–13], to name a few. Finally, methods based on energy efficient error correcting codes are proposed by some as a method of energy reduction. In this thesis, we will focus on the later and propose methods of our own to reduce energy consumption of WSNs.

1.2 Contributions

Our contributions can be divided in to two main groups. First, we propose a low complexity cooperative method that takes advantage of the correlated nature of the data measured to minimize transmission power of WSNs. We will then suggest an analytic method based on linear programing to minimize the transmission power while maintaining successful transmission links. To drive our low complexity cooperative method based on correlated sources, we use two typical properties of WSNs, namely spatial correlation and spatial diversity. Our analytic technique for minimizing the transmission power while maintaining a successful link is based on Extrinsic Information Transfer Chart (EXIT Chart) of Low-Density Parity-Check

(LDPC) codes.

1.2.1 Using Correlated Sources

One typical property of WSNs is their spatial distribution. This means that the nodes are typically distributed over a large area compared to transmission range of each node. Spatially distributed sensor networks typically have two benefits: first, they have spatial diversity which means the fading on different links are independent; and second, measured data are spatially correlated. The latter can be due to nature. For example in Figure 1.3 we can see a sensor network measuring seismic activities around a volcano. It is highly probable that two neighboring nodes measure a highly correlated seismic curves.

Spatial diversity can be generalized to cooperative diversity [14], where each node can assist its neighboring nodes in transmitting their information to a common receiver or a data sink. This can result in an increase in system throughput which can lead to a more efficient network. On the other hand, spatial correlation can be exploited by the Slepian-Wolf theorem [15] where the output of two correlated sources can be compressed without any communication between them. This compression permits a reduction in the number of transmitted bits, and hence a power efficient system. Meanwhile, from the node's perspective, Slepian-Wolf compression can be accomplished with relatively low complexity [16].

Based on these two properties of WSNs we propose a novel coding technique with the following advantageous features:

1. *Higher diversity order.* The source-destination link diversity order is dependent on the number of relays assisting the source in transmitting its information bits to the destination. However, if we introduce correlated sources where their information bits are correlated with the source's information bits, we can show that the diversity order of the system can still be increased by increasing the number of correlated sources. Therefore in a WSN if a source has a limited number of relays we can still increase the diversity order if we utilize the nodes measuring correlated data.
2. *Decreased power in transmission and computation.* The relay nodes need to either decode the source's information bits (decode-and-forward) or make hard-decisions (demodulate-and-forward) before forwarding to the destination. This requires computational power. By using correlated sources the decode or hard decision step at the relay is eliminated resulting in reducing the overall energy consumption of the system. Also the use of correlated sources will permit Slepian-Wolf compression of sources's information bits before transmission. This will in turn reduce the number of bits transmitted by the source and relayed by the relays thereby further reducing the overall

energy on the system.

3. *Better Transmission Error Rate.* Using correlated sources it is possible to achieve better error rates, specially at low signal-to-noise ratio (SNR). One reason behind this is that, at low SNR, the source-relay link can get worst than the correlation factor between the source and the correlated source. There using the correlated source will be advantageous.
4. *Applicable to Fractional Cooperation.* Fractional cooperation was proposed in [17] as energy saving technique for WSNs. In fractional cooperation relay would select a small fraction of source's transmission bits at random and would relay them to the destination. Similar idea can be applied to correlated sources, where each correlated source selects a small fraction of its correlated bits and transmits them to the destination.

1.2.2 Analytic Energy Optimization

In [17], Eckford et al. proposed fractional cooperation where each relay would select a small fraction of source's transmission bits at random and would relay them to the destination. However, it is not clear what fraction each relay must select and forward to ensure successful decoding at the destination. To solve this problem we propose Extrinsic Information Transfer Chart (EXIT chart) analysis of Low-Density

Parity-Check (LDPC) codes based on channel mean. We use similar EXIT chart analysis proposed in [18] by Ardakani et al. From this analysis we can get a close estimate of the channel mean required for successful decoding at the destination.

We will then propose a linear programming solution to the problem of relay's forward fraction that will minimize the overall transmission power of the WSN system while maintaining a successful transmission. Our method has the following benefits:

1. *Minimizes energy.* This analytic method will select the minimum transmission power by selecting the fractions of bits forwarded by each relay with the constraint that the transmission is successful. This is achieved by ensuring that channel means are within the threshold derived by EXIT chart analysis.
2. *Flexibility.* There can be other constraints added to this method and the process will still find a solution that will minimize the overall power of the system with respect to those constraints. For example one might add the constraint that each relay can select a maximum of n -bits to forward to the destination. Then our analytic method will minimize the overall transmission power with respect to these constraints while ensuring successful decoding at destination. This process can also be applied to decode-and-forward scheme, demodulate-and-forward scheme as well as any other scheme as long as it

can be represented as a linear programming constraint. More complicated constraints are also possible and the method is very flexible.

3. *Dynamic.* The method is also dynamic meaning it can adjust to the changes in channel instantly. For example, if a source-relay channel SNR changes the number of bits forwarded by this relay can be calculated and adjusted instantly. This dynamic properly makes this method ideal for real world applications where channel SNRs are constantly changing.

1.3 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 provides a literature survey of some of the related work in the area. The system models based on both single-nodes and multi-nodes are described in Chapter 3. In Chapter 4, we propose the use of correlated-sources and compression and study the effects on the rate, diversity order, and transmission error rate. Our simulation results are presented to show the advantages of this method. Our analytic method based on linear programming is presented in Chapter 5. We also present experimental results to demonstrate the correctness and flexibility of this technique. Chapter 6 concludes the thesis and outlines some directions for future work.

2 Literature Survey

In this chapter, a literature survey of some of the background and related works is presented. As discussed in the introduction, two main properties of WSNs, namely spatial diversity and spatial correlation will result in cooperative diversity and Slepian-Wolf compression. In this thesis, we propose a novel scheme for combining these two techniques to achieve a better optimized WSN with excellent performance. To the best of our knowledge there has not been a study on combining these two properties for energy optimization of WSNs. As the result, we first survey some of the work on cooperative diversity followed by some of the work on Slepian-Wolf compression.

2.1 Cooperative Diversity

In this section, we will focus of cooperative diversity [21]. In the first subsection, an overview of basic cooperative techniques will be presented in detail. In particular, coded cooperation, amplify-and-forward (AF), Decode-and-Forward(DF), and

demodulate-and-Forward (DemF) are described in detail. In the second subsection, some of the other related works are mentioned.

2.1.1 Cooperation Overview

It is well known that multiple-input multiple-output (MIMO) that is, systems with many transmit and receive antennas, are advantageous compared to single-input single-output (SISO) systems because of better transmission diversity. Transmission diversity in wireless systems is defined as the number of independent fading paths from a source to a destination. Transmission diversity is usually achieved through the use of multiple antennas at the transmitter. However, in some wireless networks such as cellular networks, some ad hoc networks and sensor networks, the size of the devices is typically small for incorporation of multiple antennas at each node. One way to overcome this limitation is through the use of cooperative communication. In cooperative communication, each single-antenna device can borrow its antenna to other devices in the network and create a virtual MIMO system.

Typically wireless channels, and especially mobile wireless channels, suffer from fading, which means that signal attenuation can vary significantly over the course of transmission. Therefore, transmitting independent copies of the signal over independent channels will generate diversity and can effectively combat the effects of fading. In spatial diversity, transmitting the signal from different locations will

generate diversity by allowing independently faded versions of the signal at the receiver.

In its simplest form, cooperative communication achieves diversity through a relay channel system. Figure 2.1 shows an overview of this basic cooperation scheme. In this figure User 1 or the Source wants to send some information to the Base Station or the Destination. User 2 or the Relay assists User 1 by relaying its information to the destination. The Base Station will then use the signals received from both User 1 and User 2 to decipher source's information. The groundbreaking work of Cover and El Gamal [19] analyzed the information theoretic capacity of this simple relay scheme. They assumed that all nodes operate in the same band, so the system can be decomposed into a broadcast channel from the viewpoint of the source and a multiple access channel from the viewpoint of the destination. Many ideas from this paper were later appeared in other cooperation literature.

Cover and El Gamal mostly analyzed the capacity in an additive white Gaussian noise (AWGN) channel, while recent developments are motivated by the concept of diversity in the fading channel. Moreover, in most cooperative schemes the total system resources are fixed and users act both as information sources as well as relays. This difference is shown in Figure 2.2, where the dotted arrows indicate the transmission of User 2 through User 1, and the solid arrows indicate the transmission of User 1 through User 2. We now review different cooperative signaling

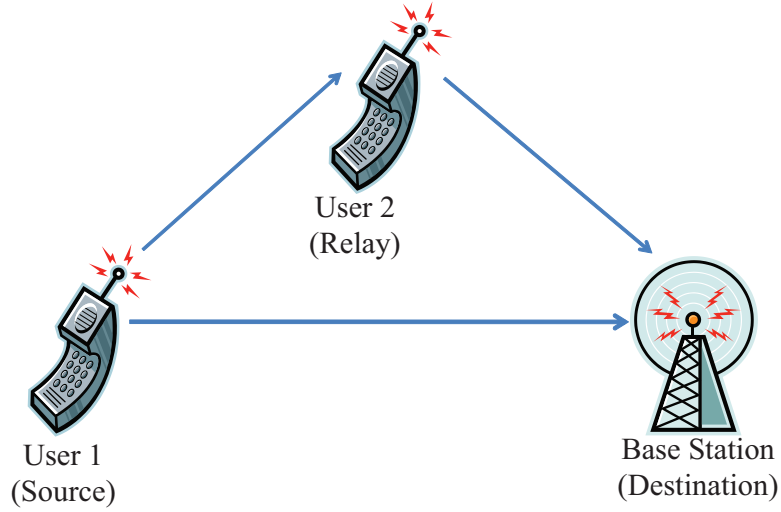


Figure 2.1: Basic Relay Channel Cooperation

schemes.

2.1.1.1 Decode-and-Forward

In this method, the relay decodes the signal coming from the source, re-encodes it using error correcting codes, and retransmits to the destination. Figure 2.3 summarizes this cooperative scheme. In [14] and [20], user cooperation diversity based on decode-and forward (DF) was proposed by Sendonaris et al. as a form of transmission diversity for mobile users. The authors showed that this version of DF signaling cooperation can achieve higher data rates and decreased sensitivity to channel variations. The increase in data rate can also be translated to reduced power for the user (better battery life) or increase in cell coverage (fewer Base

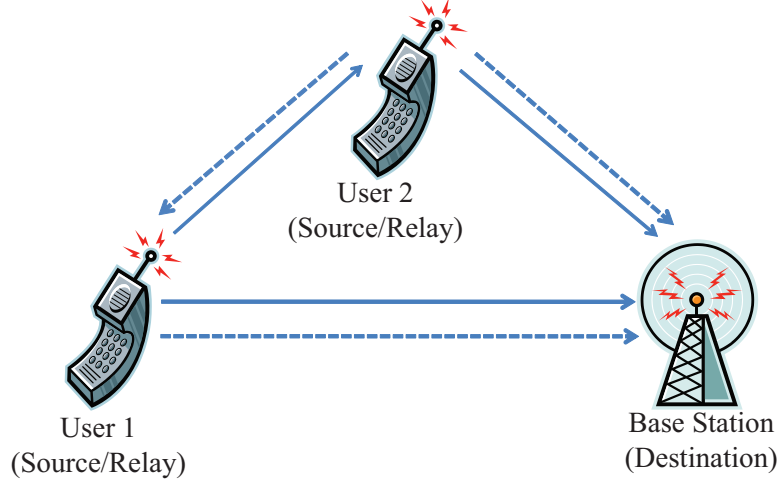


Figure 2.2: Cooperation where each user is both a source and a relay.

Stations).

Sendonaris et al. implemented the DF signaling scheme on a conventional code-division multiple-access (CDMA) system. In this method, each user has its own spreading code, denoted by $c_1(t)$ and $c_2(t)$. The two user's data bits are represented by $b_i^{(n)}$ where $i = 1, 2$ are the user indices and n represents the time index of the information bits. Each signaling period consists of three bit intervals. Therefore, we have

$$X_1(t) = [a_{11}b_1^{(1)}c_1(t), a_{12}b_1^{(2)}c_1(t), a_{13}b_1^{(2)}c_1(t) + a_{14}\hat{b}_2^{(2)}c_2(t)]$$

$$X_2(t) = [a_{21}b_2^{(1)}c_2(t), a_{22}b_2^{(2)}c_2(t), a_{23}\hat{b}_1^{(2)}c_1(t) + a_{24}b_2^{(2)}c_2(t)],$$

where $X_1(t)$ is the signal from user 1, $X_2(t)$ the signal from user 2, and $a_{i,j}$ are the signal amplitudes. Therefore, in the first and second intervals, each user transmits

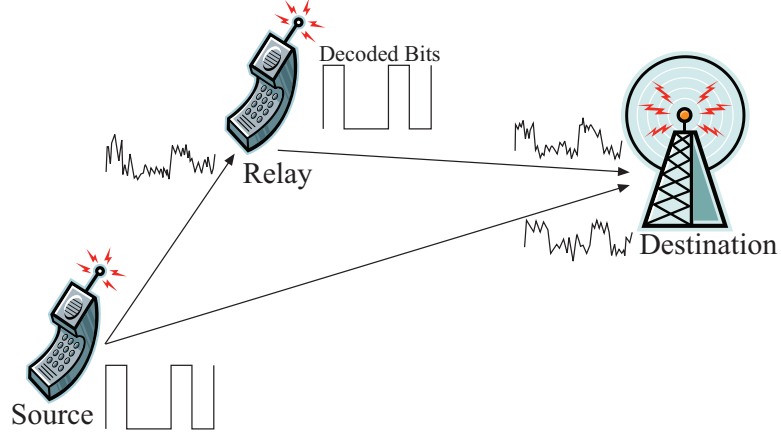


Figure 2.3: Decode-and-Forward (DF)

its own information bits. Each user will then estimate (decode) the other user's second bit denoted by $\hat{b}_i$. In the third interval, both users transmit a linear combination of the their own second bit as well as the estimate of their partner's second bit, each multiplied by corresponding spreading code. The individual signal amplitudes for the first, second and third intervals are used to adjust the transmission power according to the conditions of each link, and provides adaptability to channel conditions.

Sendonaris et al. also considered the performance analysis of an optimal and suboptimal receiver design in the implementation in [14]. In [20], the authors considered a high rate CDMA implementation and a cooperation strategy when assumptions about channel state information at the transmitter was relaxed. The results showed that, in all scenarios considered, cooperation increased the system

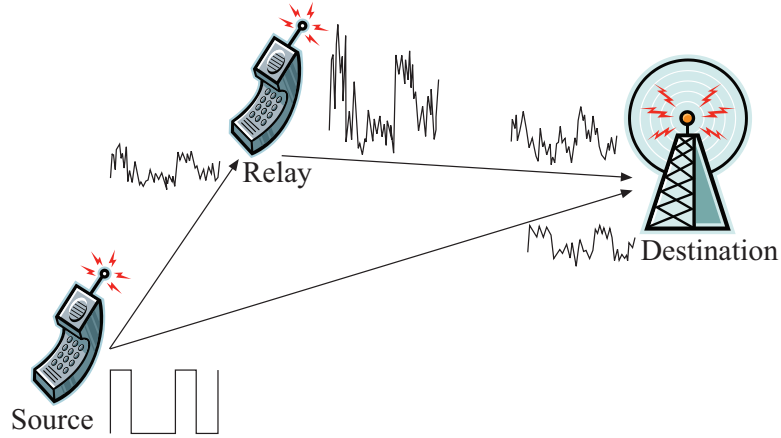


Figure 2.4: Amplify-and-Forward (AF)

throughput and cell coverage and also decreased sensitivity to channel variations.

One of the major disadvantages of this signaling scheme is that it is possible that the decoding at the relay is unsuccessful, in which case cooperation can be detrimental to the decoding process at the destination. To avoid this problem of error propagation, Laneman et al. in [22], proposed hybrid decode-and-forward. In this method, if the relay can not decode the source's transmission successfully (e.g. at times where the SNR between source and relay is low), noncooperative scheme is used.

2.1.1.2 Amplify-and-Froward

Amplify-and-forward (AF) is another simple cooperative signaling scheme. The relay in this method receives a noisy version of the signal transmitted by the source.

The relay will then amplify and retransmit this noisy signal to the destination as depicted in Figure 2.4. The destination combines the signals received from the relay and the source to decode the information bits. Although noise is amplified in this scheme, the destination receives two independently faded versions of the signal and can make better decision at detecting information bits. This work was first presented by Laneman et al. in [22], where they showed that this method achieves diversity order two, which is the best possible outcome at high SNR.

One disadvantage of AF is that it is assumed that the destination knows the interuser channel coefficients to do optimal decoding. Therefore, some method for exchanging or estimating this information must be included in the implementation. Moreover, in a time division scheme, sampling, amplifying, and retransmitting analog values is technologically non trivial.

2.1.1.3 Coded Cooperation

In coded cooperation [27,28], channel coding and cooperation are integrated, where different portions of the code word are sent through different fading paths. In its simplest form, each user tries to add redundancy to its partner's code, when this is not possible, the user will revert back to noncooperative mode. The major advantage of coded cooperation is that the cooperation is managed automatically through coding and no feed back is necessary between users.

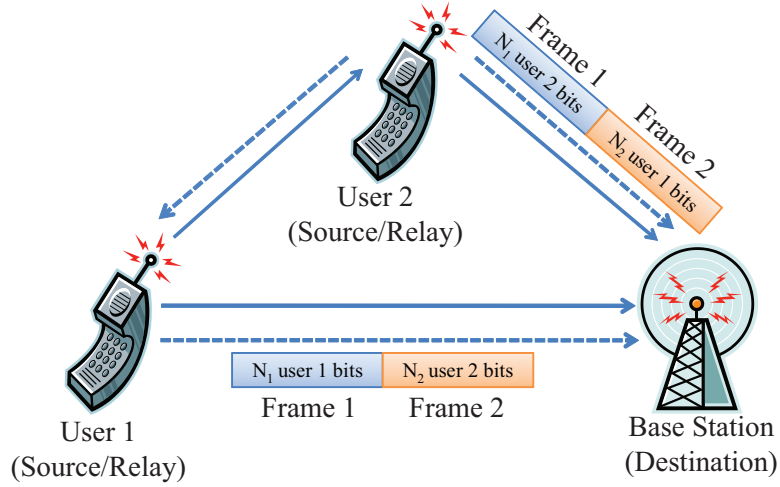


Figure 2.5: Coded Cooperation.

Error correcting codes must be used by users in coded cooperation, and since most communications use some sort of error correcting code, coded cooperation does not add any overhead to the system. It is easier to explain coded cooperation using an example depicted in Figure 2.5. In this example both users encode their information using error correcting codes and a length- N transmission codeword is generated. This length- N codeword is then partitioned into two segments of length N_1 and N_2 , where $N = N_1 + N_2$. For example, one way of partitioning is by puncturing the codeword down into N_1 bits, where exactly N_2 bits are punctured. Therefore, these N_1 bits are still a valid and weaker codeword.

The data transmission period for each user is also divided into two time segments, where N_1 bits are transmitted in the first time segment or frame and N_2

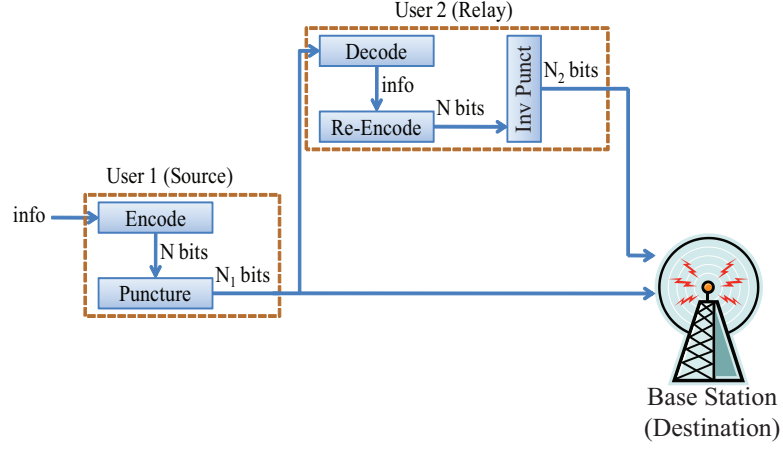


Figure 2.6: Relay Channel Coded Cooperation.

bits are transmitted in the second time segment or frame. In the first frame each user transmits its punctured down N_1 bits codeword. Since the wireless channels broadcast, each user will receive the N_1 bits codeword of its partner and attempts to decode this codeword. If the decoding attempt is successful, in the second transmission frame each user will re-encode and transmit the N_2 punctured bits of its partner. However, if the decoding is unsuccessful by a user, it will transmit its own N_2 punctured bits in the second frame instead of its partner's. Therefore, each user will transmit a total of $N = N_1 + N_2$ bits over two frames, where N_1 bits code from one fading channel and N_2 bits come from another fading channel if the decoding is successful by the partner. Finally, the level of cooperation can be defined as N_2/N , the percentage of the bits each user transmits for its partner. Figure 2.6 shows coded cooperation in a relay system where user 1 acts as the source and user 2 acts

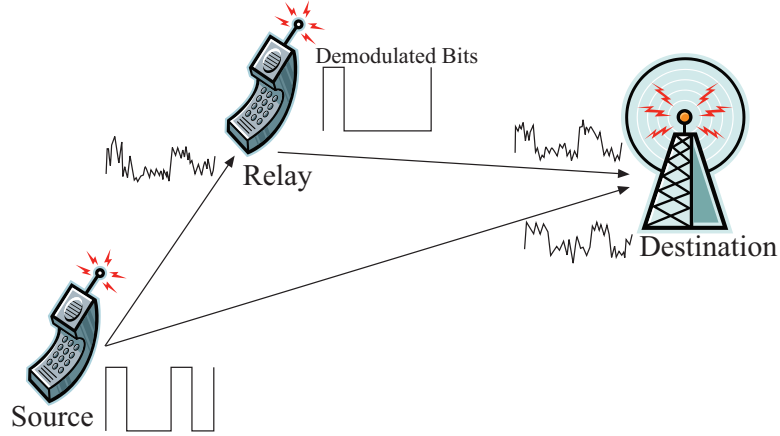


Figure 2.7: Demodulate-and-Forward

as the relay. One advantage of coded cooperation is that various channel coding schemes, such as block or convolutional codes, can be used. Moreover, the code bits for two frames can be selected through puncturing, product codes, or other forms of concatenation.

2.1.1.4 Demodulate-and-Forward

Demodulate-and-forward [24, 25] is another cooperative signaling scheme. This method is very similar to decode-and-forward, but instead of decoding the source's signal, the relay demodulates the signal, re-encodes it and transmits to the destination. This method is more suitable for WSNs since the process of demodulation is less costly than decoding in terms of computational complexity. Figure 2.7 summarizes this scheme. Demodulate-and-forward was first proposed by Chen and

Laneman in [24], and was used in [26] to discuss the capacity of relay systems. In [17] re-encoding of the demodulated data by the relay was proposed and used to drive a low complexity fractional cooperation scheme. Independently, demodulate-and-forward was proposed to implement cooperative diversity in sensor networks in [25].

In [24], Chen and Laneman compared some aspects of the performance of uncoded cooperative diversity with different relay processing, particularly decode-and-forward (DF) and amplify-and-forward (AF), and different demodulations, namely coherent and noncoherent. The paper's main contribution was its focus on DF processing and noncoherent demodulation. For purposes of comparison, Chen and Laneman developed a general framework for maximum likelihood (ML) demodulation in cooperative diversity. A simple piecewise-linear combiner was proposed as an accurate approximation of the nonlinear ML combiner for DF and leads to tight closed-form BER approximations for the noncoherent ML combiner. This paper also develops tight bounds on diversity order for DF, revealing that DF loses about half of the diversity order compared to AF.

2.1.2 Other Related Works

In this section, some of the other related works in cooperative diversity will be surveyed. We include only the works that are most recent and most relevant to

this thesis. For more specific information the reader is referred to references of each paper.

In [23], Laneman et al., developed low-complexity, cooperative protocols that enabled a pair of wireless terminals, each with a single antenna, to fully exploit spatial diversity in the channel. These protocols combined different fixed relaying modes, namely AF and DF, with strategies based on adapting to channel state information between cooperating source terminals as well as utilizing limited feedback from the destination terminal. The costs associated with the proposed cooperative protocol were also discussed. It was noted that cooperation with half-duplex operation required twice the bandwidth of direct transmission of the same rate. This led to larger effective SNR losses for increasing spectral efficiency. It was also mentioned that additional receiver hardware may be required for relaying. There may also had been additional power costs due to relays operating instead of powering down. However, despite all these costs, it was shown that significant performance enhancements are achieved in practice.

In [29], it was shown that under the aggregate power constraint, cooperative relays can be useful even when they do not retransmit but cooperatively listen, giving priority to the transmission of a single opportunistic relay. The authors showed the equivalence of opportunistic DF relaying to the outage bound of the optimal DF. They also presented that opportunistic AF relaying as the outage optimal solution

for single relay selection and showed the significant gain over equal power multi relay SF relaying. The authors therefore concluded that cooperation should be viewed not only as a transmission problem but also as a distributed relay selection task. They also mentioned that the opportunistic relaying required no simultaneous same-frequency transmissions and its simplicity allowed implementation with existing low-complexity radio front ends.

Azarian et al., in [30], considered the design of cooperative protocols for a system consisting of half-duplex nodes. Three different scenarios were considered. First for the relay channel, AF and DF protocols were investigated. Azarian et al. developed a dynamic DF protocol (DDF) and showed its dominance over all known full diversity cooperation strategies and its optimality in a certain range of multiplexing gains. It was shown that in the cooperative broadcast channel, the gains offered by the DDF strategy was more significant compared to the relay channel. Lastly, for the multiple-access scenario, AF cooperative protocol where an artificial ISI channel was created was proposed. Azarian et al. proved the optimality (in the sense of diversity multiplexing tradeoff) of this protocol by showing that it achieved the same trade off curve as the genie-aided point to point system.

In [31], Hong and Scaglione, analyzed the fundamental gain in terms of energy efficiency that is achievable with a novel form of cooperative broadcasting, in which each receiver uses the accumulation of signal energy from multiple transmit-

ting nodes. In previous methods, when a broadcast tree was created each receiver received a signal from one transmitter. The authors suggested using the wireless broadcast advantage to accumulate the signal from all possible transmitters at the receiver for better detection and decoding. This cooperation was provided through a system called the Opportunistic Large Array (OLA) where network broadcasting is done through signal processing techniques at the physical layer. The authors showed that the OLA strategy achieves a lower minimum energy solution compared to another scheme where no user cooperation is considered.

Coded cooperation was considered by Janani et al. in [32]. In the original coded cooperation framework users transmitted their partners' data in the second transmission frame whenever possible. The authors however proposed a new method, called space-time cooperation, where users sent both their own as well as their partners' parity bits in the second frame. In the second part of the paper cooperation through use of turbo codes was considered. The authors showed that better performance in terms bit-error-rate (BER) is achieved by using these cooperative schemes.

A new coded cooperation method using punctured convolutional codes was presented by Stefanov et al. in [33]. In order to study the error performance of these cooperative codes, an analytical framework was considered that showed the potential diversity and coding gains as a function of the inter-user channel quality.

Stefanov et al. showed that for good inter-user channels, cooperative coding indeed achieves diversity. However, when the inter-user channel is very noisy, while there are no diversity gains, there is still some coding gain over direct transmission. These analytical results were confirmed by simulations. Cooperative gains for a symmetric scenario (both nodes have similar quality channels toward their destination) and an asymmetric scenario (one node has a better channel than the other) for a wide range of inter-user channel qualities were studied. The effects of cooperation on the routing decisions in wireless networks were also discussed.

In [34], a turbo-based coding framework for both SISO and MIMO relay systems with various encoding and decoding approaches was proposed. Relay node was assumed to operate in a full-duplex mode, i.e., it transmits and receives signals simultaneously. The destination observed a superposition of the transmitted codewords from the source and the relay nodes, and used a joint decoding scheme working iteratively on all of the available blocks. When the source-to-relay link is perfect, the performance of the proposed coding and decoding technique can be as close as 1.0 dB to the theoretical limits for relay channel. When the source-to-relay link is noisy it is about 1.5 dB of the theoretical limit for the relay channel. Employing this practical method can significantly improve system performance compared to direct and multi-hop transmission schemes.

Practical implementation of the DF strategy for the relay channel was considered

in [35]. The Gaussian relay channel at low signal-to-noise ratios (SNRs) for which binary linear codes are suitable was considered. It was shown that the binning strategy in which a bin index of the codeword is transmitted by the relay to the destination can be interpreted as a parity forwarding scheme. Moreover, the optimal code design for the DF strategy was achieved by design of a Low-Density Parity-Check (LDPC) code working at two different channel SNRs: a high SNR at the relay and a low SNR at the destination. This novel LDPC code construction was named bilayer LDPC coding by the authors and was shown to have good performance results.

