COLLABORATIVE DECODING: ACHIEVING COOPERATIVE DIVERSITY IN WIRELESS NETWORKS USING SOFT-INPUT SOFT-OUTPUT DECODERS

By

ARUN AVUDAINAYAGAM

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2006

To my parents, Hema and Nayagam, and my wife, Krithi.

ACKNOWLEDGMENTS

Numerous individuals have directly and indirectly contributed to my graduate experience in general and this dissertation in particular. The following few words cannot fully convey my gratitude for all the help and guidance I have received over the years.

Any acknowledgment should start with my advisor, Dr. John Shea. I have greatly benefited from his expertise, valuable insights and approach to tackling problems. His mentoring has made me strive to become a better engineer and researcher. I thank him for all his help, guidance, and patience.

I would also like to thank Dr. Tan Wong for many fruitful and interesting discussions. Special thanks go to Dr. Fang for taking me under his wing when I first began my graduate studies and for encouraging me to continue at UF for my Ph.D. I must also mention my fellow graduate students who made my stay at WING a very memorable experience. Abhinav Roogta, Hanjo Kim and Jangwook Moon deserve a special mention for various successful and some not-so-successful (but interesting nonetheless) collaborations. I am also grateful to the Office of Naval Research for sponsoring part of my research.

This dissertation would not have been possible without the support and encouragement of my parents, Hema and Nayagam. I thank you for all your sacrifices that has moulded me into this person that I am today.

And to end with a “last but not least” cliché; my most loving and sincere thanks go to my wife, Krithi. She was probably the most affected by my schedule during the course of this work and I cannot express how invaluable her support has been over the years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

COLLABORATIVE DECODING: ACHIEVING COOPERATIVE DIVERSITY IN WIRELESS NETWORKS USING SOFT-INPUT SOFT-OUTPUT DECODERS

By

Arun Avudainayagam

May 2006

Chair: John M. Shea
Major Department: Electrical and Computer Engineering

Spatial diversity techniques use multiple transmit and receive antennas (antenna arrays) to improve performance in wireless environments without requiring additional bandwidth or loss in throughput. However, the spacing between antenna elements depends on the carrier wavelength, and this might often exceed the size of modern mobile radios. Thus, alternative approaches are required to harness spatial diversity in small terminals. Recently, cooperation among users has been proposed as an alternative means to achieve diversity in wireless networks with small radios. In this proposal, we develop collaboration schemes for scenarios in which radios in a network cooperate to improve performance. Since radios in a wireless network are typically separated in space, the different nodes can pool their resources together to form a virtual antenna array. The elements of the antenna array can then collaborate by exchanging information with each other in order to achieve diversity gains. The information exchanged by the collaborating nodes is called cooperation overhead. Our schemes are targeted towards bandwidth-limited systems in which the cooperation overhead should be small. We provide a framework called collaborative decoding to help design schemes that have low cooperation overhead and still achieve performance close to that of optimal combining schemes. We present a collaboration technique called

improved least-reliable bits (I-LRB) collaborative decoding that provides a higher level of adaptation than previously proposed cooperative schemes. The I-LRB scheme utilizes reliability information and information about competing paths in soft-input soft-output decoders to adaptively select the amount of information that is needed to correct a particular part of a message, as well as which bits should be exchanged. Simulation results show that the proposed approach offers a significant performance advantage over existing cooperation techniques. For example, I-LRB can provide a 30%-60% improvement in throughput with respect to traditional cooperation schemes in bandwidth-constrained systems.

CHAPTER 1
INTRODUCTION

Multipath fading is one of the most common problems associated with wireless communications. Reflections from and refraction through various objects in the channel cause multiple attenuated and delayed copies of the transmitted signal to constructively or destructively combine at the destination. Fading can cause severe fluctuations in the signal-to-noise ratio (SNR), which in turn affects system performance. Various techniques like equalization, error-control coding and diversity combining are used independently or in conjunction to combat fading. Diversity techniques typically improve performance by making multiple independent copies of the transmitted signal available to the destination. These multiple copies can then be optimally combined using various techniques like maximal-ratio combining (MRC) or equal-gain combining (EGC) [1]. Temporal diversity is typically achieved through error-control coding. Frequency diversity is achieved by using various physical layer techniques like frequency hopping or multi-carrier modulation.

Recent advances in space-time coding have proven that processing in the spatial domain is an efficient approach to achieve diversity in delay-limited and bandwidth-limited applications. Space-time codes exploit the multipath nature of the wireless medium to combat the detrimental effects of fading. Spatial diversity can be achieved by using multiple antennas at the transmitter and/or receiver. However, for significant gains the spacing between the antenna elements should be at least half the wavelength of the RF carrier. This prohibits the use of antenna arrays in most small, portable radios. This is the reason why antenna arrays are not used in the cellular downlink or in ad hoc networks.

In recent years, a number of *network-assisted* diversity techniques have been studied [2-9]. In these approaches, the users depend on the network to provide diversity at the physical layer. The broadcast nature of the wireless channel, wherein any node within range of the transmitter can listen to the transmission, is exploited in this network-based approach to spatial processing. This property of the wireless medium is referred to as the *wireless broadcast advantage* (WBA) [10]. Since nodes in a network are spatially separated, the different nodes that receive the transmission from a source can be considered to be the elements of a virtual *antenna array*. Since the elements are not physically connected, this is referred to as a *distributed array*. The different users can then collaborate with each other to achieve diversity gains. Diversity achieved when users in a network collaborate to improve each other's performance has been termed *cooperative diversity* or *multiuser diversity*. We will to use the term *user cooperation* [2] to refer to the process of collaboration between the various users of the network.

A lot of work on user cooperation (including most information-theoretic and a few practical schemes) is based on simple repetition coding [2-6]. The basic idea of these schemes is that any user that listens to the transmission from the source forwards the information (either the coded bits after decoding or quantized versions of the received symbol values) to the destination. The amount of information exchanged by the collaborating nodes is referred to as the cooperation overhead. The use of repetition codes makes these techniques inefficient in terms of the overhead. Cooperation through the use of more powerful error correction codes has also been proposed in [7, 8]. The disadvantage of these schemes is that they do not easily scale to large networks (with more than two cooperating nodes).

1.1 Objectives and Main Contributions

Since the wireless medium is bandwidth-limited, the cooperation overhead is a very important issue and one that has not been addressed so far in the literature. The objective of this work is to provide a framework to help develop cooperation strategies that are efficient in terms of the cooperation overhead. It is also important that the collaboration techniques extend naturally to multiple cooperating nodes. Using this framework, called *collaborative decoding*, we develop strategies that provide close-to-optimal performance with only a fraction of the overhead required by conventional cooperation schemes. Unlike previous cooperation strategies, collaborative decoding provides a convenient approach to trade performance for overhead, and collaborative decoding scales easily to multiple cooperating nodes.

Conventional coded cooperation strategies are based on distributed encoding of a message among the collaborating nodes. Collaborative decoding is based on a distributed decoding of an encoded message among collaborating nodes. All schemes in this dissertation use soft-input soft-output (SISO) decoders. The magnitude of the output of the SISO decoder is called the *reliability* and is an indication of the correctness of the decoded bit. In all our schemes, the nodes exchange information for only a fraction of the message bits based on the reliability information. The design of a collaborative decoding scheme then consists of the choice of the bits to be exchanged and what information is to be exchanged among the nodes. In existing cooperation strategies, the messages exchanged by collaborating nodes is predetermined and fixed. Collaborative decoding adapts the content involved in cooperation to each channel instantiation. Thus, by tailoring the messages exchanged by collaborating nodes to the potential bit errors, collaborative decoding aims to lower the cooperation overhead.

The contributions of this work are two-fold. First the dissertation furthers understanding of the fundamental operation of the maximum *a posteriori* (MAP) convolutional decoder with the max-log-MAP implementation. In this context the main contributions in this dissertation are the following:

1. We provide a closed-form approximation to the density and distribution function of reliabilities at the output of a max-log-MAP SISO decoder. This closed-form estimate is parameterized by a single numerical quantity that can be determined analytically. The estimates can be can be used to analyze reliability-based systems.

2. Using these closed form approximations we provide an approximation to the bit error rate of a max-log-MAP decoder in terms of a single Q-function. This is the first such result in the literature.

3. We investigate the correlated nature of the soft-output of a max-log-MAP decoder. The max-log-MAP decoder computes the soft-output for a trellis section by considering the maximum-likelihood (ML) path and a competing path that differs from the ML path in the input for that trellis section. We show that the time-correlated reliabilities occur in a max-log-MAP decoder because the same competing path is considered in computing the soft-output for adjacent bits.

4. We provide an efficient approach to explicitly compute the ML and competing paths by using computations that are already performed in the decoder.

The second area of contribution in this dissertation is in applying the knowledge gained about the operation of the decoder to the design of collaborative decoding. The main contributions in this area are the following:

1. We design a cooperation strategy called *improved least-reliable-bits* (I-LRB) collaborative decoding that has the following features:
   - Achieves full diversity in the number of cooperating nodes.
   - Requires a fraction of the overhead involved in full maximal-ratio-combining.
   - Easily scales to multiple relays.
   - Offers the ability to easily trade performance for overhead.

2. I-LRB is adaptive in two levels. The trellis sections for which information is combined are adapted to each channel instantiation. For each trellis section, the amount of information combined is adapted to the reliability of that trellis section.

3. I-LRB exploits correlated bit reliabilities by computing competing paths in the decoder, and utilizes knowledge of the competing paths to reduce cooperation overhead.

1.2 Outline of the Dissertation

This dissertation is organized as follows. In Chapter 2, we summarize various important results in the literature that pertain to the idea of user cooperation. These results provide a basis for comparison with our schemes, and makes it easier to emphasize the distinction between our approach and the existing schemes. We provide an introduction to soft-input soft-output (SISO) decoders in Chapter 3. A good understand of SISO decoding is important to understand the operation of our techniques. We also provide a mathematical characterization/approximation of the statistics of the SISO decoder output. In Chapter 4, we introduce the concept of *collaborative decoding* in which various users cooperate in the decoding process and achieve spatial diversity. Though these schemes are suitable for AWGN channels, we show that these schemes are not suitable for fading channels. We develop guidelines to help design cooperative diversity protocols for fading channels. We also develop various guidelines for the design of cooperative protocols in this chapter. In Chapter 5, we study the correlated nature of the output of the SISO decoder. We show that error events encountered in the SISO decoder can be used to capture this correlation. We also present a technique to efficiently compute these error events with minimal modifications to the decoder. Based on the design guidelines presented in Chapter 4 and the technique presented in Chapter 5, in Chapter 6 we design a collaboration scheme that utilizes the correlated output of the SISO decoder to reduce the cooperation overhead. The dissertation is concluded in Chapter 7.

CHAPTER 2
BACKGROUND AND RELATED RESEARCH

In this chapter the idea of cooperative communications is introduced and important references relating to this broad area are summarized. The objective of the chapter is to familiarize the reader with the different approaches to user cooperation and the various issues involved in the design of such schemes. We start by introducing the very first ideas of cooperation and some recent techniques that were proposed in the information theory community. We then highlight some of the more practical approaches that have been studied in recent years.

2.1 Information-Theoretic Strategies

Studies on the relay channel in the late 1960s can be considered to contain the first instances of cooperation. The relay channel (shown in Figure 2-1) was first introduced and studied by van der Meulen in 1968 [11]. In this setting, an intermediate node, called the *relay*, listens to the transmission from the source to the destination, processes this information, and transmits additional information about the initial transmission to the destination. The destination uses the relay transmission to resolve any ambiguity about the original transmission. The transmission from the relay is done jointly with the source; i.e., relay transmission for block i is super-imposed on block $i+1$ sent by the source. Thus, the relay cooperates with the source to improve reception at the destination.

In 1979, Cover and El Gamal [12] studied the capacity of relay channels under different scenarios. Cover and El Gamal put forward three different approaches to achieve user cooperation in a relay channel. In *facilitation*, the relay passively aids the communication between a source and destination by not transmitting, thereby reducing interference to the original communication. In *cooperation*, the relay decodes the transmission from

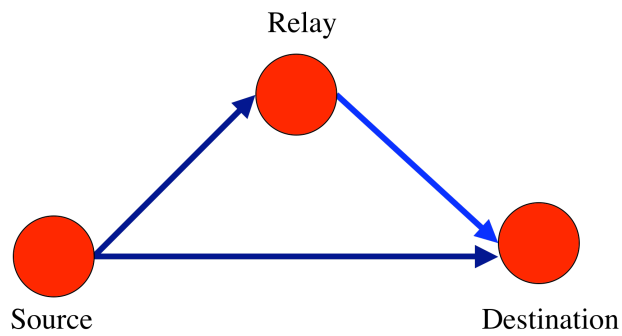

Figure 2-1: The relay channel.

the source and provides additional information about the initial transmission to the destination to aid in recovering the original message. In *observation*, the relay just forwards the observed symbol values to the destination.

The observation scheme was introduced to overcome a problem with the cooperation scheme. In cooperation, the relay partitions the set of valid codewords into bins using the Slepian-Wolf partitioning technique [13] and transmits the bin index of the partition containing the source message. The destination then uses the set of codewords in the corresponding partition to resolve any ambiguity about the transmitted message. However, the computation of the bin index requires correct decoding at the relay, and thus this scheme is limited by the rate between the source and the relay. The observation scheme can overcome this problem because it does not require correct decoding at the relay.

Recent studies of the relay channel can be found in [14-18]. The Cover and El Gamal scheme is extended to multiple nodes in Gupta and Kumar [14]. In Cover and El Gamal [12] and Gupta and Kumar [14], the nodes cooperate using block-Markov encoding and Slepian-Wolf partitioning [13]. The destination decodes the message using two transmitted blocks; i.e., upon receiving block i, the decoder estimates the message in the previous block $(i-1)$. The nodes use codebooks of different sizes at the relay and the source, making it difficult to extend the this strategy to multiple nodes. A technique that uses codebooks of the same size is proposed in Willems [19] and is extended to multiple relays in Kramer *et al.* [15]. The use of equal size codebooks makes it easier to extend

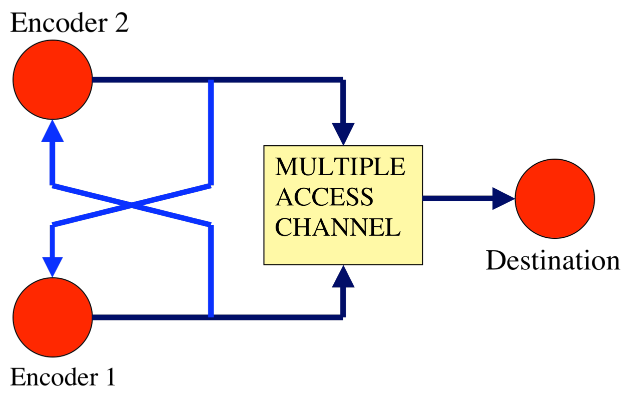

Figure 2-2: The multiple access channel with cooperating encoders.

this scheme to multiple relays. Another advantage of this scheme is that the use of the complicated Slepian-Wolf binning technique is avoided. The disadvantage manifests itself as a large decoding delay incurred due to the *backwards decoding* technique used at the destination. The coding is done over B blocks and the decoding process can start only after receiving all the B blocks. The decoding is done in the backward direction, starting with the last block and ending at the first block. A strategy that avoids Slepian-Wolf partitioning and in which all relays use codebooks of the same size but in which the decoder lag is reduced to just one block (as in the Cover and El Gamal scheme) is proposed in Xie and Kumar [16]. An extension of this scheme to multiple relays has also been studied [17, 18].

The information theory community also studied cooperation from the view of a multiple access channel. In the early 1980s user cooperation in the setting of a multiple access channel was studied [20, 21]. Unlike the relay channel, both the encoders have data to send to the destination. In Willems [20], the users cooperate over a separate channel before sending their messages. The need for a separate set of channels for cooperation is eliminated in Willems and van der Meulen [21]. The scenario considered is shown in Figure 2-2. The cooperative transmission works as follows. Each encoder cooperates with the other and learns the codeword that the other encoder is going to send in the current transmission. Thus, in transmission i, each encoder has knowledge of all the previous $i-1$ codewords of the other encoder. The encoding process works as follows. Without loss of generality, we assume that encoder 1 starts the encoding process for transmission i. Encoder 1 forms its codeword as a function of its own data and the codewords sent by encoder 2 in all previous transmissions. Encoder 2 learns the current codeword of encoder 1 and forms its own codeword as a function of its own data and the current and previous codewords of encoder 1. This strategy requires perfect cooperation between the encoders; i.e., each encoder should learn the codewords of the other encoder without errors. Note that the model of Willems *et al.* reduces to the relay channel if one of the encoders has no data of its own to send. Then that particular encoder will act as a relay for the other encoder. The authors prove that encoder cooperation increases the capacity of the multiple-access channel when compared to non-cooperative transmission.

The techniques described in this section depend on information theoretic concepts like random binning, typical-set and backwards decoding. These are not viable for practical implementation. In the next section, we review a few practical cooperation strategies based on simple repetition coding ideas.

2.2 Repetition Based Cooperation

After the work of Willems, the idea of user cooperation was largely ignored until the late 90s. The advances in space-time coding [22, 23] proved that exploiting spatial diversity with the use of multiple transmit and receive antennas can lead to significant improvements in data rate. However, small portable radios do not permit the use of multiple antennas. User cooperation is a natural way to achieve spatial diversity by pooling the resources of many radios, each equipped with a single antenna. User cooperation in a wireless scenario was first investigated by Sendonaris, Erkip and Aazhang [2, 5, 6, 24]. Sendonaris *et al.* study the idea of user cooperation in the setting of a cellular CDMA system. The system model they use is identical to the model shown in Figure 2-2. However, due to the wireless setting the links between the two encoders (the *cooperation channels*) are imperfect channels that experience fading.

The cooperation model is based on the idea of Willems *et al.*, wherein a codeword sent by one user depends on the codeword sent by the other user. In Sendonaris *et al.* [2], the authors consider more practical aspects of user cooperation in this scenario. In particular, the authors begin by using information theory to evaluate the effects of cooperation on outage probability, diversity, and cellular coverage. Then the authors propose and analyze the following practical cooperation scheme.

Let $X_i(t)$ and $c_i(t)$ denote the signals transmitted by user i and the spreading code used by user i at time t. Then the signals transmitted by the two users are

$$\begin{aligned} X_1(t) &= a_{11}b_1^{(1)}(t)c_1(t),\ \ a_{12}b_1^{(2)}(t)c_1(t),\ \ a_{13}b_1^{(2)}(t)c_1(t)+a_{14}\hat{b}_2^{(2)}(t)c_2(t) \\ X_2(t) &= a_{21}b_2^{(1)}(t)c_2(t),\ \ a_{22}b_2^{(2)}(t)c_2(t),\ \ a_{23}\hat{b}_1^{(2)}(t)c_1(t)+a_{24}b_2^{(2)}(t)c_2(t), \end{aligned}$$

where b_i^j is user i's j^{th} bit and $\hat{b}_i^{(j)}$ is the corresponding estimate at the other node. The parameters $\{a_{ji}\}$ represent how much power is allocated to each bit. Thus, in the first two periods each user transmits its own bits. In the third period, each encoder sends a linear combination of its own bits of the second period and its estimate of its partner's bits of the second period. Since the basis of cooperation is an estimate of the other encoder's codeword, the authors allocate rate and power to guarantee error-free communication on the cooperation channels. Thus, each encoder's estimate of the codeword of the other encoder is perfect. With this system and cooperation model the authors prove that it is possible to increase the maximum sum-capacity of the network if the transmitter has knowledge of the channel phase.

Laneman *et al.* [3, 4], introduce two broad classes of cooperation techniques called *decode-and-forward* and *amplify-and-forward*. Their approach is similar to the relay-channel-based cooperation techniques of Cover and El Gamal. In the *decode-and-forward* scheme, the relays first decode the source message and then forward the re-encoded information bits to the destination. This is akin to Cover *et al.*'s *cooperation* scheme with the

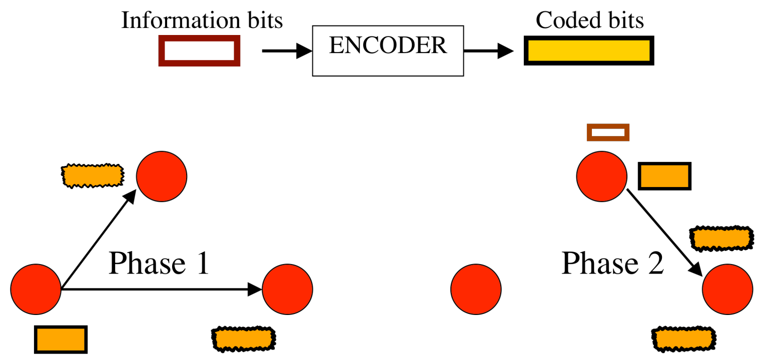

Figure 2-3: The *decode-and-forward* cooperation scheme.

relay transmission consisting of re-encoded information bits instead of Slepian-Wolf bin indices.

The operation of the *decode-and-forward* scheme is shown in Figure 2-3. The scheme works in two phases as in the conventional relay channel. In the first phase, the source encodes the information bits (represented by an empty rectangle), and transmits the coded bits (represented by a solid rectangle). The destination and the relay receive noisy versions of the coded bits. In the second phase, the relay decodes the information bits, re-encodes them using the same code used at the source. The re-encoded codeword is then sent to the destination.

At the end of the second phase, the destination has two independent noisy copies of the codeword sent by the source (assuming that the relay decoded correctly). These two independent copies can be combined using various combining schemes like maximal-ratio combining (MRC) or equal-gain combining (EGC) [1]. This scheme can be considered as an instance of rate-$1/2$ repetition coding since the destination receives two independent copies of the same message. However, the repetition is done by the relay instead of the source itself.