In [36], the authors, considered the design and analysis of the Low-Density Parity-Check (LDPC) coded relay systems based on time division half duplex relaying. In particular, a series of coding strategies for different channel conditions was proposed and suitable receiver structures were developed. Also, the convergence behavior of the LDPC coded relay system was analyzed and predicted by employing the measure of average mutual information. Both the convergence analysis and simulation results had shown that relay systems based on LDPC codes can approach the theoretical information limits for the relay channel very closely with appropriate code design. In addition, the optimization problem of the time division parameters and the related bit allocation strategies were discussed. It was shown that by choosing suitable parameters and designing the codes accordingly,

the system performance can be improved significantly.

Chakrabarti et al., in [37], presented Low-Density Parity-Check (LDPC) code designs for the half-duplex relay channel. The authors first presented a near-optimal LDPC coding scheme in which side information was conveyed through additional parity bits. The key challenge of relay code design lied in the utilization of side information from the relay to decode the source transmission. Chakrabarti et al., then proposed three simplifications to reduce the complexity of encoding and decoding without significantly compromising performance. Since the gain of relaying over direct communication was maximum at low SNR, the use of binary modulation in conjunction with binary codes on each channel dimension was considered. The relay coding scheme in multiple access mode was a superposition of two fundamental extremes. In one, the source and the relay sent identical messages, and combined their signals coherently at the destination. In the other, the source and relay sent completely independent information. Any source-relay correlation can be achieved by a combination of these two cases, but the interesting observation was that excellent performance can be achieved if we can simply choose the better of these two schemes. Finally, it was shown that successive decoding was optimal in MAC mode of the relay channel, an observation that simplified the decoder's design.

In [17], Eckford et al., proposed fractional cooperation. In this scheme a source

was considered to have several relays where each relay forwards a randomly selected portion of the source's transmission bits to the destination. Using fractional cooperation, it was shown that energy-efficient diversity gain can be achieved by a system that was flexible enough to be used in a distributed network. Moreover, the derivation of fractional cooperation had made practical assumptions about the capabilities of wireless networking hardware; specifically a relay node does not necessarily need to decode the source's transmission. Practical implementations of this scheme based on Low-Density Generator-Matrix (LDGM) codes and Punctured Systematic Repeat-Accumulate (PSRA) codes were presented. It was shown that fractional cooperation can have transmission diversity of order two or more if a certain number of relays were used. This work was of significance since it showed that the transmission power can be distributed over a large number of relays, that retransmit only a small fraction of the source's bits, while the same diversity gains present in regular cooperation can be achieved.

2.2 Slepian-Wolf Compression

In this section, we survey the works on Slepian-Wolf Compression [15]. WSNs are typically distributed spatially over a geographic area. Moreover, usually the phenomena being measured are spatially correlated, i.e., measurements taken in different places will be similar or close (e.g. Temperature). Therefore, neighbor-

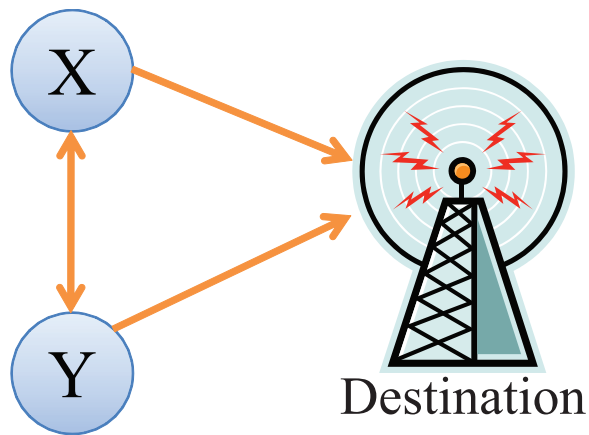


Figure 2.8: Two correlated sources that communicate with each other.

ing nodes measure correlated data and as the results compression can be applied to reduce the number of transmission bits and therefore reduce the power of the system. Although compressing two correlated sources that communicate with each other is trivial, as we will discuss, the task proves to be difficult when correlated sources do not communicate with each other. In this section, we will first present an overview of the Slepian-Wolf concept in which compression is achieved by non-communicating sources, and then a practical implementation known as Distributed Source Coding Using Syndromes (DISCUS) in detail. We will then present some of the other related work in this area.

2.2.1 Slepian-Wolf Compression Overview

A general overview of the Slepian-Wolf compression concept will be presented in this section. To formulate the problem, assume we have two correlated sources, X and Y , that want to transmit their information to a common destination, as shown in Figure 2.8. If the two correlated sources communicate with each other as shown in Figure 2.8 the compression is trivial. For example, assume each correlated source has three information bits to send to the destination. Moreover, let's assume X and Y are correlated such that the Hamming distance between them is at most one (i.e. they differ at most at one bit position). The compression can then be achieved using the following step.

1. Y broadcasts its information bits to destination and X
2. X receives the information bits from Y and XORs them with its own information bits. Since X and Y are correlated such that the Hamming distance between them is at most one there are four possible outcomes,

$$X \oplus Y = \left\{ \begin{array}{l} 000 \longrightarrow 00 \\ 001 \longrightarrow 01 \\ 010 \longrightarrow 10 \\ 100 \longrightarrow 11 \end{array} \right. ,$$

which can be encoded using two bits. Therefore, X has been compressed from three bits down to two bits

3. X transmits these two bits to the destination.
4. At the destination the information from Y is readily available. X is decoded by first mapping the two bits back to three bits as follows,

$$X \oplus Y = \begin{cases} 00 \longrightarrow 000 \\ 01 \longrightarrow 001 \\ 10 \longrightarrow 010 \\ 11 \longrightarrow 100 \end{cases} .$$

find the place of one in these three bits and flip the corresponding bit position in Y to get X

If correlated sources do not communicate, as shown in Figure 2.9, the problem will be nontrivial. In 1973, Slepian and Wolf [15], presented their famous work and showed the compression is still possible even if two correlated sources are independent of each other (i.e. they do not communicate with each other). They determined the minimum number of bits per source character required for the two encoded message streams, from the two correlated sources, in order to ensure accurate reconstruction by the decoder of the outputs of both information sources.

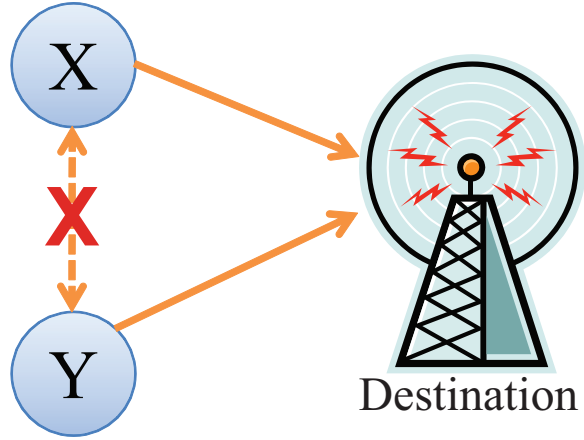


Figure 2.9: Two correlated sources that do not communicate with each other.

In particular, they showed that the overall rate of transmission of the system which includes rate of transmission of X , R_X , and rate of transmission from Y , R_Y can be as low as

$$\min(R_X + R_Y) = H(X, Y) \leq H(X) + H(Y), \quad (2.1)$$

where $H(\cdot)$ is the entropy. In other words the achievable rate of the system is given by

$$R_X + R_Y \geq H(X, Y). \quad (2.2)$$

We can extend this results to our example with three information bits that are correlated such that the Hamming distance is at most one (i.e. they differ at most at one bit position). Lets assume as in the previous example Y transmits all of its information bits to the destination, which means $R_Y = H(Y) = 3$. From statistics

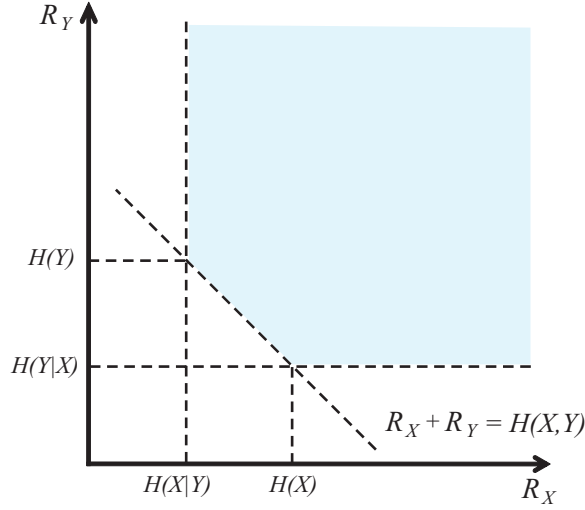


Figure 2.10: Achievable rate region defined by the Slepian-Wolf bounds.

it is clear that $H(X, Y) = H(X|Y) + H(Y)$, and therefore using Equations (2.1) and (2.2) we get $R_X = H(X|Y)$. Since, X is correlated with Y such that the Hamming distance is at most one, we have $H(X|Y) = 2$. Therefore, according to Slepian-Wolf theorem same level of compression can be achieved when correlated sources are independent of each other (i.e. they do not communicate with each other).

In the example above, $R_Y = H(Y)$, which means that Y is not compressed while X is compressed. This is called a “corner case” and is represented by a corner point in Figure 2.10. In general however, both X and Y can be compressed as long as Equation 2.1 is satisfied. This property can be represented as a two dimensional rate region graph shown in Figure 2.10. In this graph the y-axis is R_Y and the x-

axis is R_X , and the borderline shows the maximum rate of compression achievable.

This graph is known as the achievable Slepian-Wolf rate region.

Although Slepian and Wolf proved that it is possible to compress two correlated sources independently, they did not provide any insight on how this can be achieved in a practical sense. In the next section, we will describe a practical implementation known as Distributed Source Coding Using Syndromes (DISCUS), presented by Pradhan and Ramchandran in 1999.

2.2.1.1 DISCUS

Pradhan and Ramchandran, in [16, 38], proposed distributed source coding using syndromes (DISCUS), as a practical method of implementing Slepian Wolf theorem. They focused on a special case when the correlation between X and Y is specified as a prescribed maximal Hamming distance. We now present the general idea behind DISCUS using an example. Lets assume we have a system similar to the one in the previous section where both X and Y have three bits to send and are correlated such that the maximum Hamming distance between them is at most one. The following process can be used to compress and decompress the information bits without any communication between correlated sources.

1. Just like before Y transmits all its information to destination. However, these transmitted information bits are not available at X .

2. X creates four cosets such that the elements at each coset have Hamming distance 3 as shown bellow.

$$\begin{bmatrix} 000 \\ 111 \end{bmatrix} \rightarrow 00, \begin{bmatrix} 001 \\ 110 \end{bmatrix} \rightarrow 01, \begin{bmatrix} 010 \\ 101 \end{bmatrix} \rightarrow 10, \begin{bmatrix} 100 \\ 011 \end{bmatrix} \rightarrow 11.$$

These four cosets can be encoded using two syndrome bits as shown. For example, if the information bits at X is 100 or 011 then syndrome bits is 11.

3. X transmits the two bit syndrome bits to the destination.
4. At the destination the information bits from Y is readily available. To determine the information of X the syndrome bits are mapped back to the corresponding coset. Since each coset has two elements the Hamming distance between each element and information bits of Y is measured, and the one with the smallest Hamming distance is selected.

As an example, lets assume information at X is 100 and at Y is 101. Therefore, Y transmits 101 while X transmits 11. At the destination the information from Y is readily available. From mapping of the coset to syndrome bits we know that X 's information is in the fourth coset. To determine which of the two element in the coset corresponds to X , the Hamming distance between information bits of Y , 101, and the two elements in the coset, 100 and 011, is measured. The Hamming distance with respect to the first element, 100, is 1 while the Hamming distance

with respect to the second element, 011, is 2. Therefore, the destination can decide that information bits of X is 100.

Pradhan and Ramchandran [16, 38], also proposed a low complexity encoding and decoding method based on linear codes that achieved all points in the achievable rate region of Slepian-Wolf theorem. The extension of these concepts to the construction of Euclidean space codes is also studied and analyzed for the case of trellis and lattice codes. The performance of these symmetric methods for encoding with a fidelity criterion was shown to be the same as that of asymmetric encoding.

2.2.2 Other Related Works

In this section, some of the other related works in Slepian-Wolf compression will be surveyed. We include only the works that are most recent and most relevant to this thesis. For more specific information the reader is referred to references of each paper.

The use of turbo codes as applied to the Slepian-Wolf problem was investigated in [39]. Finite-state machine (FSM) encoders, concatenated in parallel, were used at the transmit side and an iterative turbo decoder was applied at the receiver. Simulation results of system performance were presented for binary sources with different amounts of correlation. Obtained results showed that the proposed technique outperforms by far both an equivalent uncoded system and a system coded

with a single FSM encoder and BCJR decoding.

The use of punctured turbo codes for compression of correlated binary sources was considered by Garcia-Frias et al. in [40]. They achieved the compression by puncturing turbo codes. In the encoding process, no information about the correlation between sources was required. The proposed decoder used an iterative scheme, and performed well even when the correlation between the sources was not known at the decoder since this correlation can be estimated jointly with the iterative decoding process. The resulting performance of the proposed scheme was close to the theoretical limit provided by the SlepianWolf theorem.

In [41], Liveris et al., presented a way of doing distributed compression with side information using binary Low-Density Parity-Check (LDPC) codes. They focused on the asymmetric use of compression with side information. This means one source is compressed fully while the other source is not compressed (also known as a corner point in Slepian-Wolf rate region). Their approach was based on viewing the correlation as a channel and applying the syndrome concept. The encoding and decoding procedures, i.e. the compression and decompression, were explored in detail. The authors showed through simulation that the performance results were better than most of the existing turbo code results and very close to the Slepian-Wolf limit in case of irregular LDPC codes.

Systematic irregular repeat accumulate (IRA) codes were used as joint source-

channel codes for the transmission of an equiprobable memoryless binary source with side information at the decoder, in [42]. A special case of this problem under consideration was joint source-channel coding for a nonequiprobable memoryless binary source. The theoretical limits of this problem were derived by combining the Slepian-Wolf theorem, the source entropy in the special case, with the channel capacity. The authors viewed the correlation between the binary source output and the side information as a separate channel or an enhancement of the original channel. They then described the encoding and decoding procedures, used as joint source-channel coding, based on systematic IRA codes, in detail. The simulated performance results were then presented and shown to be better than some turbo code results and very close to the theoretical limits of Slepian-Wolf theorem.

In [43], the use of Low-Density Generator-Matrix (LDGM) codes for both channel coding and joint source-channel coding of correlated sources over noisy channels was presented. In particular the concatenated schemes was used in order to avoid the error floors associated with LDGM codes. The encoding and decoding complexity of the proposed scheme presented advantages with respect to turbo and standard LDPC codes. For channel coding, the performance over BSCs, AWGN channels and ideally interleaved Rayleigh fading channels with perfect channel state information (CSI) at the receiver was comparable to that of turbo codes and standard irregular LDPC codes, and close to the theoretical limits of Slepian-Wolf theorem. In the

case of correlated sources the proposed system also achieved a performance close to the theoretical limits and similar to those of turbo codes.

A Slepian-Wolf cooperation scheme was proposed to improve the inter-user outage performance in wireless cooperative communications in [44]. To do this the authors applied Slepian-Wolf cooperation to a relay system. Since the data sent through the relay was corrupted by noise, and therefore correlated with original data, the authors suggested compressing source's information bits and transmitting them with the original uncompressed sources bits. The authors called this method Slepian-Wolf cooperation. It was shown that this scheme can significantly increase the performance compared to the case where no cooperation is used.

In [45], the transmission of information from two correlated sources to a common destination through a Rayleigh fast fading multiple access channel was considered. Each source was encoded independently using a turbo-like code and neither the correlation model nor the channel state information was assumed to be known at the encoder. A novel iterative decoding technique was performed at the destination that exploits the correlation between the two senders. Because of the iterative coding method, the noise variance and correlation model did not need to be known at the decoder site, since they could be estimated jointly. However, when perfect channel state information was available at the decoder, the resulting performance was very close to theoretical limits given by Slepian-Wolf theorem. When the

channel state information was not available at the decoder, the performance loss would depend on the correlation strength, becoming smaller for highly correlated sources.

In [46], Low-Density Parity-Check (LDPC) codes were used in providing reliable data transmission and developing aggregation techniques for correlated data in wireless sensor networks. This work, first proposed forward error correction (FEC) technique using LDPC codes that can reduce the transmission power. The simulation results showed that this approach was significantly more efficient than using BCH codes and convolutional codes. The authors then studied the problem of source and channel coding for two and three correlated nodes. They also proposed the use of non-uniform LDPC codes for distributed source coding of two correlated nodes. This scheme improved the performance of the source coding considerably. They also extended the results to the case of three correlated nodes and showed similar performance gains.

2.3 Thesis Overview

In this chapter, we surveyed some of the background and related works of this thesis. In this section, we will give an overview of the thesis with respect to the materials presented in the previous sections. As it was mentioned in Chapter 1, WSNs are typically distributed spatially over an area. Because of this spatial distribution

independent fading paths exist from a sensor node to the data sink. Therefore, cooperative diversity can be applied to ensure a better communication and lower overall system power consumption. On the other hand, because naturally occurring phenomena are usually spatially correlated, Slepian-Wolf theorem can be applied to compress the information bits before transmission. This will reduce the number of bits transmitted by sensor node, thereby saving valuable battery power.

Given the beneficial effect of, cooperation and Slepian-wolf compression, it is natural to study the effects of one on the other and the overall effects of combining these two schemes to make a more energy efficient WSN. However, as it was shown in the survey section there is no work in the literature that studies these effects. Therefore, in this thesis we combine cooperation and Slepian-Wolf compression and study the effects on a WSN system. Also, in the second part of this thesis, we try to provide an analytic technique for a special class of cooperation known as Fractional Cooperation, surveyed in section 2.1.2, that minimizes the overall energy of the WSN while it maintains the needed diversity.

The rest of this thesis is organized as follows. In Chapter 3, we present the system model, the relay model, channel codes and codes for Slepian-Wolf compression. Then, in Chapter 4, we combine cooperation with Slepian-Wolf compression and study the effects of compression on cooperation and the diversity order. We will also analyze the performance and energy consumption when both cooperation

and compression are combined. An analytical method based linear programming for minimizing the energy of a fractional cooperative WSN will be presented in Chapter 5. Finally, we conclude the thesis with Chapter 6.

3 System Model

In this chapter we will present our system model, which consists of a relay system and coding methods and techniques. In general, our system models are inspired by the model used in [17]. We consider a sensor network with multiple sensors and one information sink. Each sensor measures a phenomenon and transmits its measurements to the data sink. We assume the sensors have simple two-way radios, processors of limited complexity, and limited power resources; while the data sink possesses virtually unlimited radio, computational, and power resources.

The sensors' limited capabilities, and their ability to communicate with each other, imply that they should co-operate in conveying information to the sink. We also assume that the measured data across different sensors in close vicinity of each other are correlated. With these assumptions in mind, the rest of this chapter is organized as follows. In Section 3.1 we will discuss our relay model that represents the cooperation between nodes. Then, in Section 3.2, we will discuss different coding techniques used throughout the rest of this thesis for transmission

and compression of information.

3.1 Relay Model

Our relay model consists of a source, a destination, zero or more relays, and zero or more correlated sources. The purpose of the system is to convey sensor measurements from the source to the destination. The relays receive the source's transmission and assist the source in transmitting its information to the destination. The correlated sources, which observe a physical phenomenon that is correlated with the source's phenomenon, also assist the source in transmitting its information, but *have no radio link with the source*. We start by giving a detailed description of a single-relay, single-correlated-source model and then extend the results to a multi-relay, multi-correlated-source system.

3.1.1 Single correlated source and relay

Here we give a detailed description of a single-relay, single-correlated-source models. Our four-node model has a source, a single relay, a correlated source and a destination. As shown in Figure 3.1, there are four radio links: source to relay (SR), source to destination (SD), relay to destination (RD), and correlated source to destination (CSD). We assume these communication links use binary phase shift keying (BPSK) for data modulation. A particular realization of the channel is parame-

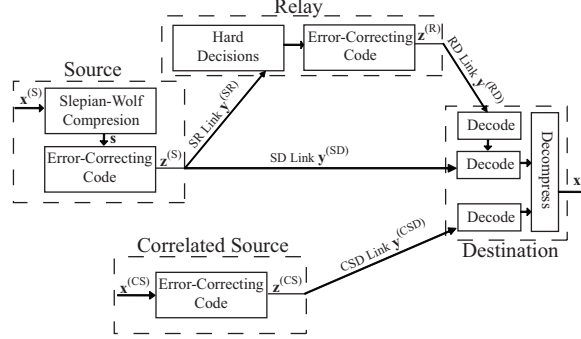


Figure 3.1: Single relay, single correlated source model.

terized by representing the amplitude on the four links with $(a_{SD}, a_{SR}, a_{RD}, a_{CSD})$, where a_{SD} is the amplitude over SD link, a_{SR} is the amplitude over SR link, a_{SR} is the amplitude over RD link and a_{CSD} is the amplitude over CSD link.

The length- n information strings at the source and correlated source sensors are represented by $\mathbf{x}^{(S)} = \{x_1^{(S)}, x_2^{(S)}, \dots, x_n^{(S)}\}$ and $\mathbf{x}^{(CS)} = \{x_1^{(CS)}, x_2^{(CS)}, \dots, x_n^{(CS)}\}$, respectively, where $x_i^{(S)}, x_i^{(CS)} \in \{0, 1\}$. To represent the correlation between the source and correlated source we assume $\Pr[x_i^{(S)} \neq x_i^{(CS)}] = p < 0.5$.

The source transmits its information bits in two phases. In the first phase DISCUS [13] source coding is used to compress source's information bits according to Slepian-Wolf theorem. As we described the operation of discus in Chapter 2, we will have a length- k vector of syndrome bits $\mathbf{s} = \{s_1, s_2, \dots, s_k\}$ where $k \geq nH(x_i^{(S)} | x_i^{(CS)})$. The source will then encode the resulting syndrome bits with error correcting codes, which results in a length- m codeword $\mathbf{z}^{(S)} = \{z_1^{(S)}, z_2^{(S)}, \dots, z_m^{(S)}\}$.

The correlated source will also encode all its information bits with error correcting codes and the length- l codeword $\mathbf{z}^{(CS)} = \{z_1^{(CS)}, z_2^{(CS)}, \dots, z_l^{(CS)}\}$ will be transmitted to the destination. Since we are using BPSK we assume $z_i^{(S)}, z_i^{(CS)} \in \{+1, -1\}$ and we define $\sigma : \{0, 1\} \rightarrow \{+1, -1\}$ as a function which translates zero and one to one and minus one respectively. The relay and destination will therefore observe the real valued vectors

$$\mathbf{y}^{(SR)} = a_{SR}\mathbf{z}^{(S)} + \mathbf{n}^{(SR)}, \quad (3.1)$$

$$\mathbf{y}^{(SD)} = a_{SD}\mathbf{z}^{(S)} + \mathbf{n}^{(SD)}, \quad (3.2)$$

$$\mathbf{y}^{(CSD)} = a_{CSD}\mathbf{z}^{(CS)} + \mathbf{n}^{(CSD)}, \quad (3.3)$$

where $\mathbf{n}^{(SR)}$, $\mathbf{n}^{(SD)}$, and $\mathbf{n}^{(CSD)}$ represent unit-variance additive white Gaussian noise (AWGN) vectors at the relay and destination links respectively.

The relay receives $\mathbf{y}^{(SR)}$ and uses a processing function $\phi : \mathbb{R}^m \rightarrow \{0, 1\}^m$ on $\mathbf{y}^{(SR)}$ to estimate the data sent by the source. The result of the processing function,

$$\mathbf{x}^{(R)} = \phi(\mathbf{y}^{(SR)}), \quad (3.4)$$

is encoded using error correcting codes and the resulting length- h codeword $\mathbf{z}^{(R)} = \{z_1^{(R)}, z_2^{(R)}, \dots, z_h^{(R)}\}$, $z_i^{(R)} \in \{+1, -1\}$, is transmitted to the destination. Therefore the signal received by the destination is given by

$$\mathbf{y}^{(RD)} = a_{RD}\mathbf{z}^{(R)} + \mathbf{n}^{(RD)}. \quad (3.5)$$

To keep the processing function energy efficient and simple, we make hard decisions on $\mathbf{y}^{(SR)}$ at the relay. Therefore the processing function is given by

$$\mathbf{x}^R = \phi(\mathbf{y}^{(SR)}) = \sigma^{-1}(\text{sign}(\mathbf{y}^{(SR)})), \quad (3.6)$$

where the $\text{sign}(\cdot)$ function returns +1 if the argument is positive and -1 if the argument is negative. Furthermore, $\sigma^{-1}(\cdot)$ is the inverse of the $\sigma(\cdot)$ function explained before.

The Rayleigh fading model is parameterized by $\bar{\gamma}$, the average signal-to-noise ratio, assuming unit noise power. Therefore, in Rayleigh fading the channel amplitude is a random variable with probability distribution function (PDF)

$$P_A(a) = \frac{2a}{\bar{\gamma}} \exp\left(\frac{-a^2}{\bar{\gamma}}\right). \quad (3.7)$$

As a result, $E[A^2] = \bar{\gamma}$, and the average signal-to-noise ratio is $\bar{\gamma}$. Thus, in a fading channel, the four amplitudes $(a_{SD}, a_{SR}, a_{RD}, a_{CSD})$ are a four-dimensional vector of independent Rayleigh-distributed random variables, parameterized by $(\bar{\gamma}_{SD}, \bar{\gamma}_{SR}, \bar{\gamma}_{RD}, \bar{\gamma}_{CSD})$.

3.1.2 Multi-relay and multi-correlated-source

Here we consider a system with r relays and q correlated sources, as shown in Figure 3.2. Each relay and correlated source behaves as described in the last section, and the notion is similar to the one-relay case, with the following generalizations:

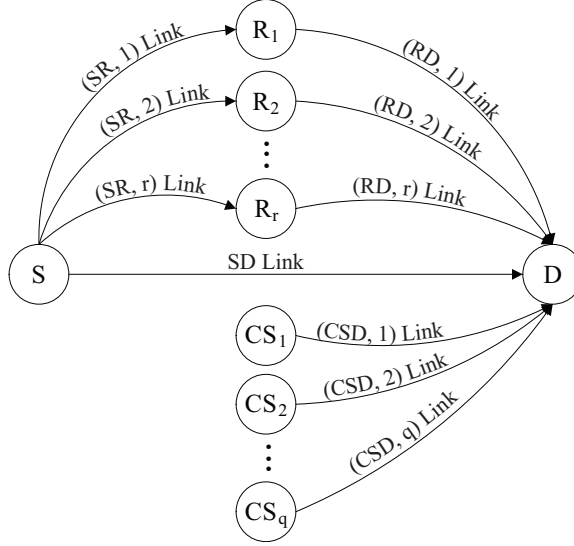


Figure 3.2: Multi-relay, multi-correlated-source model.