The effectiveness of cooperative communication schemes is often assessed in terms of the effects on capacity and on the diversity achieved. The ability to achieve diversity is quantified in terms of the *diversity order*, which is defined as the asymptotic slope of the bit or block error rate curve on log-log scale. If there are M elements in an antenna array, then

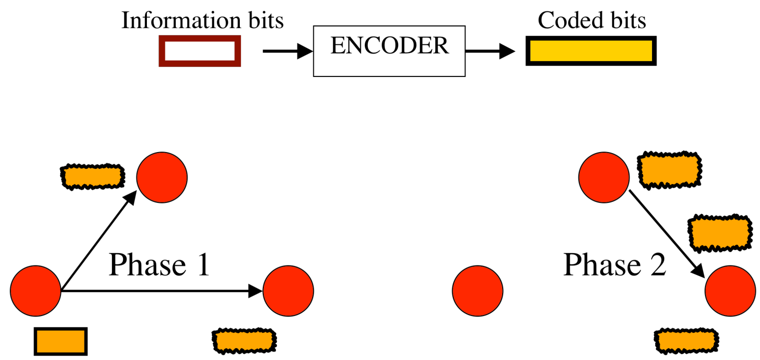

Figure 2-4: The *amplify-and-forward* cooperation scheme.

at most M independent copies of the message are received. Hence, the maximum diversity order that can be achieved is M. Any scheme that achieves diversity order that is equal to the number of cooperating nodes is said to achieve *full* diversity.

As in the *cooperation* scheme proposed by Cover *et al.*, the *decode-and-forward* scheme depends on correct decoding at the relay. It is proved that the *decode-and-forward* schemes can provide all of the capacity benefits offered by cooperative transmission, but cannot achieve full diversity (in the number of collaborating nodes) [3, 4]. The reason is that a diversity channel is created on the link between the relay and the destination only when the relay decodes successfully, and hence this scheme is limited by the channel between the source and the relay. Note that the scheme of Sendonaris *et al.* also falls under the *decode-and-forward* class of cooperation schemes since a perfect estimate of the partner's bits is required for cooperation.

The operation of the *amplify-and-forward* scheme is illustrated in Figure 2-4. The first phase is identical to first phase in the *decode-and-forward* scheme. In the second phase, the relay does not perform decoding. Instead, the relay amplifies/scales its observations (the received symbol values) subject to a power constraint and forwards it to the destination. If $\mathbf{y}$ is the message received from the source, the transmission of the relay can be expressed as

$$\mathbf{x} = \beta \mathbf{y}, \tag{2.1}$$

where the amplification factor β is constrained by

$$\beta \leq \sqrt{\frac{P}{|\alpha|^2 P + N_0}}. \tag{2.2}$$

Here P is the maximum transmit power of the source (and relay), α represents the fading amplitude between the source and relay, and $N_0/2$ is the noise variance. On average, the scaling factor β constrains the transmission power of the relay to its maximum allowed value P. Unlike the *decode-and-forward* scheme, the relay also amplifies its own receiver noise. This is identical to Cover *et al.*'s *observation* scheme if the amplification factor is set to unity i.e., if the relays have no power constraint. As in the *decode-and-forward* scheme, the destination has two independent noisy versions of the original codeword that can be optimally combined. Thus, from the perspective of the destination, it still appears as though a rate-$1/2$ repetition code is used at the source. It is shown that *amplify-and-forward* schemes can achieve full diversity in the number of cooperating nodes [3, 4] .

In the *decode-and-forward* scheme the relay just transmits the binary codeword. In the *amplify-and-forward* scheme, the relay must amplify the received symbols, and retransmit these amplified soft values. This soft-amplification process will not be practical in many real systems. Instead, the relay would have to quantize the received symbol values and then transmit the quantized bits to the receiver. The information exchanged by the relays in order to improve performance is referred to as the *cooperation overhead*. Then if B bits are used for quantization, then the cooperation overhead of the *amplify-and-forward* schemes is B times the *decode-and-forward* cooperation overhead. However, the *amplify-and-forward* scheme does not depend on correct decoding at any of the relays. Thus, the *amplify-and-forward* scheme achieves full diversity at the cost of overhead.

The schemes introduced in this section are based on simple repetition coding ideas. The relays just repeat their estimate of the original codeword or their received symbol values. In the next section we review a few cooperation strategies that are based on better error-correction codes.

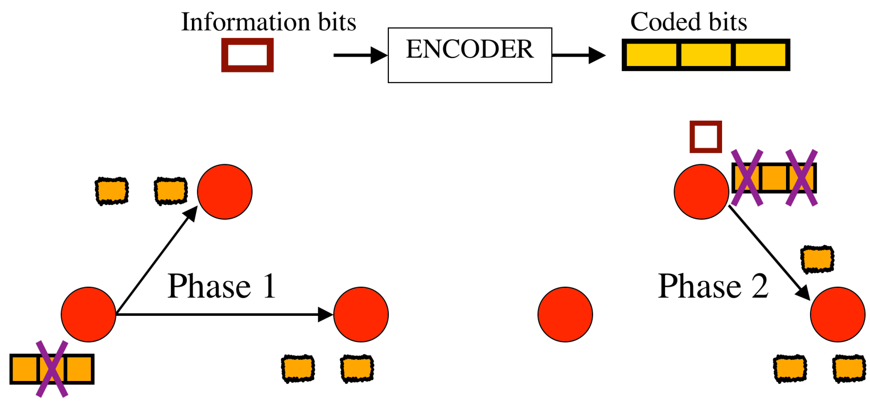

Figure 2-5: Coded cooperation using rate-compatible punctured convolutional codes.

2.3 Coded Cooperation

Cooperative diversity through the use of better error-correction codes is called *coded cooperation* [7]. Other schemes for coded cooperation have also been proposed [8, 9]. Coded cooperation schemes can be divided into two main classes. In the first class of techniques, distributed encoding is performed among the cooperating nodes, and decoding takes place only at the destination. In the second class, encoding is performed only at the source, and decoding takes place in a collaborative manner among the cooperating nodes. In this section, we review two coded cooperation schemes that belong to the former class. Techniques belonging to the latter category will be introduced in the following chapters.

Hunter and Nosratinia study the idea of user cooperation using rate-compatible punctured convolutional (RCPC) codes [7, 25, 26]. RCPC codes were introduced by Hagenauer [27] as a means to achieve incremental redundancy in ARQ schemes. The operation of the RCPC-based coded cooperation scheme is illustrated in Figure 2-5. For this example, we have shown the use of a rate-$1/3$ convolutional mother code. The data is encoded with the mother code in the source. Then a part of the codeword (the center part in the example) is punctured out and the remaining code bits are transmitted. Thus, the relay and the destination receive noisy versions of a rate-$1/2$ codeword. The relay decodes this high-rate transmission and then re-encodes the information bits using the mother code. Then the relay punctures those sections of the codeword that were transmitted by the source in

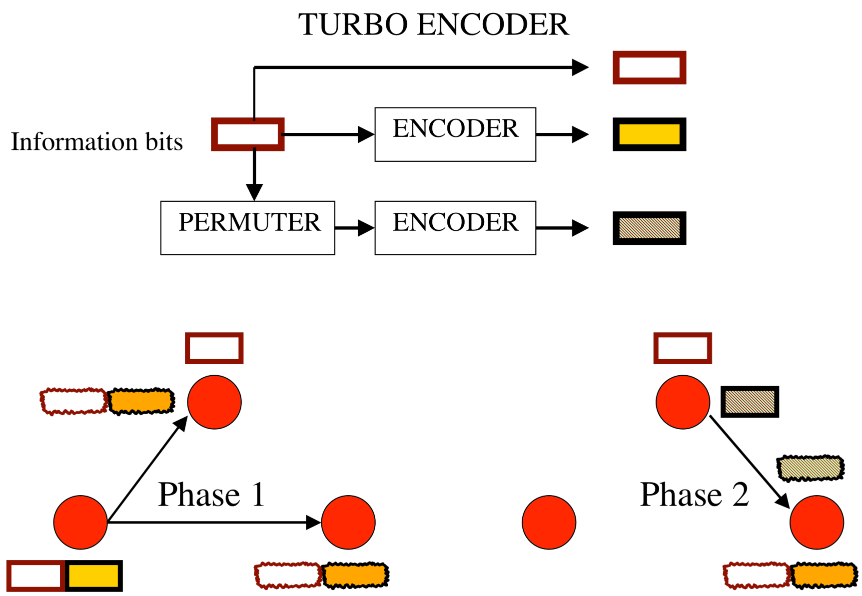

Figure 2-6: Coded cooperation using turbo codes.

the first phase (the first and last parts in the example). The relay then transmits the remaining part of the codeword (the part that was punctured in the first phase) to the destination. Thus, the destination effectively has all parts of the original mother code. Thus, the relay transmission helps transform the initial high rate code (rate-$1/2$) into a decodable lower rate (rate-$1/3$) transmission thereby improving performance at the destination.

Zhao and Valenti [8] investigate cooperation using turbo codes [28, 29]. Their scheme is shown in Figure 2-6. A turbo encoder (also shown in Figure 2-6) consists of two recursive systematic convolutional (RSC) encoders. The information bits are fed directly into one of the encoders and a permuted version of the information bits is fed into the other. The two sets of parity bits along with the systematic (information) bits form the codeword. The cooperation scheme works as follows. The source encodes the information bits with an RSC encoder and transmits the parity bits along with the systematic bits. The relay decodes this transmission using a convolutional decoder. It permutes its estimate of the information bits and re-encodes it with the same RSC code used at the transmitter. This produces the second set of parity bits that make up the turbo code. This second set of parity bits is sent to the destination by the relay. Thus, the destination has in effect received a codeword encoded by a turbo code. It can make use of the powerful, iterative turbo decoding algorithm [28, 29] to improve performance.

Note that both the schemes relies on correct decoding at the relay and thus fall in the *decode-and-forward* category. Thus, like any of the *decode-and-forward* schemes, these techniques are not guaranteed to achieve full diversity. It is proved that the RCPC-based cooperation scheme is capable of achieving full diversity only when the relay is able to decode correctly [26].

Another big drawback of the coded cooperation schemes that utilize distributed encoding is scalability. These schemes do not scale easily to multiple relays. When there is more than one relay, it is not immediately obvious on how the distributed encoding should be done. In the sequel, we present collaborative decoding, which is a coded cooperation scheme that is based on distributed decoding that scales naturally to any number of cooperating nodes.

CHAPTER 3
SOFT-INPUT SOFT-OUTPUT DECODING

This chapter presents a brief overview of soft-input soft-output (SISO) decoding. A good grasp of SISO decoding concepts is required to understand our collaborative decoding scheme that is presented in the following chapters. A mathematical characterization of the output of a particular implementation of the maximum *a posteriori* (MAP) decoder is also presented. This characterization aids in the analysis of one of our approaches to user cooperation.

3.1 The Log-MAP and Max-log-MAP Algorithms

Decoders that operate on floating point (soft) inputs and produce floating point outputs are called SISO decoders. The sign of the soft-output is the hard-decision and the magnitude of the soft-output is called the *reliability* of the hard-decision [30]. The reliability is an indication of the correctness of the hard-decision; i.e., a high value of the reliability implies a high probability of the decision being correct and vice-versa. In this proposal, we restrict our attention to SISO MAP decoders. It is well known that bit-by-bit MAP decoding produces the minimum probability of bit error among all decoding algorithms. The inputs to a typical MAP decoder are *a priori* probabilities of the information bits and channel symbols. The *a priori* probabilities are usually initialized to equally likely values. The soft-output of a MAP decoder corresponds to the *a posteriori* probability (APP) of an information bit u_i being 0 (or 1), $P(u_i = 0|\mathbf{r})$ (or $P(u_i = 1|\mathbf{r})$). Due to reasons of speed and numerical stability, MAP decoders are typically implemented in the log-domain (Log-MAP decoders). The output of a log-MAP decoder corresponds to log-likelihood ratios (LLRs) of the APPs. The LLR for each information bit u_i is computed as follows

$$L(u_i|\mathbf{r}) = \ln \frac{\mathrm{P}(u_i = 0|\mathbf{r})}{\mathrm{P}(u_i = 1|\mathbf{r})} = \ln \frac{\sum_{\mathbf{c}\in C_+} \mathrm{P}(\mathbf{c}|\mathbf{r})}{\sum_{\mathbf{c}\in C_-} \mathrm{P}(\mathbf{c}|\mathbf{r})}, \tag{3.1}$$

where $\mathbf{r}$ is the received codeword, C_+ is the set of all codewords with $u_i = 0$ and C_- is the set of all codewords with $u_i = 1$. Note that $\mathbf{c}_k \in \{+1, -1\}$. The output LLR is also referred to as the soft information. Assuming that all the codewords are equally likely and using Baye's rule, the soft information for codewords transmitted on a additive white Gaussian channel (AWGN) with noise variance $\sigma^2 = N_0/2$ can be written as

$$\begin{aligned} L(u_i|\mathbf{r}) &= \ln \sum_{\mathbf{c}\in C_+} \mathrm{P}(\mathbf{r}|\mathbf{c}) - \ln \sum_{\mathbf{c}\in C_-} \mathrm{P}(\mathbf{r}|\mathbf{c}), \\ &= \ln \left[\sum_{\mathbf{c}\in C_+} \exp\left(-\frac{\|\mathbf{r}-\mathbf{c}\|^2}{2\sigma^2} \right) \right] - \ln \left[\sum_{\mathbf{c}\in C_-} \exp\left(-\frac{\|\mathbf{r}-\mathbf{c}\|^2}{2\sigma^2} \right) \right]. \end{aligned} \tag{3.2}$$

A suboptimal implementation of the Log-MAP decoder, called the Max-Log-MAP decoder, is obtained by using the approximation $\ln(\sum x_i) = \max(\ln(x_i))$ to evaluate the LLR in (3.2). Thus, for a Max-Log-MAP decoder the soft-output is given by,

$$L(u_i|\mathbf{r}) = \min_{\mathbf{c}\in C_+} \left(\frac{\|\mathbf{r}-\mathbf{c}\|^2}{2\sigma^2} \right) - \min_{\mathbf{c}\in C_-} \left(\frac{\|\mathbf{r}-\mathbf{c}\|^2}{2\sigma^2} \right), \tag{3.3}$$

Since the union of C_+ and C_- spans the space of all valid codewords, one of the terms in (3.3) corresponds to the Euclidean distance between $\mathbf{r}$ and the maximum-likelihood (ML) decoding solution. Thus, the reliability for bit i (Λ_i) can be expressed as

$$\Lambda_i \triangleq |L(u_i|\mathbf{r})| = \frac{1}{2\sigma^2} \min_j \left\{ \|\mathbf{r} - \mathbf{c}_i^{(j)}\|^2 - \|\mathbf{r} - \mathbf{c}_{ML}\|^2 \right\}, \tag{3.4}$$

where $\mathbf{c}_i^{(j)}$ is a codeword corresponding to an input sequence that differs from the ML input sequence in the ith bit. Since the distance between $\mathbf{r}$ and the ML codeword is smaller than the distance between $\mathbf{r}$ and any other codeword, the difference in (3.4) is always positive. Thus, the Max-Log-MAP decoder associates with the ith bit, the minimum difference between the metric associated with the ML path and the best path that differs from the ML path in the input of the ith trellis section [31]. A high value of reliability implies that the ML path and the next best path are far apart, and hence there is a lower probability of choosing the other path and making a bit error. It has also been shown via simulation in [32, 33] that reliability is a measure of the correctness of the bit decision. Thus, a bit with high reliability is more likely to have decoded correctly than a bit with low reliability.

Note that the scaling of the reliability by the noise variance in (3.4) does not affect the performance of the Max-Log-MAP decoder and is just an implementation consideration. If channel estimates are available to the decoder, the scaling can be performed.

3.2 The Density Function of Reliabilities Associated with a Max-Log-MAP Decoder

Reggiani and Tartara [34] provided the first characterization of the soft information in terms of its probability density function (PDF). Reggiani and Tartara [34] examine the projection of noise in the direction corresponding to an error event and interpret this random variable as a distance in Euclidean space to derive the PDF. Here an *error event* denotes a sequence that translates one codeword into another, where the path through the code trellis that is induced by the error sequence is only in the same state as the original codeword at the endpoints of the sequence. For the rest of the paper, the random variable resulting from the projection of noise onto a direction specified by an error event will be referred to as the *projection random variable* (PRV). Reggiani and Tartara [34] present two approaches to obtain the PDF. In the first approach, the PDF is derived based on the assumption that different PRVs (projection of noise onto directions specified by different error events) are independent. The PDF obtained using the independence assumption results in conservative reliability estimates that are lower than the actual values. The authors suggest incorporating the correlation between the PRVs into the PDF to avoid conservative estimates. In the second approach, the authors obtain a covariance matrix involving the correlation between different PRVs and use it in a joint multivariate distribution to obtain the PDF. Though the PDF obtained using the second approach produces good reliability estimates, the expression for the density function is very complicated. Even with the independence assumption, the PDF obtained using this technique cannot be expressed in closed-form and involves products and summations that depend on the enumeration of all possible error events. Thus, the PDF given in Reggiani and Tartara [34] is not attractive for use in the analysis of reliability-based techniques.

We now present a streamlined derivation of the densities of reliabilities at the output of a max-log-MAP decoder by working in the conventional Hamming space (Reggiani and Tartara [34] work in Euclidean space) and use a high signal-to-noise ratio (SNR) approximation to obtain the PDF. A simple technique to account for the correlation between the PRVs is also presented. Using this technique, we can avoid the use of complicated joint multivariate distributions. Though this approach produces good estimates of the density function and other statistics of the reliability, the expression is still complicated to be of further use in analysis. To this end, we also present an ad hoc estimate of the PDF that is mathematically tractable. This closed-form estimate of the PDF is parameterized by a single quantity that can be numerically evaluated. We show that our technique produces an accurate approximation of the true PDF.

3.2.1 A High SNR Approximation to the Density Function of Reliabilities

The reliability of the output of a max-log-MAP decoder is given in (3.4). Since $\mathbf{c}_i^{(j)}$ is a codeword corresponding to an input sequence that differs from the ML input sequence in the ith bit, $\mathbf{c}_i^{(j)}$ can be expressed as,

$$\mathbf{c}_i^{(j)} = \mathbf{c}_{ML} + \mathbf{e}_i^{(j)}, \tag{3.5}$$

where $\mathbf{e}_i^{(j)}$ is an error event generated by an input sequence with bit i equal to 1. Since the symbols of $\mathbf{c}_i^{(j)}$ and $\mathbf{c}_{ML}$ take on values in $\{+1, -1\}$ and the error event transforms one codeword into another, the components of $\mathbf{e}_i^{(j)}$ take on values in $\{+2, 0, -2\}$. Using (3.5) in (3.4), we get

$$\Lambda_i = \min_j \left\{ \|\mathbf{e}_i^{(j)}\|^2 - 2(\mathbf{r} - \mathbf{c}_{ML})^{\mathrm{T}} \cdot \mathbf{e}_i^{(j)} \right\}. \tag{3.6}$$

Note that we have dropped the scaling by the noise variance $(1/2\sigma^2)$ in (3.6). After deriving the density and distribution functions using (3.6), a simple transformation can be used to account for the scaling in (3.4). At high SNRs, the ML decoder will find the correct codeword (input sequence). Thus for high SNRs we can express the received sequence as

$$\mathbf{r} = \mathbf{c}_{ML} + \mathbf{e}, \tag{3.7}$$

where $\mathbf{e} \sim \mathcal{N}\big(0, \frac{N_0}{2}\mathbf{I}\big)$. This assumption is similar to the approach in [34], in which the authors obtain the conditional density function given correct decoding of a bit. Using (3.7) in (3.6) we get

$$\Lambda_i = \min_j \left\{ \|\mathbf{e}_i^{(j)}\|^2 + 2\mathbf{e}^{\mathrm{T}} \cdot \mathbf{e}_i^{(j)} \right\}. \tag{3.8}$$

Note that according to our terminology, $\mathbf{e}_i^{(j)}$ is an error event, whereas $\mathbf{e}^{\mathrm{T}} \cdot \mathbf{e}_i^{(j)}$ is the PRV i.e., the projection of the noise onto the direction of the error event $\mathbf{e}_i^{(j)}$. Let

$$Z_j \triangleq \|\mathbf{e}_i^{(j)}\|^2 + 2\mathbf{e}^{\mathrm{T}} \cdot \mathbf{e}_i^{(j)}. \tag{3.9}$$

Since Z_j is just a linear combination of Gaussian noise samples, Z_j is also a Gaussian random variable. It is easy to see that

$$Z_j \sim \mathcal{N}\left(4d_j, 16d_j \frac{No}{2}\right), \tag{3.10}$$

where d_j is the Hamming weight (number of non-zero elements) of $\mathbf{e}_i^{(j)}$.

Thus, the reliability can be expressed as the minimum over a sequence of Gaussian random variables with distributions given by (3.10). Assuming that all the Z_js are independently distributed, the cumulative density function (CDF) of Λ can be written as

$$\begin{aligned} F_\Lambda(\lambda) &= 1 - \prod_j \text{Prob}(Z_j > \lambda) \\ &= 1 - \prod_{d=d_{min}}^{d_{max}} \left\{ Q\left(\frac{\lambda - 4d}{\sqrt{16d\sigma^2}}\right)\right\}^{a(d)}, \end{aligned} \tag{3.11}$$

where $a(d)$ is the multiplicity of error events of weight d and $Q(x)$ represents the Gaussian complementary distribution function. The PDF can be obtained by differentiating the CDF.