1. There are two index sets $I^{(R)} = \{1, 2, \dots, r\}$ and $I^{(CS)} = \{1, 2, \dots, q\}$ containing a unique index for each relay and correlated source, respectively.
2. Channel amplitudes are $(a^{(SD)}, \mathbf{a}^{(SR)}, \mathbf{a}^{(RD)}, \mathbf{a}^{(CSD)})$, where the vectors elements $\mathbf{a}^{(SR)} = [a_1^{(SR)}, a_2^{(SR)}, \dots, a_r^{(SR)}]$ and $\mathbf{a}^{(RD)} = [a_1^{(RD)}, a_2^{(RD)}, \dots, a_r^{(RD)}]$ represent the source-to-relay and relay-to-destination amplitude for each relay in $I^{(R)}$, respectively. Similarly, $\mathbf{a}^{(CSD)} = [a_1^{(CSD)}, a_2^{(CSD)}, \dots, a_q^{(CSD)}]$ is the vector of correlated source-to-destination amplitudes for each correlated source in $I^{(CS)}$.
3. The equations (3.1), (3.3) and (3.5) are modified as follows

$$\mathbf{Y}^{(SR)} = \mathbf{A}^{(SR)} \mathbf{Z}^{(S)} + \mathbf{N}^{(SR)}, \quad (3.8)$$

$$\mathbf{Y}^{(CSD)} = \mathbf{A}^{(CSD)} \mathbf{Z}^{(CS)} + \mathbf{N}^{(CSD)}, \quad (3.9)$$

$$\mathbf{Y}^{(RD)} = \mathbf{A}^{(RD)} \mathbf{Z}^{(R)} + \mathbf{N}^{(RD)}, \quad (3.10)$$

while the equation (3.2) remains unchanged For (3.8), we define

$$\mathbf{Y}^{(SR)} := \begin{bmatrix} \mathbf{y}^{(SR,1)} \\ \mathbf{y}^{(SR,2)} \\ \vdots \\ \mathbf{y}^{(SR,r)} \end{bmatrix}, \mathbf{A}^{(SR)} := \text{diag}(\mathbf{a}^{(SR)}),$$

$$\mathbf{Z}^{(S)} := \begin{bmatrix} \mathbf{z}^{(S)} \\ \mathbf{z}^{(S)} \\ \vdots \\ \mathbf{z}^{(S)} \end{bmatrix}, \mathbf{N}^{(SR)} := \begin{bmatrix} \mathbf{n}^{(SR,1)} \\ \mathbf{n}^{(SR,2)} \\ \vdots \\ \mathbf{n}^{(SR,r)} \end{bmatrix}.$$

The superscript (R, i) for $i \in I^{(R)}$ refers to processes at the i th relay. Using a similar technique, we can define $\mathbf{Y}^{(CSD)}$, $\mathbf{Y}^{(RD)}$, $\mathbf{A}^{(CSD)}$, $\mathbf{A}^{(RD)}$, $\mathbf{N}^{(CSD)}$ and $\mathbf{N}^{(RD)}$ for equations (3.9), (3.10). For equation (3.9) the index i is $i \in I^{(CS)}$ instead of $I^{(R)}$. The $\mathbf{Z}^{(CS)}$ and $\mathbf{Z}^{(R)}$, however are defined as

$$\mathbf{Z}^{(R)} := \begin{bmatrix} \mathbf{z}^{(R,1)} \\ \mathbf{z}^{(R,2)} \\ \vdots \\ \mathbf{z}^{(R,r)} \end{bmatrix}, \mathbf{Z}^{(CS)} := \begin{bmatrix} \mathbf{z}^{(CS,1)} \\ \mathbf{z}^{(CS,2)} \\ \vdots \\ \mathbf{z}^{(CS,q)} \end{bmatrix}.$$

where, $\mathbf{z}^{(R,i)}$ and $\mathbf{z}^{(CS,j)}$ are the bits to be transmitted to the destination by the i th relay and j th correlated source respectively.

4. In Rayleigh fading, $(\bar{\gamma}^{(SD)}, \bar{\gamma}^{(SR)}, \bar{\gamma}^{(RD)}, \bar{\gamma}^{(CSD)})$, represents the average SNR on each link.

3.1.3 Multi-Source, Multi-Relay Fractional Cooperation

In this section we consider a multi-source, multi-relay system with no correlated source and compression. The relay model that we use is an extension of the fractional cooperation model proposed in [17]: we consider s sources, r relays, and a single destination. The r relays are shared amongst all s sources (i.e. each single source has r relays that assists in its transmission), as shown in Figure 3.3. Each source measures a phenomenon, encodes it using LDPC codes, and broadcasts the encoded codeword to the r relays, as well as the destination. The relays are assumed to employ both demodulate-and-forward (DemF), as well as decode-and forward (DF) cooperative schemes. After decoding or demodulating the i th source's signal, the j th relay, selects a small fraction $\epsilon_{(i,j)}$ for retransmission to the destination. The destination will then decode each source's information bits using the received signal from the r relays, as well as the source itself.

Each source has a length- n information sequence to transfer to the destination represented by $\mathbf{x}^{(S_i)} = \{x_1^{(S_i)}, x_2^{(S_i)}, \dots, x_n^{(S_i)}\}$, where $x_k^{(S_i)} \in \{0, 1\}$ and S_i repre-

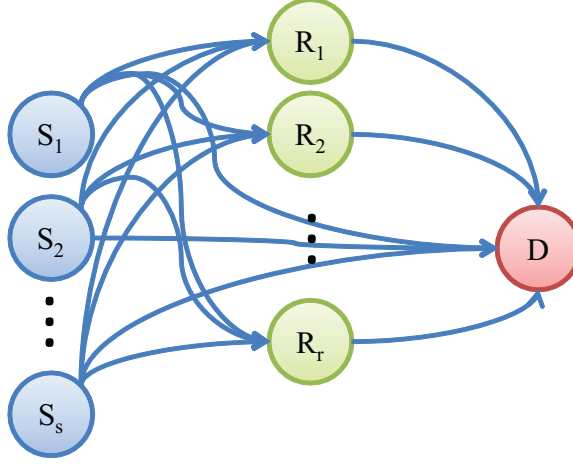


Figure 3.3: Multi-Source, Multi-relay fractional cooperation model.

sents the i th source. Each information sequence is encoded by an LDPC code for each source. Let $R_1, R_2, \dots, R_s$ be the code rates at each source. Therefore the codeword ready for transmission at the i th source is represented by $\mathbf{z}^{(S_i)} = \{z_1^{(S_i)}, z_2^{(S_i)}, \dots, z_{m_i}^{(S_i)}\}$, where $m_i = n/R_i$ is the length of the codeword.

As shown in Figure 3.3, there are $s \times r$ source to relay (S-R), s source to destination (SD), and r relay to destination (R-D) links. We assume these communication links use binary phase shift keying (BPSK) for data modulation. We define the function $\phi : \{0, 1\} \rightarrow \{+1, -1\}$ as the modulation function where 0 is mapped to a +1 and 1 is mapped to -1. Also, the demodulation function is defined as

$$\phi^{-1}(y) = \begin{cases} 0 & \text{if } y \geq 0 \\ 1 & \text{otherwise} \end{cases}, \quad (3.11)$$

with slight abuse of the inverse notation.

The S-D links are therefore given by

$$\mathbf{y}^{(S_i,D)} = \phi(\mathbf{z}^{(S_i)}) + \mathbf{n}^{(S_i,D)}, \quad (3.12)$$

where S_i corresponds to the i th source and $\mathbf{n}^{(S_i,D)}$ is AWGN with variance $\sigma_{(S_i,D)}^2$.

The channel SNRs for each of the s S-D links are represented by $\gamma_{(S_i,D)} = 1/(2\sigma_{(S_i,D)}^2)$.

The S-R links are also given by

$$\mathbf{y}^{(S_i,R_j)} = \phi(\mathbf{z}^{(S_i)}) + \mathbf{n}^{(S_i,R_j)}, \quad (3.13)$$

where S_i and R_j correspond to the i th source and the j th relay respectively and $\mathbf{n}^{(S_i,R_j)}$ is AWGN with variance $\sigma_{(S_i,R_j)}^2$. Therefore, all the S-R links can be represented by $s \times r$ channel SNRs $\gamma_{(S_i,R_j)} = 1/(2\sigma_{(S_i,R_j)}^2)$.

Decode-and-forward. In DF a relay will first decode a source's information bits, then re-encode the information bits using the exact same encoding process used by the original source. Therefore, if the decoding process for the i th source at a relay is successful the information sequence, $\mathbf{x}^{(S_i)}$, is recovered by that relay and the re-encoding process will reproduce the same codeword, $\mathbf{z}^{(S_i)}$, transmitted by the source. For simplicity, we assume that the relays can always decode the source's information bits successfully. In practice this assumption is achievable by using higher code rates. Therefore, we assume the codeword $\mathbf{z}^{(S_i)}$ is available at each relay.

The j th relay will select a small fraction, $\epsilon_{(i,j)}$, of codeword, $\mathbf{z}^{(S_i)}$, to forward to the destination assisting the i th source. We define a vector $\mathbf{b}^{(S_i, R_j)} \in \{0, 1\}^{m_i}$ such that if a codeword position k is selected by the j th relay to be forwarded for the i th source, the k position is set to one and zero otherwise. Therefore, the vector, $\mathbf{b}^{(S_i, R_j)}$, has a Hamming distance of $m_i \epsilon_{(i,j)}$. The R-D channels can be formulated using $\mathbf{b}^{(S_i, R_j)}$ as

$$\mathbf{y}_{DF}^{(S_i, R_j, D)} = \mathbf{b}^{(S_i, R_j)} \odot [\phi(\mathbf{z}^{(S_i)}) + \mathbf{n}^{(R_j, D)}] \quad (3.14)$$

where $\odot$ is element wise multiplication of vectors, and $\mathbf{n}^{(R_j, D)}$ is AWGN with variance $\sigma_{(R_j, D)}^2$. The channel SNRs for each of the r R-D links are represented by $\gamma_{(R_j, D)} = 1/(2\sigma_{(R_j, D)}^2)$.

Demodulate-and-forward. In DemF, a relay first demodulates the signal received from a source. This process can be formulated as

$$\mathbf{z}^{(S_i, R_j)} = \phi^{-1}(\mathbf{y}^{(S_i, R_j)}), \quad (3.15)$$

where $\mathbf{z}^{(S_i, R_j)}$ is the results of hard decisions (demodulation) for the j th relay assisting i th source.

Each relay will then select a fraction of the demodulated signal, re-encodes it using error correcting codes and retransmits to the destination. In DemF any type of code (such as RA codes or irregular LDPC codes) can be used. For simplicity we assume that using powerful and capacity approaching codes such as irregular

RA [51], and irregular LDPC [57] codes, over R-D links, can result in perfect recovery of demodulated bits at the destination, at rates close to capacity. Thus, for any DemF system, we will assume that a capacity-approaching code is used in the R-D link, and is decoded without error.

The vector $\mathbf{b}^{(S_i, R_j)}$ represents the demodulated bit positions selected for retransmission to the destination: if $b_k^{(S_i, R_j)} = 1$, then the k th bit is relayed; if $b_k^{(S_i, R_j)} = 0$, then the k th bit is not relayed. Therefore, the demodulated sequence resulting from the j th relay assisting i th source is available at the destination as

$$\mathbf{y}_{DemF}^{(S_i, R_j, D)} = \mathbf{b}^{(S_i, R_j)} \odot \phi(\mathbf{z}^{(S_i, R_j)}), \quad (3.16)$$

where $\odot$ is element-wise multiplication of vectors, $\mathbf{z}^{(S_i, R_j)}$ is given by equation (3.15), and $\mathbf{y}^{(S_i, R_j, D)}$ represents the results of demodulations available at the destination. The elements of $\mathbf{y}_{DemF}^{(S_i, R_j, D)}$ can take three possible values: +1 (representing a demodulated 0 bit), -1 (representing a demodulated 1 bit), and 0 (representing a position that is not selected for relaying).

3.2 Coding Methods

In this section we will look at different coding techniques used in the rest of this thesis. We use two well known classes of codes namely, Low-Density Parity-Check (LDPC) codes and Repeat-Accumulate (RA) codes. The LDPC codes are used

for correlated source compression as well as in channel coding as a form of error correcting code. The Repeat-Accumulate (RA) codes are used as error correcting codes for channel coding. The rest of this section is organized as follows. In section 3.2.1, RA codes are discussed as well as a special class of RA codes called Punctured Systematic Repeat-Accumulate (PSRA) codes. LDPC codes for both compression and channel coding are discussed in section 3.2.2.

3.2.1 Repeat-Accumulate Codes

Repeat-Accumulate codes (RA Codes), proposed by Divsalar et al. [47], are a low complexity class of error correcting codes that are competitive alternative to turbo codes [48] and Low-Density Parity-Check (LDPC) [49] codes. Throughout the years different extensions are made to RA codes. In 2000, Jin et al. [51], proposed Irregular RA (IRA) codes that consist of an outer code that is a mixture of repetition codes and an inner code that consists of parity-check and an accumulator. IRA codes are shown to out perform turbo codes and achieve limits close to Shannon's limits. In [17,53,54], puncturing was proposed for adjusting the rate of transmission and Punctured Systematic RA (PSRA) codes were developed.

RA codes are an excellent choice for WSNs since they have a low complexity encoder and perform well at low signal-to-noise ratio. Because typically sensors nodes have limited computational power, the RA code's low complexity encoder

can be implemented easily. Also, because RA codes perform well at low signal-to-noise ratio, the sensor node can use less power in transmission and save energy. In the rest of this section we will discuss the RA code's encoder and iterative decoder as well as use of PSRA codes for adjusting the rate of transmission.

3.2.1.1 Encoder

The encoder is explained in the following algorithm.

1. Take $x_1x_2x_3 \cdots x_n$, a sequence of n information bits to be encoded.
2. Repeat each bit three times to form a sequence of length $N = 3n$

$$x_1x_1x_1x_2x_2x_2x_3x_3x_3 \cdots x_nx_nx_n$$

3. Permute these N bits using a random permutation and call the permuted sequence $\mathbf{v}$

$$v_1v_2v_3v_4v_5v_6v_7v_8v_9 \cdots v_N$$

4. Create the final encoded sequence $\mathbf{z}$ by accumulating using modulo two ad-

dition as follows.

$$\begin{aligned}
z_1 &= v_1 \\
z_2 &= z_1 \oplus v_2 \\
&\vdots \\
z_n &= z_{n-1} \oplus v_n \\
&\vdots \\
z_N &= z_{N-1} \oplus v_N
\end{aligned}$$

At step 2 of this algorithm, information bits were repeated three times. However, it is noted that this repetition factor can be any number in practice. It is also noted in [47] that the repetition factor of at least three is needed in noisy environments, and hence the use of three in this algorithm.

3.2.1.2 Decoder

In this section an iterative decoding algorithm for RA codes based factor graphs and sum-product algorithm [52] is presented. The factor graph of an RA code is presented in Figure 3.4. This factor graph consists of variable nodes, the circular purple nodes, and check nodes, the square blue nodes. The small red circular nodes represent messages received from the channel. The information bit sequence $\mathbf{x}$ is presented by the top variable nodes. The number of edges connected to the top variable nodes define the repetition factor. For example, Figure 3.4 represents an

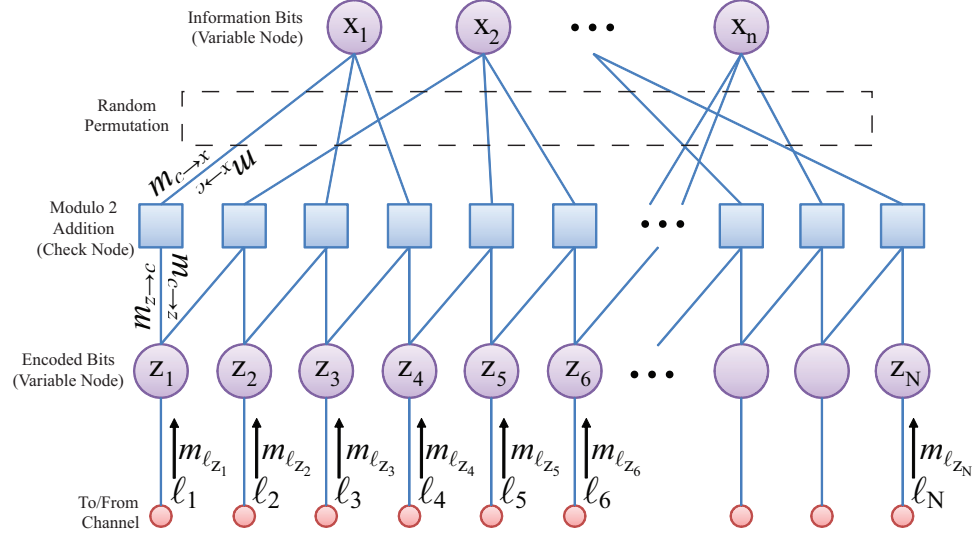


Figure 3.4: The factor graph representation of RA codes.

RA code with repetition factor 3. These edges are connected to the check nodes beneath according to the random permutation step described in encoding section. The check nodes in the middle represent a modulo-2 addition of the incoming edges. The variable node at the bottom represents the sequence $\mathbf{z}$ of the resulting encoded bit. From the figure we can see that for this particular RA code $z_1 = x_1$, $z_2 = z_1 \oplus x_2$ and $z_3 = z_2 \oplus x_1$ and so on.

From Section 3.1 we know that the received signal at the receiver is given by

$$\mathbf{y} = a\sigma(\mathbf{z}) + \mathbf{n}, \quad (3.17)$$

where $\mathbf{n}$ represents unit-variance additive white Gaussian noise (AWGN) vectors, a the channel amplitude, and the function $\sigma(\cdot)$ maps 0 to 1 and 1 to -1. Therefore, $\mathbf{y}$,

the signal received at the destination, is real valued. We can convert each element in vector, $\mathbf{y}$, to a corresponding log-likelihood-ratio (LLR) as follows

$$\ell_i = \ln \left[\frac{\Pr(y_i|z_i = 0)}{\Pr(y_i|z_i = 1)} \right], \quad (3.18)$$

where

$$\Pr(y_i|z_i = 0) = \frac{1}{\sqrt{2\pi}} e^{-(y_i-a)^2/2}, \text{ and, } \Pr(y_i|z_i = 1) = \frac{1}{\sqrt{2\pi}} e^{-(y_i+a)^2/2}. \quad (3.19)$$

Therefore, substituting equation 3.19 into equation 3.18 and simplifying, we get

$$\ell_i = 2ay_i. \quad (3.20)$$

The small red circles at the bottom of Figure 3.4 represent these LLR values.

Let the repetition factor be q , the top variable nodes be represented by x , the middle check nodes be represented by c , and the bottom variable nodes be represented by z . According to sum-product algorithm there are four possible messages on the edges of the graph as shown in Figure 3.4, namely $m_{z \rightarrow c}$, message from variable nodes z to check nodes c , $m_{x \rightarrow c}$, message from variable nodes x to check nodes c , $m_{c \rightarrow z}$, message from check nodes c to variable nodes z , and $m_{c \rightarrow x}$, message from check nodes c to variable nodes x . Let m_{ℓ_z} be the message corresponding to channel LLR corresponding to node z as shown in Figure 3.4. The decoder algorithm will take the channel LLRs, ℓ , as an input and decode the information bits as follows.

1. Initialize all messages on all edges to zero.

2. Update $m_{z \rightarrow c}$ for all the edges between nodes z and nodes c as follows

$$m_{z \rightarrow c} = \begin{cases} m_{\ell_z} & \text{if } z = z_{qn} \\ m_{\ell_z} + m_{c' \rightarrow z} & \text{o/w, where } (c', z) \text{ is an edge and } c' \neq c \end{cases} .$$

3. Update $m_{x \rightarrow c}$ for all the edges between nodes x and nodes c as follows

$$m_{x \rightarrow c} = \sum_{c'} m_{c' \rightarrow x} \text{ where } (c', x) \text{ is an edge and } c' \neq c .$$

4. Update $m_{c \rightarrow z}$ for all the edges between nodes c and nodes z as follows

$$m_{c \rightarrow z} = \begin{cases} m_{x \rightarrow c} & \text{if } c = c_1 \text{ and } (x, c) \text{ is an edge} \\ 2 \tanh^{-1} [\tanh(m_{x \rightarrow c}/2) \tanh(m_{z' \rightarrow c}/2)] & \text{o/w, where } z' \neq z \end{cases} .$$

5. Update $m_{c \rightarrow x}$ for all the edges between nodes c and nodes x as follows

$$m_{c \rightarrow x} = \begin{cases} m_{z \rightarrow c} & \text{if } c = c_1 \text{ and } (z, c) \text{ is an edge} \\ 2 \tanh^{-1} [\tanh(m_{z \rightarrow c}/2) \tanh(m_{z' \rightarrow c}/2)] & \text{o/w, where } z' \neq z \end{cases} .$$

6. Repeat step 1 to 4 until K iterations or until all parity check nodes are satisfied.

7. If after K iterations the parity check nodes are not satisfied return decoding failure.

8. If parity check nodes are all satisfied, calculate the sum of all incoming messages to nodes x as follows

$$s(x) = \sum_c m_{c \rightarrow x}.$$

The bits are then estimated (decoded) from this sum as shown below

$$\hat{\mathbf{x}} = \begin{cases} 0 & \text{if } s(x) \geq 0 \\ 1 & \text{Otherwise} \end{cases}.$$

3.2.1.3 PSRA Codes

Although RA codes are powerful and simple error correcting codes, they are not flexible in terms of code rate. The only way to adjust the rate of an RA code is through changing the repetition factor and only repetition factors greater than or equal to 3 are proved to perform well as shown in [47]. However, code rates for repetition factors above 4 are considered very low and require a lot of transmission power and are highly inefficient. Therefore, for some wireless applications where the channel conditions change relatively fast more flexible code rates are needed. To overcome this problem the use of puncturing has been proposed in a number of papers, e.g., [17, 53, 54]. Since it is known that for better performance punctured codes must be systematic, we will describe the Punctured Systematic RA (PSRA) codes in this section.

The encoder of a PSRA code performs like the RA encoder described in section 3.2.1.1 with the following steps added to the end.

1. Puncture the RA encoded sequence, $\mathbf{z}$, (achieved at step 4 of the encoder in section 3.2.1.1) by randomly selecting M elements where $M \leq N$ as follows.

$$z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8 z_9 \cdots z_N$$

$$\downarrow \quad \downarrow \text{ RAND } \downarrow \quad \downarrow$$

$$u_1 \quad u_2 \quad \quad \quad u_3 \cdots u_M$$

2. Create the new PSRA encoded sequence by concatenating the original n information bit sequence, $\mathbf{x}$, with the punctured sequence $\mathbf{u}$.

$$x_1 x_2 x_3 \cdots x_n u_1 u_2 u_3 \cdots u_M$$

From this algorithm it is clear that by selecting the number of bits to be punctured at step 1 the code rate can be changed easily. For example if we want to achieve a code rate of $2/5$, then $M = N/2$ or only one of every two bits from $\mathbf{z}$ is selected at random in step 1.

The decoder for a PSRA code will also perform similarly to the iterative decoder of RA codes described in section 3.2.1.2. Since, instead of N bits, $n + M$ bits are received, there would be $n + M$ channel LLRs where the first n correspond to the systematic bits and the last M correspond to coded parity bits. Moreover, for the punctured bits we can assume the channel LLRs are zero. Figure 3.5 shows

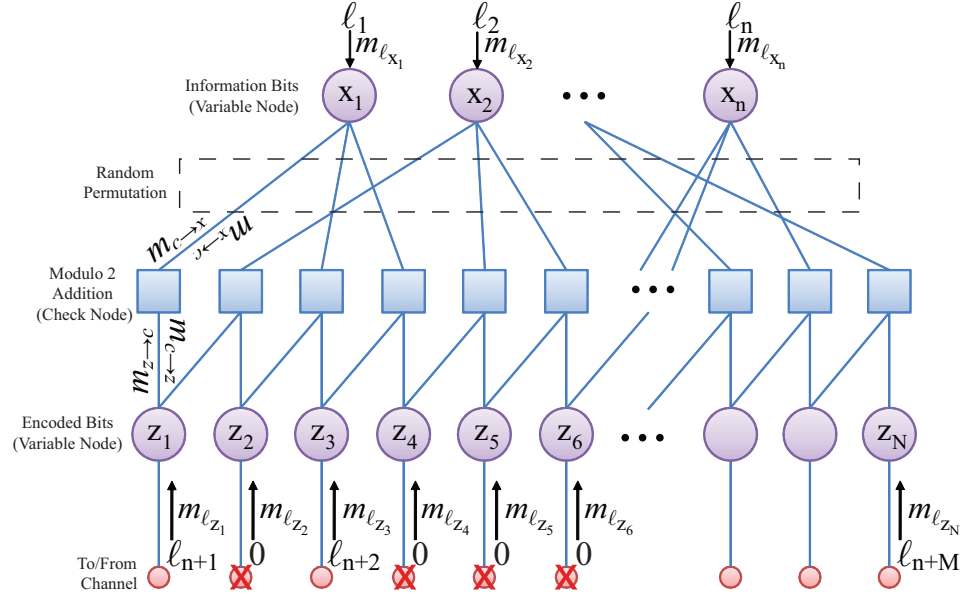


Figure 3.5: The factor graph representation of PSRA codes.

the factor graph of a PSRA code. This factor graph is exactly similar to the factor graph of regular RA code with the exception of channel LLRs. In Figure 3.5, channel LLRs are available at the top variable nodes, $\mathbf{x}$, because the code is systematic. Also, zero is used for z_2 , and z_4 through z_6 since those parity bits were punctured by the encoder and were not transmitted.

Let m_{ℓ_z} be the message corresponding to the channel LLR corresponding to node z . Moreover, if node z was punctured by the encoder, then $m_{\ell_z} = 0$ as shown in Figure 3.5. Similarly, m_{ℓ_x} be the message corresponding to the channel LLR corresponding to node x as shown in Figure 3.5. The decoder algorithm will

now take as its input $n + M$ channel LLRs and performs similar to the algorithm presented in section 3.2.1.2 with the following modifications.

1. At step 2, $m_{x \rightarrow c}$ is updated with addition of the channel LLR.

$$m_{x \rightarrow c} = \sum_{c'} m_{c' \rightarrow x} + m_{\ell_x} \text{ where } (c', x) \text{ is an edge and } c' \neq c .$$

2. At step 7, when the sum of incoming message to node x is calculated the channel LLR is also added.

$$s(x) = \sum_c m_{c \rightarrow x} + m_{\ell_x} .$$

Both the RA codes and PSRA codes have a low complexity encoder and decoder in the order of $O(n)$. They are also very powerful error correcting codes and perform very well at low SNRs. PSRA codes are also very flexible in terms of code rate because of puncturing.

3.2.2 Low-Density Parity-Check Codes

Low-density parity-check (LDPC) codes and a corresponding iterative decoding algorithm like the one described in the previous section for RA codes, were first introduced by Gallager [49, 50] in 1960s. However, because computers of the time could not simulate the performance of these codes, for the next decades LDPC codes were forgotten until their rediscovery through the work of McKay and Neal [55, 56]

in the late 1990s. LDPC codes had numerous advantages over most of the previously discovered codes such as block codes and trellis codes. First, they were constructed in a random manner; second, they had a decoding algorithm whose complexity was linear in the block length of the code. Finally, when combined with their spectacular performance, this made LDPC codes a compelling class of codes for many applications. Therefore, a lot of research has been done on LDPC codes after their rediscovery in 1996.

Because of its popularity, there has been a lot of different work on LDPC codes and as the result the their performance [57], the convergence criteria for iterative decoding algorithm [18, 57], and their use in Slepian-Wolf compression [41] have been studied in detail. These criteria makes LDPC codes suitable for both channel coding (error-correcting codes) and Slepian-Wolf compression used throughout this thesis. Therefore, in this section we will describe the LDPC encoder and decoder. Then, we will describe their use for Slepian-Wolf compression as described in [41]. Finally, we will take a look at convergence criteria of LDPC codes based on the mean value of channel log-likelihood-ratios (LLRs).

3.2.2.1 Encoder

Since LDPC codes are very similar to linear block codes, we will first review basics of linear block codes. An (n, k) block code C is a mapping from k -bit information

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

Figure 3.6: Parity-check matrix of an LDPC code.

vector $\mathbf{x}$, to a length- n codeword vector $\mathbf{z}$. The code C is a k -dimensional subspace of an n -dimensional binary vector space V_n since it is linear. Therefore, the code can also be viewed as a mapping of k -space to n -space by a $k \times n$ generator matrix $\mathbf{G}$, where $\mathbf{z}^T = \mathbf{x}^T \mathbf{G}$. The rows of $\mathbf{G}$ create a basis of the code subspace. The dual space, C^* consists of all the vectors in V_n orthogonal to C , which means that for all $\mathbf{z} \in C$ and all $\mathbf{d} \in C^*$, $\mathbf{z} \cdot \mathbf{d} = 0$. Let the rows of an $(n - k) \times n$ parity-check matrix $\mathbf{H}$ forms a basis for C^* . Therefore, it follows that for all $\mathbf{z} \in C$, $\mathbf{H}\mathbf{z} = 0$. Moreover, a code is completely defined by either $\mathbf{G}$ or $\mathbf{H}$ since they can be reduced to $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$ and $\mathbf{H} = [(-\mathbf{P})^T \mid \mathbf{I}_{n-k}]$ respectively and converted back and forth.