Using the product rule of differentiation, the PDF is obtained as

$$f_\Lambda(\lambda) = \sum_{d_j=d_{min}}^{d_{max}} \left\{ \frac{a(d_j)}{4\sqrt{(2\pi d_j \sigma^2)}} \exp\left(-\frac{(\lambda - 4d_j)^2}{32 d_j \sigma^2} \right) \right.$$
$$\left. \times\, Q\left(\frac{\lambda - 4d_j}{\sqrt{16 d_j \sigma^2}} \right)^{a(d_j)-1} \prod_{\substack{d_i=d_{min} \\ d_i \neq d_j}}^{d_{max}} Q\left(\frac{\lambda - 4d_i}{\sqrt{16 d_i \sigma^2}} \right)^{a(d_i)} \right\}. \tag{3.12}$$

Thus, even under the simplifying assumption of independent PRVs, the density function obtained from first principles is very complicated and not suited for use in the analysis of techniques involving reliabilities. For the Max-Log-MAP decoder with the noise scaling implemented (as in (3.4)), the CDF and PDF of the reliability can be obtained by a simple transformation as

$$F_{\Lambda,\sigma}(\lambda) = F_\Lambda\big(2\sigma^2\lambda\big), \quad f_{\Lambda,\sigma}(\lambda) = 2\sigma^2 f_\Lambda\big(2\sigma^2\lambda\big). \tag{3.13}$$

The subscript σ is used in the above expressions to indicate that the soft-information is scaled by the noise variance in the Max-Log-MAP decoder. Since Λ is non-negative and continuous, the mean of the reliability can then be evaluated numerically as

$$\begin{aligned} \mu(\sigma^2) \triangleq E[\Lambda] &= \int_\lambda \big[1 - F_{\Lambda,\sigma}(\lambda)\big] d\lambda \\ &= \int_0^\infty \prod_{d=d_{min}}^{d_{max}} \left\{ Q\left(\frac{2\sigma^2\lambda - 4d}{\sqrt{16 d \sigma^2}} \right) \right\}^{a(d)} d\lambda. \end{aligned} \tag{3.14}$$

3.2.2 On the Correlation Between Output Error Events

In Section 3.2, we model the reliability as the minimum of a number of Gaussian random variables that are assumed to be independent. This assumption is valid only if all the PRVs are independent. However, this is not a valid assumption. It is possible that the different error events ($\mathbf{e}_i^{(j)}$) associated with the PRVs ($\mathbf{e}^{\mathrm{T}} \cdot \mathbf{e}_i^{(j)}$) share the same path through the trellis at certain time instants. At each of these trellis sections, the PRVs share common noise samples from the vector $\mathbf{e}$ and thus, the PRVs are correlated. Because of this correlation, the expressions for the PDF and mean of the reliabilities given by (3.13) and (3.14) can significantly differ from the simulation results, as will be shown in Section 3.6. Thus, the correlations among the output error events should be considered in order for the analytical expressions to agree with the simulation results. In [34], the authors account for this correlation by obtaining the joint multivariate distribution of Z_j and using this distribution to compute the density function. However, this approach would involve computing a covariance matrix involving different pairs of error events and using this covariance matrix in the density function. This approach results in a very complicated expression. Even with the independence assumption, the density function in (3.12) is complicated. Further, the approach using the multivariate distribution offers no further insight into the behavior of the reliabilities.

Note that the correlation between different PRVs arise because they share common noise samples, which is a consequence of the associated error events differing from the correct codeword in a common set of symbols. We introduce a simple approach to account for the correlation between PRVs by computing the correlation among different error events. We first define the correlation between two error events $\mathbf{e}_1$ and $\mathbf{e}_2$ of lengths l_1 and l_2 respectively as

$$C_{\mathbf{e}_1,\mathbf{e}_2} = \frac{\sum_{i=1}^{\min(l_1,l_2)} \mathbf{e}_{1,i} \odot \mathbf{e}_{2,i}}{\max(l_1, l_2)}, \tag{3.15}$$

where $\mathbf{e}_{j,i}$ refers to the ith bit of error event $\mathbf{e}_j$ and the '$\odot$' operator denotes the XNOR operation. For example, '11 10 10 11' and '11 10 10 00 01 11' are two error events of length 8 and 12 respectively, and the correlation between the two events can be computed using (3.15) to be 0.5. We account for the correlation between output error events by eliminating some of the error events that are highly correlated and using the reduced set of error events to compute the PDF/CDF of reliabilities. We define a correlation threshold T_{Corr}, and whenever two error events have a correlation value greater than T_{Corr}, the longer of the two events is eliminated from the event set. The longer of the two events is removed from the event set because performance is usually dominated by the low weight error events. This process is continued until all remaining pairs of error events have correlation less than T_{Corr}. We normalize the correlation by the longer of the two error event lengths to ensure that events with very dissimilar lengths have a low value of the correlation. This eliminates the possibility of discarding a long event which may share a common initial path through the trellis with a small error event. Thus, a condensed event set is obtained within which the events have low correlation value. We expect the small correlation between the events in the condensed set to have a negligible effect on the independence assumption used in deriving the PDF. It will be shown in Section 3.6 that if the summation in (3.12) is performed over the condensed event set, the resulting values are strikingly close to the simulation results for properly chosen values of T_{Corr}. Thus, the need for joint multivariate distributions involving the covariance matrix of output error events is avoided using this technique.

3.3 A Mathematically Tractable Density Function

The expressions for the density function of the reliability given by (3.12) and (3.13) are complicated and not convenient for use in mathematical analysis of reliability-based techniques. We address this issue with an ad hoc estimate of the PDF based on the following observations:

- The mean of the reliabilities obtained from (3.14) is very close to the simulation results. (This fact will be substantiated in Section 3.6).
- Given the correct decoder output, the conditional distribution of the soft output for a bit is approximately Gaussian with variance approximately equal to twice the mean (cf. [35, 36]).

Thus, we suggest modeling the reliability as the absolute value of a Gaussian random variable that satisfies the symmetry condition, i.e.,

$$\Lambda = |X|\,, X \sim \mathcal{N}(\mu, 2\mu), \tag{3.16}$$

where μ is the mean obtained by numerically evaluating (3.14). The cumulative distribution function (CDF) can easily found to be

$$F_\Lambda(\lambda) = \begin{cases} Q\left(\frac{\mu-\lambda}{\sqrt{2\mu}}\right) - Q\left(\frac{\mu+\lambda}{\sqrt{2\mu}}\right), & \lambda > 0 \\ 0, & \text{otherwise}. \end{cases} \tag{3.17}$$

Differentiating with respect to λ, the PDF of the reliability is,

$$f_\Lambda(\lambda) = \begin{cases} \frac{\exp\left(-\frac{(\mu-\lambda)^2}{4\mu}\right) + \exp\left(-\frac{(\mu+\lambda)^2}{4\mu}\right)}{2\sqrt{\pi\mu}}, & \lambda > 0 \\ 0, & \text{otherwise}. \end{cases} \tag{3.18}$$

Unlike the expression in (3.13), the density function in (3.18) does not involve summations and products and is expressed in closed-form. Indeed, we have to resort to numerical computation to obtain the mean, μ, but for problems involving explicit probabilities of reliabilities, (3.18) is mathematically more tractable than (3.13). In Section 3.6 we provide results that show this Gaussian approximation is extremely accurate.

3.4 A Closed-Form Expression for the Bit-Error-Rate of SISO Decoders

In this section, we demonstrate one application of the approximate density function of reliabilities presented in the previous chapter (see 3.18). We will use the density function to derive a closed form expression for the bit-error-rate of max-log-MAP decoding. The probability of a bit decoding incorrectly conditioned on its reliability is given by

$$P_{b|\lambda} = \frac{1}{1 + e^{\lambda}}. \tag{3.19}$$

The probability of bit error can be obtained by integrating $P_{b|\lambda}$ over the density of Λ given in (3.18).

$$\begin{aligned} P_b &= \int_0^\infty \frac{1}{1+e^\lambda} f_\Lambda(\lambda) && (3.20) \\ &= \frac{1}{\sqrt{4\pi\mu}} \int_0^\infty \frac{\exp\left(-\frac{(\mu-\lambda)^2}{4\mu}\right) + \exp\left(-\frac{(\mu+\lambda)^2}{4\mu}\right)}{1+e^\lambda}, && (3.21) \\ &= \frac{1}{\sqrt{4\pi\mu}} \int_0^\infty \frac{\exp\left(-\frac{(\mu+\lambda)^2}{4\mu}\right)\left\{1+e^\lambda\right\}}{1+e^\lambda}, && (3.22) \\ &= Q\left(\frac{\mu}{\sqrt{2\mu}}\right) = Q\left(\sqrt{\frac{\mu}{2}}\right). && (3.23) \end{aligned}$$

Thus, the bit error rate can be expressed as a Q-function[1] with the argument depending solely on the mean of reliability. The mean of the reliability can be calculated using (3.14). The probability of bit error depends on the weight distribution of the error events of a code. The mean of the reliability encapsulates all properties of the code into a single quantity thereby leading to a convenient expression for the bit-error-rate. The expression of the bit error rate given in (3.23) can be used in various receiver-driven strategies that rely on the receiver having an estimate of the bit-error-rate. For example, consider a real-time streaming audio/video streaming application. These applications are loss-tolerant but delay-intolerant. Thus, an ARQ scheme for such a scenario can be designed as explained below. After decoding, the receiver estimates the number of bits that have decoded in error by using (3.23). Since these applications can tolerate some loss, the receiver requests for a re-transmission only when the number of bits in error exceeds a certain threshold.

3.5 Extension to Block-Fading Channels

We now extend the results presented to block-fading (quasi-static fading) channels. In a block-fading environment, all bits in a packet experience the same channel gain. For

[1] we consider this to be closed form since the Q-function is widely used in communication theory and can be computed accurately and efficiently

such a scenario, we characterize the reliability at the output of a max-log-MAP decoder conditioned on a particular realization of the channel. Assuming coherent detection (the channel gains are known to the decoder), the entire code space is rotated and scaled by the channel gain. Then ML or MAP decoding is performed on the received vector in the new code space. Thus, the reliability conditioned on the block-fading channel gain α can be expressed using (3.4) as

$$\Lambda_i|\alpha = \frac{1}{2\sigma^2}\min_j\left\{\|\mathbf{r}-\alpha\mathbf{c}_i^{(j)}\|^2 - \|\mathbf{r}-\alpha\mathbf{c}_{ML}\|^2\right\}. \tag{3.24}$$

Note that $\alpha\mathbf{c}_i^{(j)}$ and $\alpha\mathbf{c}_{ML}$ are codewords in the rotated code space. Using the same approach as before, we can assume that the ML codeword is the true transmitted codeword. Thus we have,

$$\begin{aligned}\mathbf{r} &= \alpha\mathbf{c}_{ML}+\mathbf{e}, \qquad && \mathbf{e}\sim\mathcal{N}(0,\sigma^2\mathbf{I})\\ &= \alpha(\mathbf{c}_{ML}+\mathbf{e}'), && \mathbf{e}'\sim\mathcal{N}(0,\frac{\sigma^2}{\alpha^2}\mathbf{I})\end{aligned} \tag{3.25}$$

Using (3.25) and (3.5) in (3.24), the reliability conditioned on the fading coefficient is

$$\Lambda_i|\alpha = \frac{\alpha^2}{2\sigma^2}\min_j\left\{\|\mathbf{e}_i^{(j)}\|^2 + 2\mathbf{e}'^{\mathrm{T}}\cdot\mathbf{e}_i^{(j)}\right\}. \tag{3.26}$$

Proceeding as in (3.8)-(3.13), the conditional CDF and PDF of reliabilities can be obtained as

$$F_{\Lambda|\alpha,\sigma}(\lambda) = F_\Lambda\left(\frac{2\sigma^2}{\alpha^2}\lambda\right), \quad f_{\Lambda|\alpha,\sigma}(\lambda) = \frac{2\sigma^2}{\alpha^2}f_\Lambda\left(\frac{2\sigma^2}{\alpha^2}\lambda\right), \tag{3.27}$$

where $F_\Lambda(\lambda)$ and $f_\Lambda(\lambda)$ are defined in (3.11) and (3.12) respectively. The conditional mean of soft information can then be obtained as

$$E[\Lambda|\alpha] = \mu\left(\frac{\sigma^2}{\alpha^2}\right), \tag{3.28}$$

where $\mu(x)$ is defined in (3.14).

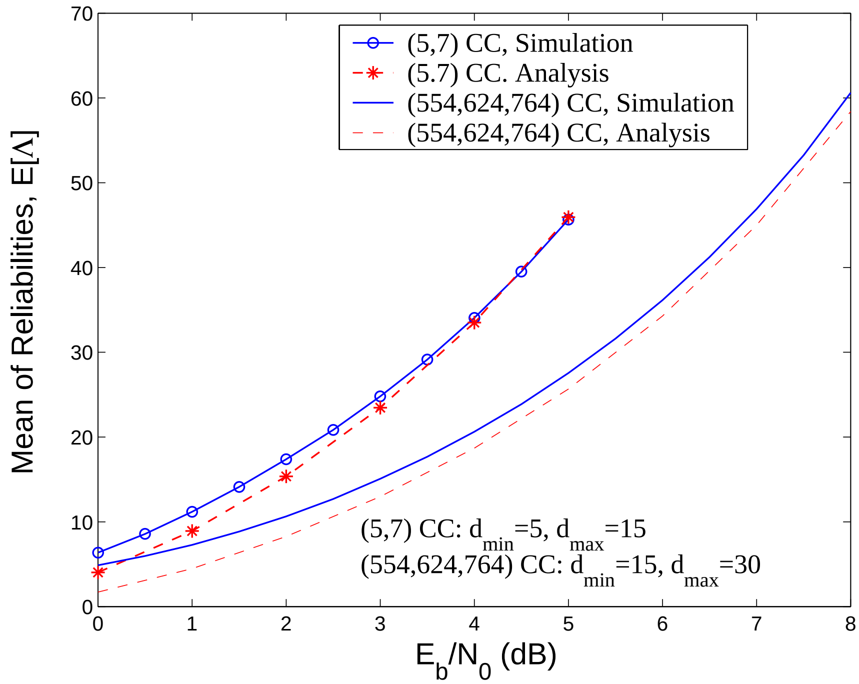

Figure 3-1: The mean of reliabilities as a function of the signal-to-noise ratio when the correlation between the output error events are ignored.

3.6 Numerical Results

In this section, the analytical expressions derived in the previous sections are compared with simulation results. For all results, non-recursive, non-systematic convolutional codes (CC) with a block size of 1000 bits are used. The Max-Log-MAP implementation of the BCJR [37] algorithm is used for decoding. In our implementation the Max-Log-MAP metric is scaled by the noise variance as in (3.4). In Figure 3-1, the means of the reliabilities obtained using (3.14) are compared with the actual means obtained from simulation. The comparison is shown for a rate $1/2$, constraint length 3 convolutional code and a rate $1/3$, constraint length 7 convolutional code. The constraint length 3 CC has generator polynomials $1 + D^2$ and $1 + D + D^2$ or $(5, 7)_8$ in octal notation. The constraint length 7 CC has generator polynomials $(554, 624, 764)_8$. It is observed that the analytical expression produces estimates that are smaller than the actual values. As explained in Section 3.2.2 and in Reggiani and Tartara [34], the assumption that all the output error events are independent leads to over-counting which causes the analytical results to produce conservative results.

Table 3-1: Error event multiplicity of the $(5, 7)$ convolutional code

d	a(d) All Events	a(d) T_{Corr}=0.7
d_{min}=5	1	1
6	2	2
7	4	4
8	8	8
9	16	9
10	32	5
11	64	11
12	128	5
13	256	14
14	512	13
d_{max}=15	1024	20

At low SNRs, there is a larger gap between analytically obtained values and the simulation results when compared to high SNRs. This is because the performance at low SNRs is dominated by blocks that decode incorrectly and hence the assumption (3.7) is violated.

To tighten the gap between the analytical and simulation results, it is required to consider the correlation between error events. The number of error events with weight d (event multiplicity) is shown in Table 3-1 for the $(5, 7)_8$ CC It is seen that eliminating events that have a correlation value higher than the correlation threshold ($T_{Corr} = 0.7$ in this case) results in a condensed set of error events. We expect that using this condensed set of events with low correlation will reduce the over-counting problem caused by the independence assumption.

The mean of the reliabilities after accounting for the correlation between output error events (as explained in Section 3.2.2) is shown in Figure 3-2. If the summation in (3.12) is performed over a condensed set of error events (as shown in Table 3-1), and the mean then computed using (3.14), it can be seen from Figure 3-2 that the analytical results are very close to the true values even at low SNRs.

The PDF of reliabilities (eqn. (3.13)) for the $(5, 7)_8$ code is compared with the true PDF in Figure 3-3. The true PDF was obtained experimentally by simulating the decoding

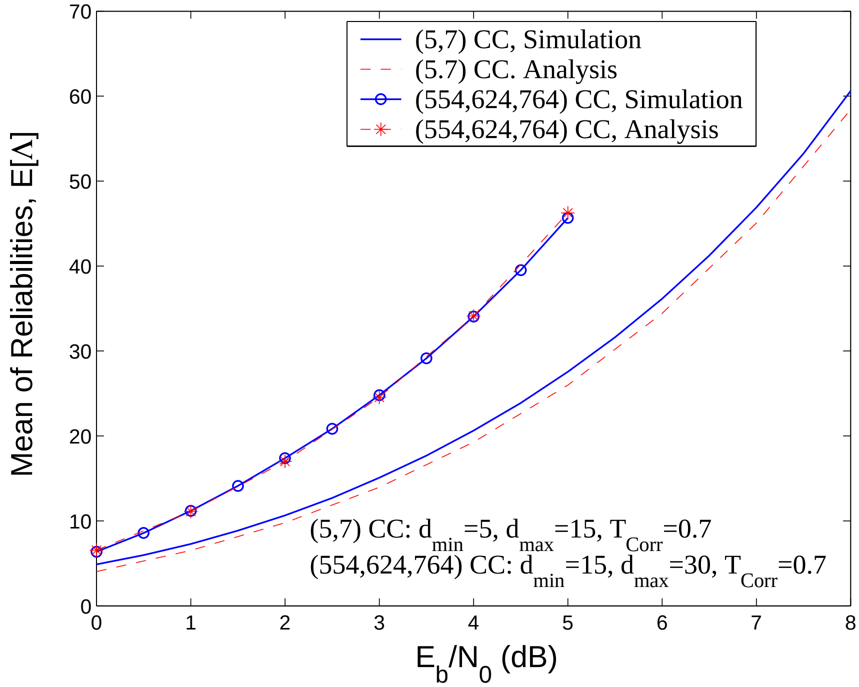

Figure 3-2: The mean of reliabilities as a function of signal-to-noise ratio after taking into account the correlation between output error events.

of a number of blocks that were transmitted over an AWGN channel. The reliability of each bit was recorded and the true PDF was estimated from the histogram of the recorded reliabilities. It can be seen from Figure 3-3 that results close to the true PDF can be obtained when the correlation between output error events is considered in the computation of the density function. The correlation is considered by evaluating the PDF in (3.12) over the condensed set of error events shown in Table 3-1. Note that the analytical PDF is much closer to the true PDF at higher SNRs.

The PDF obtained using the simple, ad hoc estimate in (3.18) is shown in Figure 3-4. The mean, μ, that specifies the PDF is obtained numerically from (3.14). It is observed that this ad hoc expression produces results that are closer to the true PDF when compared to the expression in (3.12). Unlike the PDF given in (3.12), the ad hoc estimate produces results that are very close to the true PDF even at low SNRs. The correlation between error events can be accounted for in the ad hoc PDF estimate by evaluating μ over a condensed set of low-correlation error-events (as shown in Table 3-1). As before, accounting for the

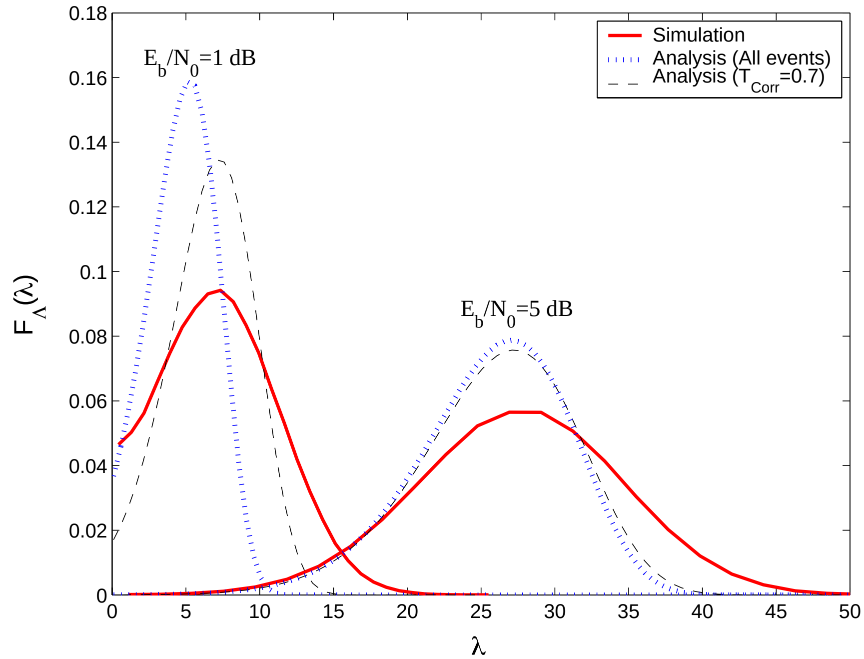

Figure 3-3: The PDF of reliabilities of the $(5, 7)_8$ CC for two different signal-to-noise ratios.

correlation produces better results when compared to treating the PRVs as independent random variables.

The closed-form approximation for the probability of bit error (P_b) given in (3.23) is compared to simulation results in Figure 3-5. Results are shown for both the memory–2 and memory–6 codes. Since P_b depends solely on μ, results are shown for two approaches to compute μ. In the first approach, μ is calculated analytically using the expression given in (3.14). In the second approach, μ is obtained through simulation. This approach shows the utility of the approximation to P_b in applications which require the receiver to have an estimate of its bit-error-rate. For both scenarios it is seen that the closed-form approximation is very close to the simulation results. For the results in Figure 3-5, all the error events were considered (correlation between error events were ignored) when the mean of the reliabilities was calculated analytically using (3.14). If the mean is computed after accounting for the correlation between error events, the closed-form approximation produces

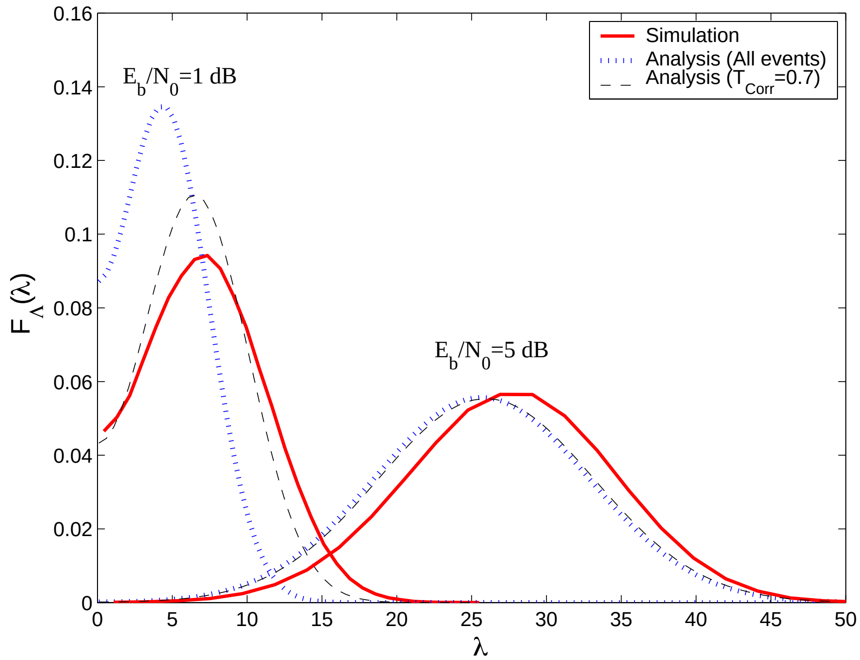

Figure 3-4: The PDF of reliabilities of the $(5,7)_8$ CC obtained using the simpler, mathematically tractable expression given in (3.18).

better results (when compared to ignoring the correlation). This is not shown in Figure 3-5 for the sake of clarity.

The mean of reliabilities obtained when the coded bits are transmitted over a Rayleigh block-fading channel is shown in Figure 3-6 for the $(5,7)$ convolutional codes. The analytical values are obtained by integrating the conditional mean in (3.28) over the density of the fading amplitudes given by $f(\alpha) = 2\alpha e^{-\alpha^2} u(\alpha)$ where $u(\alpha)$ is the unit step function.

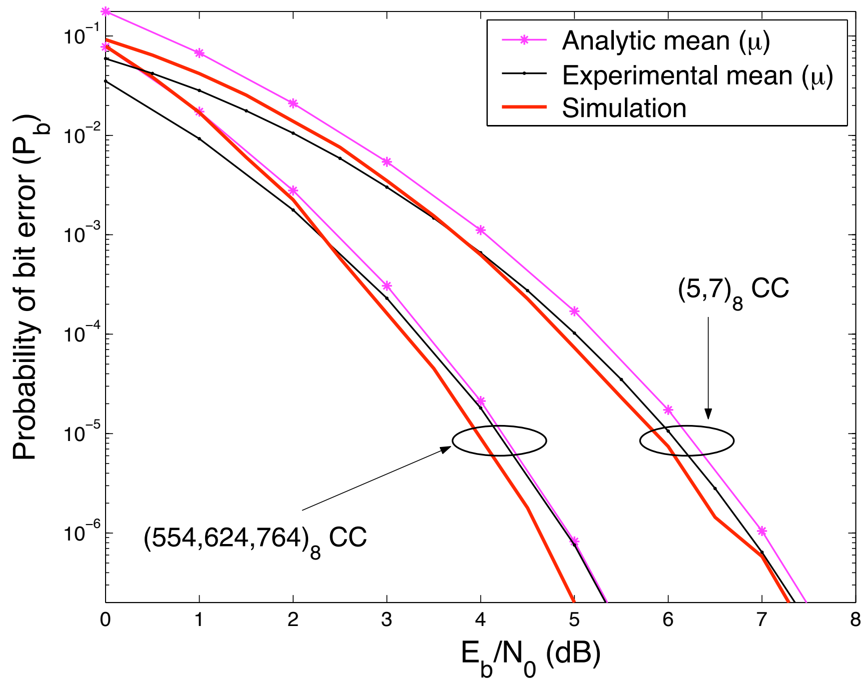

Figure 3-5: The probability of bit error for max-log-MAP decoding of convolutional codes evaluated using the closed-form approximation given in (3.23)

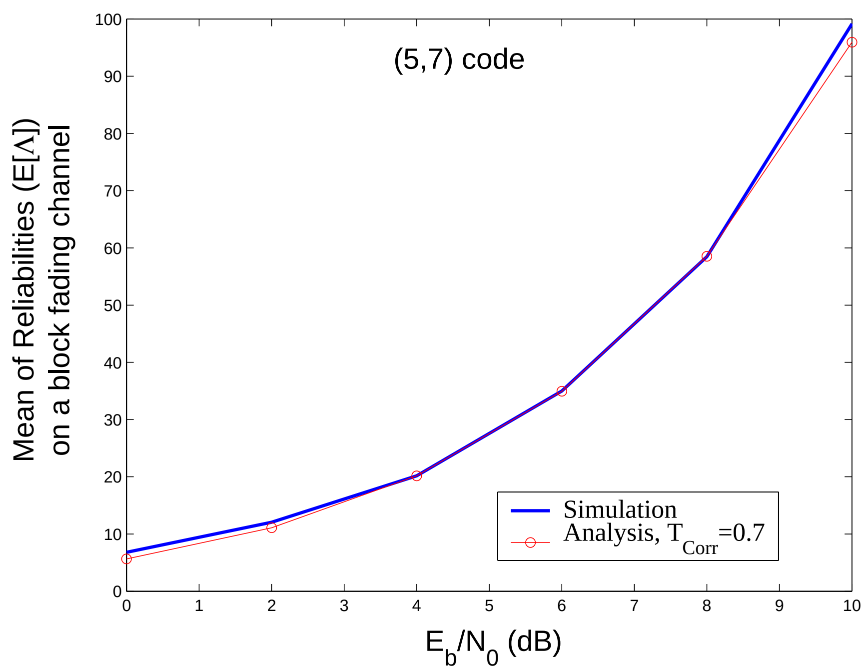

Figure 3-6: The mean of reliabilities of the $(5, 7)$ convolutional codes as a function of signal-to-noise ratio of a block-fading channel.

CHAPTER 4
CODED COOPERATION THROUGH COLLABORATIVE DECODING

The coded cooperation schemes summarized in Section 2.3 exploit the encoder structure of the error control code that is used; i.e., the nodes cooperatively encode the message to form a more powerful codeword at the receiver. In this chapter we introduce a cooperative strategy that involves collaboration in the decoding process; i.e, the destination and the relay(s) cooperatively decode the message from the source. The system model is shown in Figure 4-1. A distant transmitter broadcasts a message to a cluster of receiving nodes, one (or many) of which could be the intended destination. If any of the other nodes decodes correctly, it can use of one of the traditional *decode-and-forward* schemes to forward the message to the destination. The more interesting scenario is when none of the nodes decodes correctly. In this case, we cannot use any scheme that depends on correct decoding at the relays. The *amplify-and-forward* scheme could be used, but the cooperation overhead is very high. Thus, alternative techniques are required to minimize the overhead.

The fundamental drawback of the *decode-and-forward* based approach is that each relay forms its transmission based on its own decoding decisions and not on the decoding decisions at the other nodes. For example, if the intended destination or one of the other nodes in Figure 4-1 has decoded all but two of the bits correctly, then it is not necessary for the other relays to forward information as in the previously described *decode-and-forward* schemes. If the relays have some information about the decisions made at the other nodes, they can accordingly schedule their transmissions to minimize the collaboration overhead. Thus, in the model shown in Figure 4-1, there is no distinction made between the relay and the destination. The nodes cooperate with each other to obtain some information about the decisions made at the other nodes. This information helps minimize the cooperation overhead required for correct decoding at one of the nodes.

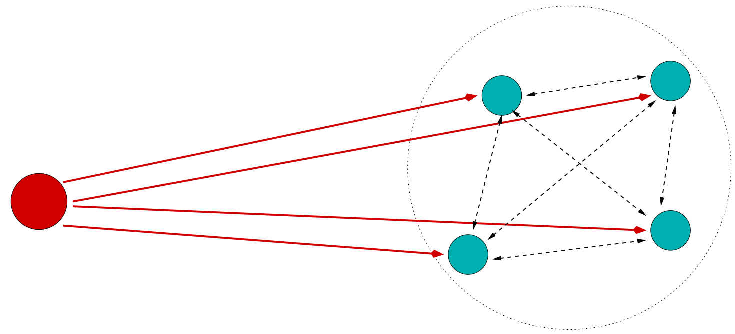

Figure 4-1: System model for collaborative decoding.

The system model also represents a broadcast scenario wherein all nodes are interested in the message from the transmitter. After collaboration, the node that decodes correctly can pass this information onto other nodes. An iterative technique to achieve cooperation in this broadcast scenario with two cooperating nodes was proposed by Wong *et al.* [9, 38]. The basic principle of this technique with two cooperating nodes is shown in Figure 4-2. The scheme proceeds in two stages as in the previous cooperation schemes (see Chapter 2). In the first stage, the distant transmitter broadcasts its message to the cooperating cluster. Stage 2 proceeds in multiple iterations with the iterations continuing until one of the nodes decodes correctly or a fixed number of iterations elapse. In each iteration, the nodes first exchange some *coordination* information. The coordination information is shown as the solid diamond in Figure 4-2. The coordination information gives each node some information about decoding at the other node. Based on this information, each node transmits some information about the initial message sent by the source to the other node. On receiving this message, each node performs decoding and if either node decodes correctly, cooperation is terminated. If not, the cooperation procedure is repeated as shown in Figure 4-2. The process of iterating between decoding and information exchange is referred to as *collaborative decoding*.

An information-theoretic study of the system model shown in Figure 4-1 with two cooperating nodes was studied by Draper *et al.* [39]. The nodes start collaborating after

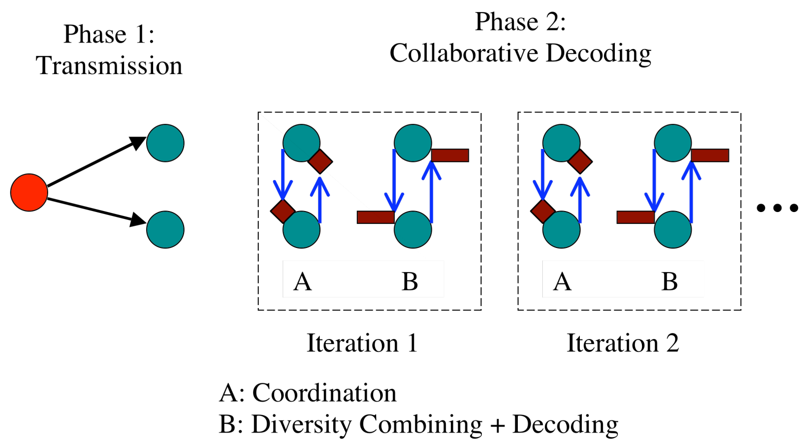

Figure 4-2: Principle of interactive/collaborative decoding with two nodes.

they receive the entire block sent by the transmitter. A round of conversation is defined as a message exchange between the nodes with one message transmitted from each node to the other. The sum of the transmission rates (sum-rate) between the nodes required for correct decoding is used as the performance criteria. A low sum-rate implies low cooperation overhead. The authors prove that collboration with multiple rounds of conversation between the two nodes can guarantee correct decoding (in the Shannon sense: arbitrarily low error probability as the block length goes to infinity) with a lower sum-rate (overhead) than collaboration consisting of one round of conversation. This is because information sent in an earlier iteration serves as side-information at the receiving node and the transmitting node can use more efficient coding techniques that use side-information at the transmitter and receiver to encode future conversations. With reference to Figure 4-2, note that there is no specific coordination information in this scheme. The information transmitted by a node in one iteration serves as coordination information for the next iteration. Thus, each node tailors its transmission based on information about decoding at the other node that it received in the previous iteration.

4.1 Collaborative Decoding through Reliability Exchange

Wong *et al.* [9, 38] present an iterative approach to cooperation for the scenario depicted in Figure 4-1. The basic principle of their approach is as follows. On receiving the message from the transmitter, the nodes perform MAP decoding on their received symbols. Each node uses the bit reliabilities to determine which bits are unreliable and requests additional information about these bits from other nodes. The other nodes transmit their estimates of the *a posteriori* LLR for these bits. The original requester uses this information as the corresponding *a priori* information in its MAP decoder and performs decoding again. This process of information request and decoding is repeated for a few iterations.

For simplicity, the process of exchanging soft-information will be henceforth referred to as *reliability exchange*. The information that passes between the different nodes will be referred to as the *overhead* in collaborative decoding. We build on the scheme of Wong *et al.* [9, 38] by investigating the performance and overhead for two different classes of reliability exchange schemes for multiple nodes (greater than two).

4.1.1 Collaborative Decoding through the Reliability Exchange of the Least Reliable Bits

In this section, we provide results for an extension of the scheme proposed in [9]. For all the results in this chapter, a rate 1/2 nonrecursive convolutional code with generator polynomials $1+D^2$ and $1+D+D^2$ is used to encode the information sequence. For convenience, we refer to this code as the $(5,7)$ code, where 5 and 7 are the octal representations of the generator polynomials. The encoded messages are transmitted over additive white Gaussian noise channels using binary antipodal signaling and are coherently demodulated. Each receiver decodes the received message using the BCJR [37] algorithm. Each node then requests reliability information for a certain percentage of the least reliable information bits by broadcasting the bit indices of those bits. Each node that receives the bit indices replies with its estimate of the soft information for those bits. The node that requested the information then uses these reliabilities as *a priori* information and runs the BCJR algorithm again. In Wong *et al.* [9] it is shown that for a packet size of approximately 1000 bits encoded with a $(5,7)$ convolutional code, collaborative decoding with two receivers provides performance very close to MRC at values of E_b/N_0 greater than 5 dB. Three iterations of collaborative decoding was performed by requesting soft information for 7.5% of the least reliable information bits in each iteration. The reason for requesting the least reliable bits (LRBs) is that most of the bits that decode incorrectly have low reliability values. Using MRC would require exchanging all of the received coded symbols. The overhead in bits, denoted by Θ_{MRC} can be calculated as

$$\Theta_{MRC} = \frac{N}{R_c} \times q, \tag{4.1}$$

where N is the size of the information message in bits, R_c is the code rate and q is the number of bits required to represent a (floating point) channel symbol. Note that (4.1) represents the overhead contribution of a single node. Using the collaborative decoding scheme mentioned above, the overhead contribution of a single receiver can be split into two parts. The first part consists of the bit indices that a receiver broadcasts to request the soft information of the LRBs, and the second part consists of the soft information that a node transmits each time it receives an LRB request from another node. Thus, the overhead for this scheme can be expressed as

$$\Theta_{LRB} = N_I \times a \times N \times (\lceil \log_2 N \rceil + (N_R - 1) \times q), \tag{4.2}$$

where N_I is the number of iterations of collaborative decoding, a is the fraction of information about which reliability information is requested, N_R is the number of receivers involved in collaborative decoding and $\lceil x \rceil$ is the smallest integer greater than or equal x. The first term in the summation on the R.H.S of (4.2) accounts for the bit indices that need to be transmitted to request soft information, and the second term in the summation accounts for the bits required to send out the soft information (each node receives $N_R - 1$ LRB requests). Note that the size of the requests can be further reduced through source coding or by exploiting the time-correlation between the reliabilities of the bits in error [33]. Both (4.1) and (4.2) refer to the overhead per receiver. All the schemes in this paper will be compared using the overhead contribution per receiver.

Table 4-1: Overhead of LRB-1 for different number of nodes.

Number of nodes	Overhead (bits)	% reduction (relative to MRC)
2	2025	77.5 %
5	4050	55.0 %
10	7425	17.5 %

Generally five bits are enough to represent a (floating point) channel symbol accurately [40], [41]. For a packet size of 900 bits, the overhead for MRC can be calculated using (4.1) as 9000 bits. Ten bits are required to represent each bit index in packet of 900 bits, and if we perform three iterations of collaborative decoding with soft information of 5 % of the LRBs being requested, the overhead is 2025 bits for two nodes (using (4.2)). Thus we see that performing collaborative decoding reduces the overhead by 77.5 % when compared to the MRC overhead.

Performing collaborative decoding with three iterations of 5% LRB exchange will be henceforth referred to as scheme LRB-1. LRB-1 has collaborative decoding overhead of 22.5 % of MRC overhead for a packet size of 900 bits and a cluster size of two nodes. For the rest of the chapter, the overhead for collaborative decoding will be reported as a percentage with reference to the MRC decoding overhead. Note that the overhead per receiver for LRB-1 increases with the number of receivers. The overhead for LRB-1 for different number of nodes is shown in Table 4-1. The reduction in overhead relative to MRC decreases with an increase in the number of nodes.

In Figure 4-3, the performance of LRB-1 is shown for different number of collaborating receivers. We note that the performance saturates for more than four receivers. This indicates that biasing the least reliable bits with a lot of *a priori* information from too many receivers will not improve the performance significantly. This is because there are some incorrectly decoded bits that may have relatively high reliabilities. This is again substantiated in the next section. When least-reliable bits are exchanged, the incorrect bits with

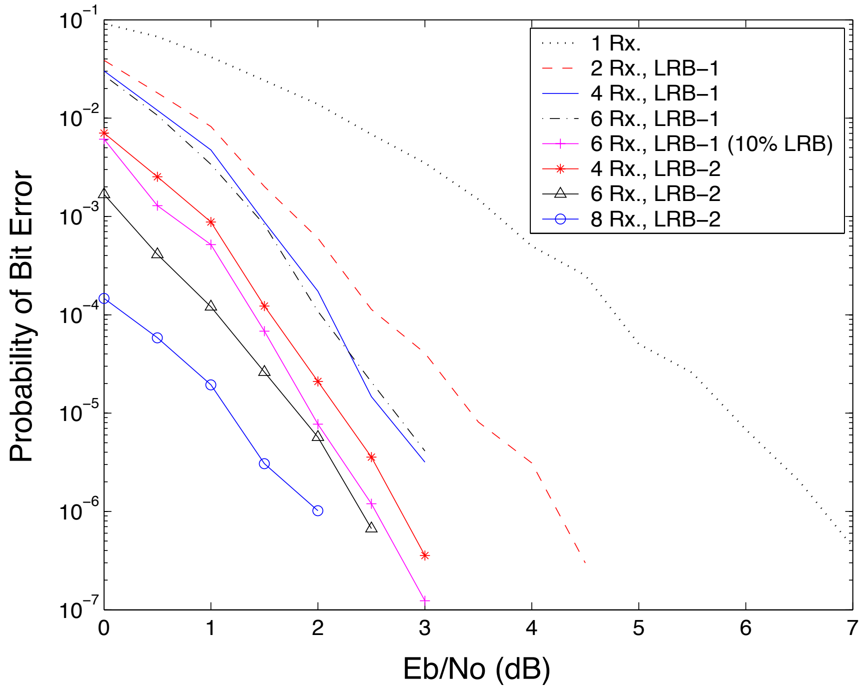

Figure 4-3: Performance of two collaborative decoding schemes in which receivers request information for a set of least-reliable bits.

high reliabilities may never be corrected regardless of how much information is provided for the LRBs.

An obvious method to improve the performance of LRB-1 is to increase the percentage of LRBs requested. From Figure 4-3, we see that requesting 10% of LRB reliabilities instead of 5% gives an improvement in performance of approximately 1 dB for a cluster of six collaborating nodes. However, this increases the collaborative decoding overhead, and our simulations show that the performance saturates for more than four receivers even in this case. Another disadvantage of requesting more information is that as the number of receivers increases, the time for information exchange also increases. Each receiver has to send out a set of bit indices requesting reliability information, and then all the other receivers have to respond. To coordinate this information exchange, a good MAC protocol will have to be designed. This latency would not be acceptable in certain applications.

A simple extension to LRB-1 is to transmit all the soft information via a broadcast channel and to have each node use *all* the received soft information, even if that node was not the original requester. Since the nodes other than the one that requested information also receive the soft information, they can make use of it as *a priori* information in their next round of SISO decoding. Thus, for the nodes that did not request the information, reliability information for a set of bits with random reliabilities is obtained. This scheme with 5% request and three iterations will be referred to as LRB-2. LRB-2 has the same overhead as LRB-1. The results in Figure 4-3 show that LRB-2 even outperforms LRB-1 with 10% LRB exchange. Further, LRB-2 does not suffer from the saturation problem like LRB-1. Hence, LRB-2 would be a better choice if exchanging LRBs was the scheme chosen to perform collaborative decoding.

The biggest disadvantage of this scheme is that the per-receiver overhead grows linearly with the number of receivers (cf. (4.2)). Thus, if the number of nodes is large, even requesting a very small percentage of LRB soft information might cause the overhead to become larger than the MRC overhead. In the next section, an exchange scheme is presented that has an overhead which is independent of the number of receivers.