In its simplest form, an LDPC code is a linear code with a parity-check matrix that is sparse, which means it has a small number of non zero entries. Gallager [49, 50] proposed randomly placing 1's and 0's in a $m \times n$, $m = n - k$, parity-check matrix $\mathbf{H}$ such that each row of $\mathbf{H}$ has d_c 1's and each column of $\mathbf{H}$ has d_v 1's. Figure 3.6 shows a $m = 4 \times n = 8$ parity-check matrix where $d_v = 2$ and $d_c = 4$. Therefore, this code represents a (8,4) linear block code. In general, codes of this

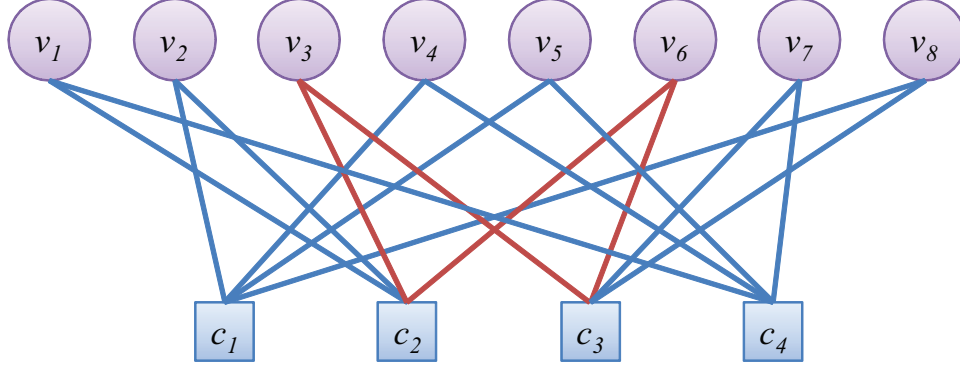


Figure 3.7: The factor graph representation of an LDPC codes.

form are referred to as regular (d_v, d_c) LDPC codes of length n which means all the rows and columns have the same number of 1's. Although in the example in Figure 3.6 the matrix is not sparse because of lack of space (i.e. $d_c \ll n$ and $d_v \ll m$ do not hold), the fraction of 1's given by $\frac{md_c}{mn} = \frac{d_c}{n}$ approaches zero as the block length gets large and the matrix will become sparse.

Just like RA codes LDPC codes can be represented with factor graphs. Figure 3.7 shows the factor graph (also known as Tanner Graph) representation of the parity-check matrix in Figure 3.6. The purple circular nodes at the top are variable nodes that represent the columns of the parity-check matrix and the blue square nodes at the bottom are check nodes and represent the rows of the parity-check matrix. If there is a 1 at position h_{ij} in the parity-check matrix, where i indicates the row and j the column, there will be an edge between v_j variable node and c_i check node. For example, since there is a 1 at h_{14} there is an edge connecting v_4 to

c_1 . The degree of all variable nodes (i.e. number of edges connected to the variable nodes) is $d_v = 2$ and the degree of all check nodes is $d_c = 4$, which makes the code regular.

The encoder of the LDPC codes typically creates the parity-check matrix by placing 1's at random positions in the matrix subject to the constraint that there are d_v ones at each column and d_c ones at each row. A similar equivalent approach would involve randomly permuting edges between variable nodes and check nodes in the factor graph subject to the constraint that the degree of all variable nodes are d_v and the degree of all check nodes are d_c . This constraint is not enough because this random permutation might connect the same variable node to the same check node through more than one edge. For example, in Figure 3.7 changing the edge (v_1, c_4) to (v_1, c_2) and the edge (v_2, c_2) to (v_2, c_4) would constitute a valid random permutation. However, in this permutation variable node v_1 is connected to check node c_2 with two edges and this will make the code irregular. Therefore, another constraint that must be satisfied is that each variable node is connected to the same check node with only one edge. Another constraint that is usually satisfied is avoiding short cycles. In Figure 3.7, the path $v_3 \rightarrow c_2 \rightarrow v_6 \rightarrow c_3 \rightarrow v_3$ (the red path), is an example of such a cycle. It is usually preferred that these short cycles are avoided for better decoder performance.

Having explained all the constraints we now provide an algorithm for encoding

LDPC codes bellow.

1. First a parity-check matrix is generated by randomly placing ones in the matrix such that there are d_v 1's at each column and d_c 1's at each row.
2. The equivalent factor graph is generated and adjustments to the edges are made such that there are not more than one edge between each variable and check node pair. Moreover, the adjusted edges might be managed in such a way that there will be no short cycles for better performance at the decoder.
3. The resulting factor graph is converted back to corresponding parity-check matrix $\mathbf{H}$.
4. Using Gaussian elimination the parity-check matrix is converted to the form $\mathbf{H} = [(-\mathbf{P})^T \mid \mathbf{I}_{n-k}]$.
5. The parity-check matrix is then converted to the equivalent generator matrix $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$.
6. The generator matrix is then used to encode the information bit vector $\mathbf{x}$, as follow $\mathbf{z}^T = \mathbf{x}^T \mathbf{G}$, where $\mathbf{z}$ is a vector of the encoded bits.

At the first glance the encoder algorithm explained above performs in $O(N^2)$ or quadratic complexity with respect to the code length. The main reason is conversion of parity-check matrix to generator matrix. However, there has been number of

papers on more efficient encoding schemes. If we assume the code is systematic then it is possible to encode directly using the parity check matrix. Suppose the encoded codeword has the structure $\mathbf{z} = [\mathbf{z}_u \mathbf{z}_p]$, where $\mathbf{x} = \mathbf{z}_u$ and $\mathbf{z}_p$ is the vector of parity bits. Similarly the parity-check matrix can be split into $\mathbf{H} = [\mathbf{H}_u | \mathbf{H}_p]$ such that

$$\mathbf{H}_p \mathbf{x}_p^T = \mathbf{H}_u \mathbf{x}_u^T \Rightarrow \mathbf{x}_p^T = \mathbf{H}_p^{-1} \mathbf{H}_u \mathbf{x}_u^T \quad (3.21)$$

The authors of [59] proposed an efficient way of achieving this task and showed that using such a method the encoder complexity can be reduced to $O(n)$ or linear time. Moreover, for simulation purposes, since the all-zero codeword is part of any LDPC code, there is no need for an encoder (i.e. in simulations we always use the all-zero codeword without encoding).

3.2.2.2 Decoder

In this section we explain the LDPC iterative decoder algorithm based on factor graphs and sum-product algorithm. Just like RA codes we need to calculate the channel LLRs. This calculation is similar the one explained in section 3.2.1.2 which derived equation 3.20, and therefore $\ell_i = 2ay_i$. Figure 3.8, shows these LLRs as inputs to variable nodes as well as the two messages passed along each edge, namely $m_{v \rightarrow c}$, message from variable nodes to check nodes, and $m_{c \rightarrow v}$.

Let m_{ℓ_v} be the message corresponding to channel LLR corresponding to node

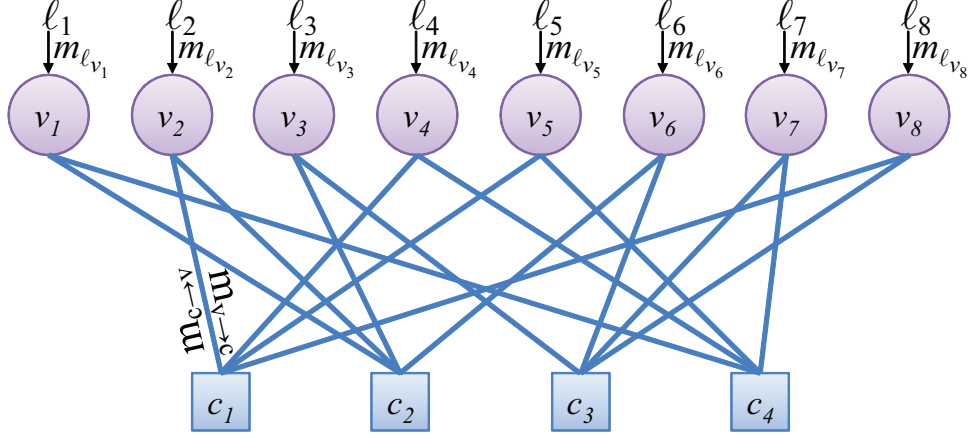


Figure 3.8: The factor graph representation of an LDPC codes.

v as shown in Figure 3.8. The decoder algorithm takes the channel LLRs as input and performs as follows.

1. Initialize all messages on all edges to zero and all $m_{v \rightarrow c} = m_{\ell_v}$.
2. Calculate $m_{c \rightarrow v}$ for all edges as

$$m_{c \rightarrow v} = 2 \tanh^{-1} \left(\prod_{v'} \tanh(m_{v' \rightarrow c}/2) \right) \text{ where } (v', c) \text{ is an edge and } v' \neq v$$

3. Calculate $m_{v \rightarrow c}$ for all edges as

$$m_{v \rightarrow c} = \sum_C m_{c' \rightarrow v} + m_{\ell_v} \text{ where } (c', v) \text{ is an edge and } c' \neq c$$

4. Repeat step 2 to 3 until K iterations or until all parity check nodes are satisfied.

5. If after K iterations the parity check nodes are not satisfied return decoding failure.
6. If parity check nodes are all satisfied, calculate the sum of all incoming messages to nodes v as follows

$$s(v) = \sum_c m_{c \rightarrow v} + m_{\ell_v}.$$

The bits are then estimated (decoded) from this sum as shown below

$$\hat{\mathbf{v}} = \begin{cases} 0 & \text{if } s(v) \geq 0 \\ 1 & \text{Otherwise} \end{cases}.$$

7. An estimate of the original information bits, $\hat{\mathbf{x}}$, is recovered by extracting the systematic parts of $\hat{\mathbf{v}}$.

The iterative decoder algorithm described above perform in $O(n)$ which makes it suitable for many applications. In the next section we will describe how LDPC codes can be used in Slepian-Wolf compression to compress and decompress correlated information.

3.2.2.3 LDPC Slepian-Wolf Compression

It is possible to use the LDPC codes for compression and decompression of binary correlated sources as explained in [41]. In this section we will describe this

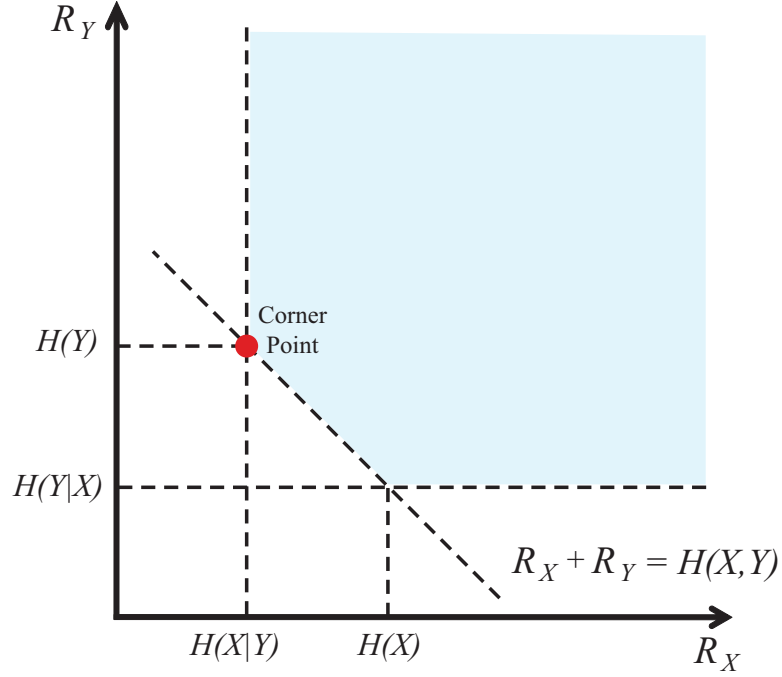


Figure 3.9: The corner point of Slepian-Wolf rate region ($\mathbf{x}$ compressed fully).

process in details. Let's assume that there are two sources, $\mathbf{x} = [x_1, x_2, \dots, x_n]$ and $\mathbf{y} = [y_1, y_2, \dots, y_n]$, where x_i 's and y_i 's are binary independent and identically-distributed (i.i.d.) equiprobable random variables. The correlation is defined as $\Pr[x_i \neq y_i] = p < 0.5$. We try to compress at the corner point of Slepian-Wolf rate region, the red dot in Figure 3.9. This means $\mathbf{x}$ is compressed fully while $\mathbf{y}$ is not compressed at all. Therefore, the rate of $\mathbf{y}$ is its entropy and the number of bits transmitted is $nR_Y = nH(y_i) = n$ bits. The number of bits $\mathbf{x}$ is compressed to is $nR_X \geq nH(x_i|y_i) = nH(p) = n[-p \log_2 p - (1-p) \log_2 (1-p)]$.

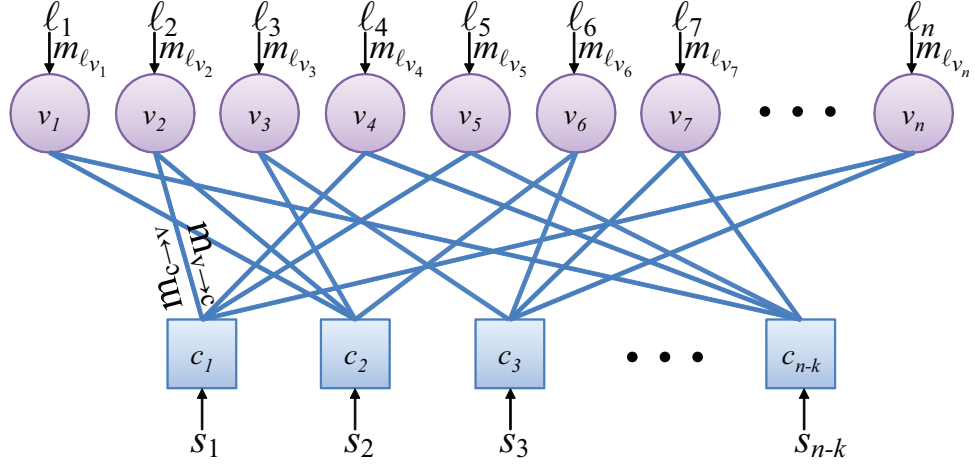


Figure 3.10: The factor graph used in LDPC compression and decompression algorithms.

In the compression step first a valid LDPC parity-check matrix $\mathbf{H}$ is generated similarly with the techniques discussed in section 3.2.2.1. The syndrome bits are then calculated as $\mathbf{s} = \mathbf{x}\mathbf{H}^T$. Therefore, the length- n information bits are compressed to length- $(n - k)$ syndrome bits. Alternatively, this compression process can be viewed as binary addition of all the variable nodes that are connected to the same check node in the factor graph representation.

To decompress the syndrome bits, $\mathbf{y}$ is required to calculate the LLR of variable nodes similar to the channel LLRs calculated in section 3.2.2.2 for LDPC decoder. This time the LLRs are calculated as

$$\ell_i = \log \frac{\Pr[x_i = 0|y_i]}{\Pr[x_i = 1|y_i]} = (1 - 2y_i) \log \frac{1-p}{p}. \quad (3.22)$$

Let the function $L(v) = \ell$ return the channel LLR corresponding to node v which is a simplified notation for $L(v_i) = \ell_i$. Also, let $S(c)$ be the function that returns the syndrome bit corresponding to a check node, which is a simplified notation for $S(c_j) = s_j$ as shown in Figure 3.10. The decompression algorithm takes the LLRs ℓ and the compressed syndrome bits $\mathbf{s}$ as input and performs exactly like the LDPC iterative decoder algorithm described in section 3.2.2.2 with the following modifications.

1. At step 2, $m_{c \rightarrow v}$ is calculate for all edges as

$$m_{c \rightarrow v} = 2 \tanh^{-1} \left([1 - 2S(c)] \prod_v \tanh(m_{v' \rightarrow c}/2) \right), (v', c) \text{ is an edge, } v' \neq v$$

2. At step 6, the algorithm ends since $\hat{\mathbf{x}} = \hat{\mathbf{v}}$ and step 7 is not executed.

The above compressor and decompressor algorithms both run in $O(n)$ just like the regular LDPC encoder and decoder. Another property of this compression and decompression algorithm is that there are decompression errors associated with the process. In other words the decompression is not guaranteed to be successful always. We will present some simulations demonstrating this property in the next chapter.

3.2.2.4 LDPC Convergence Requirements: EXIT Charts

It is extremely useful to know what the requirements would be for successful decoding of LDPC codes under iterative decoding algorithm. For example, the encoder can determine, given the current channel condition, whether the decoder is able to decode the message in advance and adjusts the rate accordingly. As a result, there has been a number of studies on the convergence of the iterative decoding algorithms in the literature. Generally, there are two methods for studying the convergence, one based on density evolution [58], and one based on extrinsic information transfer (EXIT) charts [18]. In this section, we use EXIT charts to analyze the convergence requirements of LDPC codes, based on channel LLR's mean, with a techniques used in [18]. Although, density evolution is exact, EXIT chart analysis can be optimized using linear programming.

In order to analyze the convergence of the iterative decoder, we must first study the input to the decoder which is a set of channel LLRs. Let's assume that the all zero codeword $\mathbf{z}$ was transmitted. Therefore, we have $\mathbf{y} = \sigma(\mathbf{z}) + \mathbf{n}$ where $\sigma(\cdot)$ maps 0 to +1 and 1 to -1. For a single symbol then $y = 1 + n$ with the conditional pdf

$$p(y|z=0) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-(y-1)^2/2\sigma_n^2}$$

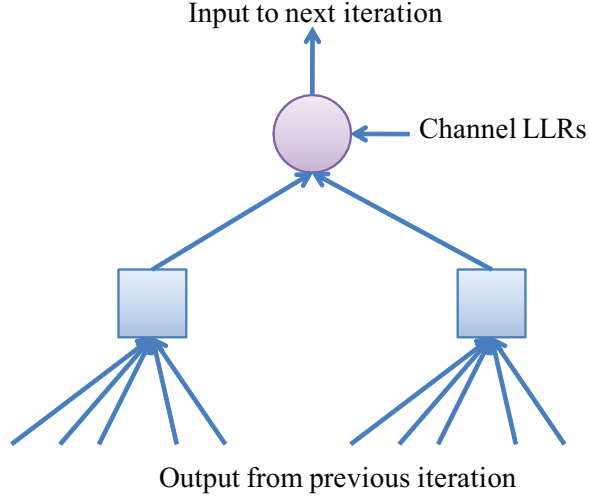


Figure 3.11: (3,6) regular LDPC code tree.

where σ_n^2 is the variance of Gaussian noise. The LLR value ℓ is

$$\ell = \ln \frac{p(y|z=0)}{p(y|z=1)} = \frac{2}{\sigma_n^2} y = \frac{2}{\sigma_n^2} (1+n) \quad (3.23)$$

which is a Gaussian random variable. The mean and variance of ℓ are given by $m_\ell = 2/\sigma_n^2$ and $\sigma_\ell^2 = 4/\sigma_n^2$ respectively. Since $\sigma_\ell^2 = 2m_\ell$ then the channel LLRs are Gaussian symmetric. In [58], it was shown that if the channel LLRs have a symmetric pdf, using iterative decoding, the pdf of the message LLR in the decoder will remain symmetric.

Using this property of channel LLRs and the iterative decoding algorithm, we can study the convergence by simulating one iteration of decoding algorithm as follows. Assume we have a (3,6) regular LDPC code, which means $d_v = 3$ and

$d_c = 6$. A single iteration can then be simulated on a tree of depth-one taken from the factor graph and shown in Figure 3.11. In this tree, since the code is (3,6), each variable node has a 3 edges and one input from channel LLR. Each check node also has 6 edges and uses 5 input edges to calculate the output to the variable node. The variable node also needs messages from two check nodes to calculate its output.

The tree in Figure 3.11, can be treated as a function with 10 random variables as inputs and 1 random variable as output, and is the simplest block in a (3,6) regular LDPC code that is repeated many times. Let this function be represented by $f(\mathbf{x})$ where $\mathbf{x}$ is a vector of 10 random variables. Since for a symmetric Gaussian density $\sigma^2 = 2m$, only the mean or variance is sufficient to describe the random variable; we choose the mean. Therefore, each element in vector x is a random variable with pdf $\mathcal{N}(m_{in}, 2m_{in})$. The function $f(\mathbf{x})$ will output a symmetric Gaussian random variable with pdf $\mathcal{N}(m_{out}, 2m_{out})$. To find the mapping between the input pdf and the output pdf, we run the following algorithm subject to the input mean m_{in} , on the tree in Figure 3.11 for a large number of times and calculate m_{out} by averaging the outputs.

1. Generate 10 symmetric Gaussian random variables with mean $m = m_{in}$ (i.e. Gaussian variable with $\mathcal{N}(m_{in}, 2m_{in})$).
2. These random variables are inputted into the tree and the output is calculated

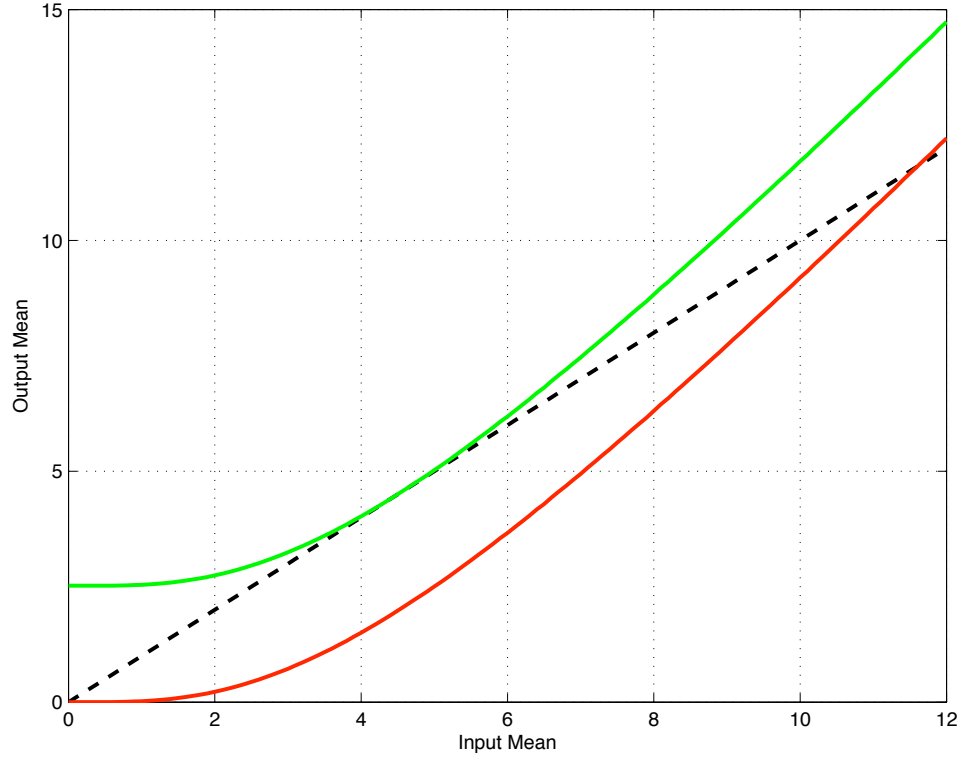


Figure 3.12: EXIT chart of (3,6) regular LDPC code.

based on the decoding rules described in section 3.2.2.2.

3. Repeat step 1 and 2 until large number of iterations have been reached.
4. After a large number of iterations have been reached calculate the mean of all the obtained outputs m_{out} .
5. Start from step one with a different m_{in} .

The result of this algorithm is shown in Figure 3.12. The x-axis represents

the input mean while the y-axis represents the output mean. The green curve at the bottom represents the function $f(\mathbf{x})$ which is obtained by running the above algorithm. Since in the real LDPC iterative decoder after each iteration the output mean m_{out} would become the input mean m_{in} in the next iteration, in order for the algorithm to converge the curve must be above the line $m_{in} = m_{out}$ at all times, otherwise, decoding failure will occur. Let the function $g(\mathbf{x}, m_\ell) = f(\mathbf{x}) + m_\ell$ represent the addition of the mean of channel LLR m_ℓ to the tree in Figure 3.11. Therefore, the iterative decoding algorithm converges when the mean of the channel LLR is big enough such that the green curve at the bottom of the Figure 3.12 is moved up to a tangent position of line $m_{in} = m_{out}$.

3.3 Summary

In this chapter we first explained the system model that will be used in the next two chapters. We also explained two different codes namely RA, and LDPC codes. For each code we explained the encoding and the iterative decoding algorithms. We also described how LDPC codes can be used for compression and decompression of correlated binary sources. The convergence of LDPC codes using EXIT charts were also discussed. In the next two chapters we will use these models and coding methods to increase the efficiency of WSNs through two different novel techniques.

4 Low-Complexity Cooperation Using Correlated Sources

As was mentioned in previous chapters, phenomena under measurement by WSNs are often spatially correlated. Therefore, Slepian-Wolf compression can be applied to compress the number of bits transmitted by two correlated sources, thereby reducing the transmission power of the system. Moreover, because of the spatial distribution of sensor nodes, cooperative diversity can be used to increase the efficiency of transmission. Therefore, both cooperation and compression of correlated data are beneficial to WSNs.

As was explained in detail in Chapter 2, in wireless networks, it is best if the system has a high diversity order, since if one link fails because of deep fading, there will be other links with possibly a better fading and successful transmission. In WSNs, diversity can be achieved by relaying transmissions through neighbouring sensor nodes towards the data sink. Therefore, one of the main issues that will be discussed in this chapter is the effect of compression of correlated sources on the

diversity order of a low-complexity WSN system.

To answer this question, we first consider the effects of Slepian-Wolf compression on diversity order of a cooperative network with a single source and a single destination, along with relays and correlated sources. We will apply the Slepian-Wolf compression and cooperation together (explained in Chapters 2 and 3) to study the effects on the diversity order of the system, system performance and transmission rates. Our main assumption is that there exist constraints on the nodes' computational power. We will show that in spite of the simplicity and adaptability of this scenario, excellent and robust performance can be achieved. Although these results are applicable to general wireless networks, they are perfectly suited for networks with low complexity nodes, such as WSNs.

The rest of this chapter is organized as follows. We will explain the system under study in details in section 4.1, using the relay models and coding techniques described in Chapter 3. We will then present our theoretical results in section 4.2, where we will show that the diversity order of the system under study will drop asymptotically to the minimum of the number of relays and number of correlated sources. In section 4.3 we will present our simulation results and show that diversity orders of greater than the theoretical asymptotic bound can be achieved at lower SNRs.

4.1 System Setup

In this section, we describe the system to be studied and set up the problem to be investigated. The relay models, along with some of the coding techniques described in Chapter 3, will be used to set up the system. A simple example will then be presented to show how different components of the system work together. We will then describe our contribution by formulating some of the questions we will answer in the rest of the chapter.

The relay model that we will use in this chapter is based on the multi-relay multi-correlated-source model described in section 3.1.2 in Chapter 3. Therefore, our system consists of a source and a destination along with r relays and q correlated sources. The source measures a phenomenon and compresses the measurements using the LDPC compression technique described in Chapter 3, encodes the compressed bits using PSRA codes and broadcasts to the r relays as well as to the destination. Each relay makes a hard decision on the received bits and re-encodes them using PSRA codes and transmits to the destination. Each correlated source takes its own measurements, and encodes them using PSRA codes and transmits to the destination.

Although in the system described above we specify the coding methods (i.e. LDPC codes for compression and PSRA codes as error correcting codes for channel

coding), in general we can use any asymptotically good coding scheme, and the theoretical results that will be presented in the next section are independent from the coding method used. However, since the system under study is a WSN, and we need low-complexity coding with good performance and flexibility in terms of code rate, we choose LDPC and PSRA codes for our simulations as an example of practical implementation.

4.1.1 An Example

We now provide an example of a system with $r = 1$ and $q = 1$ (i.e. one relay and one correlated source) in detail. The example provides just an outline of the processes at the source, relay, correlated source, and destination and does not work in practice because of its small size (i.e. the small size of the information bits, encoder, decoder, etc.). We divide the procedure into five parts: compression at the source, channel encoder at the source and correlated source, processing and retransmission at the relay, decoding source's compressed bits at the destination, and finally decompression of source's information bits.