4.1.2 Collaborative Decoding through the Reliability Exchange of the Most Reliable Bits

One way to significantly reduce the overhead is to prevent a node from transmitting soft information more than once. From (4.2), we see that for block sizes of approximately 1000 and more than ten receivers, multiple transmissions of soft information contributes towards more than 82% of the overhead per receiver. Suppose that, after SISO decoding, each receiver selects a certain set of bits and broadcasts the reliabilities of these bits to the other nodes. It is important to ensure that the nodes broadcast "good" reliability information, i.e., reliability information about bits that are decoded correctly. The critical step in this scheme is to determine the set of bits for which a node will broadcast the soft information. Since each node only sends out soft-information only once, the collaborative decoding overhead per receiver is given by

$$\Theta_{MRB} = N_I \times a \times N \times (\lceil \log_2 N \rceil + q). \tag{4.3}$$

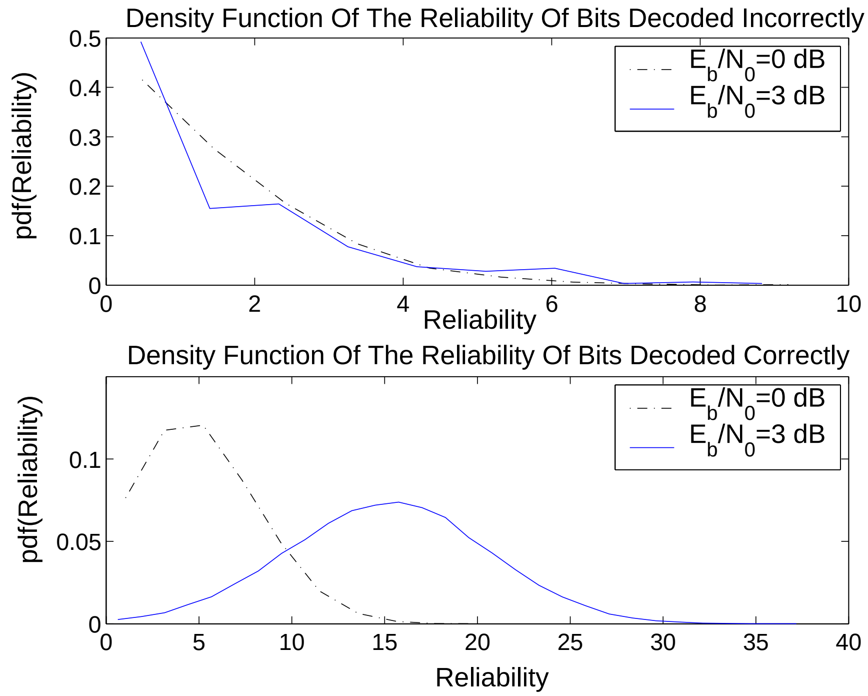

Figure 4-4: Reliability density functions associated with correctly and incorrectly decoded bits.

Note that for this scheme, unlike the LRB-based schemes, the overhead per receiver is independent of the number of receivers.

If a node broadcasts the soft information for a bit that was decoded incorrectly, using this value as *a priori* information would degrade the performance of the other nodes. The motivation behind our approach for selecting bits comes from observing the density functions of the reliabilities associated with correctly and erroneously decoded bits. Figure 4-4 shows the density function of the reliabilities for a $(5, 7)$ convolutional code with a block size of 900 bits.

We note that the trend followed by the density functions is as expected; i.e., most of the incorrectly decoded bits have low reliabilities and the correctly decoded bits have a relatively high reliability. We observe that at an E_b/N_0 of 0 dB, the maximum value of the reliability of a bit that decodes incorrectly is about half of the maximum value of the reliability of a bit that decodes correctly. For values of E_b/N_0 greater than 3 dB, more than 50% of the bits that decode correctly have reliabilities greater than the maximum reliability

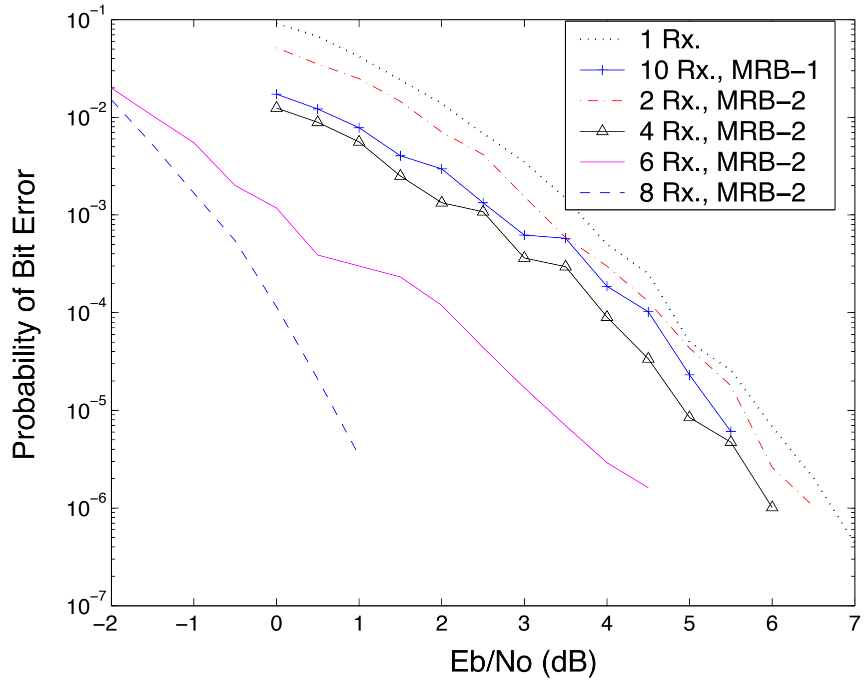

Figure 4-5: Performance of two collaborative decoding schemes in which receivers broadcast information about a set of most-reliable bits.

of the incorrectly decoded bits. Hence, if a node broadcasts a small percentage of its most reliable bits (MRBs), it is very likely to send out “good” soft information. These bits will correspond to a set of bits with random reliabilities at the other nodes. Performing three iterations of 10% MRB reliability exchange will be referred to as scheme MRB-1. The collaborative decoding overhead (per receiver) of MRB-1 is calculated, using (4.3), as 45% that of MRC. Though reliability information is exchanged for more bits than in LRB-1 and LRB-2, the overhead is still smaller than in LRB-1 and LRB-2 for more than five nodes.

Our simulations showed that the performance improvement is less than 2 dB even with ten nodes when compared to the performance of a single receiver. This is shown in Figure 4-5. The smaller performance improvement can be attributed to the set of bits that are broadcast in each iteration. At the end of the first SISO decoding, reliabilities of 10% of the most reliable bits are broadcast. Since we are biasing certain bits with “good” *a priori* information, at the end of the next SISO decoding, the reliabilities for these bits will

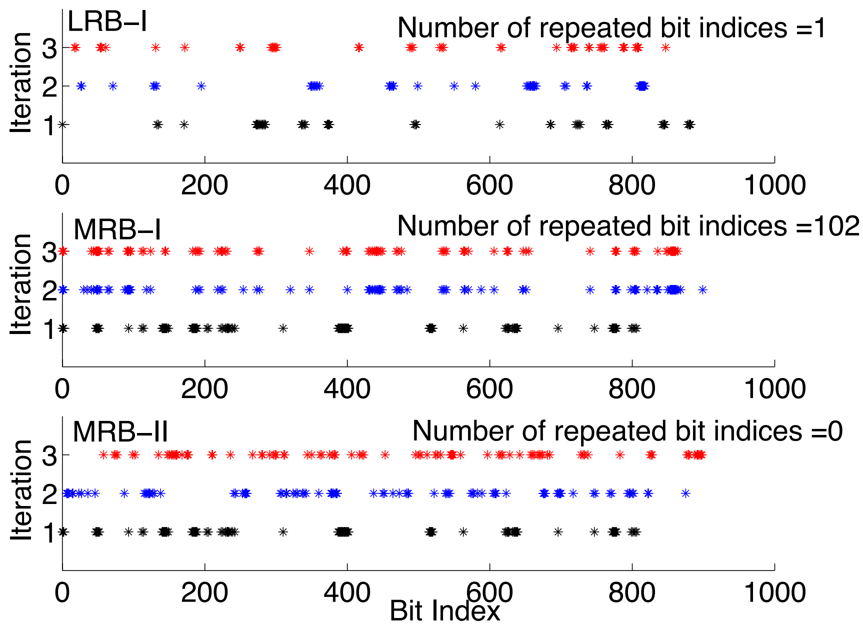

Figure 4-6: Bit indices of reliabilities exchanged as a function of iteration.

become large, and it is very likely that these bits will lie in the 10% MRB set. So reliabilities of these bits will be broadcast in the next iteration. But since the other nodes have already received the information about these bits, biasing them with more *a priori* information will not improve the performance significantly. This can be observed in Figure 4-6 in which we show the bit indices broadcast in each iteration for one packet of 900 bits. An asterisk on a bit position implies that reliability information about that bit was either requested (LRB-1) or transmitted (MRB schemes). We see that for MRB-1, a very large portion of bits are broadcast again in every iteration. In three iterations, the reliabilities of 102 bits are broadcast again among the total of 270 bits transmitted. This constitutes around 37% of the total bits sent. As the value of E_b/N_0 increases, there are fewer bits in error and in order to improve the performance, these erroneously decoded bits need to be biased with reliable *a priori* information. If a good percentage of the bits are repeated, there will be a low probability that *a priori* information will be received for all of the bits that are in error.

A simple method to eliminate this problem is to give the nodes memory to remember the set of bits for which soft information is transmitted or received. This ensures that bits that are already likely to have good reliabilities after one iteration do not get biased with more *a priori* information in the next iteration. Other bits are now given an opportunity to receive reliability information. This scheme, which is just MRB-1 with memory, will be referred to as MRB-2. In MRB-2, each node sorts its bits in ascending order of reliability after the first SISO decoding. Then each receiver broadcasts 10% of the MRBs for which soft information was not transmitted by any node in the previous iterations. Thus, in each iteration a new set of bits get reliability information. This is illustrated in Figure 4-6. In MRB-2, there are no bits for which soft information is transmitted in more than one iteration. The performance of MRB-2 is shown in Figure 4-5. If in any of the iterations, a node is not able to find a bit about which *a priori* information has not been transmitted earlier, it does not send out any reliabilities. Thus, the overhead in MRB-2 is less than or equal to the overhead in MRB-1, but the performance of MRB-2 is much better than that of MRB-1.

Note that adding memory to LRB-1 will not improve the performance significantly. This is because in each iteration, *a priori* information biases the least reliable bits and their reliability increases after SISO decoding. Thus, in the next iteration a new set of bits will constitute the set of LRBs. Hence, there is only a negligible overlap in the set of LRBs in each iteration. This can be observed in Figure 4-6, in which LRB-1 has just one bit that is repeated in three iterations.

A good suboptimal variant of MRB-2 sends hard-decisions of the MRBs instead of the soft decisions. This reduces the overhead for transmitting soft information from q (cf. (4.3)) bits per information bit to one bit per information bit. Thus, for MRB-2 with three iterations of 10% reliability exchange, the collaborative decoding overhead is only 33% for a packet of 900 bits. For a reasonably large number of receivers, the hard decisions from different receivers form *a priori* information that is sufficient to bias the information bits to produce correct decisions at the output of the SISO decoder. The performance of this scheme for six receivers is illustrated in Figure 4-7.

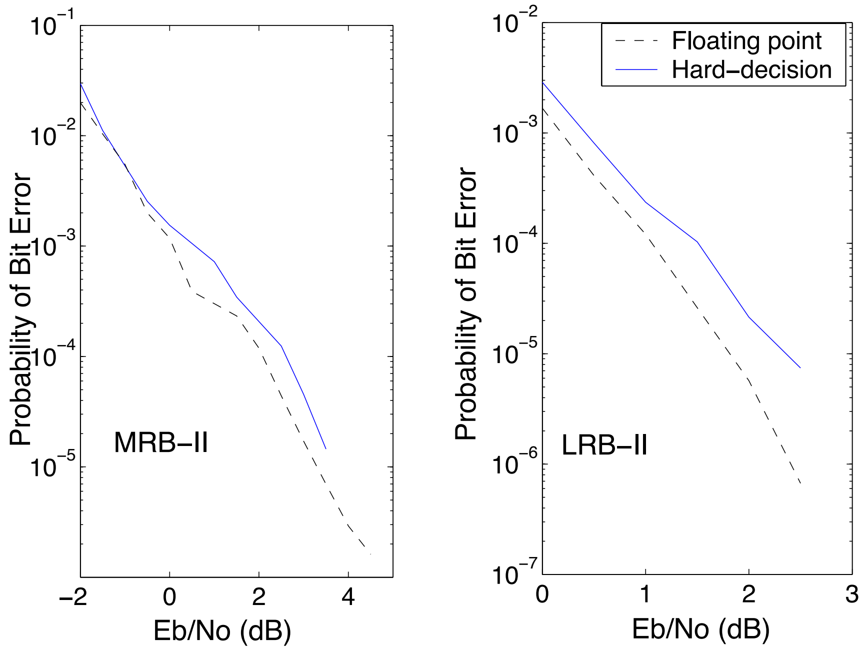

Figure 4-7: Performance of suboptimal variants two collaborative decoding schemes in which hard decisions are exchanged instead of soft information.

Note that this technique can be extended to any of the schemes discussed earlier. The performance of LRB-2 with hard-decisions is also shown in Figure 4-7. We see that a loss of approximately 0.5 dB can be expected for the suboptimal scheme when compared to the original scheme.

By comparing the performance of the LRB and MRB schemes in Figure 4-3 and Figure 4-5, it is seen that LRB-2 outperforms MRB-2 for a small number of receivers and MRB-2 performs better when the number of cooperating nodes is large. This is because an improvement in performance is obtained when bits that are decoded incorrectly get good *a priori* information. In the LRB schemes, the bits that are likely to be decoded incorrectly are specifically targeted leading to a improvement in performance. The MRB schemes are more optimistic in nature. Each node broadcasts reliable information that may or may not be useful to the other nodes. Thus, at one of the other nodes, good reliabilities are received for a set of bits with random reliabilities. There may or may not be an incorrectly decoded bit in these bit positions. When a few receivers collaborate it is not likely that all the unreliable bits receive *a priori* information from other nodes. Thus, the performance of MRB-2 is relatively worse for a small number of cooperating users. However, for a large number of receivers, it is likely that many of the bits in error are covered. For example, with 10% of MRB exchange and more than eight receivers, it is likely that information will be exchanged for almost all the bits in a block of 900 bits, and hence the performance of MRB-2 is better than LRB-1 for a large number of receivers.

The schemes that work with the MRBs also require less-complex channel access techniques. If the number of nodes are fixed, a simple round robin of all the nodes can be used to allow them to broadcast reliabilities of a certain percentage of their MRBs. For dynamically formed *ad hoc* networks, a cluster head could be chosen that assigns the order in which the nodes broadcast the reliabilities. When coupled with the fact that the overhead of LRB schemes can become prohibitive for large cooperating groups, the simplicity of MRB-2 and its performance in reasonably big cooperating groups makes it more suited for practical implementation.

4.2 Guidelines for the Design of Collaborative Decoding Schemes

Note that the reliability exchange schemes described earlier can be considered to lie in the realm of the *decode-and-forward* schemes with the relays transmissions consisting of the soft information. However, unlike the *decode-and-forward* schemes mentioned earlier, collaborative decoding does not depend on correct decoding at the cooperating nodes. The SISO decoders in our schemes use bit-by-bit MAP decoding like the BCJR [37] algorithm, and hence correct decoding is not needed to extract useful information for a small subset of bits. Thus, our reliability exchange schemes are an improvement over the *decode-and-forward* schemes since they are not limited by the capacity between the source and the cooperating nodes.

All the results shown in this chapter correspond to transmission over AWGN channels. There is no diversity in its true sense on AWGN channels since all the channels are equivalent. Thus, in order to study the diversity benefits of collaborative decoding using

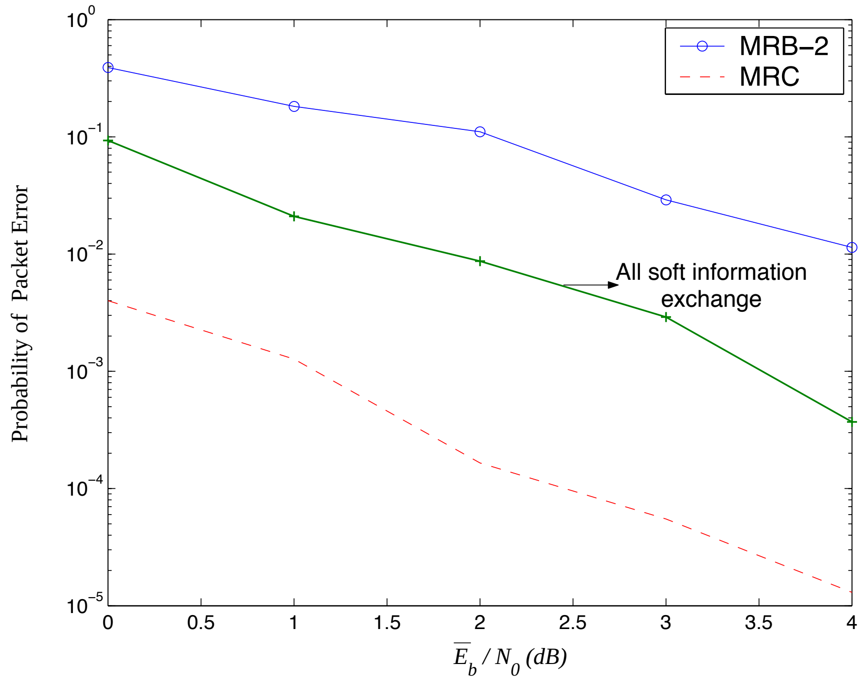

Figure 4-8: Performance of the MRB-2 scheme with eight nodes on a block-fading channel.

reliability exchange, we need to study the performance of the reliability exchange schemes over fading channels. The probability of packet error for collaboration using MRB-2 with eight receivers experiencing block-fading is shown in Figure 4-8. It is seen from the simulation results that the MRB-2 scheme does not achieve full diversity (the curves for MRC and MRB-2 are not parallel). It is observed that the performance of MRB-2 scheme with eight receivers is over 5 dB worse when compared to MRC. For the sake of comparison, the performance of a scheme that exchanges soft-information for all the information bits is also shown. This is the best performance that any of the MRB schemes can achieve. The performance of this scheme is around 3 dB worse than MRC. Thus, the best MRB scheme requires more than twice the SNR to achieve the same performance as MRC. The reason for the poor performance of the MRB scheme is as follows. If all nodes experience severe fading, then it is difficult to extract useful soft information for use in the MRB scheme. The reason for not achieving full diversity can be attributed to the fact that reliability exchange falls under the realm of the *decode-and-forward* schemes. Laneman [3] proved that the decode-and-forward schemes are not capable of providing all the diversity advantages associated with cooperative schemes.

However, *amplify-and-forward* schemes are guaranteed to achieve full diversity advantages [3]. Thus, it seems necessary to exchange information for the coded bits as in the *amplify-and-forward* schemes (information is exchanged for the information bits in reliability exchange) in order to achieve full diversity. MRC, which is an instance of this category, also exchanges received symbol values. However, MRC combines information in an inefficient manner with respect to overhead. Because of the use of error correction codes, there are certain bits (trellis sections) about which reliable decisions can be made without the exchange of information. The LRB schemes make use of this observation to request for information for only those trellis sections that are likely to be in error. But the LRB schemes request for the same amount of information for all the trellis sections. When the index of a trellis section is transmitted by the node requesting information, all the other nodes in the cooperating cluster will respond. Thus, all trellis sections in the set of LRBs receive the same amount of information. However, it is not clear if a trellis section that decodes incorrectly with a high reliability requires the same amount of information in order to correct the decision as a trellis section that decodes incorrectly with a low reliability. Ideally, the amount of information requested should be adapted to the reliability.

In fading channels, some nodes have better channels to the original transmitter and hence have made a greater number of correct bit decisions. Such nodes should share more information with other nodes when compared to the relays with bad channels. All the nodes transmit an equal amount of information if MRC or one of the LRB/MRB schemes is used.

These observations lead to three principles that should be kept in mind while designing cooperative protocols:

P1. In order to obtain full diversity advantages, it is necessary to exchange information closest to the RF front end, i.e., the received symbol values (soft demodulator outputs).

P2. The information exchanged in the cooperating cluster should be adapted to the errors in each packet. If reliabilities are used to adapt the collaboration content, the amount of information requested for each trellis section should be based on the reliability.

P3. Nodes with good channels should share more information than nodes with bad channels.

Note that MRC and the reliability exchange schemes described in this chapter violate all of these principles. In Chapter 6, we present an improved-LRB (I-LRB) scheme that is based on these principles. It will be shown that I-LRB achieves full diversity (same as MRC) in the number of cooperating nodes. The performance of I-LRB is better than comparable *amplify-and-forward* based collaborative approaches while still achieved performance close to that of MRC with a fraction of the collaboration overhead. The I-LRB scheme exploits the correlated reliabilities at the output of a SISO decoder in order to reduce the overhead. In the next chapter, we investigate the underlying decoder mechanism that leads to correlated bit reliabilities. This understanding will prove useful in the design of the I-LRB scheme.

CHAPTER 5
ON CORRELATED BIT ERRORS AT THE OUTPUT OF A MAX-LOG-MAP DECODER

We begin this chapter by first motivating the need to understand the correlated nature of reliabilities. We will use LRB as an example to show how correlated reliabilities can help in decreasing the overhead associated with collaborative decoding. In the LRB schemes, the output of the SISO decoder is used to identify the bits with low reliabilities, and information is requested for such bits because these bits are more likely to be in error. However, as noted in [33], errors at the output of a decoder are typically time-correlated. Since the LRBs are more likely to be in error, the LRBs are also correlated i.e., if a bit has decoded with a low reliability it is likely that the adjacent bits also have a low reliability. This can also be observed in Figure 4-6 where it is seen that in each iteration, information is requested for sets of consecutive bits. Thus, in LRB a node will request for information about a set of consecutive trellis sections with low reliabilities. This is a conservative approach because correcting one LRB will have an effect on the neighboring LRBs due to the correlated nature of the output of the decoder. In other words, if two bits are strongly correlated, it is likely that combining information for one bit will influence the decision at the other bit. Thus, it is not necessary to request for additional information for the entire set of consecutive LRBs. In order to decrease the cooperation overhead associated with collaborative decoding, it is necessary to understand the interaction between decoded bit reliabilities. In this chapter, we show that the error event that separates the ML codeword and a competing codeword in the max-log-MAP decoder can succinctly capture the correlated nature of bit errors. We also show how this error event can be efficiently computed using computations that are already performed in the BCJR max-log-MAP decoder. In the next chapter, we use this error event that separates the ML and competing path to design a collaborative decoding scheme that improves on the performance of LRB.

5.1 Terminology and Notation

The terminology and notation introduced here are specific to rate $1/2$ convolutional codes. It is straight-forward to generalize these to rate k/n codes.

- *input and output labels*: An *input label* is used to indicate the input that causes a particular state transition in the code-trellis, and an *output label* is used to indicate the corresponding output caused by that state transition.
- *path and event*: A sequence of valid state transitions in the trellis is called a *path* through the trellis. Note that every codeword represents a path through the trellis. Because the code is linear, the difference between any two codewords is a *path* through the trellis. Such a path is also called an *event*.
- *valid state*: A valid state lies on any *path* through the trellis. Because the trellis starts and stops in the all-zeros state, not every state is a valid state near the ends of the trellis.
- *metric*: The Euclidean distance between the received vector $\mathbf{r}$ and any codeword $\mathbf{c}$, $||\mathbf{r} - \mathbf{c}||^2$, is referred to as the metric[2] . Note that the metric is a maximum-likelihood (ML) decision statistic for additive white Gaussian noise (AWGN) channels.

The notation used in this Chapter is given in Table 5-1.

5.2 Revisiting Max-log-MAP Decoding of Convolutional Codes

The soft-output of a max-log-MAP decoder for codewords transmitted on an AWGN channel with noise variance σ^2 can be written as (see Chapter 3, eq. (3.3))

$$L(u_i|\mathbf{r}) = \min_{\mathbf{c} \in C_+^i} \left(\frac{||\mathbf{r} - \mathbf{c}||^2}{2\sigma^2} \right) - \min_{\mathbf{c} \in C_-^i} \left(\frac{||\mathbf{r} - \mathbf{c}||^2}{2\sigma^2} \right). \tag{5.1}$$

[2] Note that a metric is associated with a particular codeword. In other words, each codeword has a different metric.

Table 5-1: Notation used in this chapter

N	Block-size (the number of sections in the code-trellis).
u_i	The input to the encoder at time i; i.e., the input label for trellis section i. For binary codes considered in this paper $u_i \in \{0,1\}$. We will refer to u_i as the information bit.
$\underline{c}_i = [c_i^0, c_i^1]$	The output of the encoder at time i. This is a two-dimensional vector consisting of the two parity bits output by the encoder at each time. If BPSK is used for modulation then $\underline{c}_i^j \in \{-1,1\}, \forall j \in \{0,1\}$. Since every $\underline{c}_i$ corresponds to a particular branch in the trellis , $\underline{c}_i$ will used as the output labels for the branches in the trellis at time i. We will use parity bits or coded bits to refer to the output labels at any particular time in the trellis.
$\mathbf{c} = [\underline{c}_1, \ldots, \underline{c}_N]$	A valid codeword (output labels on a path through the trellis). Appropriate subscripts will be used to indicate the codeword being considered.
$\underline{r}_i = [r_i^0, r_i^1]$	The received vector corresponding to $\underline{c}_i$.
$\mathbf{r}$	The received vector corresponding to $\mathbf{c}$.
$\mathbf{c}_a^b$	$[\underline{c}_a, \underline{c}_{a+1}, \ldots, \underline{c}_{b-1}, \underline{c}_b]$. $\mathbf{r}_a^b$ is similarly defined.
$u^l(\mathbf{c})$	Input label at trellis section l in codeword $\mathbf{c}$.
$\mathbf{c}(l)$	Component l in codeword $\mathbf{c}$. Note that this refers to a particular bit in the corresponding output label.
C_+^i	$\{\mathbf{c} : u^i(\mathbf{c}) = 0\}$ i.e., the set of all codewords with input label 0 at trellis section i.
C_-^i	$\{\mathbf{c} : u^i(\mathbf{c}) = 1\}$ i.e., the set of all codewords with input label 1 at trellis section i.
$\mathcal{C}$	The set of all valid codewords. $\mathcal{C} = C_-^i \bigcup C_+^i$.
$\mathcal{S}$	Set of states in the trellis. For the memory-two code considered in this paper, there are four states. Therefore, $\mathcal{S} = \{0,1,2,3\}$.
s_k	State of the encoder at time k. Note $s_k \in \mathcal{S}$.
$\mathcal{S}(\rightarrow s)$	The set of valid states at time $k-1$ that have branches leading into state s at time k.
$\mathcal{S}(s \rightarrow)$	The set of valid states at time $k+1$ that have branches emerging from state s at time k.
$s_k(\mathbf{c})$	The state that codeword $\mathbf{c}$ passes[1] through at time k.
$\alpha_i(s)$	$\log(P(s_i = s), \mathbf{r}_1^i)$
$\gamma_i(s', s)$	$\log(P(s_i = s, \underline{r}_i \vert s_{i-1} = s'))$
$\beta_i(s)$	$\log(P(\mathbf{r}_{i+1}^N \vert s_i = s))$
$\mathcal{N}(\mu, \sigma^2)$	represents a Gaussian distribution with mean μ and variance σ^2.

Note that the *maximum-likelihood (ML) codeword/path* $\mathbf{c}_{\text{ML}}$ is a codeword that is closest to the received vector $\mathbf{r}$,

$$\mathbf{c}_{\text{ML}} = \underset{\mathbf{c} \in \mathcal{C}}{\operatorname{argmin}} \|\mathbf{r} - \mathbf{c}\|^2.$$

It is possible that there is more than one ML codeword (although this occurs with probability zero for the unquantized AWGN channel), in which case we arbitrarily choose one of the paths as the ML codeword.

$\diamondsuit$ *Definition 1. Competing codeword/path* $\mathbf{c}^i_{comp}$: The competing path at trellis section i is the path that is closest to the received vector among all paths that differ from the ML path in the input label for trellis section i,

$$\mathbf{c}^i_{\text{comp}} = \underset{\{\mathbf{c} \in \mathcal{C} : u^i(\mathbf{c}) \neq u^i(\mathbf{c}_{\text{ML}})\}}{\operatorname{argmin}} \|\mathbf{r} - \mathbf{c}\|^2. \tag{5.2}$$

As in the case of the ML codeword, there may be more than one codeword that satisfies (5.2), in which case the tie is broken by choosing one of the codewords arbitrarily. Note that although there is only one $\mathbf{c}_{\text{ML}}$, there may be many different $\mathbf{c}^i_{\text{comp}}$ for different values of i.

Then the reliability for bit i, which is the magnitude of the soft information in (5.1), can be expressed as

$$\Lambda_i \triangleq |L(u_i|\mathbf{r})| = \frac{1}{2\sigma^2}\left\{ \|\mathbf{r} - \mathbf{c}^i_{\text{comp}}\|^2 - \|\mathbf{r} - \mathbf{c}_{ML}\|^2 \right\}. \tag{5.3}$$

Note that $\mathbf{c}^i_{\text{comp}}$ is referred to as $\mathbf{c}^{(j)}_i$ in Chapter 3. We have replaced the notation to simplify exposition, and to stress the fact that $\mathbf{c}^i_{\text{comp}}$ is *competing* with $\mathbf{c}_{\text{ML}}$ for the hard-decision on trellis section i. Since the distance between $\mathbf{r}$ and the ML codeword is smaller than the distance between $\mathbf{r}$ and any other codeword, the difference in (5.3) is always positive. A high value of reliability implies that the ML path and the next best path with the opposite input label for bit i are far apart, and hence there is a lower probability that the decoder chose the wrong path and made a bit error. Thus, reliability is a measure of the correctness of the bit decision. This has also been shown via simulation results in [32, 33]. A bit with high reliability is more likely to have decoded correctly than a bit with low reliability.

The I-LRB scheme that is described in Section 6.4 utilizes both the bit reliabilities and knowledge of $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ in determining which information should be exchanged in the collaborative decoding process. In the next section, we detail how $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ can be determined for a particular trellis section.

5.2.1 Obtaining the ML and Competing Path using the BCJR Algorithm

Following the development in [42], the soft information in (5.1) can be expressed as

$$L(u_i|\mathbf{r}) = \max_{C^i_+}\left(\alpha_{i-1}(s')+\gamma_i(s',s)+\beta_i(s)\right) - \max_{C^i_-}\left(\alpha_{i-1}(s')+\gamma_i(s',s)+\beta_i(s)\right), \quad (5.4)$$

where $\alpha_k(s)$, $\gamma_k(s',s)$, and $\beta_k(s)$ are defined in Table 5-1.

It can also be shown that (see [42])

$$\begin{aligned} \alpha_i(s) &= \max_{s'\in\mathcal{S}(\to s)}\left(\alpha_{i-1}(s')+\gamma_i(s',s)\right) && (5.5)\\ \beta_{i-1}(s) &= \max_{s'\in\mathcal{S}(s\to)}\left(\beta_i(s')+\gamma_i(s,s')\right) && (5.6)\\ \gamma_i(s',s) &\propto -\|\underline{\mathrm{r}}_i-\underline{\mathrm{c}}_i\|^2, && (5.7) \end{aligned}$$

where $s'\in S(\to s)$ and $s'\in S(s\to)$ are defined in Table 5-1, $\alpha_0(0)=0$ and $\beta_N(0)=0$. Thus, it is seen from (5.7) that $\gamma_i(s',s)$ is proportional to the branch metric (cf. [43]), $P(\underline{\mathrm{r}}_i|\underline{\mathrm{c}}_i)$, used in the Viterbi algorithm (where the constant of proportionality depends on only the channel coefficient and signal-to-noise ratio).

Let the ordered pair of states (s_{i-1},s_i) that maximizes the first term in (5.4) be (s^+_{i-1},s^+_i). Let (s^-_{i-1},s^-_i) be the ordered pair of states that maximizes the second term. By comparing (5.1) and (5.4), it is seen that one of the ordered pairs of states (s^+_{i-1},s^+_i) or (s^-_{i-1},s^-_i) corresponds to $\mathbf{c}_{\text{ML}}$, while the other ordered pair corresponds to $\mathbf{c}^i_{\text{comp}}$. For example, if

$$\max_{C^i_+}\left(\alpha_{i-1}(s')+\gamma_i(s',s)+\beta_i(s)\right) > \max_{C^i_-}\left(\alpha_{i-1}(s')+\gamma_i(s',s)+\beta_i(s)\right),$$

then $s_{i-1}(\mathbf{c}_{\text{ML}}) = s_{i-1}^{+}$, $s_i(\mathbf{c}_{\text{ML}}) = s_i^{+}$, and $s_{i-1}(\mathbf{c}_{\text{comp}}^{i}) = s_{i-1}^{-}$, $s_i(\mathbf{c}_{\text{comp}}^{i}) = s_i^{-}$. Thus, when computing soft-output for trellis section i, it is possible to identify the branches through the trellis at time i that correspond to the ML path and the competing path.

We now introduce two theorems that will enable us to obtain $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}_{\text{comp}}^{i}$ in a straight-forward manner using the computations performed by the decoder.

Theorem 1: *The branch selection theorem*

Given the state in the code trellis at time k, $s_k = s'$ and the vector of received symbols $\mathbf{r}$, the following statements are true:

(a) *Trace-back:* The state-transition $s^* \rightarrow s'$, where $s_{k-1} = s^* = \underset{s \in \mathcal{S}(\rightarrow s')}{\operatorname{argmax}} \{\alpha_{k-1}(s) + \gamma_k(s, s')\}$, is a branch on the codeword $\mathbf{c}^*$ given by, $\mathbf{c}^* = \underset{\{\mathbf{c} \in \mathcal{C}: s_k(\mathbf{c}) = s'\}}{\operatorname{argmin}} \|\mathbf{r}_1^k - \mathbf{c}_1^k\|^2$.

(b) *Trace-forward:* The state-transition $s' \rightarrow s^*$, where $s_{k+1} = s^* = \underset{s \in \mathcal{S}(s' \rightarrow)}{\operatorname{argmax}} \{\gamma_{k+1}(s', s) + \beta_{k+1}(s)\}$, is a branch on the codeword $\mathbf{c}^*$ given by $\mathbf{c}^* = \underset{\{\mathbf{c} \in \mathcal{C}: s_k(\mathbf{c}) = s'\}}{\operatorname{argmin}} \|\mathbf{r}_{k+1}^N - \mathbf{c}_{k+1}^N\|^2$.

Proof: To prove the Trace-back procedure in Theorem 1, we first prove the following Lemma.

Lemma: $\alpha_k(s) \propto \underset{\mathbf{c} \in \mathcal{C}: s_k(\mathbf{c}) = s}{\min} \|\mathbf{r}_1^k - \mathbf{c}_1^k\|^2$ for any state s at time k that is on the path of a valid codeword

Proof: By mathematical induction.

Note that $\alpha_0(0) = 0$. Then $\alpha_1(0)$ is computed using (5.5) as

$$\alpha_1(0) = 0 + \gamma_1(0, 0) \tag{5.8}$$

because there is only one valid state leading into state 0 at time 1. Similarly, $\alpha_1(2) = 0 + \gamma_1(0, 2)$. The lemma does not apply to the other states at time 1 because they are not valid states for a rate 1/2 convolutional code initialized to state 0 at time 0. So using (5.7), the lemma holds for $k = 1$.

Assume that the lemma holds for time $k-1$. Then

$$\begin{aligned}
\alpha_k(s^*) &= \max_{s\in\mathcal{S}(\to s^*)} \left(\alpha_{k-1}(s) + \gamma_k(s,s^*)\right) && (5.9)\\
&\propto \max_{s\in\mathcal{S}(\to s^*)} \left(\max_{\mathbf{c}\in\mathcal{C}:s_{k-1}(\mathbf{c})=s} -\|\mathbf{r}_1^n - \mathbf{c}_1^{k-1}\|^2 + \gamma_k(s,s^*)\right) && (5.10)\\
&\propto \min_{\mathbf{c}\in\mathcal{C}:s_k(\mathbf{c})=s^*} \|\mathbf{r}_1^k - \mathbf{c}_1^k\|^2, && (5.11)
\end{aligned}$$

where (5.10) follows from the assumption about the claim, and the last equation follows from (5.7). Thus, the claim is true for time k. The principle of induction completes the proof.

Remark: From the lemma, $\alpha_k(s)$ is proportional to the partial-path metric ($\log P(\mathbf{r}_1^k|\mathbf{c}_1^k)$) [43] of the surviving path at state s at time k in the Viterbi algorithm when the branch metric is the Euclidean distance.

Proof of the trace-back theorem:

Compare the trace-back theorem and (5.5). The trace-back theorem chooses the previous state (s_{i-1}) that corresponds to the branch involved in computing the alpha for the current state (s_i). Since $\alpha_k(s)$ is proportional to the partial path metric of the surviving path leading to $s_k = s$, the branch involved in computing $\alpha_k(s)$ is part of the corresponding surviving path.

Thus, conditioned on the current state, the trace-back theorem chooses the previous state as the state at time $k-1$ on the surviving path at time k. The proof of the trace-back procedure follows because the surviving path has the best partial-path metric ($\min \|\mathbf{r}_1^k - \mathbf{c}_1^k\|^2$) among all paths $\mathbf{c}$ that pass through $s_k = s$.

The trace-forward theorem can be proved in a similar manner by comparing the trace-forward theorem with (5.6).

Theorem 2: *The conditional path selection theorem.*

Given a state transition at time i, i.e., $s_{i-1} = s'$ and $s_i = s^*$, let C_* represent the set of all paths through the trellis (codewords) passing through this transition at time i. That is, $C_* = \{\mathbf{c} \in \mathcal{C} : s_{i-1}(\mathbf{c}) = s', s_i(\mathbf{c}) = s^*\}$. Then the sequence of state transitions

$\{s'_0, s'_1, \ldots, s'_{i-2}, s', s^*, s^*_{i+1}, \ldots, s^*_N\}$ given by

$$s'_{k-i} = \operatorname*{argmax}_{s \in \mathcal{S}(\rightarrow s'_{k-i+1})} \{\alpha_{k-i}(s) + \gamma_{k-i+1}(s, s'_{k-i+1})\}, \quad i = 2, 3, \ldots, k \tag{5.12}$$

$$s^*_{k+i} = \operatorname*{argmax}_{s \in \mathcal{S}(s^*_{k+i-1})} \{\beta_{k+i}(s) + \gamma_{k+i}(s^*_{k+i-1}, s)\}, \quad i = 1, 2, \ldots, N-k \tag{5.13}$$

corresponds to the codeword $\mathbf{c}^*$ that is closest to the received vector $\mathbf{r}$ among all the codewords in C_*, $\mathbf{c}^* = \operatorname*{argmin}_{\mathbf{c} \in C_*} \|\mathbf{r} - \mathbf{c}\|^2$.

Proof: The proof follows by repeated application of the trace-back and trace-forward theorems.

As mentioned earlier, the state transitions from time $i-1$ to i that correspond to the ML path and the competing path can be obtained during the computation of the soft-output for bit i. Given the states $s_{i-1}(\mathbf{c}_{\text{ML}})$, and $s_i(\mathbf{c}_{\text{ML}})$, the ML codeword $\mathbf{c}_{\text{ML}}$ can be obtained using the conditional path selection theorem. The codeword output by the conditional path selection theorem is closest in Euclidean distance to the received vector among all paths that pass through $s_{i-1}(\mathbf{c}_{\text{ML}})$, and $s_i(\mathbf{c}_{\text{ML}})$, and is thus the ML path. Similarly, the competing path can be obtained using the conditional path selection theorem given $s_{i-1}(\mathbf{c}^i_{\text{comp}})$, and $s_i(\mathbf{c}^i_{\text{comp}})$.

As noted in the Lemma, the trace-back theorem always chooses the previous state (s_{i-1}) that corresponds to the branch involved in computing the alpha for the current state (s_i). Similarly, the trace-forward theorem always chooses the next state (s_{i+1}) that corresponds to the branch involved in computing the beta for the current state(s_i). This observation enables an efficient modification of the BCJR algorithm that enables computing $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ for any trellis section i. During the computation of the $\alpha_i(s),\ s \in \mathcal{S},\ i \in \{1, \ldots, N\}$, record the state $s_{i-1} = s'$ that maximizes $\alpha_{i-1}(s') + \gamma_i(s', s)$ as the previous state for s. Similarly, during the computation of the $\beta_i(s),\ s \in \mathcal{S},\ i \in \{1, \ldots, N\}$, record the state $s_{i+1} = s'$ that maximizes $\beta_{i+1}(s') + \gamma_i(s, s')$ as the next state for s. Then given the branch in the code-trellis corresponding to $\mathbf{c}_{\text{ML}}$ or $\mathbf{c}^i_{\text{comp}}$ for at time i, the entire codeword can be obtained by recursively following the states recorded in this way. Thus no additional computations are required to compute $\mathbf{c}_{\mathrm{ML}}$ and $\mathbf{c}^i_{\text{comp}}$. By recording information about the states that lead to the maximum values in (5.5) and (5.6) during the BCJR algorithm, $\mathbf{c}_{\mathrm{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ can easily be obtained through a series of table-lookups.

During the trace-back (or trace-forward) procedure, if $s_{i-k}(\mathbf{c}_{\mathrm{ML}}) = s_{i-k}(\mathbf{c}^i_{\text{comp}})$ for some k, then the sequence of state-transitions obtained for any time before k will be the same for $\mathbf{c}_{\mathrm{ML}}$ and $\mathbf{c}^i_{\text{comp}}$. Similarly, if or $s_{i+k}(\mathbf{c}_{\mathrm{ML}}) = s_{i+k}(\mathbf{c}^i_{\text{comp}})$, then the sequence of state-transitions will be the same for $\mathbf{c}_{\mathrm{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ for any time after k. It will be shown in Chapter 6 that I-LRB only requires knowledge of trellis sections where $\mathbf{c}^i_{\text{comp}}$ and $\mathbf{c}_{\mathrm{ML}}$ differ. Thus, it is sufficient to execute the trace-back and trace-forward procedures until $s_{i\pm k}(\mathbf{c}_{\mathrm{ML}}) = s_{i\pm k}(\mathbf{c}^i_{\text{comp}})$.

It is well known that the soft-output/reliabilities of adjacent bits in a convolutional code are correlated [34]. For max-log-MAP decoders, it was found through simulations that groups of neighboring bits have the same reliability. Since $\mathbf{r}$ and $\mathbf{c}_{\mathrm{ML}}$ are the same for all trellis sections, (5.3) implies that bits decoding with the same reliability should have the same competing path. By using the technique described above, and observing the competing paths for adjacent bits that decoded with the same reliability it is verified that the competing paths are indeed the same for those bits. Thus, for max-log-MAP decoders, the strong correlation between the reliabilities of adjacent bits is reflected in the choice of the same competing path in the code-trellis for those bits.