Compression at the source. Assume the number of information bits at the source and correlated source is $n = 8$. Also assume the information bits at source

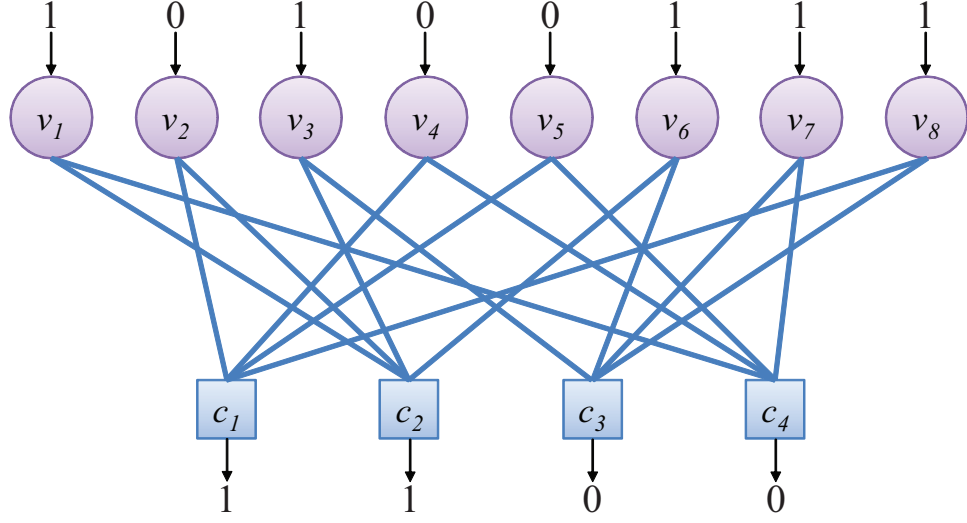


Figure 4.1: Example of compression by a (2,4) regular LDPC code.

and correlated source are given by

$$\mathbf{x}^{(S)} = 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1, \text{ and}$$

$$\mathbf{x}^{(CS)} = 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1,$$

respectively. Therefore, the source and correlated source differ only at the first bit position on the left. The source compresses its information bits using a (2,4) regular LDPC code, shown in Figure 4.1. The resulting syndrome bits are given by

$$\mathbf{s} = 1\ 1\ 0\ 0.$$

Channel encoder at the source and correlated source. The resulting length-4 syndrome bits are encoded using PSRA codes with repetition factor of

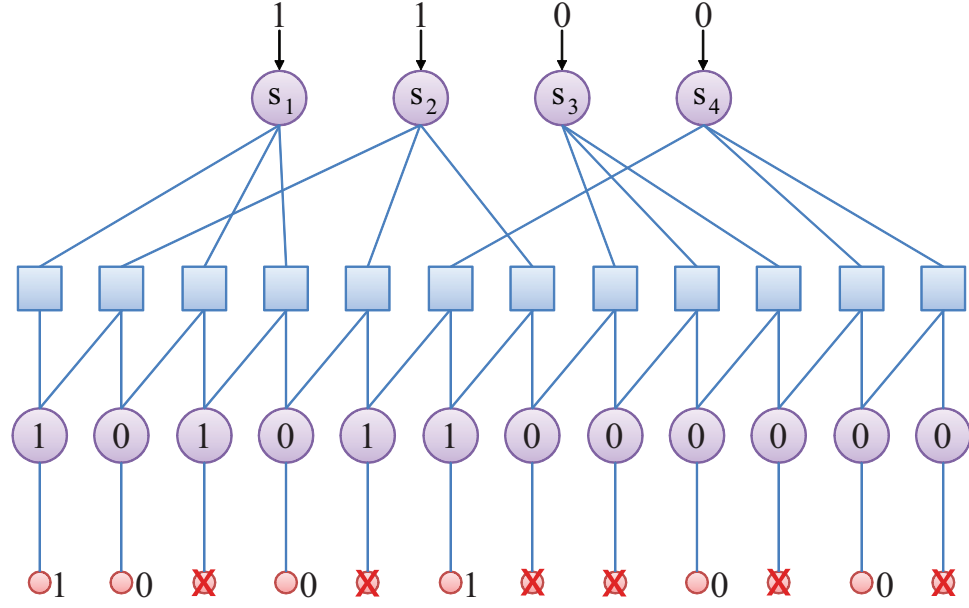


Figure 4.2: Example of PSRA code.

three where half of the parity bits are punctured as shown by the red cross over the red variable nodes in Figure 4.2. Therefore, the resulting transmission bits from the source are

$$\sigma^{-1}(\mathbf{z}^{(S)}) = 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0,$$

where the first four bits on the left are the systematic bits and the last six bits are the parity bits. Similarly, the 8-bit information at the correlated source are encoded using the PSRA codes to create the transmission sequence $\mathbf{z}^{(CS)}$. The transmission sequence, $\mathbf{z}^{(S)}$, is broadcast to the relay and the destination by applying Equations (3.1,3.2) and the transmission sequence, $\mathbf{z}^{(CS)}$, is transmitted to the destination by

applying Equation (3.3).

processing and retransmission at the relay. The relay makes hard decisions on the received signal. Suppose the result is,

$$\mathbf{x}^{(R)} = 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1,$$

that is, there is only one bit error in transmission to relay, namely the first bits from the right. The 10-bit $\mathbf{x}^{(R)}$ is then encoded with PSRA codes to create the transmission sequence $\mathbf{z}^{(R)}$, which is transmitted to the destination by applying equation (3.5).

decoding source's compressed bits at the destination. At the destination, three signals are received: $\mathbf{y}^{(SD)}$, $\mathbf{y}^{(RD)}$ and $\mathbf{y}^{(CSD)}$, from the source, the relay and the correlated source respectively. The signal from the relay, $\mathbf{y}^{(RD)}$, is first decoded using the PSRA decoder and the channel LLRs are calculated as described in Equation (3.20). Let the decoded bits be represented by $\hat{\mathbf{x}}^{(R)}$, which is an estimate of $\mathbf{x}^{(R)}$. These estimated bits along with the signal from the source $\mathbf{y}^{(SD)}$, are used jointly to decode the syndrome bits $\mathbf{s}$. To do this, the channel LLRs inputed to the PSRA decoder are calculated as a combination of LLR from the relay and the LLR from the source. The LLR from the source, $\ell^{(S)}$, is calculated using the equation (3.20) in chapter 3. That is,

$$\ell_i^{(S)} = \ln \frac{\Pr[y_i^{(SD)} \mid z_i^{(S)} = +1]}{\Pr[y_i^{(SD)} \mid z_i^{(S)} = -1]} = 2a_{SD}y_i^{(SD)} \quad (4.1)$$

The LLR from the relay, however, is calculated as

$$\ell_i^{(R)} = \ln \frac{\Pr[\hat{x}_i^{(R)} \mid z_i^{(S)} = +1]}{\Pr[\hat{x}_i^{(R)} \mid z_i^{(S)} = -1]} = (1 - 2\hat{x}_i^{(R)}) \log \left[\frac{1 - p_{HD}}{p_{HD}} \right] \quad (4.2)$$

where p_{HD} is the probability that the hard decision at the relay was incorrect.

Therefore, p_{HD} can be calculated as

$$p_{HD} = \frac{1}{2} \text{erfc}(\sqrt{a_{SR}}) \quad (4.3)$$

where a_{SR} is the channel amplitude or the average SNR. The LLR that is input top the PSRA decoder is given by

$$\ell_i = \ell_i^{(S)} + \ell_i^{(R)}, \quad (4.4)$$

and the result of this decoding process is an estimate of the syndrome bits sent by the source $\hat{\mathbf{s}}$.

In general, if we have r relays, first the signal from each relay is decoded using a PSRA decoder to get an estimate of hard decision at the j th relay, $\hat{\mathbf{x}}^{(R,j)}$. Then, to decode the syndrome bits, $\mathbf{s}$, sent by the source, the joint LLR from r relays and the S-D link is calculated as

$$\ell_i = \ell_i^{(S)} + \sum_{j=1}^r \ell_i^{(R,j)}, \quad (4.5)$$

and is input to the PSRA decoder.

decompression of source's information bits. The signal from the correlated source $\mathbf{y}^{(CSD)}$ is also decoded using PSRA decoder just like the signal from the relay,

$\mathbf{y}^{(RD)}$. The result of this decoding process is an estimate of the information bits at the correlated source given by $\hat{\mathbf{x}}^{(CS)}$. Using this estimate, the decompression of the earlier estimated syndrome bits is achieved by running the decompression algorithm described in section 3.2.2.3, with LLR

$$\ell_i^{(CS)} = \log \frac{\Pr[x_i^{(S)} = 0 \mid \hat{x}_i^{(CS)}]}{\Pr[x_i^{(S)} = 1 \mid \hat{x}_i^{(CS)}]} = (1 - 2\hat{x}_i^{(CS)}) \log \frac{1-p}{p}, \quad (4.6)$$

where p is the correlated factor described in Chapter 3. The result of the decompression is an estimate of the information bits at the source $\hat{\mathbf{x}}^{(S)}$. The communication process is successful when $\hat{\mathbf{x}}^{(S)} = \mathbf{x}^{(S)}$.

In general if there are q correlated sources, the signal from each correlated source is decoded individually. Therefore, there are q estimates, where the k th estimate is given by $\hat{\mathbf{x}}^{(CS,k)}$. The input LLR to the decompression algorithm is therefore calculated as sum of all the individual LLR, calculated separately using equation (4.6),

$$\ell_i = \sum_{k=0}^q \ell_i^{(CS,k)}. \quad (4.7)$$

4.1.2 Problem Setup

Although in that system we also specify coding methods (i.e. PSRA and LDPC codes) in general, we can use any coding technique for channel coding and compression. With these assumptions in mind, we will try to answer the following questions in the rest of this chapter.

1. Can the use of correlated sources increase the diversity order of the system?
2. Is it better to use correlated sources to increase the diversity than using relays?
3. How well does the process of compression and decompression perform in the presence of noise?

The most important question is question 1, since diversity is very important in wireless systems. Because the system under study is complicated, it is hard to obtain an exact theoretical result for diversity order of the system. Instead, we have to rely on asymptotic bond (i.e. as the SNR approaches infinity). Therefore, in the next section, we will try to obtain this asymptotic bond. Then in section 4.3, through simulation, we try to answer these questions.

4.2 Theoretical Result

In this section, we present our theoretical contributions. In particular we will show that the diversity order of the multi-relay multi-correlated-source system described in Section 4.1 will asymptotically drop to $\min(r + 1, q)$ (Theorem 1), where r is the number of relays and q is the number of correlated sources.

Although our theoretical results indicate that the diversity order will asymptotically drop to $\min(r + 1, q)$ our simulations show that the diversity order can be greater than $\min(r + 1, q)$ for $q > r + 1$, over SNR ranges of practical interest.

Therefore, using correlated sources we can compress source's information bits and reduce the amount of information that needs to be relayed by the system while increasing the diversity order of the system. The correlated sources are other sensor nodes in the network transmitting their own informations. Therefore, there is no increase in the system's total power. Finally we mention that, using correlated sources the system can outperform a system with no correlated source in terms of frame error rate (as will be shown in our simulations).

4.2.1 Key Assumptions and Definitions

We consider the multi-relay multi-correlated-source model given in Section 3.1.2. We assume for convenience that the average SNR $\bar{\gamma}$ on every link is the same (but as shown in [17], relaxing this assumption makes no difference to our results). As before, let r represent the number of relays in the system and q the number of correlated sources.

We define a system outage as the event during which the overall probability of frame error between the source and destination falls below some minimum. To describe asymptotic outage probabilities, designated P_{out} , we will be using the Θ order notation, where $g(x) = \Theta(f(x))$ means that there exists a constant c such that $\lim_{x \rightarrow \infty} f(x)/g(x) = c$.

For a single link in Rayleigh fading, and given a minimum SNR γ_{min} to avoid

system outage, it is easy to show that P_{out} is given by

$$P_{out} = \Pr(\gamma < \gamma_{min}) = 1 - e^{-\eta\gamma_{min}/\gamma}, \quad (4.8)$$

where η is a positive constant. Given the Taylor series expansion of e^x , it is easy to show that $1 - e^{-\eta\gamma_{min}/\gamma} = \Theta(\bar{\gamma}^{-1})$. Therefore in Rayleigh fading, a system with diversity order d has probability of system outage $P_{out} = \Theta(\bar{\gamma}^{-d})$.

When using error-correcting codes, we assume that there exists an SNR $\gamma > 0$ so that the code (and its decoding algorithm) is equal or superior to using no coding for every SNR greater than γ . In other words, for sufficiently high SNR, the use of the code does not result in a higher probability of error than using no code. Also, since we are using DISCUS as a method of compression at the source, there is a certain probability of error in decompression, P_{out_Decomp} , given the system parameters, p (probability of decorrelation) and frame size. In other words, if the source's compressed syndrome bits, and the correlated source's information bits are perfectly available at the destination, there is still some probability P_{out_Decomp} that we cannot retrieve the source's information bits successfully. We will demonstrate this effect through simulation in the next section. However, to derive the asymptotic theoretical results, we generally assume that $P_{out_Decomp} = \epsilon$, a small negligible number. In practice, this assumption can be implemented by increasing the frame size or adjusting the rate the compression code.

4.2.2 Main Theoretical Results

Our asymptotic theoretical result is expressed through the following theorem assuming serial decoding at the destination (i.e. procedure outlined in section 4.1:

Theorem 1 *The diversity order of the multi-relay, multi-correlated-source model with r relays and q correlated sources, described in section 4.1, will asymptotically drop to $\min(r + 1, q)$ for $0 \leq r, q < \infty$ as $SNR \rightarrow \infty$.*

Proof: Let there be $\gamma_{\circ}^{(SD)}$ such that whenever $\gamma^{(SD)} < \gamma_{\circ}^{(SD)}$ the S-D link will be in outage. Also, let there be $\gamma_{\circ}^{(SRD,i)}$ for any relay i such SRD link through relay i is in outage whenever $\gamma^{(SRD,i)} < \gamma_{\circ}^{(SRD,i)}$. Here the SRD link is the combination of SR link and RD link. Therefore we can consider SRD link as $\gamma^{(SRD,i)} = \min \gamma^{(SR,i)}, \gamma^{(RD,i)}$. Then using serial decoding, sufficient condition for system outage is that $\gamma^{(SD)} < \gamma_{\circ}^{(SD)}$ and $\gamma^{(SRD,i)} < \gamma_{\circ}^{(SRD,i)}$ for all i . We therefore have

$$P_{synd_out} = \Pr(\gamma^{(SD)} < \gamma_{\circ}^{(SD)}) \prod_{i=1}^r \Pr(\gamma^{(SRD,i)} < \gamma_{\circ}^{(SRD,i)}) \quad (4.9)$$

where P_{synd_out} is the probability that syndrome bits are not received at the destination. Since, by assumption, the average SNR for every channel is equal $\bar{\gamma}$, we have

$$\begin{aligned} P_{synd_out} &= (1 - e^{-\eta \gamma_{\circ}^{(SD)} / \bar{\gamma}}) \prod_{i=1}^r (1 - e^{-\eta \gamma_{\circ}^{(SRD,i)} / \bar{\gamma}}) \\ &= \Theta(\bar{\gamma}^{-1}) \Theta(\bar{\gamma}^{-r}) = \Theta(\bar{\gamma}^{-(r+1)}) \end{aligned}$$

Similarly, let there be $\gamma_{\circ}^{(CSD,j)}$ for any correlated source j such that whenever $\gamma^{(CSD,j)} < \gamma_{\circ}^{(CSD,j)}$ the CS-D link j will be in outage. Therefore the probability that the CS bits are not received at the destination is given by

$$P_{cs_out} = \prod_{j=1}^q \Pr(\gamma^{(CSD,j)} < \gamma_{\circ}^{(CSD,j)}) \quad (4.10)$$

where P_{cs_out} that correlated source information bits are not received at the destination. Again, since the average SNR for every channel is equal $\bar{\gamma}$ we have

$$P_{cs_out} = \prod_{j=1}^q (1 - e^{-\eta \gamma_{\circ}^{(CSD,i)} / \bar{\gamma}}) = \Theta(\bar{\gamma}^{-q})$$

In order to decode sources information bits both the syndrome and information bits from at least one of the correlated source must be available to the decoder (please note that we have assumed if these information are available perfectly at the decoder, the source's information bits can be decoded without any error). Therefore the system is in outage as a whole when either systematic bits are not available or correlated source's information bits are not available. We have

$$\begin{aligned} P_{out} &= P_{synd_out} + P_{cs_out} \\ &= \Theta(\bar{\gamma}^{-(r+1)}) + \Theta(\bar{\gamma}^{-q}) = \Theta(\bar{\gamma}^{-\min(r+1,q)}) \end{aligned}$$

Therefore the system has diversity order equal to $\min(r+1, q)$. ■

Although the system's diversity order drops asymptotically to $\min(r+1, q)$ we will show in the next section through simulations that at practical SNR regions (i.e. lower SNR regions) it can be greater for $q > r+1$.

4.3 Experimental Results

The theoretical results in the previous section are derived as SNR approaches infinity. However, in practice such high SNR values do not exist in nature. Since, deriving exact theoretical results is extremely difficult at lower SNRs, we have to rely on simulations for investigation. Therefore, in this section we will study the effects of compression through the use of correlated sources, on the diversity and the performance of WSNs, with different simulations. To do this, we use the slope of the curve that represents the frame error rate versus the SNR on a semilog plot. In the Rayleigh fading channel, this slope represents the diversity order of the system. Therefore, by running different simulations and plotting these graphs we can study diversity order of the systems. The rest of this section is organized as follows. We first present general information about our implementation and simulations. We will then present and discuss our simulation results.

4.3.1 System Model Implementation

In our simulations, the length- n information bits at the source, $\mathbf{x}^{(S)}$, are selected at random with an equiprobable density. The correlated sources' information bits, $\mathbf{x}^{(S)}$, are then selected in such a way that there are exactly $K = np$ bits that are different from $\mathbf{x}^{(S)}$. In other words $\Pr[x_i^{(S)} \neq x_i^{(CS)}] = p$. Similar to the example

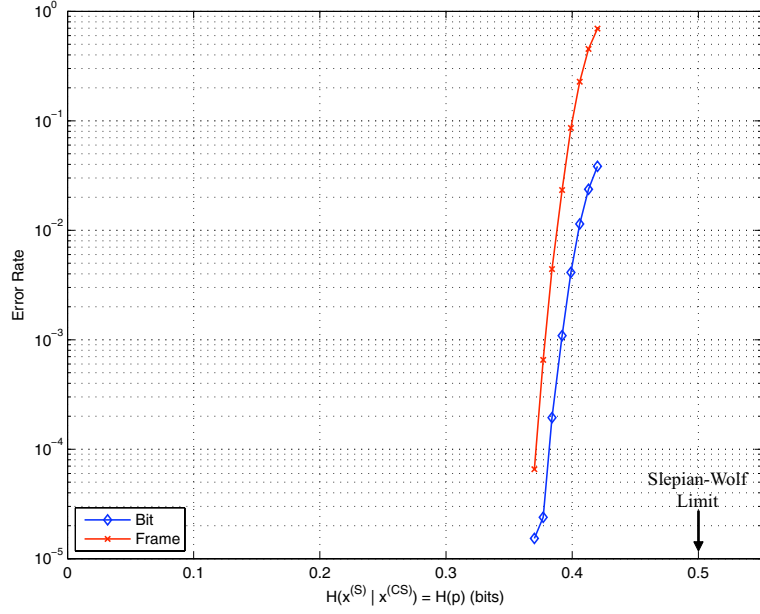


Figure 4.3: Error rate of LDPC compression.

described in section 4.1, we use LDPC codes for compression and PSRA codes as error correcting codes for channel coding.

As we mentioned in sections 4.1 and 4.2, LDPC compression is not error free. In our simulations we use (3,6) regular LDPC codes. To show the error associated with the compression and decompression process, we ran a simulation using the frame size of 10000 on different decorrelation factors (i.e. different p values). The results are presented in Figure 4.3, where the blue curve is the bit error rate and the red curve is the frame error rate. Since a (3,6) regular LDPC code compressed the original length- n information bits to syndrome bits of length- $\frac{n}{2}$, the Slepian-Wolf

limit is at $H(x_i^{(S)} | x_i^{(CS)}) = 0.5$. However, it is clear from Figure 4.3 that the (3,6) LDPC code performs over this limit. Also, it is clear that there is an error associated with the decompression process that quickly approaches zero as p and in turn $H(x_i^{(S)} | x_i^{(CS)})$ gets smaller.

In [41], the performance of some other LDPC codes, including irregular LDPC codes, were presented. It was shown that irregular LDPC codes can get closer to Slepian-Wolf limit. Moreover, it was shown that as the frame size increases, the code performance gets closer to the Slepian-Wolf limit. In our simulations because of limited computational power we use regular LDPC codes with small frame size. In order to capture the effects of better compression performance (i.e. better compression error rates), we use a higher correlation factor (i.e., smaller p values). Although such a high correlation might not occur in nature, the effects can be countered by use of bigger frame size (frame sizes as high as 100000 is possible in practice) and stronger codes such as irregular LDPC codes.

Our simulations are all performed with the following parameters, unless specified otherwise. The number of information bits to be transmitted by the source and correlated source at each iteration is 4096 (i.e. frame size). For compression, (3,6) regular LDPC codes are used. The repeat factor for PSRA code is 3. The SNR on all the links are the same. We define the rate as number of information bits per transmitted bit. Let R_S be the rate of the source, R_R the rate of the relay, R_{CS}

the rate of the correlated source and R_{sys} the overall rate of the systems. Since the overall rate can be different for each system under comparison we use normalized SNR which means $E_b/N_o = 1/(2R_{sys}\sigma_n^2)$ where R_{sys} is the total rate of the system, σ_n^2 the noise variance. As usual let r be the number of relays and q the number of correlated sources. With these coding schemes and parameters we are ready to present our simulation results in the next section.

4.3.2 Simulation Results

Since we want to study the effects of number of correlated sources have on the diversity order of the system, we first consider a system with zero relays and one to three correlated sources. Therefore, according to Theorem 1, the asymptotic diversity order of the system is 1 for all $q \geq 1$. Figures 4.4, 4.5, and 4.6, show the frame error rate (FER) and bit error rate (BER) of systems with $q = 1$, $q = 2$ and $q = 3$ in Rayleigh fading channel. The ‘NCS’ curve in all the graphs represents the system where there is no correlated source and no compression (i.e. there is a source and a destination node only). In this system, the source encodes its information bits using PSRA code and transmits them at the rate $R_S = 1/4$. Clearly the diversity order of such a system is one since there is only one link from source to destination. Therefore, the slope of ‘NCS’ curve is one. The ‘NCS’ curve has been included in all the graph as a reference.

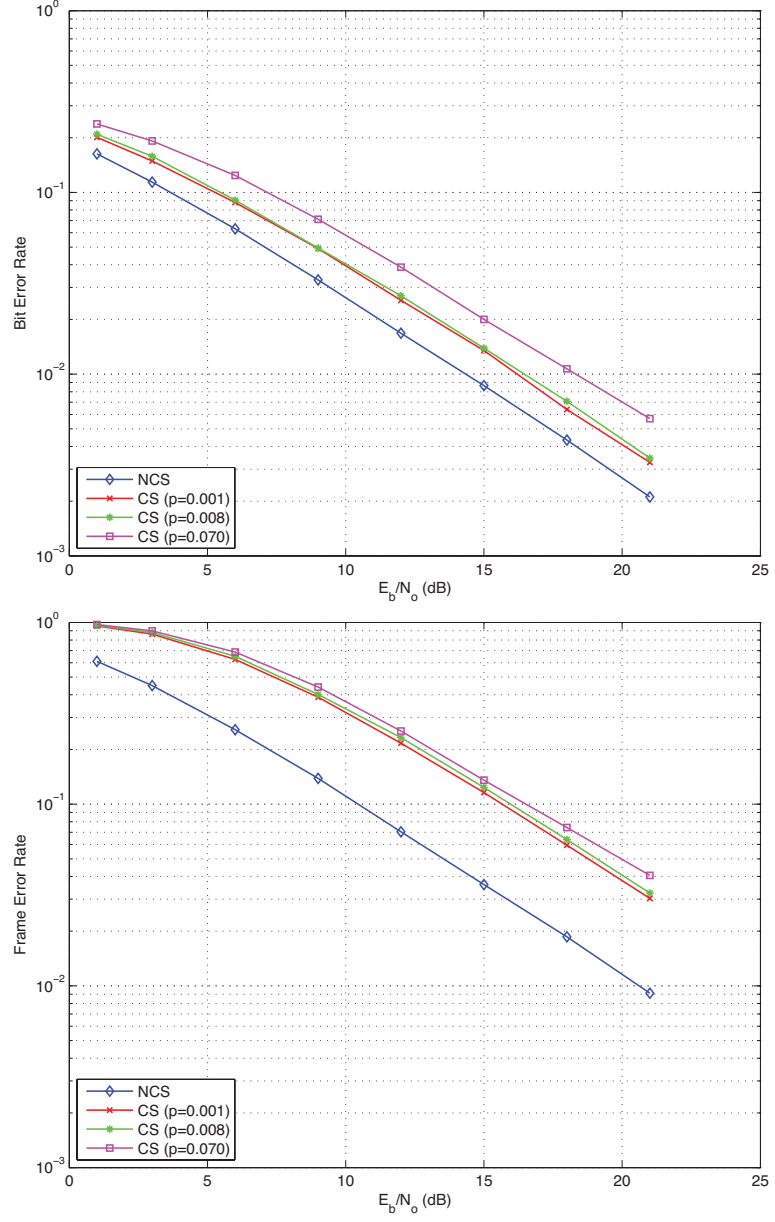


Figure 4.4: FER and BER of systems with no relays and one correlated source in Rayleigh fading.

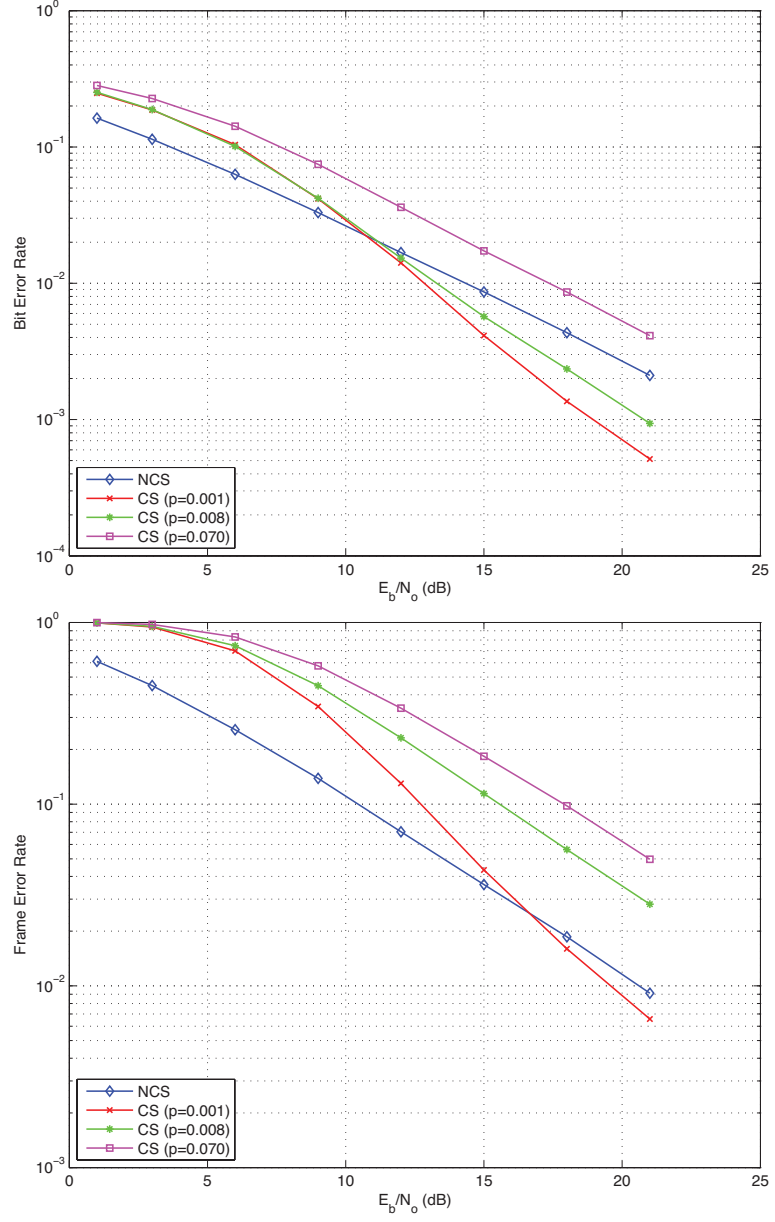


Figure 4.5: FER and BER of systems with no relays and two correlated sources in Rayleigh fading.

The other three curves in all the graphs present systems with correlated sources (i.e. a source, a destination and one to three correlated sources). The decorrelation factor p is the differentiating factor in these curves. The purple curves represent $p = 0.07$, the green curves represent $p = 0.008$, and the red curves represent $p = 0.001$. The source's information bits are first compressed using the LDPC codes. The resulting syndrome bits are encoded using PSRA codes, and are transmitted at the rate $R_S = 1/2$. Each correlated source will also encode its information bits using PSRA codes and transmit them at rate $R_{CS} = 1/2$. Therefore, the overall rate of the system for each 'CS' curve is $R_{sys} = 1/3$ for one correlated source, $R_{sys} = 1/5$ for two correlated sources and $R_{sys} = 1/7$ for three correlated sources.

From Figure 4.4, we can see that for one correlated source the diversity order is one. The overall rate of the system is lower when we add the correlated source and compress the data at the source. However, the performance in term of BER and FER is worse. From Figure 4.5, it is clear that the diversity order can go above one for intermediate SNRs since the red curve, representing two correlated sources with $p = 0.001$, and the blue curve (our reference curve) intersect (i.e. the slopes are not the same). We also notice that as the correlation factor between the source and the correlated sources increase the diversity and the performance in terms of FFR and BER increases. This increase in diversity is temporary and as SNR increases approaches the theoretical asymptotic results which is one. This effect can be seen

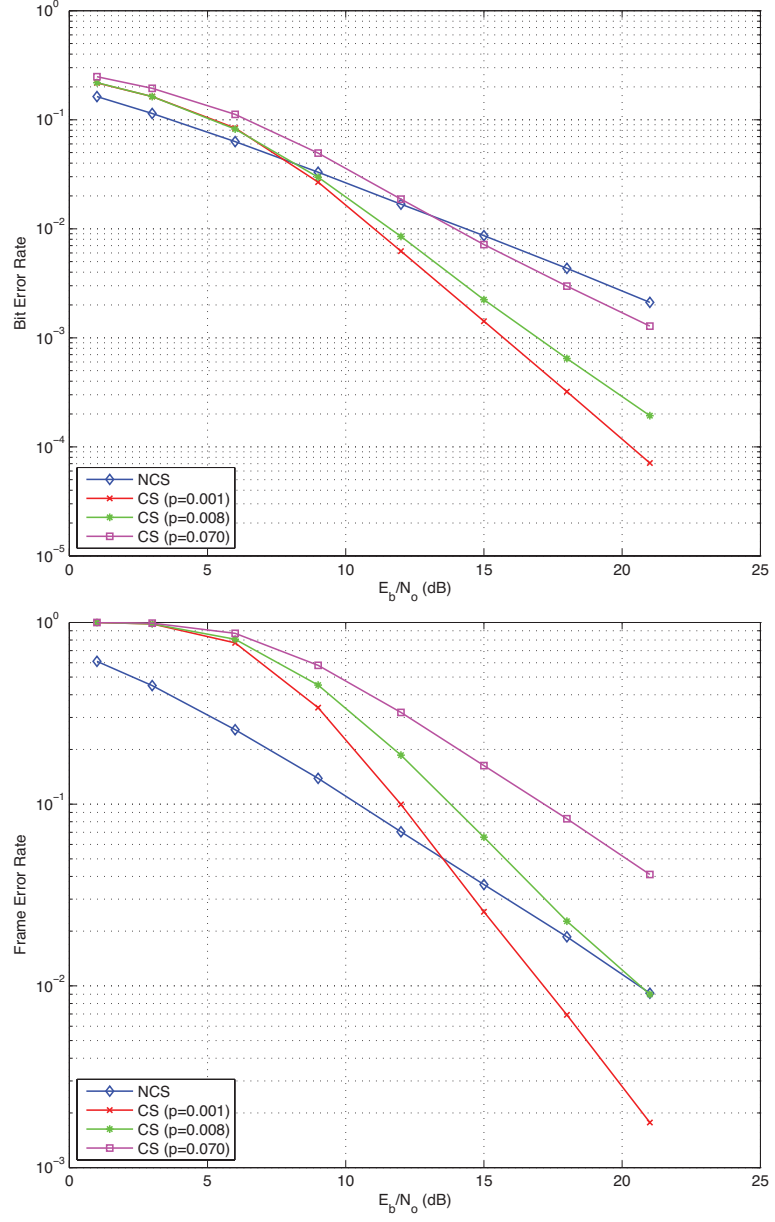


Figure 4.6: FER and BER of systems with no relays and three correlated sources in Rayleigh fading.

to a certain degree at the end of the red curve in FER graph in Figure 4.5 where the slope seems to be changing towards 1. We also notice from BER graph that for higher correlations and SNR values the BER is better than our reference with no correlated sources. Finally, from 4.6, we can see higher diversity orders and better performance in terms of BER and FER to a degree that one order of diversity increase is observed for the red curve representing $p = 0.001$.

A system with a relay is considered next, where the only difference is the introduction of the relay. Figures 4.7, 4.8, and 4.9, show the BER and FER such system with zero to three correlated sources in Rayleigh fading channel. Again the ‘NCS’ curve represents a system that does not have a correlated source and does not use compression, while the ‘CS’ curves represent the systems with correlated sources and compression. As explained before, the source encodes its transmission bits (compressed syndrome bits or uncompressed information bits) using PSRA codes, and transmits them at the rate $R_S = 1/2$. The relay makes hard decisions on the received bits, re-encodes them using PSRA codes, and transmits at the rate $R_R = 1/2$. Each of the possible correlated sources will also encode their information bits using PSRA codes and transmits at the rate $R_{CS} = 1/2$.

The overall rate of the ‘NCS’ curve is therefore $R_{sys} = 1/6$, while the overall rate of the ‘CS’ curves are $R_{sys} = 1/5$ for one correlated source, $R_{sys} = 1/7$ for two correlated sources and $R_{sys} = 1/9$ for two correlated sources. Again, the purple

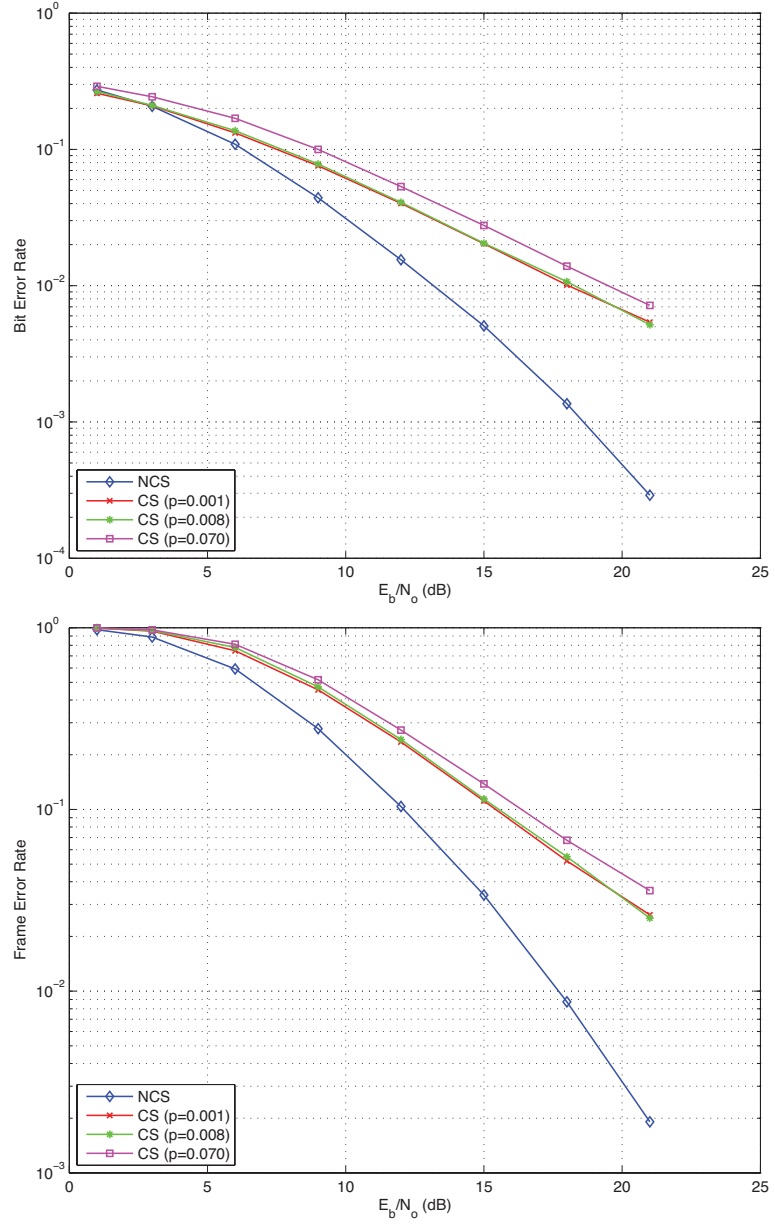


Figure 4.7: FER and BER of systems with one relay and one correlated source in Rayleigh fading.

curves represent $p = 0.07$, the green curves represent $p = 0.008$, and the red curves represent $p = 0.001$. It is clear that a system with a single relay (i.e. no correlated sources or compression) has a diversity order of two since there are two possible paths from the source to destination, the S-D path and the S-R-D path. This is reflected through the slope of the blue curve (i.e. slope is 2). The ‘NCS’ curve is therefore included as a reference in all the graphs.

Again we start by looking at Figure 4.7, where the slope of the ‘CS’ curves are all one for a single correlated source. This is because, according to Theorem 1, the minimum of the number of relays and the number of correlated sources is one, and hence the slopes and the diversity orders are one. In Figure 4.8, where we have two correlated sources the slopes are two or higher. For example, the red curve where $p = 0.001$ touched the blue curve (reference ‘NCS’ curve) at around 15dB which can happen only if its slope is greater than that of the reference curve. Also, we see as correlation factor increases the slopes and hence the diversity order increases. Figure 4.9, represent curves with three correlated sources. Therefore, according to Theorem 1, the asymptotic diversity must be 2. However, again we see greater diversity orders since both the red and the green curves ($p = 0.001$ and $p = 0.008$) intersect and pass the reference curve.

From previous results it is clear that although asymptotically diversity order can not be increased by introducing correlated sources, it can be increased at practical

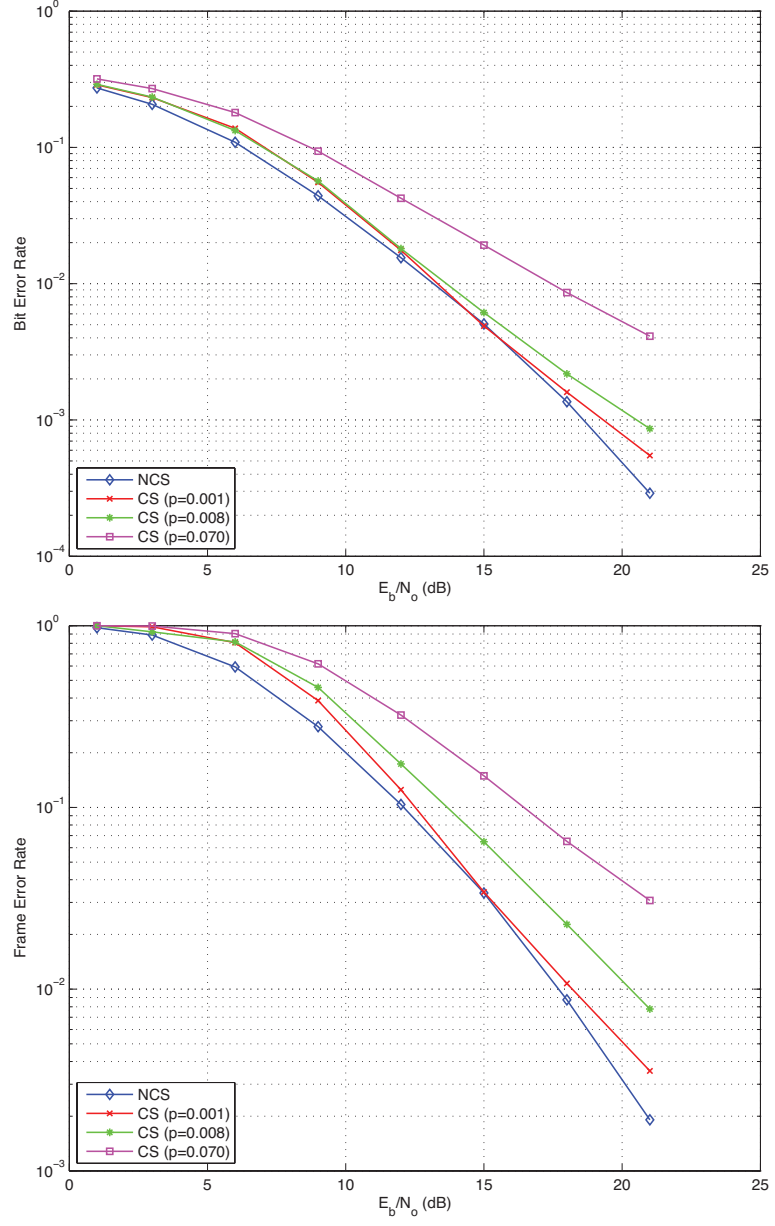


Figure 4.8: FER and BER of systems with one relay and two correlated sources in Rayleigh fading.

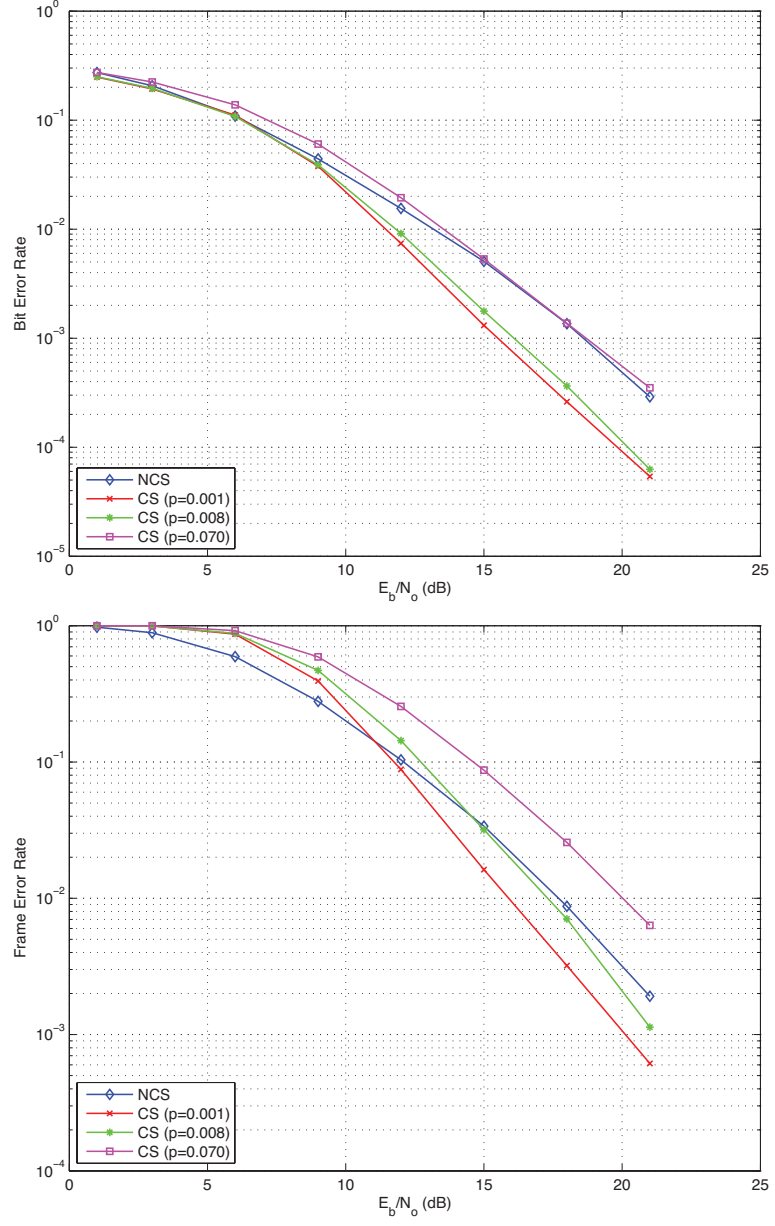


Figure 4.9: FER and BER of systems with one relay and three correlated sources in Rayleigh fading.

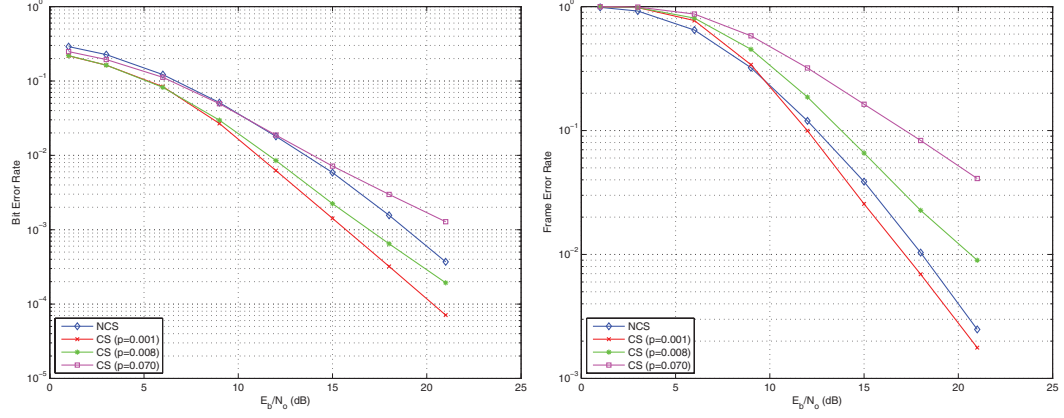


Figure 4.10: Using relays for diversity vs using correlated sources.

SNRs. Therefore, in the next set of simulations we compare the use of correlated sources for adding diversity with the use of relays. In particular two systems are considered, the ‘NCS’ source with a single source, relay and destination nodes and the ‘CS’ systems with a single source and destination as well as three correlated source. The rates of ‘NCS’ system are $R_S = 1/2$, $R_R = 2/5$ and $R_{sys} = 1/7$, while the rates for the ‘CS’ system are $R_S = 1/2$, $R_{CS} = 1/2$ and $R_{sys} = 1/7$. Since both systems have the same overall rate, the overall transmission power in both systems are the same; even though there are three correlated sources but only one relay.

Figure 4.10 shows the results of these simulations. The blue curve represents the ‘NCS’ case and the purple, green and red curves represent the correlated source curves with decorrelation factors of $p = 0.07$, $p = 0.008$ and $p = 0.001$, respectively. As we can see from the curves, for high enough correlation factors (or possibly

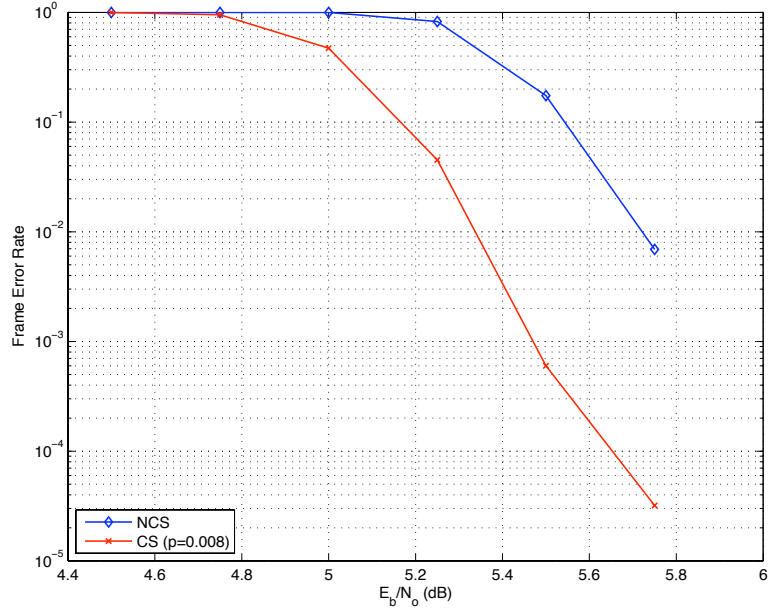


Figure 4.11: Frame error rate of a system with a relay and three correlated sources in AWGN channel.

larger frame size and stronger compression codes), the use of correlated sources can perform as well (and maybe even better than) using relays for adding diversity. For example, the red curve representing three correlated source with $p = 0.001$ can be seen to outperform the blue curve that represents the relay.

The final figure, Figure 4.11, compares the performance of a system with no correlated source and no compression versus a system with a correlated source and compression in AWGN channel. The ‘CS’ curve has a single relay and a single correlated source while ‘NCS’ curve has a single relay and no correlated source.

The rates of transmission are $R_S = 1/2$, $R_R = 1/2$ and $R_{CS} = 1/2$. The overall rate of the ‘NCS’ curve is therefore $R_{NCS} = 1/6$, while the overall rate of the ‘CS’ curve is $R_{sys} = 1/5$. The correlation between the source and correlated source is given by $p = 0.008$. Although in this case the ‘CS’ curve has higher rate (and hence lower power requirements), we can see it outperforms the ‘NCS’ curve. Therefore, by introducing a correlated source and compression not only the transmission power is lowered but also the performance of the system is improved.

4.4 Summary

In this chapter, we proposed a system for studying the effects of introducing correlated sources and compression on the diversity and performance of the WSN. We first derived an asymptotic theoretical result that showed adding correlated sources will not increase the diversity as the SNR approaches infinity. We then used simulation to study these effects under practical SNR values. We showed that using correlated sources can increase the diversity of the system at practical SNRs. We also showed that the use of correlated sources can be as good or maybe better than the use of relay in terms of diversity and error rates at similar transmission power. Although at first glance this might not seem like a big improvement, a closer look reveals that the correlated sources in WSNs are other sensor nodes transmitting their information. Therefore, the use of correlated sources to increase diversity

might not come at extra cost of the overall transmission power of WSN while the use of relay does. We then showed that introduction of correlated sources and compression can also improve the performance in AWGN channel, also increasing the rate and reducing the transmission power. In the next chapter we introduce an analytic method for optimizing the transmission power of WSN with lots of nodes, using fractional cooperation.

5 Analytic Optimization

WSNs typically consist of large number of sensor nodes, which both send their own information to the data sink and assist their neighbours via cooperation. Therefore, since there are possibly a large number of relays in a WSN, the use of fractional cooperation [17], described in section 3.1.3, is appropriate. However, the challenge in using fractional cooperation is that it is difficult to know what fraction needs to be retransmitted to the destination, by each relay, that helps minimize the transmission power while ensuring successful decoding at the destination. In this chapter, we present an analytic method based on linear programming for this optimization task.

To derive this analytic method we first study, what fraction must be retransmitted at each relay to ensure decoding success given the channel SNR and the coding method being used. This question can be answered by using EXIT chart analysis, as described in Chapter 3. We will then consider a multi-source multi-relay system and pose a linear programming problem, where the answer to the problem is the optimized fractions to be retransmitted by each relay for each source.

The rest of this chapter is organized as follows. We will explain the system under study in detail in section 5.1, using the relay models and coding techniques described in Chapter 3. We will then derive our theoretical analytic method in section 5.2. In section 5.3 we will present our simulation results to illustrate our method.

5.1 System Setup

In this section, we describe the system to be studied and the problem to be investigated. The relay models, along with some of the coding techniques described in Chapter 3, will be used to set up the system. A simple example will then be presented to show how different components of the system work together. We will then describe our contribution by formulating some of the questions we will answer in the rest of the chapter.

The relay model that we use in this chapter is based on the fractional cooperation model described in section 3.1.3. The fractional cooperation system consists of s sources, and r relays, and a destination. Each source measures a phenomenon, encodes it using LDPC codes, and broadcasts the encoded codeword to the r relays, as well as the destination. Each relay makes hard decisions (or decodes the received bits), and selects a small fraction for retransmission to the destination. The destination will then decode each source's information bits using the received

signal from the r relays, as well as the source itself.

Next we will provide an example for both decode-and-forward (DF) method where sources' informations are decoded at the relays and demodulate-and-forward (DemF) method where the relays make hard decisions.

5.1.1 An Example

Consider a system with $r = 3$ relays and $s = 2$ sources. For simplicity, let's assume that there are $n = 3$ information bits that need to be sent by each source. The information bits are given by $\mathbf{x}^{(S_1)} = [1 \ 0 \ 1]$ for the first source and $\mathbf{x}^{(S_2)} = [1 \ 1 \ 0]$ for the second source. Let

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

represent a (6,3) LDPC parity-check matrix, which each source uses to encode its information bits. This parity-check matrix is too small to be useful as an LDPC encoder. However, it is sufficient for this example. The corresponding generator matrix is given by

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

The resulting encoded bits for the two sources are given by $\mathbf{z}^{(S_1)} = \mathbf{x}^{(S_1)}\mathbf{G} = [1 \ 0 \ 1 \ 0 \ 1 \ 1]$ and $\mathbf{z}^{(S_2)} = \mathbf{x}^{(S_2)}\mathbf{G} = [1 \ 1 \ 0 \ 0 \ 1 \ 0]$. These encoded transmission sequences are then broadcast to destination and each of the r relays.

5.1.1.1 Demodulate-and-Forward

Each relay selects a small fraction of the signal from each source and demodulates them by making hard decisions. Table 5.1 summarizes this process for our example. In this example, each relay selects two bits from each source. For example, the first relay selects bit positions 1 and 4 from the first source which were 1 and 0, respectively. It makes an error in demodulating the second bit, which results in bits 1 and 1, respectively. Similarly, the second and third relays, randomly select two bit positions and demodulate those as shown in table 5.1. The same process occurs when the second source broadcasts to relays.

As explained in chapter 3, if DemF is used by the relays, for simplicity we assume powerful and capacity approaching codes (such as irregular RA [51] and irregular LDPC [57] codes) are used on R-D links, which can result in perfect recovery of demodulated bits at the destination. As explained in section 3.1.3, let $\mathbf{y}_{DemF}^{(S_i, R_j, D)}$ be the signal representation of the demodulation values for i th source through j th relay available at the destination. Therefore, $\mathbf{y}_{DemF}^{(S_i, R_j, D)}$ contains: a -1 if a bit position is demodulated as 1, +1 if a bit position is demodulated as 0, and 0 if a bit position

Table 5.1: Relay selections and demodulations

Relay	Positions Selected	Actual Position Values	Demod Values
$R_1^{(S_1)}$	1 , 4	1 , 0	1 , 1
$R_2^{(S_1)}$	2 , 5	0 , 1	0 , 1
$R_3^{(S_1)}$	1 , 6	1 , 1	1 , 1
$R_1^{(S_2)}$	2 , 3	1 , 0	0 , 1
$R_2^{(S_2)}$	1 , 4	1 , 0	1 , 0
$R_3^{(S_2)}$	5 , 6	1 , 0	1 , 1

is not selected to be forwarded to the destination. In our example, the $\mathbf{y}_{DemF}^{(S_i, R_j, D)}$ values are given by

$$\begin{aligned}
\mathbf{y}_{DemF}^{(S_1, R_1, D)} &= \begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} , \\
\mathbf{y}_{DemF}^{(S_1, R_2, D)} &= \begin{bmatrix} 0 & +1 & 0 & 0 & -1 & 0 \end{bmatrix} , \\
\mathbf{y}_{DemF}^{(S_1, R_3, D)} &= \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} , \\
\mathbf{y}_{DemF}^{(S_2, R_1, D)} &= \begin{bmatrix} 0 & +1 & -1 & 0 & 0 & 0 \end{bmatrix} , \\
\mathbf{y}_{DemF}^{(S_2, R_2, D)} &= \begin{bmatrix} -1 & 0 & 0 & +1 & 0 & 0 \end{bmatrix} , \\
\mathbf{y}_{DemF}^{(S_2, R_3, D)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} .
\end{aligned}$$

At the destination the channel LLRs are calculated similar to the example in chapter 4 section 4.1. Therefore for the signals coming from the sources, the channel

LLRs are given by

$$\begin{aligned}\ell^{(S_1,D)} &= 4\gamma_{(S_1,D)}\mathbf{y}^{(S_1,D)} , \\ \ell^{(S_2,D)} &= 4\gamma_{(S_2,D)}\mathbf{y}^{(S_2,D)} .\end{aligned}\tag{5.1}$$

Meanwhile, the channel LLRs from the j th relay are given by

$$\begin{aligned}\ell_{DemF}^{(S_1,R_j,D)} &= \mathbf{y}_{DemF}^{(S_1,R_j,D)} \log \left[\frac{1 - p_{Dem}^{(S_1,R_j)}}{p_{Dem}^{(S_1,R_j)}} \right] , \\ \ell_{DemF}^{(S_2,R_j,D)} &= \mathbf{y}_{DemF}^{(S_2,R_j,D)} \log \left[\frac{1 - p_{Dem}^{(S_2,R_j)}}{p_{Dem}^{(S_2,R_j)}} \right] ,\end{aligned}\tag{5.2}$$

where $p_{Dem}^{(S_1,R_j)}$ and $p_{Dem}^{(S_2,R_j)}$ are probabilities that the hard decisions at the relays were incorrect. Therefore, $p_{Dem}^{(S_i,R_j)}$ is given by

$$p_{Dem}^{(S_i,R_j)} = \frac{1}{2} \text{erfc} \left(\sqrt{\gamma_{(S_i,R_j)}} \right) ,\tag{5.3}$$

where $\gamma_{(S_i,R_j)}$ is the average SNR of the link between the i th source and the j th relay. Finally, the input LLRs to the LDPC iterative decoder algorithms at the destination are calculated as

$$\begin{aligned}\ell_{DemF}^{(S_1)} &= \ell^{(S_1,D)} + \sum_{j=1}^r \ell_{DemF}^{(S_1,R_j,D)} , \\ \ell_{DemF}^{(S_2)} &= \ell^{(S_2,D)} + \sum_{j=1}^r \ell_{DemF}^{(S_2,R_j,D)} .\end{aligned}\tag{5.4}$$

in general, the channel LLR for the S-D link is calculated as

$$\ell^{(S_i,D)} = 2\mathbf{y}^{(S_i,D)} / \sigma_{(S_i,D)}^2 = 4\gamma_{(S_i,D)}\mathbf{y}^{(S_i,D)} ,\tag{5.5}$$

and for the S-Rs link as

$$\ell_{DemF}^{(S_i,R_j,D)} = \mathbf{y}_{DemF}^{(S_i,R_j,D)} \log \left[\frac{1 - p_{Dem}^{(S_i,R_j)}}{p_{Dem}^{(S_i,R_j)}} \right] ,\tag{5.6}$$

where $p_{Dem}^{(S_i, R_j)}$ is the probability of demodulation error at the j th relay assisting i th source given by

$$p_{Dem}^{(S_i, R_j)} = \frac{1}{2} \text{erfc} \left(\sqrt{\gamma_{(S_i, R_j)}} \right). \quad (5.7)$$

Consequently, the message LLR input to the iterative LDPC decoder of the i th source can be calculated as

$$\ell_{DemF}^{(S_i)} = \ell^{(S_i, D)} + \sum_{j=1}^r \ell_{DemF}^{(S_i, R_j, D)}. \quad (5.8)$$

5.1.1.2 Decode-and-Forward

In this scheme, each relay decodes each source's signal, and re-encodes it using exactly the same LDPC encoder as the one used by that source. As explained in chapter 3, if DF is used by the relays, for simplicity we assume powerful and capacity approaching codes (such as irregular RA [51] and irregular LDPC [57] codes) are used on S-R links, which can result in perfect recovery of each source's information bits at each relay. Since we have assumed the decoding step is successful and we re-encode using exactly the same encoding process, each relay will have the exact transmission sequence sent by the source and will select only a fraction for retransmission to the destination. In this example each relay selects two bits from each source. For example, the first relay selects bit positions 1 and 4 from the first source (which were 1 and 0) and retransmits them to the destination. Similarly, the second and third relay randomly select two bits positions and retransmits them

as shown in table 5.2. The same process occurs when the second source broadcasts to relays.