5.2.2 On the Utility of Competing Paths in the Design of Collaborative Decoding

In this section, we provide a brief outline of how explicit knowledge of these competing paths can help reduce cooperation overhead. It is well known that errors at the output of a convolutional code are bursty, and similarly the soft-output/reliabilities are temporally correlated [34, 33]. It was shown that the reason for this correlation is that bits that are close to each other in the trellis may often share the same competing codeword/path. For max-log-MAP decoding, such bits have exactly the same reliability, as can be seen from (5.3).

A particular bit decodes incorrectly if the ML path is not the transmitted codeword. Thus, all the neighboring bits that also choose the same ML path will also decode in error leading to time-correlated errors. The fact that errors in convolutional codes occur in bursts was also noted in [33].

Consider the use of the LRB scheme. Consider a LRB (say bit i) that decoded in error with $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^{i}_{\text{comp}}$ as the competing paths. This means that bit i chose the wrong path through the trellis as the ML path. In this case, it is likely that the next closest path (with respect to the received vector) with the opposite bit decision corresponds to the true transmitted codeword i.e., it is likely that $\mathbf{c}^{i}_{\text{comp}}$ is the true codeword. Now suppose that receiving additional information about bit i from other nodes is able to correct the decision. This implies that the additional information changed the ML path to be $\mathbf{c}^{i}_{\text{comp}}$. Assume that there were adjacent trellis sections ($i \pm k$), that had originally decoded with the same reliability (same choice of $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^{i}_{\text{comp}}$) before requesting additional information for bit i. Since additional information changed the choice of the ML path to $\mathbf{c}^{i}_{\text{comp}}$ for bit i, then it is likely that all the adjacent trellis sections that originally decoded with the same reliability as bit i will also choose $\mathbf{c}^{i}_{\text{comp}}$ as the ML path. This will also correct the errors at these adjacent trellis sections. Thus, by combining information for one trellis section it is possible to correct bit decisions at other trellis sections also. The I-LRB technique introduced in the next chapter uses this idea to reduce overhead by not requesting information for all the trellis sections that decode with the same competing path. I-LRB requests minimal additional information that will flip the decision from $\mathbf{c}_{\text{ML}}$ to $\mathbf{c}^{i}_{\text{comp}}$, thereby correcting all bit errors associated with the incorrect choice of $\mathbf{c}^{i}_{\text{comp}}$.

CHAPTER 6
IMPROVED LEAST-RELIABLE-BITS COLLABORATIVE DECODING FOR BANDWIDTH-CONSTRAINED SYSTEMS

In this section we describe the Improved LRB (I-LRB) collaborative decoding scheme for convolutionally encoded communications. We modify the LRB scheme described in Chapter 4 to satisfy the design guidelines mentioned in Section 4.2. We use LRB as our baseline scheme because the information exchanged in the LRB schemes targets the trellis sections that are most likely to have decoded in error. In addition to conforming with the design guidelines of Section 4.2, the I-LRB scheme also exploits correlated reliabilities to reduce the cooperation overhead.

The system model for collaborative decoding is shown in Figure 4-1. A distant transmitter broadcasts a packet to a cluster of receiving nodes. ARQ is not possible because of the power limitations of the mobiles and the distance to the transmitter. Cooperation overhead is critical in bandwidth-constrained systems. In these systems, the cooperation overhead is upper bounded by a maximum value. This constraint may be necessary in order to provide a minimum throughput guarantee to the distant transmitter. Since there is no feedback channel to the distant transmitter, it will continue to transmit messages at a certain rate. In such a scenario, cooperation (exchange of messages) cannot continue indefinitely. If the collaboration proceeds for a long time, then the process of collaborative decoding will interfere with additional transmissions from the source. Therefore it is necessary to constrain the cooperation overhead in order to ensure that collaboration does not conflict with transmissions from the source.

We first begin with a broad perspective of collaborative decoding in bandwidth-constrained environments. After introducing an baseline *amplify-and-forward* scheme (MRC variant) for such systems, we develop the I-LRB scheme.

6.1 Collaborative Decoding with Constrained Overheads

In collaborative decoding, the message at the source is packetized and encoded with a code that permits SISO decoding. The codeword is then broadcast to a cluster of receiving nodes that will attempt to decode the message. The received message for symbol i at node j can be modeled as

$$r_{i,j} = a_j x_i + n_{i,j}, \tag{6.1}$$

where x_i is the transmitted symbol at time i; a_j is the channel coefficient at receiving node j, which we assume is fixed over each packet; and $n_{i,j}$ is white Gaussian noise. In all that follows, we consider rate $R = 1/2$ codes, but it is straight-forward to generalize the work to other code rates.

If any node in the cluster decodes the message correctly, then we consider the message to be successfully received. If none of the nodes decodes the packet correctly, then the nodes begin the process of cooperating to receive the message. In the collaborative decoding schemes presented in Chapter 4 , the nodes use the outputs of the SISO decoders to select which information should be exchanged and which nodes should transmit that information. The *a posteriori* probability (APP) log likelihood ratio (LLR) at the output of a SISO decoder is a real number and is commonly referred as the soft output. The sign and magnitude of the soft output for an information bit represent the hard decision and the reliability of that decision, respectively [30]. The sample mean of the reliabilities at node j, μ_j, is a measurement of the overall reliability of the decoder's decision. We assume that the nodes exchange the μ_js after the first decoder iteration and that combining occurs at the node with the largest μ_j, which we refer to as the "best" node.

The nodes then broadcast information about a selected set of the received symbols (as in A-F) to the best node. The cooperative process can go through several iterations, each of which consists of three parts. In the first part of the iteration, the nodes identify information to be exchanged. In the second part, a selected group of nodes will transmit that information to the best node. In the final part of each iteration, the nodes decode the message and check whether it has decoded correctly. The process stops if any of the nodes has decoded the message correctly or if the limit on the number of iterations is reached.

In each iteration, we constrain the maximum number of bits that can be transmitted in the cooperative process. This may be necessary in many systems to ensure that the cooperative process does not conflict with the transmission of additional packets from the source. We specify the constraint as a portion of the total information exchanged in maximal ratio combining (MRC). Let N be the information block size, R be the code rate, N_{rx} be the number of receivers, and q be the number of bits used to quantize the channel observations. Then the cooperation overhead for MRC is $\theta_{MRC} = NqN_{rx}/R$ bits. The large θ_{MRC} will be not acceptable for many applications. Hence, we constrain the amount of information that can exchanged in the cooperating cluster to be a fraction p of θ_{mrc}. Note that this places a limit on the maximum amount of information exchange in the cooperative process for a particular packet; however, the actual amount of information exchange for any particular packet may be much less because we allow the cooperative process to terminate whenever the packet is decoded correctly. In each iteration, we constrain the overhead to $p\theta_{mrc}/N_{iter}$, where N_{iter} is the total number of iterations allowed.

Note that there are three main differences between the collaborative decoding scheme described in Chapter 4 and the collaborative decoding scheme described above for bandwidth-constrained systems.

- In the collaborative decoding scheme of Chapter 4, combining is performed at all nodes. For example, all nodes in the LRB scheme request for additional information about their LRBs. In collaborative decoding for bandwidth-constrained systems, combining is performed only at the best node. This solves the problem of overhead being proportional to the size of the cooperating cluster in the LRB schemes.
- In collaborative decoding scheme of Chapter 4, information from all other nodes is combined in each iteration. For example, all nodes broadcast the APPs for the LRBs

requested by a node in the LRB scheme. In collaborative decoding for bandwidth-constrained systems, information is carefully chosen from a select subset of nodes. This is keeping in accordance with design principle $P3$ given in Section 4.2.

- Unlike collaborative decoding of Chapter 4, the amount of information combined in each iteration is constrained in collaborative decoding for bandwidth-limited systems.

We next describe the two main cooperative schemes that will be compared in this chapter. The first, which we call constrained-overhead incremental MRC (COI-MRC), is an iterative form of maximal-ratio combining in which the overhead is constrained as explained above. The second scheme is a collaborative decoding scheme called the improved least-reliable bits (I-LRB) scheme. Because of the complexity of this scheme, we first provide an overview of it in Section 6.1.2, and contrast it with the LRB scheme described in Section 4.1.1. We then develop the tools required for I-LRB, and provide a detailed description of I-LRB in Section 6.4.

6.1.1 Constrained-overhead Incremental MRC

Consider first an implementation of full MRC in a group of collaborating radios. Each node (other than the best node) scales its received symbols by the fading gain, quantizes them, and transmits them to the best node. As mentioned above, this would result in a large overhead. A variant of this scheme that can offer even better performance than MRC with lower overhead is incremental MRC (I-MRC). In incremental MRC, the cooperation is done over several iterations.[1] In iteration i, the node with the $i+1$th largest μ_i transmits information about all of its received symbols to the best node[2]. Then the best node

[1] We thank an anonymous reviewer of a previous paper for proposing this cooperative scheme.

[2] Note that for quasi-static fading channels the value of μ_i is generally dominated by the fading coefficient. If two nodes have similar fading coefficients, this approach allows us to choose the one whose received information provides more confidence in decoding.

combines that information with its own received symbols and any previously received information, decodes the message, and checks whether the message has decoded correctly. If the message decodes correctly, the cooperative procedure terminates, and thus the average overhead of I-MRC is typically much less than MRC. In addition, because decoding is performed after each information exchange, I-MRC can achieve a slightly lower error probability than MRC.

Although I-MRC has a lower average overhead than MRC, the overhead in each iteration consists of all of the received symbols from one node, and the maximum overhead is the same as MRC. As explained above, it may be necessary to constrain the maximum overhead. Thus, we introduce a constrained-overhead I-MRC (COI-MRC) scheme. In COI-MRC, the overhead is constrained to $pNqN_{rx}/R$ bits. We allow a total of $N_{iter} = N_{rx} - 1$ iterations, so in each iteration, $pNqN_{rx}/(RN_{iter})$ bits are exchanged. The information in each iteration represents a set of $pNN_{rx}/(RN_{iter})$ received symbols from the best node that has not previously transmitted all of its received symbols. The set of symbols is uniformly selected from the remaining set of symbols at that node. Once all of the symbols at a node have been transmitted, then the next best node (in terms of μ_i) will transmit information for its received symbols.

After each round of information exchange, the best node uses MRC to combine the new information with its previously received information. The best node then decodes the message. If the message decodes correctly or if the maximum number of iterations has been reached, collaboration ends. Otherwise, another iteration of information exchange is performed.

6.1.2 Overview of Improved Least-Reliable Bits Collaborative Decoding

The MRC-based schemes are effective approaches for cooperation. However, these schemes are “dumb” schemes in the sense that they do not utilize information that is available that could improve the performance for the same constraint on the collaborative overhead. SISO decoders offer the ability to assess which bit decisions are reliable and which are unreliable. By first exchanging information that can improve the unreliable bit decisions, we may be able to achieve a better tradeoff between overhead and performance.

The scheme that we propose is based on the least-reliable bits (LRB) schemes that were described in Section 4.1.1. In these LRB schemes, each node identifies the set of bits with the least reliabilities (i.e., smallest magnitude of the APP LLR) and requests information for these bits from every other node. Our technique improves on the prior LRB schemes in several ways:

1. We request information at only the best node, so that the overhead from the information requests is reduced.
2. We utilize the fact that the set of LRBs is often correlated, and we develop techniques to avoid requesting too much information because of this correlation.
3. The set of nodes that respond to a request sent by the best nodes transmit quantized values of their received symbols and not the APPs as in the LRB scheme. This satisfies $P1$ of the design guidelines in Section 4.2.
4. The amount of information required to correct a bit depends on its reliability, so we present a technique to adapt the amount of information based on a bit's reliability. This satisfies $P2$ of the design guidelines in Section 4.2.
5. All nodes do not respond to a request sent by the best node. When the best node request for additional information about a trellis section, only the next best node that has not already transmitted information about that trellis section transmits its received symbols. Since coded symbols are combined starting with the second best node, it is likely that the second best node will transmit more coded symbols than another node. This satisfies $P3$ of the design guidelines in Section 4.2.
6. Not all bits that surround an unreliable bit will necessarily help to correct that bit, so we present a technique to select the set of bits which are most likely to correct an unreliable bit.

7. LRB attempts to reduce cooperation overhead by targeting individual trellis sections that decoded incorrectly. I-LRB reduces cooperation overhead by targeting competing paths through the trellis that potentially decoding incorrectly. By targeting the competing paths, I-LRB has the potential to correct all the bit errors associated with this path.

We refer to the new approach as the improved LRB (I-LRB) scheme. In this paper, we demonstrate how the goals of the I-LRB scheme can be achieved for convolutionally encoded communications by utilizing information generated in the max-log-MAP implementation of the BCJR decoding algorithm. The details of I-LRB with convolutional codes are given in Section 6.4.

Recall that in I-LRB, the best receiver sorts the trellis sections according to the reliabilities, and requests information from the other collaborating nodes to improve the decoding of some set of least reliable bits. The LRBs will often occur in groups because they are caused by the same error event, and thus it is only necessary to provide enough information to correct the error event to correct all of the bit errors caused by that event. Moreover, we show that some of the received symbols corresponding to a LRB may not be useful in resolving the most likely error event. In the rest of this chapter, we first propose a simple analytical technique that can be used to determine how much information needs to be transmitted for each least reliable bit. We then describe how the decoder can use information about the ML and competing paths to decide which information can most efficiently correct any bit errors in the LRBs. Finally, we provide a detailed description of the I-LRB scheme for convolutionally encoded communications.

6.2 Estimation of Request Size

During the collaborative decoding process, the decoder must act under the assumption that any LRB is in error, when in fact the error probability for even the least reliable bit is generally less than 0.5 (otherwise, we would just invert that bit decision). Given the reliability of a LRB, the decoder needs to estimate the amount of information that should be requested to correct the bit. The most likely error event for bit i is the event that separates $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$, which is given by

$$\mathbf{e}^i = \mathbf{c}_{\text{ML}} \oplus \mathbf{c}^i_{\text{comp}},$$

where $\oplus$ represents the XOR (addition or subtraction in a binary field) operator. For linear convolutional codes, as considered in this paper, $\mathbf{e}^i$ is a codeword.

The reliability in (5.3) can be further simplified as

$$\Lambda_i = \frac{1}{\sigma^2}\mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}). \tag{6.2}$$

If the channel from the distant transmitter to the collaborating cluster in Figure 4-1 does not have unit channel gains, then the reliability at the jth receiver can be expressed as

$$\Lambda_{i,j} = \frac{1}{\sigma^2} a_j^* \mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}), \tag{6.3}$$

where we have suppressed the dependence of $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ on the particular receiver number, j.

The decoder tries to estimate the amount of information required to change the decision from the ML path to the competing path (assuming that this will correct the error). Let $\mathbf{c}_{\text{ML}}(k)$ and $\mathbf{c}^i_{\text{comp}}(k)$ denote the kth parity bit on the ML path and competing path for information bit i, respectively. If $\mathbf{c}_{\text{ML}}(k) = \mathbf{c}^i_{\text{comp}}(k)$, then that parity bit does not provide any distinction between the two paths in the trellis. Thus, requesting information about such parity bits from the other collaborating nodes will not be helpful in resolving between these two paths. In the most likely case, in which either $\mathbf{c}_{\text{ML}}$ or $\mathbf{c}^i_{\text{comp}}$ is the correct codeword, the decoder will only improve its decision if additional information is received for those parity bits for which the decisions of ML and competing codeword are different.

$\diamondsuit$ *Definition 2. Candidate set of parity bits* $\mathbf{S}_i$ for trellis section i: The set of parity bits for which the decisions of the ML codeword ($\mathbf{c}_{\text{ML}}$) and competing codeword ($\mathbf{c}^i_{\text{comp}}$) are different.

$$\mathbf{S}_i = \{k : \mathbf{c}_{\text{ML}}(k) \neq \mathbf{c}^i_{\text{comp}}(k)\} = \{k : \mathbf{e}^i(k) = 1\} \tag{6.4}$$

Once the candidate set of parity bits is obtained, the decoder tries to estimate the number of parity bits from the candidate set $\mathbf{S}_i$ that have to be requested from other nodes in order for the decoder to decide in favor of $\mathbf{c}^i_{\text{comp}}$ instead of $\mathbf{c}_{\text{ML}}$.

Let $\mathbf{r}^*$ be the received vector after requesting κ coded symbols from another receiver[3] , say receiver 2. The decoder estimates the minimum number of additional coded symbols (κ) that will change the decision from $\mathbf{c}_{\text{ML}}$ to $\mathbf{c}^i_{\text{comp}}$ with probability greater than some threshold. That is, after receiving the additional information, we desire a high probability that

$$\|\mathbf{r}^* - \mathbf{c}^i_{\text{comp}}\|^2 < \|\mathbf{r}^* - \mathbf{c}_{ML}\|^2 \tag{6.5}$$

$$\Longrightarrow \quad 2\mathbf{r}^{*\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}) < 0 \tag{6.6}$$

$$\Longrightarrow \quad 2\mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}) + \sum_{\substack{l \in \eta \\ \eta \subset \mathbf{S}_i, |\eta| = \kappa}} 2\mathbf{r}^{'}(l)(\mathbf{c}_{\text{ML}}(l) - \mathbf{c}^i_{\text{comp}}(l)) < 0, \tag{6.7}$$

where η is the subset of the candidate set that has been transmitted in this iteration, and $\mathbf{r}^{'}$ corresponds to the symbols received due to those transmissions; i.e., $\mathbf{r}^* = a1^*\mathbf{r} + a2^*\mathbf{r}'$ (a_2^* is the conjugate of the fading coefficient at receiver 2). Using (6.3), we obtain $2a_1^*\mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}) = 2\sigma^2\Lambda_i$, where Λ_i is the reliability of trellis section i before combining. Note that in the above equations $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ refer to the ML path and competing path encountered in computing the soft-output for trellis section i before receiving additional coded symbols from receiver 2.

As previously mentioned, the decoder assumes that the parity bits in the candidate set are in error. Then we can calculate the required value of κ under the assumption that the

[3] $\mathbf{r}^*$ is obtained by combining the original received vector $\mathbf{r}$ and the additional symbols using maximal-ratio combining.

all-zeros CW has been transmitted, in which case $\mathbf{c}^i_{\text{comp}}(l) = 1$ and $\mathbf{c}_{\text{ML}}(l) = -1, \forall l \in \mathbf{S}_i$. Since the all-zeros CW is the true transmitted codeword, $\mathbf{r}'(l) \sim \mathcal{N}(a_2, \sigma^2)$. Thus,

$$X_i \triangleq \sum_{\substack{l \in \eta \\ \eta \subset \mathbf{S}_i, |\eta| = \kappa}} 2a_2^* \mathbf{r}'(l)(\mathbf{c}_{\text{ML}}(l) - \mathbf{c}^i_{\text{comp}}(l)) \sim \mathcal{N}(-4a_2^2\kappa, 16a_2^2\kappa\sigma^2).$$

Thus the decoder estimates that after the first-retransmission, correct decoding is made if $X_i < -2\sigma^2\Lambda_i$.

The decoder estimates the number of coded bits κ for which information is required from another receiver as follows,

$$\min_{\kappa} P(X_i < -2\sigma^2\Lambda_i) \geq \Theta \tag{6.8}$$

$$\min_{\kappa} Q\left(\frac{\sigma^2\Lambda_i - 2a_2^2\kappa}{2\sqrt{a_2^2\kappa\sigma^2}}\right) \geq \Theta, \tag{6.9}$$

where κ is the number of parity bits retransmitted and Θ is a predefined threshold. Thus, the decoder estimates the number of bits to be retransmitted as the minimum number that would cause the decoder to decide in favor of $\mathbf{c}^i_{\text{comp}}$ instead of $\mathbf{c}_{\text{ML}}$ with a probability that is at least Θ. This provides the minimum number of bits that is most likely to correct bit i if it is in error. $P(X_i < -2\sigma^2\Lambda_i)$ will be referred to as the *correction probability after combining* (P_c). Thus, the receiver requests the minimum number of coded bits such that P_c exceeds Θ. Therefore, by request coded symbols for κ parity bits, I-LRB has the potential to correct all the bits that chose the same competing path (i.e, decoded with the same reliability). For example, assume $\kappa = 1$, and that there are 3 bits that decoded with the same competing path. Suppose the additional copies of this 1 coded symbol is enough to flip the decision from $\mathbf{c}_{\text{ML}}$ to $\mathbf{c}^i_{\text{comp}}$, then the bit decision at the 3 trellis sections that originally chose $\mathbf{c}^i_{\text{comp}}$ as the competing path will also change. Thus, I-LRB has the potential to correct 3 bit errors by requesting only 1 coded symbol. Note that LRB would have requested for 3 reliability values.

Table 6-1: Instantaneous SNR estimation for trellis sections based on the average of the instantaneous SNRs of the parity bits in the candidate set

Output label on trellis section i for $\mathbf{c}_{\text{ML}}$	Output label l on trellis section i for $\mathbf{c}^i_{\text{comp}}$	Estimate of the instantaneous SNR for trellis section i
$1\ 1$	$-1-1$	$(\lvert r_i^0\rvert + \lvert r_i^1\rvert)/2$
$-1\ 1$	$1-1$	''
$-1\ 1$	$1\ 1$	$\lvert r_i^0\rvert$
$-1\ 1$	$-1-1$	$\lvert r_i^1\rvert$

6.3 Estimation of the Request Set

After the decoder estimates κ from the candidate set, it needs to select the subset of κ parity bits in $\mathbf{S}_i$ for which information will be requested from another receiver. We estimate an instantaneous SNR for each trellis section involved in the error event $\mathbf{e}_i$ that separates $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ to decide the candidate set for collaborative exchange. The receiver sorts the trellis sections in the error-event according to the instantaneous SNRs, and requests for κ parity bits from the trellis sections with low SNRs.