Table 5.2: Relay selections and retransmissions

Relay	Positions Selected	Position Values
$R_1^{(S_1)}$	1 , 4	1 , 0
$R_2^{(S_1)}$	2 , 5	0 , 1
$R_3^{(S_1)}$	1 , 6	1 , 1
$R_1^{(S_2)}$	2 , 3	1 , 0
$R_2^{(S_2)}$	1 , 4	1 , 0
$R_3^{(S_2)}$	5 , 6	1 , 0

The signals received at the decoder, through relay to destination links, are corrupted by noise, as explained in 3.1.3, and are represented by the vector, $\mathbf{y}_{DF}^{(S_i, R_j, D)}$. This vector is defined such that the values of the positions not selected by the j th relay is zero. Therefore, for our example, the values for the received signals from

the relays at destination are

$$\begin{aligned}
\mathbf{y}_{DF}^{(S_1, R_1, D)} &= \begin{bmatrix} y_{DF_1}^{(S_1, R_1, D)} & 0 & 0 & y_{DF_2}^{(S_1, R_1, D)} & 0 & 0 \end{bmatrix}, \\
\mathbf{y}_{DF}^{(S_1, R_2, D)} &= \begin{bmatrix} 0 & y_{DF_1}^{(S_1, R_2, D)} & 0 & 0 & y_{DF_2}^{(S_1, R_2, D)} & 0 \end{bmatrix}, \\
\mathbf{y}_{DF}^{(S_1, R_3, D)} &= \begin{bmatrix} y_{DF_1}^{(S_1, R_3, D)} & 0 & 0 & 0 & 0 & y_{DF_2}^{(S_1, R_3, D)} \end{bmatrix}, \\
\mathbf{y}_{DF}^{(S_2, R_1, D)} &= \begin{bmatrix} 0 & y_{DF_1}^{(S_2, R_1, D)} & y_{DF_2}^{(S_2, R_1, D)} & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{y}_{DF}^{(S_2, R_2, D)} &= \begin{bmatrix} y_{DF_1}^{(S_2, R_2, D)} & 0 & 0 & y_{DF_2}^{(S_2, R_2, D)} & 0 & 0 \end{bmatrix}, \\
\mathbf{y}_{DF}^{(S_2, R_3, D)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & y_{DF_1}^{(S_2, R_3, D)} & y_{DF_2}^{(S_2, R_3, D)} \end{bmatrix},
\end{aligned}$$

where $y_{DF_k}^{(S_i, R_j)}$ represents the noise corrupted signal of the k th transmission bit, received at the destination, from the j th relay retransmitting i th source's data.

The channel LLRs are calculated similar to the the DemF case. First the source-destination channel LLRs is calculated from the signal received from each source as in equation 5.5. Then the LLRs of the relay-destination links are calculated as follows

$$\begin{aligned}
\ell_{DF}^{(S_1, R_j, D)} &= 4\gamma_{(R_j, D)} \mathbf{y}_{DF}^{(S_1, R_j, D)}, \\
\ell_{DF}^{(S_2, R_j, D)} &= 4\gamma_{(R_j, D)} \mathbf{y}_{DF}^{(S_2, R_j, D)},
\end{aligned} \tag{5.9}$$

where $\gamma_{(R_j, D)}$ is the channel amplitude and the average SNR of the link between j th relay and the destination. The final input LLRs to the LDPC iterative decoder

algorithms at the destination are calculated by adding all the LLR as

$$\begin{aligned}\ell_{DF}^{(S_1)} &= \ell^{(S_1,D)} + \sum_{j=1}^r \ell_{DF}^{(S_1,R_j,D)} , \\ \ell_{DF}^{(S_2)} &= \ell^{(S_2,D)} + \sum_{j=1}^r \ell_{DF}^{(S_2,R_j,D)} .\end{aligned}\tag{5.10}$$

In general, the channel LLR for the S-D links can be calculated as in equation (5.5), and for the R-D links as

$$\ell_{DF}^{(S_i,R_j,D)} = 4\gamma_{(R_j,D)} \mathbf{y}_{DF}^{(S_i,R_j,D)} .\tag{5.11}$$

Consequently, the message LLR input to the iterative LDPC decoder of the i th source can be calculated as

$$\ell_{DF}^{(S_i)} = \ell^{(S_i,D)} + \sum_{j=1}^r \ell_{DF}^{(S_i,R_j,D)} .\tag{5.12}$$

5.2 Linear Programming Model

In this section we present a linear programming model that will minimize the number of transmission bits of a multi-source multi-relay system, described in the previous section, subject to the constraint of successful transmission. The relays can use both DF and DemF cooperation methods. In the next section we will confirm these methods by presenting simulations.

The main challenge in implementing a fractional cooperation system is obtaining the fraction that must be forwarded by each relay. This value must be chosen such

that the transmission is successful, while the number of transmission bits is at its minimum, which is a non-trivial design problem. To find a solution to this problem we divide the problem into two parts. First, we investigate the criteria for successful transmission using EXIT chart analysis. Then, we will try to satisfy that criteria while minimizing the number of transmission bits by presenting a linear programming model.

5.2.1 Key Assumptions and Definitions

We consider the multi-relay, multi-source system, explained in section 5.1, with r relays and s sources. For each source, we have r relays, and therefore r source-relay (S-R) and relay-destination (R-D) links (as well as a single source-destination (S-D) link). We assume that the all-zero codeword is transmitted by source which is equivalent to the all-(+1) channel codeword. We assume all links are independent AWGN channels represented with their respective channel SNR.

We define an $s \times (s \times r)$ matrix Γ_{RD} such that the rows of the matrix represent each source and the columns represent $s \times r$ R-D channels. The columns are listed in the following order

$$(S_1, R_1) \cdots (S_1, R_r) (S_2, R_1) \cdots (S_2, R_r) \cdots (S_s, R_r),$$

which represents the relays 1 through r forwarding for the first source and then for the second source and so on. For the i th row the only non zero elements are

columns (S_i, R_1) to (S_i, R_r) where the values are the SNR of corresponding R-D channels. Therefore, matrix Γ_{RD} is given by

$$\begin{bmatrix} \gamma_{(R_1,D)} & \gamma_{(R_2,D)} & \cdots & \gamma_{(R_r,D)} & \mathbf{0}_{(s \times r) - r} \\ \mathbf{0}_r & \gamma_{(R_1,D)} & \cdots & \gamma_{(R_r,D)} & \mathbf{0}_{(s \times r) - 2r} \\ \mathbf{0}_{2r} & \gamma_{(R_1,D)} & \cdots & \gamma_{(R_r,D)} & \mathbf{0}_{(s \times r) - 3r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(s \times r) - r} & \gamma_{(R_1,D)} & \gamma_{(R_2,D)} & \cdots & \gamma_{(R_r,D)} \end{bmatrix}, \quad (5.13)$$

where $\mathbf{0}_l$ is a row vector of l zeros. We also define a vector of length s , γ_{SD} , as

$$\gamma_{SD} = [\gamma_{(S_1,D)} \ \gamma_{(S_2,D)} \ \cdots \ \gamma_{(S_s,D)}]^T. \quad (5.14)$$

As explained in section 5.1, for simplicity, we assume that for DemF the R-D links are perfect and for DF the S-R links are perfect. We define $\epsilon_{(i,j)}$ as the fraction of i th source transmission that was forwarded by the j th relay. We define a vector of length $s \times r$, ϵ , as

$$\epsilon = [\epsilon_{(1,1)} \ \cdots \ \epsilon_{(1,r)} \ \epsilon_{(2,1)} \ \cdots \ \epsilon_{(2,r)} \ \cdots \ \epsilon_{(s,1)} \ \cdots \ \epsilon_{(s,r)}], \quad (5.15)$$

where the elements represent the fractions that are selected for retransmission by each relay for each source.

In [15] it was shown that using EXIT chart analysis of channel parameters such as channel LLR mean and variance, an approximate threshold of the LDPC iterative decoder convergence can be estimated. Therefore, for LDPC codes an approximate

minimum channel LLR mean required for successful decoding, written $m_{\ell_{min}}$, can be calculated using simulations and EXIT chart analysis. Hence, we can assume that if the channel LLR mean, m_ℓ , that is input to the iterative decoding algorithm, satisfies $m_\ell \geq m_{\ell_{min}}$, the iterative decoding process is successful with a very high probability.

Since for each source we can have a different minimum channel LLR mean requirements, we define a vector of length s , $\mathbf{m}_{\ell_{min}}$, as vector with s elements that are

$$\mathbf{m}_{\ell_{min}} = \left[m_{\ell_{min}}^{\{1\}} \ m_{\ell_{min}}^{\{2\}} \ \cdots \ m_{\ell_{min}}^{\{s\}} \right]^T, \quad (5.16)$$

where $m_{\ell_{min}}^{\{i\}}$ is the minimum channel LLR mean threshold for the i th source.

The density of LLR intrinsic messages for a Gaussian channel is symmetric, and remains symmetric under sumproduct iterative decoding, as shown in [14]. A Gaussian pdf with mean m and variance σ^2 is symmetric if and only if $\sigma^2 = 2m$. For a symmetric Gaussian density the SNR is defined as m^2/σ^2 where m is the mean and $\sigma^2 = 2m$ is the variance. Therefore we can simplify the expression to $m/2$. Since the channel SNR and the SNR of the channel LLR messages are the same, the channel LLR mean, m_ℓ , can be calculated as $m_\ell = 2\gamma$, where γ is the channel SNR.

5.2.2 Successful Transmission Requirements

As described in the previous section, in order to ensure successful transmission and decoding at the destination, the mean of the input channel LLR should be greater than a minimum, $m_{\ell_{min}}$. Therefore, we need to be able to calculate the mean of the input channel LLRs. In this section we present two theorems for calculation of the input LLR mean to the decoder for both DemF and DF.

Theorem 2 *For the system described in section 5.1, assuming that the relays use DemF, the channel mean that is input to the iterative decoder for the i th source, S_i , is given by*

$$m_{\ell_i}^{(DemF)} = 2\gamma_{(S_i,D)} + \sum_{j=1}^r \epsilon_{(i,j)} (1 - 2p_{Dem}^{(S_i,R_j)}) \log \left[\frac{1 - p_{Dem}^{(S_i,R_j)}}{p_{Dem}^{(S_i,R_j)}} \right],$$

where $\gamma_{(S_i,D)}$ is the channel SNR between the i th source and the destination, $p_{Dem}^{(S_i,R_j)}$ is the probability of hard decision error at the relay given by equation (5.7) and $\epsilon_{(i,j)}$ the fraction selected by each relay.

Proof: When the relays use DemF the input LLR to the iterative decoding algorithm at the destination will consist of summation of, $\ell^{(S_i,D)}$ and $\ell^{(S_i,R_j)}$ as shown in equation (5.8) in section 5.1. For the single S-D link the channel LLR mean is calculated as

$$m_{\ell}^{(S_i,D)} = 2\gamma_{(S_i,D)}, \quad (5.17)$$

where $\gamma_{(S_i,D)}$ is the channel SNR of the S-D link for the i th source. This is the first term in Theorem 2.

Since we have assumed that the R-D links are perfect, $\ell^{(S_i,R_j)}$ represents the LLR of hard decisions at the j th relay. From equation (5.6) we know that the channel LLR mean for $\ell^{(S_i,R_j)}$ depends on $p_{Dem}^{(S_i,R_j)}$ the probability that an error occurs when making hard decisions at the j th relay retransmitting i th source signal. If for example we assume $\mathbf{y}^{(S_i,R_j)}$ (the results of hard decisions mapped to +1 and -1 instead of 0 and 1) is all-(+1) the channel LLR mean is given by

$$m_{\ell}^{(S_i,R_j)} = \log \left[\frac{1 - p_{Dem}^{(S_i,R_j)}}{p_{Dem}^{(S_i,R_j)}} \right],$$

where $p^{(S_i,R_j)}$ is calculated according to equation (5.7).

In general since according to our assumption the all-zero codeword was transmitted by the source, assuming the length of the codeword is m , the expected number of +1s in the demodulated sequence is $(1 - p_{Dem}^{(S_i,R_j)})m$ while the expected number of -1s in the sequence is $p_{Dem}^{(S_i,R_j)}m$. Therefore, the LLR average is calculated as

$$\frac{\left[(1 - p_{Dem}^{(S_i,R_j)})m - p_{Dem}^{(S_i,R_j)}m \right] \log \left[\frac{1 - p_{Dem}^{(S_i,R_j)}}{p_{Dem}^{(S_i,R_j)}} \right]}{m} = \quad (5.18)$$

$$(1 - 2p_{Dem}^{(S_i,R_j)}) \log \left[\frac{1 - p_{Dem}^{(S_i,R_j)}}{p_{Dem}^{(S_i,R_j)}} \right]$$

Since the relay will only forward, $\epsilon_{(i,j)}$ fraction of the demodulated sequence which is equivalent to replacing the unselected positions with zero, the channel LLR mean

for the j th relay is given by

$$m_{\ell}^{(S_i, R_j)} = \epsilon_{(i,j)} (1 - 2p_{Dem}^{(S_i, R_j)}) \log \left[\frac{1 - p_{Dem}^{(S_i, R_j)}}{p_{Dem}^{(S_i, R_j)}} \right] \quad (5.19)$$

Finally, since all the channel LLRs from the sources and relays have a symmetric Gaussian distribution, the mean of their sum is the sum of all the channel LLR means of S-D and R-D links; That is for the i th source S_i

$$m_{\ell_i}^{(DemF)} = m_{\ell}^{(S_i, D)} + \sum_{j=1}^r m_{\ell}^{(S_i, R_j)} \quad (5.20)$$

■

Corollary. For the system described in section 5.1 assuming that the relays use DemF the transmission and the decoding process is successful whenever

$$m_{\ell_i}^{(DemF)} \geq m_{\ell_{min}}$$

This corollary follows directly from Theorem 2 and the EXIT chart analysis.

The next theorem considers the input channel LLR mean to decoder for the DF case.

Theorem 3 *For the system described in section 5.1, assuming that the relays use DF, the channel mean that is input to the iterative decoder for the i th source, S_i , is given by*

$$m_{\ell_i}^{(DF)} = 2\gamma_{(S_i, D)} + \sum_{j=1}^r 2\epsilon_{(i,j)} \gamma_{(R_j, D)},$$

where $\gamma_{(S_i,D)}$ is the channel SNR between the i th source and the destination, $\gamma_{(R_j,D)}$ is the channel SNR between the j th relay and the destination and $\epsilon_{(i,j)}$ the fraction selected by each relay.

Proof: The proof is very similar to the proof of Theorem 2. The input LLR to the iterative decoding is the summation of two main LLR from S-D and R-D links. S-D link channel LLR mean is calculated exactly as in equation 5.17. Since we have assumed that the S-R links are perfect for DF in section 5.1, the channel LLR means for the R-D links can be treated as a separate S-D link (i.e. the channel mean is calculated in the same manner as equation 5.17). Because only $\epsilon_{(i,j)}$ fraction of each transmission sequence is selected by each relay and the unselected bits can be treated as zeros, the channel LLR mean for the R-D links is given by

$$m_{\ell}^{(S_i,R_j,D)} = 2\epsilon_{(i,j)}\gamma_{(R_j,D)}. \quad (5.21)$$

Finally since all the channel LLRs from the sources and relays have a symmetric Gaussian distribution, the mean of their sum is the sum of all the channel LLR means of S-D and R-D links; That is for the i th source S_i

$$m_{\ell_i}^{(DF)} = m_{\ell}^{(S_i,D)} + \sum_{j=1}^r m_{\ell}^{(S_i,R_j,D)} \quad (5.22)$$

■

Corollary. For the system described in section 5.1 assuming that the relays use DF the transmission and the decoding process is successful whenever

$$m_{\ell_i}^{(DF)} \geq m_{\ell_{min}}$$

This corollary follows directly from Theorem 3 and the EXIT chart analysis.

5.2.3 Energy Minimization

In the previous section we discussed the criteria for successful transmission and decoding at the destination for both DemF and DF based on input channel LLR means. In this section we present a linear programming model that minimizes the number of transmission bits (i.e. transmission power), subject to the constraint that the decoding is successful. In this section we derive a linear programming model [20] that achieves this task.

In linear programming the goal is to minimize or maximize a linear function of some variables. This function is called objective function. For example the function

$$f(\mathbf{x}) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

is an objective function of decision variables x_i . In addition to the objective function the decision variables need to satisfy a number of constraints that are a linear

combination of decision variables. For example,

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b,$$

is a constraint that is a linear combination of decision variables x_i .

In our model the objective is to minimize the number of transmitted bits with the constraint of successful decoding at the destination, where decision variables are $\epsilon_{(S_i, R_j)}$, the forwarding fractions of relays. We derive the objective functions and the constraints for both DF and DemF. To solve this linear programming model, any of the known linear programming solver algorithms such as simplex algorithm can be used.

5.2.3.1 Decode-and-forward

If DF is used by the relays, and the i th source has a codeword of length m_i to transmit to the destination, the objective function is given by

$$f(\epsilon) = \sum_{i=1}^s m_i + \sum_{i=1}^s \sum_{j=1}^r \epsilon_{(S_i, R_j)} m_i,$$

where the first summation represents the number of transmission bits over the S-D and S-R links, and the second double summation term corresponds to the number of transmission bits over the R-D links. Since the first summation term is a constant

and we are trying to minimize this objective function with respect to $\epsilon_{(S_i, R_j)}$, we can drop it. Therefore, the simplified objective function is given by

$$f(\epsilon) = \sum_{i=1}^s \sum_{j=1}^r \epsilon_{(S_i, R_j)} m_i. \quad (5.23)$$

The constraints can be derived using Theorem 3 as

$$2\Gamma_{RD}\epsilon \geq \mathbf{m}_{\ell_{min}} - 2\gamma_{SD}, \quad (5.24)$$

where Γ_{RD} is a $s \times (s \times r)$ matrix defined in equation (5.13), ϵ is a vector of length $(s \times r)$ defined in equation (5.15), $\mathbf{m}_{\ell_{min}}$ is a vector of length s defined in equation (5.16) and γ_{SD} is a vector of length s given by equations (5.14).

5.2.3.2 Demodulate-and-forward

If DemF is used by the relays, and the i th source has a codeword of length m_i to transmit to the destination, the objective function is given by

$$f(\epsilon) = \sum_{i=1}^s m_i + \sum_{i=1}^s \sum_{j=1}^r \frac{\epsilon_{(S_i, R_j)} m_i}{R_{i,j}},$$

which is very similar to the objective function for DF except $R_{i,j}$, the rate of transmission for the j th relay retransmitting i th source. Since in section 5.1 we assumed the R-D links are perfect through the use of powerful capacity approaching codes, we can replace $R_{i,j}$ with the capacity of the R-D channels. We can also omit the constant terms since we are minimizing. Therefore, the objective function can

be simplified to

$$f(\boldsymbol{\epsilon}) = \sum_{i=1}^s \sum_{j=1}^r \frac{\epsilon_{(S_i, R_j)} m_i}{C(\gamma_{(R_j, D)})},$$

where $C(\gamma_{(R_j, D)})$ is the channel capacity between j th relay and the destination.

To drive the constraints for the DemF we define a variable $g^{(S_i, R_j)}$ as

$$g_{(S_i, R_j)} = (1 - 2p_{Dem}^{(S_i, R_j)}) \log \left[\frac{1 - p_{Dem}^{(S_i, R_j)}}{p_{Dem}^{(S_i, R_j)}} \right], \quad (5.25)$$

where the right term in the equation is given derived in equation (5.18) and represent the R-D link channel LLR mean before fractional selection at the relays. An $s \times (s \times r)$ matrix, G_{SR} , is defined similar to the $s \times (s \times r)$ matrix, Γ_{RD} , presented in equation (5.14), such that instead of $\gamma_{(R_j, D)}$ the elements are $g_{(S_i, R_j)}$. Therefore, matrix G_{SR} is given by

$$\begin{bmatrix} g_{(S_1, R_1)} & g_{(S_1, R_2)} & \cdots & g_{(S_1, R_r)} & \mathbf{0}_{(s \times r) - r} \\ \mathbf{0}_r & g_{(S_2, R_1)} & \cdots & g_{(S_2, R_r)} & \mathbf{0}_{(s \times r) - 2r} \\ \mathbf{0}_{2r} & g_{(S_3, R_1)} & \cdots & g_{(S_3, R_r)} & \mathbf{0}_{(s \times r) - 3r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(s \times r) - r} & g_{(S_s, R_1)} & g_{(S_s, R_2)} & \cdots & g_{(S_s, R_r)} \end{bmatrix}, \quad (5.26)$$

where $\mathbf{0}_l$ is a row vector of l zeros. The constraints can be derived using Theorem 2 and matrix G_{SR} as

$$G_{SR}\boldsymbol{\epsilon} \geq \mathbf{m}_{\ell_{min}} - 2\boldsymbol{\gamma}_{SD}, \quad (5.27)$$

where $\boldsymbol{\epsilon}$, $\mathbf{m}_{\ell_{min}}$, and $\boldsymbol{\gamma}_{SD}$ are given by equations (5.15), (5.16), and (5.14), respectively.

5.3 Experimental Results

In section 5.2 we presented a Linear programming model that will optimize the number of transmission bits subject to the constraint that the decoding is successful. Since in our derivation of this linear programming model we use EXIT chart analysis which is an estimation tool, our optimization solution is also an estimate of the real optimized solution. In this section through simulation we show that this estimation is very close to the real optimized solution.

To do this we will present two sets of simulations, one to prove that the concept works through trivial examples, and another to show that the concept works and is close to the optimized solution for a non-trivial scenario. For all of our simulations we use a (3,6) regular LDPC code to encode the data and therefore all sources have the same codeword length. Using EXIT chart analysis of the (3,6) regular LDPC code we have calculated the convergence threshold for channel LLR mean as $m_{\ell_i} = 2.53$.

5.3.1 Simple Example

For our trivial example we consider a system with 2 relays and a source as well as a system with 50 relays and a source. The system is considered trivial since we assume that all the links in the system have the same SNR. Therefore, the objective

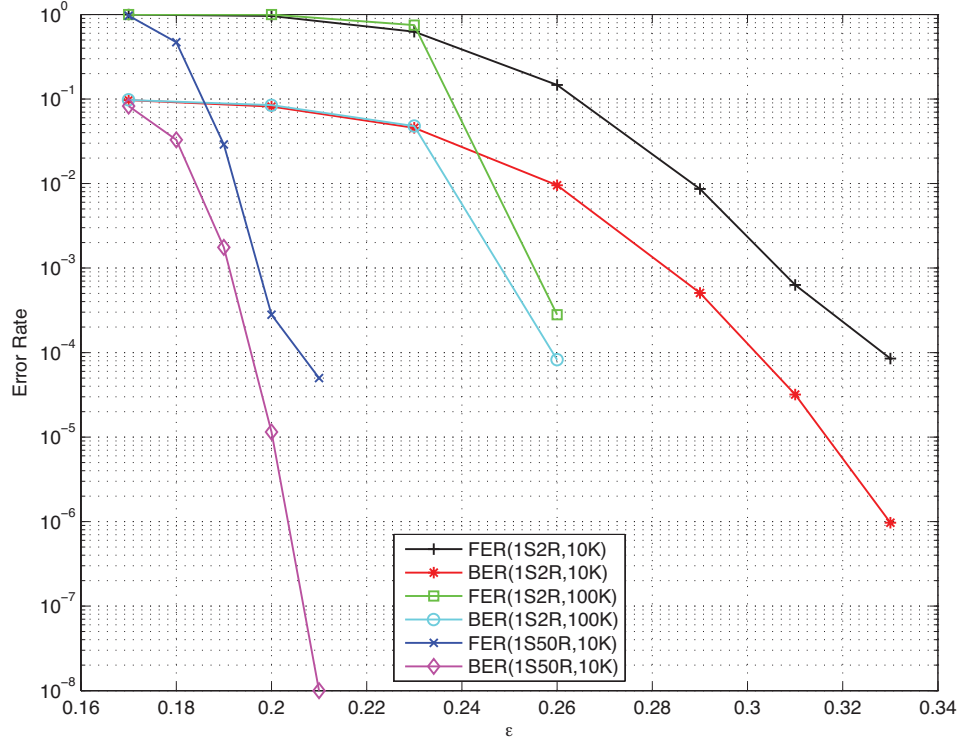


Figure 5.1: DF frame error rate and bit error rate.

function is a minimum when all the $\epsilon_{(i,j)}$ are the same. For our two relay, single source system, we assume that the normalized SNR on all the links is -1.5dB . Using our linear programming model the fraction to be forwarded by the two relays for DF is $\epsilon^{(DF)} = 0.1676$ and for DemF is $\epsilon^{(DemF)} = 0.2926$. Our 50 relay, single source model, is adjusted in such a way to have the same threshold values. Hence, for DF the normalized SNR on all the links is -9.966dB and for DemF is -10.32dB .

Figure 5.1, shows the resulting BER and FER for DF with the frame size of 10k and 100k. Since our linear programming model relies on exit chart analysis it

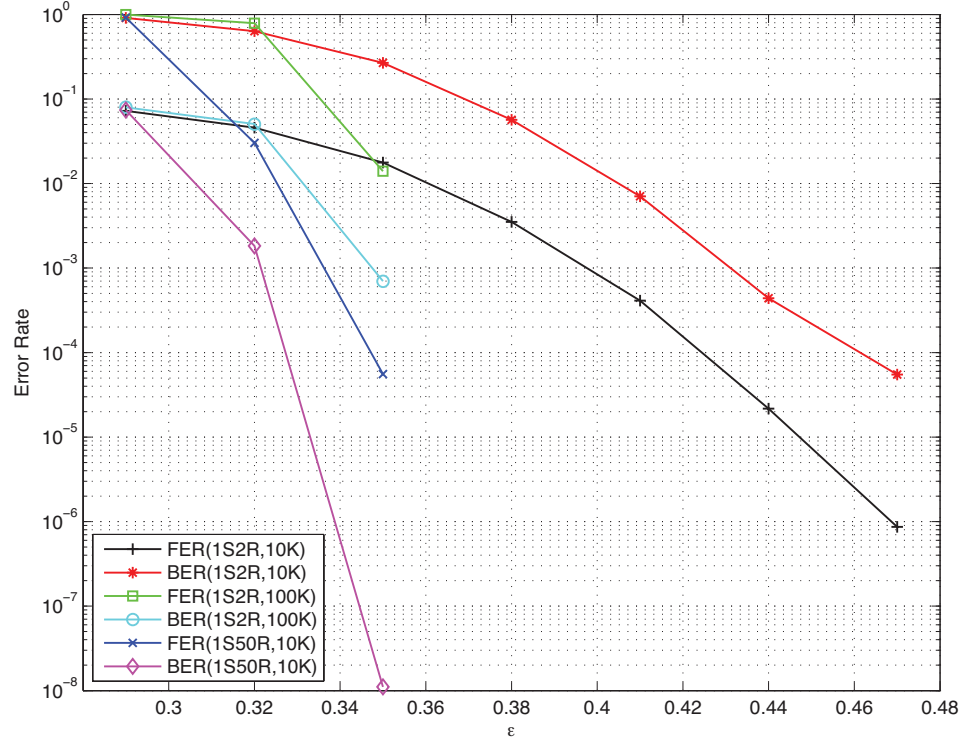


Figure 5.2: DemF frame error rate and bit error rate.

will be more accurate as the frame size or number of relays increase. This effect is captured in these graphs and it can be seen that for the case of 50 relays the curve drops close to the threshold ϵ . Similar effect can be seen in Figure 5.2 where DemF cooperative system is used.

5.3.2 Realistic Example

For our non-trivial example we consider a DemF cooperative scheme with 5 sources and 50 relays where the channel SNRs on each link is randomly selected from

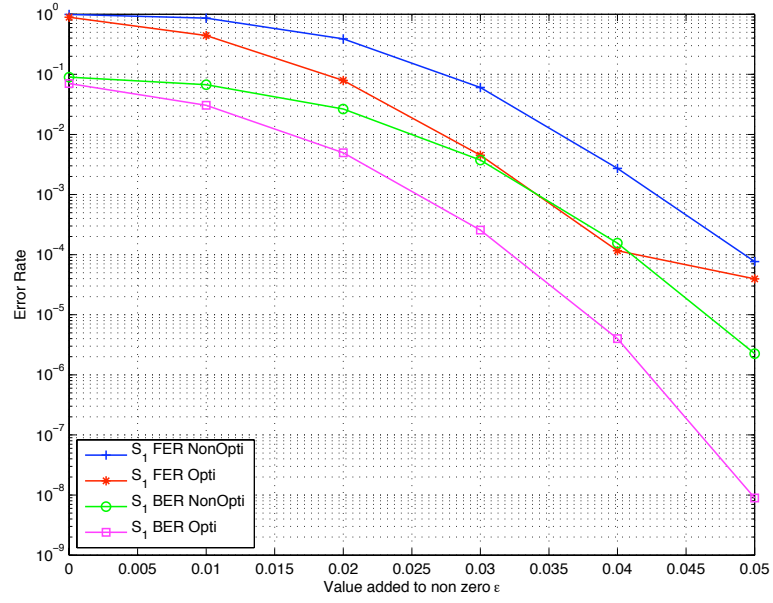


Figure 5.3: Source 1 optimal and non-optimal error rates.

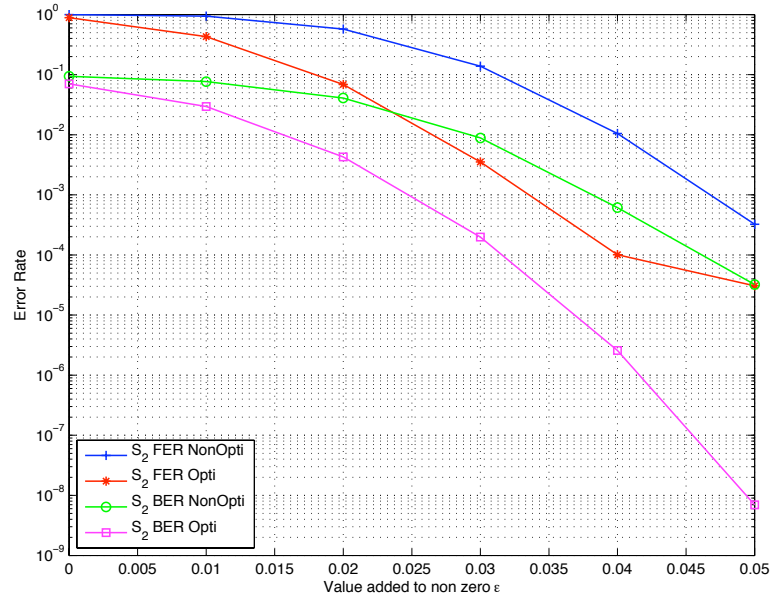


Figure 5.4: Source 2 optimal and non-optimal error rates.

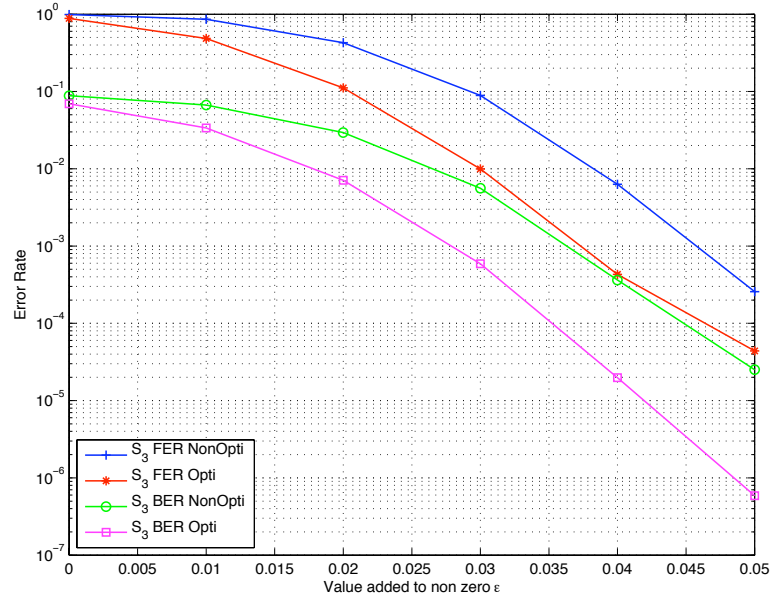


Figure 5.5: Source 3 optimal and non-optimal error rates.

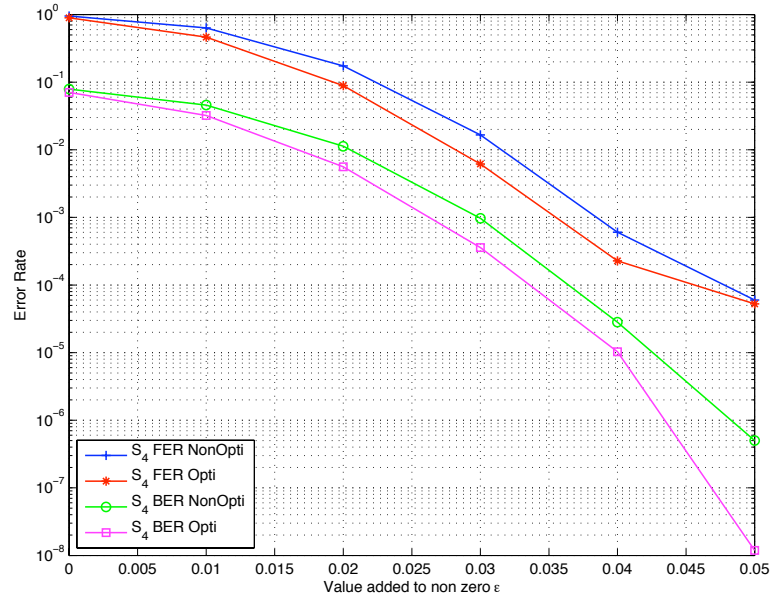


Figure 5.6: Source 4 optimal and non-optimal error rates.

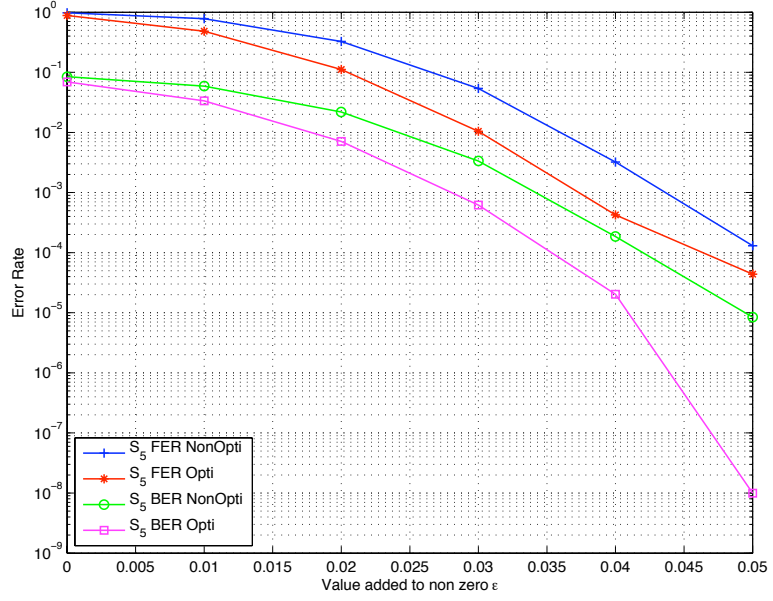


Figure 5.7: Source 5 optimal and non-optimal error rates.

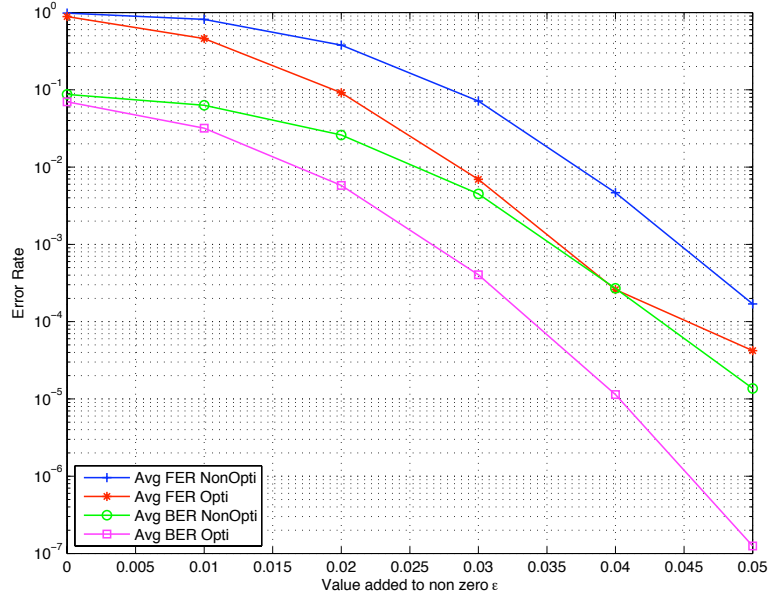


Figure 5.8: Overall system's average optimal and non-optimal error rates.

a Gaussian distribution with mean -9dB and variance 1dB . Also the maximum value that the $\epsilon_{(i,j)}$ can take is set to 0.25 instead of 1 . Since the SNRs on each link is different the problem becomes highly non-trivial. By running our linear programming model we can calculate the optimized values of $\epsilon_{(i,j)}$.

We also generate non-optimized instances by changing some of the $\epsilon_{(i,j)}$ values from the optimized solution such that most constraints are preserved. Table 5.3, at the end of the chapter, shows the fractions forwarded by each relay for each source for both the linear programming optimized and a non-optimized case. The sum of all the rows and columns of the table add up to the same number for both optimized and non-optimized case. This ensures that the constraints are the same for both the optimized and non-optimized instances.

For each source we define a measure of distance from the linear programming optimized solution as

$$\|S_i\| = \sum_{j=1}^r |\epsilon_{(i,j)} - \hat{\epsilon}_{(i,j)}| \quad (5.28)$$

where $\epsilon_{(i,j)}$ is the optimized solution and $\hat{\epsilon}_{(i,j)}$ a non-optimized instance. Therefore, using equation 5.28 and Table 5.3 we calculate this distance for each source as, $\|S_1\| = 2.1$, $\|S_2\| = 3.4$, $\|S_3\| = 2.9$, $\|S_4\| = 1.7$, and $\|S_5\| = 1.9$. Therefore, from the non-optimized instance we created, the fourth source is closest to the optimal and the second source is the farthest.

Figures 5.3-5.8 represent the result of simulating both the linear programming

optimized and the non-optimized systems given in table 5.3. The x -axis represents the value that is added to non-zero $\epsilon_{(i,j)}$ (since some relays might not be selected to forward any information for a particular source). The y -axis represents the error rates in terms of BER and FER. From all six figures we can see that for the linear programming optimized system, the BER and FER of the five sources drop quickly after the threshold value. We can also see that the optimized solution always performs better than the non-optimized solution.

Moreover, we can see that as the distance between the optimized and non-optimized systems increase the gap between their respective BER and FER curves increase. In particular, since the fourth source has the lowest distance between the optimized and non-optimized solutions, from Figure 5.6 we can see that the optimized and non-optimized curves are closer together with respect to the other graphs. Similarly, since the second source has the greatest distance between the optimized and non-optimized solutions, from Figure 5.4 we can see that the optimized and non-optimized curves are farthest with respect to the other graphs.

5.4 Summary

In this chapter we first considered a multi-source, multi-relay system that uses fractional cooperation. Then we investigated the non-trivial problem of minimizing the number of transmission bits subject to the constraint of successful decoding at the

destination. This problem was broken into two parts, successful decoding requirements and a transmission bit minimization. The successful decoding requirement was solved using Theorems 2 and 3. A linear programming model was derived for minimization of number of transmitted bits subject to successful decoding at the destination. Then through simulations, it was shown that the linear programming model presented approximates the optimized solution accurately.

Table 5.3: Optimized and non-optimized ϵ

Relay	S_1 (Opti, Non-Opti)	S_2 (Opti, Non-Opti)	S_3 (Opti, Non-Opti)	S_4 (Opti, Non-Opti)	S_5 (Opti, Non-Opti)
R_1	(0 0.25)	(0.25 0)	(0.25 0.25)	(0.25 0.25)	(0 0)
R_2	(0.25 0)	(0 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)
R_3	(0.25 0.15)	(0.25 0.25)	(0.25 0.25)	(0 0)	(0 0.1)
R_4	(0.25 0.15)	(0.25 0.25)	(0 0)	(0 0)	(0 0.1)
R_5	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)
R_6	(0 0.2)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.05)
R_7	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0)	(0 0.25)
R_8	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0.25)	(0.25 0)
R_9	(0.1107 0.1107)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.1393 0.1393)
R_{10}	(0.25 0)	(0 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{11}	(0 0.25)	(0.25 0)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{12}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0)	(0 0.25)
R_{13}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0.25)	(0.25 0)
R_{14}	(0.25 0.15)	(0.25 0.25)	(0.25 0.25)	(0 0.1)	(0.25 0.25)
R_{15}	(0 0.1)	(0.25 0.25)	(0.1674 0.1674)	(0.25 0.15)	(0.25 0.25)
R_{16}	(0.25 0.25)	(0.25 0.15)	(0 0.1)	(0.25 0.25)	(0.25 0.25)
R_{17}	(0.25 0.25)	(0.25 0.15)	(0 0.1)	(0.25 0.25)	(0.25 0.25)
R_{18}	(0 0)	(0 0.2)	(0.25 0.05)	(0.25 0.25)	(0 0)
R_{19}	(0.25 0.25)	(0.25 0.25)	(0 0)	(0 0)	(0.25 0.25)
R_{20}	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)
R_{21}	(0.25 0.25)	(0.25 0.25)	(0 0.25)	(0.25 0)	(0.25 0.25)
R_{22}	(0.25 0.25)	(0.25 0.25)	(0.25 0)	(0 0.25)	(0.25 0.25)
R_{23}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)
R_{24}	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)
R_{25}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)
R_{26}	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.1654 0.1654)	(0.25 0.25)
R_{27}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)
R_{28}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.25 0.25)
R_{29}	(0.25 0.25)	(0.0388 0.0388)	(0.25 0.25)	(0.25 0.25)	(0.2112 0.2112)
R_{30}	(0.25 0.25)	(0.0745 0.0745)	(0.25 0.25)	(0.1755 0.1755)	(0.25 0.25)

Relay	S_1 (Opti, Non-Opti)	S_2 (Opti, Non-Opti)	S_3 (Opti, Non-Opti)	S_4 (Opti, Non-Opti)	S_5 (Opti, Non-Opti)
R_{31}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.25 0.25)
R_{32}	(0 0)	(0.25 0.25)	(0.25 0.25)	(0 0)	(0 0)
R_{33}	(0 0)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{34}	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)	(0 0)
R_{35}	(0 0)	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)
R_{36}	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{37}	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{38}	(0.25 0.25)	(0 0)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{39}	(0.25 0.25)	(0.25 0.25)	(0.25 0)	(0.25 0.25)	(0 0.25)
R_{40}	(0.25 0.25)	(0.25 0.25)	(0 0.25)	(0.25 0.25)	(0.25 0)
R_{41}	(0.25 0.25)	(0.25 0)	(0 0.25)	(0.25 0.25)	(0 0)
R_{42}	(0.25 0.25)	(0 0.25)	(0.25 0)	(0.25 0.25)	(0.25 0.25)
R_{43}	(0 0.25)	(0.25 0)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{44}	(0.25 0)	(0 0.25)	(0.25 0.25)	(0.25 0.25)	(0.25 0.25)
R_{45}	(0.25 0.25)	(0.25 0)	(0 0.25)	(0.25 0.25)	(0.25 0.25)
R_{46}	(0.25 0.25)	(0 0.25)	(0.25 0)	(0.25 0.25)	(0.25 0.25)
R_{47}	(0.25 0.25)	(0.25 0)	(0 0.25)	(0.25 0.25)	(0.25 0.25)
R_{48}	(0.25 0.25)	(0 0.25)	(0.25 0)	(0.25 0.25)	(0.25 0.25)
R_{49}	(0.25 0.25)	(0.25 0.25)	(0 0)	(0 0)	(0.25 0.25)
R_{50}	(0 0)	(0.25 0.25)	(0.25 0.25)	(0 0)	(0.25 0.25)

6 Conclusion and Future Work

In this thesis, we considered different techniques for optimizing a cooperative WSN system. In particular, we proposed two novel schemes, one based on the correlation of measured data, and the other based on analytic optimization using linear programming. To study the proposed techniques two system models were presented. One was a multi-relay, multi-correlated source model and the other a multi-relay, multi-source model.

In the multi-relay, multi-correlated source model the source could compress its information bits before transmission thereby reducing the number of transmission bits. The relays forwarded all the transmission bits from the source to the destination. The destination used both the source's transmission as well as the relays transmission to decode the compressed bits, transmitted by the source. It then used the correlated source's information bits to decompress the original information bits at the source.

It was shown using Slepian-Wolf compression the diversity of such a multi-relay,

multi-correlated source can be increased by adding both relay and correlated sources at practical SNRs. Also through Theorem 1 it was shown that as the SNR goes to infinity the diversity on the system dropped to minimum of number of relays and number of correlated sources.

In the multi-source, multi-relay model, we used fractional cooperation where each relay would randomly select a small fraction of the source's transmission bits to retransmit to the destination. We also assumed LDPC codes were used for channel coding. The relays in this model used two type of cooperative schemes, namely DF and DemF. If DF was used by the relays for simplicity we assumed powerful and capacity approaching codes could be used over the source to relay links which resulted in perfect recovery of source's data at the relays. Also if DemF was used for simplicity we assumed such codes could be used over the relay to destination links which would result in recovery of the relays' demodulated bits at the destination.

Using this multi-source, multi-relay system we derived a linear programming model based on EXIT chart analysis that would optimize the number of transmission bits while ensuring successful decoding at the destination. Since EXIT chart analysis is an approximation method, the linear programming finds the approximation of the optimized solution. Through simulations it was shown that this approximation is a close estimate of the optimized solution.

For future work fractional cooperation can be used with our multi-relay, multi-correlated source model and the effect of compression can be studied on the diversity order of the system. Moreover, density evolution can be used instead of EXIT charts to derive a more accurate linear programming model. Using density evolution other error correcting codes such as RA codes can be possibly used for channel coding.

Bibliography

- [1] C.-Y. Chong and S. P. Kumar, “Sensor networks: evolution, opportunities and challenges,” *Proc. IEEE*, vol. 91, pp. 1247–1256, Aug. 2003.
- [2] P. Zhang, C. M. Sadler, S. Lyon, and M. Martonosi, “Hardware design experiences in ZebraNet,” in *Proc. 2nd International Conference on Embedded Network Sensor Systems*, Baltimore, MD, pp. 227–238, Nov. 2004.
- [3] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, “A wireless sensor network for structural monitoring,” in *Proc. 2nd International Conference on Embedded Network Sensor Systems*, Baltimore, MD, pp. 13–24, Nov. 2004.
- [4] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, “Monitoring Volcanic Eruptions with a Wireless Sensor Network,” *Proc. 2nd European Workshop Wireless Sensor Networks (EWSN 05)*, Jan. 2005.
- [5] John A. Stankovic, “Research Challenges for Wireless Sensor Networks,” *SIGBED Review: Special Issue on Embedded Sensor Networks and Wireless Computing*, 1(2), July 2004.
- [6] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” *In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.
- [7] T.V. Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” *The First ACM Conference on Embedded Networked Sensor Systems (Sensys’ 03)*, Los Angeles, CA, USA, November, 2003.
- [8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocol for wireless microsensor networks,” in *Proc. Hawaii Int. Conf. System Sci.*, Maui, HI, Jan. 2000.

- [9] I. Demirkol, C. Ersoy, F. Alagöz, “MAC protocols for wireless sensor networks: A survey,” *IEEE Communications Magazine* 44 (4) (2006), 115–121.
- [10] G. Halkes, T. V. Dam, K. Langendoen, “Comparing energy-saving MAC protocols for wireless sensor networks,” *Mobile Networks and Applications* 10 (5) (2005), 783–791.
- [11] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” *IEEE Tran. Parallel and Distributed Systems*, vol. 13, no. 9, September 2002.
- [12] Q. Han, S. Mehrotra, and N. Venkatasubramanian, “Energy efficient data collection in distributed sensor environments,” in *Proc. IEEE ICDCS04*, Mar. 2004, pp. 590–597.
- [13] X. Tang and J. Xu, “Extending network lifetime for precision constrained data aggregation in wireless sensor networks,” *INFOCOM*, 2006.
- [14] A. Sendonaris, E. Erkip, and B. Aazhang, “User cooperation diversity-part I: system description,” *IEEE Trans. Commun.*, vol. 51, pp. 1927–1938, Nov. 2003.
- [15] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.
- [16] S. Pradhan and K. Ramachandran, “Distributed source coding using syndromes (DISCUS): Design and construction,” in *Proc. IEEE DCC*, pp. 158–167, March 1999.
- [17] A. W. Eckford, J. P. K. Chu, and R. S. Adve, “Low complexity and fractional coded cooperation for wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1917–1929, May 2008.
- [18] M. Ardakani and F. R. Kschischang, “A more accurate one-dimensional analysis and design of LDPC codes,” *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.
- [19] T. M. Cover and A. A. E. Gamal, “Capacity Theorems for the Relay Channel,” *IEEE Trans. Info. Theory*, vol. 25, no. 5, Sept. 1979, pp. 572–584.
- [20] A. Sendonaris, E. Erkip, and B. Aazhang, “User cooperation diversity, Part II: Implementation aspects and performance analysis,” *IEEE Trans. Commun.*, vol. 51, pp. 1939–1948, Nov. 2003.

- [21] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 68–73, October 2004.
- [22] J. N. Laneman, G. W. Wornell, and D. N. C. Tse, "An Efficient Protocol for Realizing Cooperative Diversity in Wireless Networks," *Proc. IEEE ISIT*, Washington, DC, June 2001, p. 294.
- [23] J. N. Laneman, D. N. C. Tse, and G.W.Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
- [24] D. Chen and J. N. Laneman, "Modulation and demodulation for cooperative diversity in wireless systems," *IEEE Trans. on Wireless Commun.*, vol. 5, no. 7, pp. 1785–1794, Jul. 2006.
- [25] J. P. K. Chu and R. S. Adve, "Implementation of co-operative diversity using message-passing in wireless sensor networks," in *Proc. IEEE Global Communications Conference*, St. Louis, MO, pp. 1167–1171, Dec. 2005.
- [26] K. S. Gomadam and S. A. Jafar, "On the capacity of memoryless relay networks," in *Proc. IEEE International Conf. on Communications*, Istanbul, Turkey, pp. 1580–1585, June 2006.
- [27] T. E. Hunter and A. Nosratinia, "Cooperative Diversity through Coding," *Proc. IEEE ISIT*, Laussane, Switzerland, July 2002, p. 220.
- [28] T. E. Hunter and A. Nosratinia, "Diversity through coded cooperation," *IEEE Trans. Wireless Commun.*, no. 2, Feb. 2006.
- [29] Bletsas, H. Shin, M. Win, and A. Lippman, "Cooperative diversity with opportunistic relaying," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2006.
- [30] K. Azarian, H. E. Gamal, and P. Schniter, "On the achievable diversity-multiplexing tradeoff in half-duplex cooperative channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4152–4172, Dec. 2005.
- [31] Y.-W. Hong and A. Scaglione, "Energy-efficient broadcasting with cooperative transmission in wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, Oct. 2006.

- [32] M. Janani, A. Hedayat, T. E. Hunter, and A. Nosratinia, "Coded cooperation in wireless communications: space-time transmission and iterative decoding," *IEEE Trans. Signal Processing*, vol. 52, pp. 362–371, Feb. 2004.
- [33] A. Stefanov and E. Erkip, "Cooperative coding for wireless networks," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1470–1476, Sept. 2004.
- [34] Z. Zhang and T. M. Duman, "Capacity-approaching Turbo coding and iterative decoding for relay channels," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1895–1905, Nov. 2005.
- [35] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3723–3739, 2007.
- [36] J. Hu and T. M. Duman, "Low density parity check codes over half-duplex relay channels," in *Proc. IEEE Int. Symp. Info. Theory*, Seattle, WA, pp. 972–976, July 2006.
- [37] A. Chakrabarti, A. De-Baynast, A. Sabharwal, and B. Aazhang, "Low density parity check codes for the relay channel," *IEEE Sel. Areas Commun.: Special Issue on Cooperative Commun. and Netw.*, vol. 25, no. 2, pp. 280–291, Feb. 2007.
- [38] S. S. Pradhan and K. Ramchandran, "Distributed Source Coding: Symmetric Rates and Applications to Sensor Networks," *Proc. IEEE Data Compression Conference*, pp. 363–372, Snowbird, UT, March 2000.
- [39] J. Bajcsy and P. Mitran, "Coding for the SlepianWolf problem with turbo codes," in *Proc. IEEE Globecom 01*, vol. 2, Nov. 2001, pp. 1400–1404.
- [40] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417–419, 2001.
- [41] A. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information using low-density parity-check codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 2, pp. 1300–1304, November 2002.
- [42] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Joint source channel coding of binary sources with side information at the decoder using IRA codes," in *Proc. IEEE Multimedia Signal Processing Workshop*, pp. 53–56, St. Thomas, Virgin Islands, USA, December 2002.

- [43] W. Zhong and J. Garcia-Frias, "LDGM Codes for Channel Coding and Joint Source-Channel Coding of Correlated Sources," *EURASIP Journal on Applied Signal Processing*, pp. 942–953, May 2005.
- [44] R. Hu and J. Li, "Exploiting slepian-wolf codes in wireless user cooperation," in *IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, vol. 2005, pp. 275–279, 2005.
- [45] Y. Zhao, W. Zhong, and J. Garcia-Frias, "Transmission of correlated senders over a Rayleigh fading multiple access channel," *Signal Processing*, vol. 86, no. 11, pp. 3150–3159, Nov. 2006.
- [46] M. Sartipi and F. Fekri, "Source and channel coding in wireless sensor networks using LDPC codes," in *Proc. Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pp. 309–316, 2004.
- [47] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'Turbo-like' codes," in *Proc. 36th Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL, USA, pp. 201–210, Sept. 1998.
- [48] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. 1993 ICC*, pp. 1064–1070, May 1993.
- [49] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.
- [50] R. G. Gallager, "Low-Density Parity-Check Codes," *MIT Press*, Cambridge, MA, 1963.
- [51] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," *Proc. 2nd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 1–8, Sept. 2000.
- [52] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Inform. Theory*, vol. IT-47, no. 2, pp. 498–519, 2001.
- [53] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate repeat-accumulate codes," in *Proc. IEEE International Symposium on Information Theory*, Chicago, IL, p. 505, July 2004.

- [54] G. Yue, X. Wang, and M. Madihian, “Design of rate-compatible irregular repeat-accumulate codes,” *IEEE Transactions on Communications*, vol. 55, pp. 1153–1163, June 2007.
- [55] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” *IEE Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.
- [56] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inform. Theory*, vol. IT-45, no. 2, pp. 399–431, March 1999.
- [57] T. J. Richardson, M. Shokrollahi, and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inform. Theory*, pp. 619–637, Feb. 2001.
- [58] T. J. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inform. Theory*, pp. 599–618, Feb. 2001.
- [59] T. J. Richardson and R. Urbanke, “Efficient encoding of low-density parity-check codes,” *IEEE Trans. Inform. Theory*, pp. 638–656, Feb. 2001.
- [60] J. V. Michalowicz, J. M. Nichols, and F. Bucholtz, “Calculation of differential entropy for a mixed Gaussian distribution,” *Entropy*, vol 10, no. 2, pp. 200–206, 2008.