The concept of instantaneous SNR was proposed in [44] for use in selecting which symbols should be retransmitted in an ARQ scenario. Several different schemes were considered in [44], and the one described here was found to offer the best performance. If for a particular trellis section i, $\underline{c}_{ML}$ and $\underline{c}^i_{\text{comp}}$ differ in only one parity bit, then the instantaneous SNR of that section is equal to the absolute value of the received symbol corresponding to that parity bit. If for a particular trellis section i, $\underline{c}_{ML}$ and $\underline{c}^i_{\text{comp}}$ differ in both parity bits, then the instantaneous SNR of the trellis section is the average of the instantaneous SNRs of the two parity bits. The receiver selects κ parity bits corresponding to trellis sections with the lowest SNRs from the candidate set. The instantaneous SNR of a particular trellis section for different output labels on $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ is given in Table 6-1. Note that all possible output labels can be obtained by interchanging the output labels on the ML and competing paths in each row of Table 6-1.

6.4 Detailed Description of I-LRB Collaborative Decoding

With the above approaches to estimate the request size and the request set, we can describe I-LRB collaborative decoding in detail. Upon initiation of collaboration, the nodes broadcast their μs to determine the best receiver. Starting with the best receiver, let the receivers be numbered RX_1 to $RX_{N_{rx}}$. The second best receiver RX_2, transmits its fading coefficient a_2 to RX_1. RX_1 needs the fading coefficient to estimate the number of coded symbols that have to be requested.

Let the number of iterations in collaborative decoding be denoted by N_{iter}. For the results presented in this paper, we set $N_{iter} = N_{rx} - 1$. Given the overhead constraint, RX_1 limits the number of bits that can be exchanged in each iteration to $p\theta_{MRC}/N_{iter}$. In each iteration, RX_1 sorts the information bits according to the reliabilities, and obtains the competing path for each LRB using the technique described in Chapter 5.2.1. Then for each LRB,

1. RX_1 estimates κ using (6.9).
2. RX_1 obtains the candidate set and the set of parities to be requested based on the instantaneous SNRs.
3. RX_1 broadcasts κ, and the indices of the parity bits that need coded symbols from another node.
4. For each bit index, the best node that has not previously transmitted information for that bit will transmit information for that bit. Each node scales it received symbols by the channel coefficient and broadcasts that information for a bit. If $\kappa > |\mathbf{S}_i|$ (the number of coded symbols required is more than the size of the candidate set), then coded symbols are obtained from the next best receiver until a total of κ symbols are transmitted.

Consider an example to illustrate to illustrate step 4 above. Assume that the codeword shown in bold in Figure 6-1 is the competing path for bit i and that the ML path is the all-zeros path. Assume that this is the first iteration in which bits in this candidate set have been

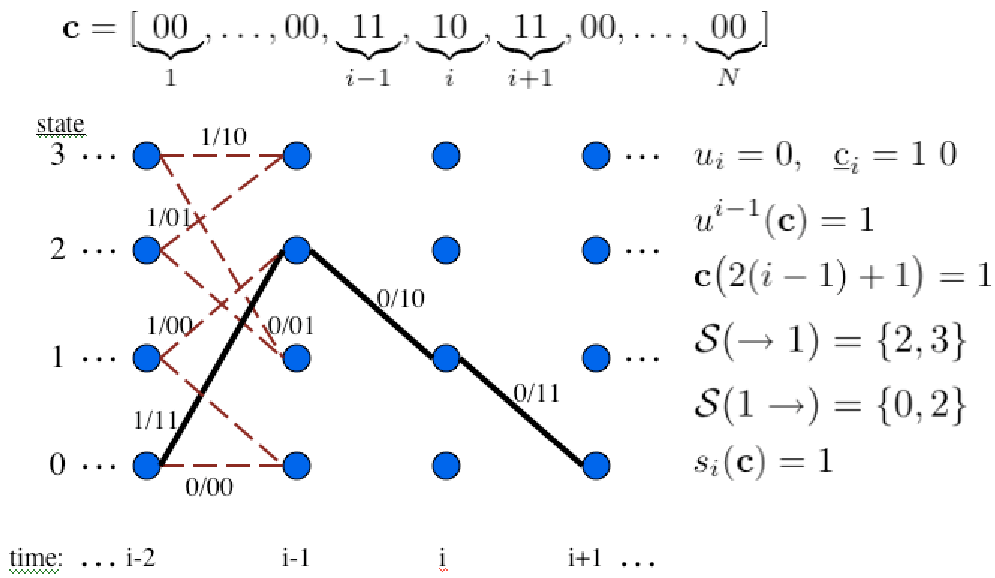

Figure 6-1: The code-trellis for the $(5, 7)$ convolutional code with examples of the notation used in this chapter.

selected to receive information from the collaborating nodes. For the sake of exposition, assume that trellis sections $i-1$, i, and $i+1$ have increasing instantaneous SNRs in that order. If $\kappa = 2$, information about $\underline{c}_{i-1}$ will be obtained from RX_2. If $\kappa = 3$, information about $\underline{c}_{i-1}$ and c_i^1 will be obtained from RX_2. If $\kappa = 7$, coded symbols for all the parity bits in the candidate set are obtained from RX_2, and coded symbols for $\underline{c}_{i-1}$ are obtained from RX_3. Once the appropriate number of coded symbols are combined for the LRB, RX_1 requests for coded symbols for the next LRB that has a *different* competing path. There may be other adjacent trellis section with the same reliability. But by requesting for κ parity bits, all the bits that had originally decoded with the same ML and competing paths are corrected with a probability that is greater than Θ. Hence, if multiple trellis sections have the same competing path (and hence the same reliability), it is enough to consider only one of them to compute the request set and request size. If the information received is able to change the decision from $\mathbf{c}_{\text{ML}}$ to $\mathbf{c}^i_{\text{comp}}$, then all the trellis section that chose the same competing path will be corrected. Thus, I-LRB exploits time-correlated reliabilities by not requesting information for all adjacent bits that decoded with the same reliability.

As previously described, coded symbols for a particular trellis section from a particular collaborating nodes are only transmitted once. Using the previous example, assume that the branch from state 2 to state 1 has already received coded symbols from RX_2 because this branch was part of a different competing path for some other bit that had a reliability less than that of bit i. Then when $\kappa = 3$, information for for $\underline{c}_{i-1}$ and c_{i+2}^1 will be obtained from RX_2 (assuming that coded symbols for these bits have not been obtained from RX_2 earlier). Also, if coded symbols for c_i^1 is required in the next iteration, it should be obtained from RX_3, and not RX_2. This procedure is repeated until a total of $p\theta_{MRC}/N_{iter}$ bits are exchanged within the cluster. Note that this includes the bits required to index the parity bits requested by RX_1. In practice, all of the information requests can be performed at the beginning of an iteration, followed by each receiver's response starting from RX_2 to $RX_{N_{rx}}$. RX_1 combines all of the received information with its previously received information using MRC (on a bit-by-bit basis). If RX_1 is able to decode correctly or the maximum number of iterations has been reached, then the collaborative decoding process terminates. Otherwise, another iteration of collaborating decoding is performed.

6.5 Results

In this section, we present the performance of our collaborative decoding scheme. For all the results in this chaper, a rate $1/2$, memory-three, non-recursive, non-systematic convolutional code with generator polynomials $1 + D^2$ and $1 + D + D^2$ ($(5, 7)$ in octal notation) is used for encoding at the distant transmitter. The message consists of $N = 900$-bit packets. For all the results, the channel between the distant transmitter and the cluster of cooperating nodes is assumed to be a quasi-static Rayleigh fading channel, where the fading is constant over each packet. For all results, the number of collaborating iterations $N_{iter} = N_{rx} - 1$.

The block error rate for I-LRB and COI-MRC is shown in Figure 6-2 for different number of collaborating nodes. For these results, a 5% overhead constraint with respect to

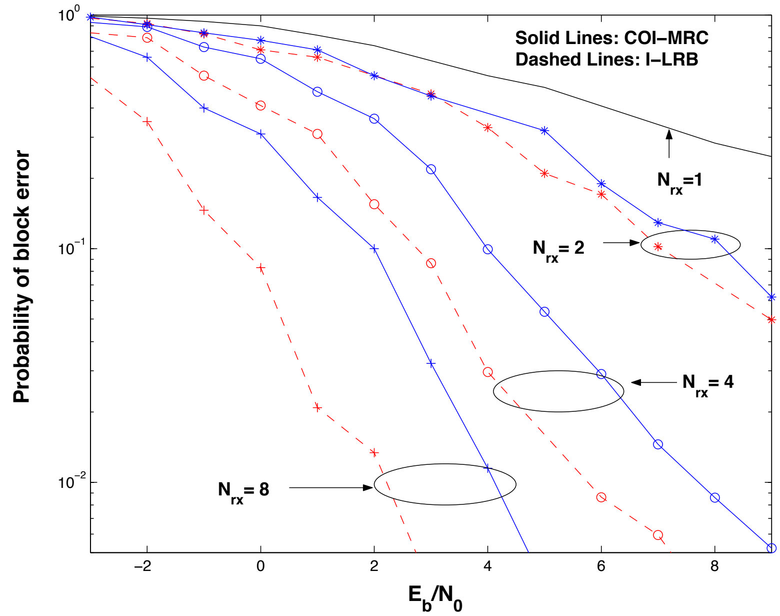

Figure 6-2: Probability of block error for different number of collaborating nodes when the overhead constraint is fixed at 5% of the overhead for MRC.

the overhead required for MRC was imposed. It is observed that I-LRB outperforms COI-MRC for all sizes of the cooperating cluster shown. It is also seen that the gain offered by I-LRB increases as the number of collaborating nodes increase. For example, with a target block error rate of 10^{-2}, I-LRB outperforms COI-MRC by approximately 2 dB when there are 8 collaborating nodes. The performance of only one node (no cooperation) is also shown for the sake of comparison. A single receiver achieves a block error rate of 10^{-2} at around 23 dB E_b/N_0. Hence, cooperation using I-LRB provides a gain of around 21 dB. The corresponding throughput for this scenario is shown in Figure 6-3. It is seen that throughput for I-LRB is larger than the throughput for COI-MRC for all the cases. At a signal-to-noise ratio (SNR) of 2 dB, and with eight collaborating receivers, I-LRB increases the throughput by almost 30% with respect to COI-MRC, and by 350% with respect to a single node.

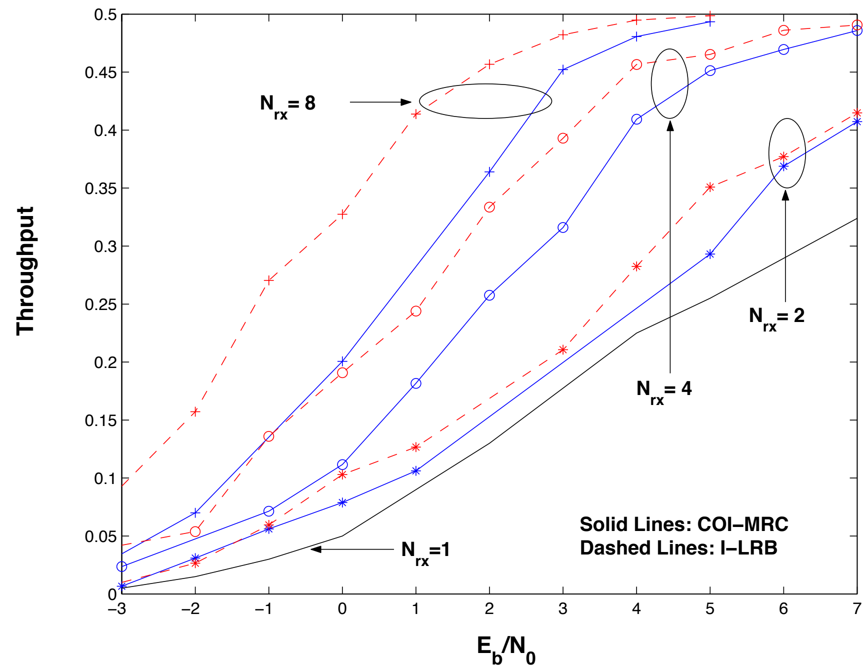

Figure 6-3: Throughput for different number of collaborating nodes when the overhead constraint is fixed at 5% of the overhead for MRC.

The block error rate of COI-MRC and I-LRB is compared in Figure 6-4 for different overhead constraints when there are eight collaborating nodes. The corresponding average cooperation overhead is shown in Figure 6-5. It is seen that I-LRB performs better than COI-MRC both in terms of block error rate and cooperation overhead. In other words, I-LRB achieves a lower block error rate with a lower cooperation overhead. The throughput of eight collaborating nodes is shown in Figure 6-6 for different overhead constraints. It is seen that I-LRB offers consistently higher throughput than COI-MRC. The throughput of I-MRC (COI-MRC with no overhead constraint) and that of a single receiver are also shown. Though I-MRC has the best block error rate among all the schemes (see Figure 6-4), it has a lower throughput when compared to I-LRB or COI-MRC. Thus, it is clear that I-MRC achieves good block error rate performance at the cost of higher overhead. It is also observed that the throughput of I-LRB decreases when the overhead constraint is relaxed. This implies that the gain in block error rate is not significant as more combining is allowed in the cooperating cluster. The increase in overhead caused by relaxing the overhead

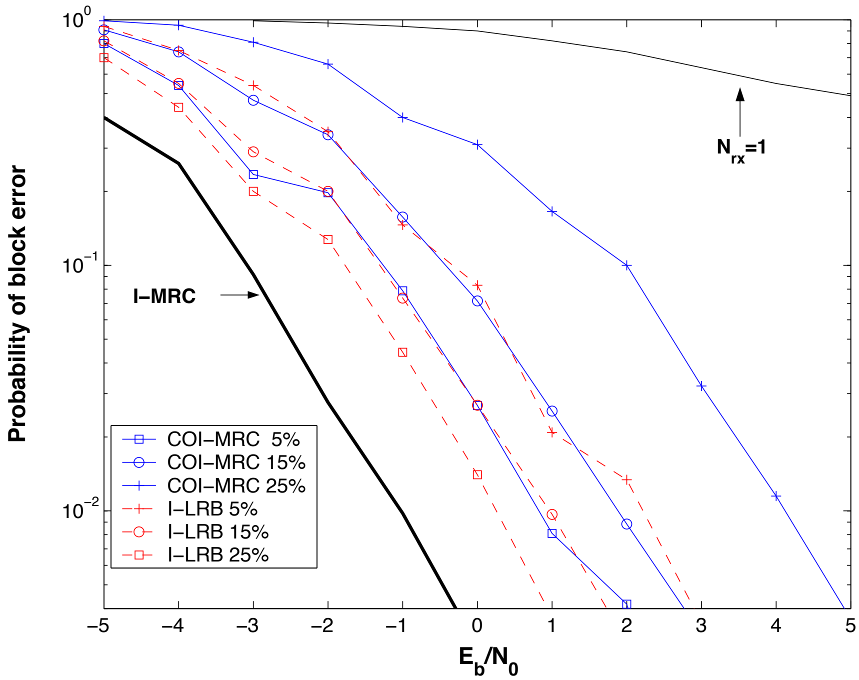

Figure 6-4: Probability of block error overhead for COI-MRC and I-LRB with eight cooperating nodes, and different constraints on the overhead.

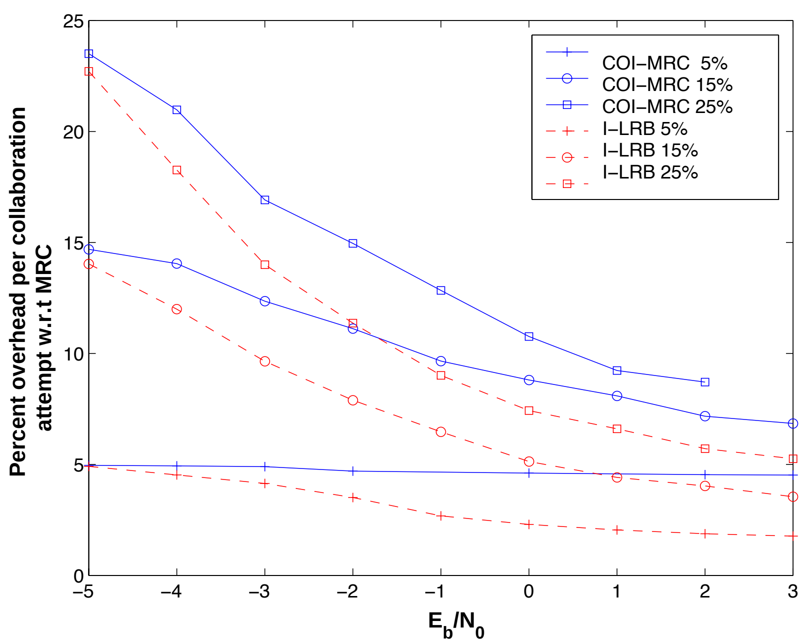

Figure 6-5: Average cooperation overhead for COI-MRC and I-LRB with eight cooperating nodes, and different constraints on the overhead.

constraint over-shadows the decrease in block error rate, leading to a lower throughput. Thus, the I-LRB scheme is capable of providing a large increase in throughput with a very

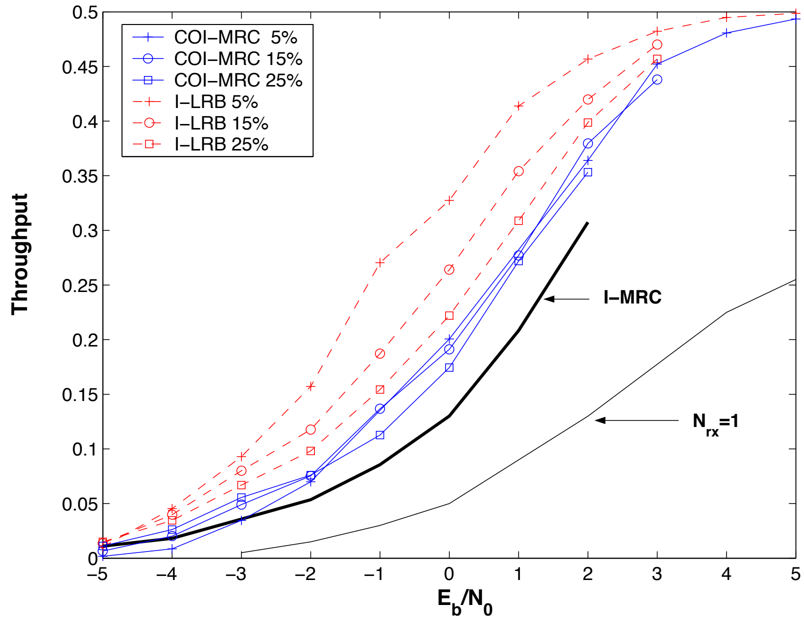

Figure 6-6: Throughput for COI-MRC and I-LRB with eight cooperating nodes, and different constraints on the overhead.

small overhead. This is because I-LRB targets the trellis-sections which are likely to be in error, and adapts the amount of information combined for these sections based on their reliabilities.

The average number of iterations required by the COI-MRC and I-LRB schemes is shown in Figure 6-7. It is seen that collaborative decoding is terminated faster in I-LRB than in COI-MRC. Since the amount of information combined in each iteration is the same in I-LRB and COI-MRC, and since I-LRB requires fewer iterations, the overhead of I-LRB is smaller than that of COI-MRC (as shown in Figure 6-5). For example, at an SNR of 0 dB and a 5% overhead constraint, I-LRB requires fewer than half the number of iterations required by COI-MRC. It can be verified from Figure 6-5 that the overhead of I-LRB is indeed around 50% of COI-MRC at 0 dB (for the 5% constraint).

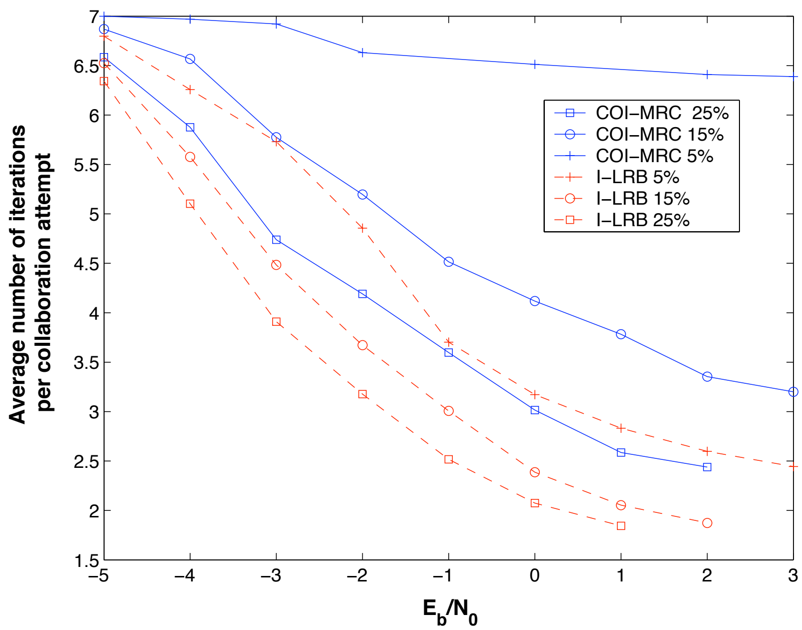

Figure 6-7: Average number of iterations per collaborative decoding attempt required by eight receivers.

CHAPTER 7
CONCLUSION AND DIRECTIONS FOR FUTURE RESEARCH

7.1 Conclusion

In this work, we have studied the idea of user cooperation from a decoding perspective. Our objective is to achieve good performance with low collaboration overhead. Our schemes are based on a very simple idea. If the cooperating nodes have some information about decoding at other nodes, the cooperation overhead can be significantly reduced. We introduced a framework called collaborative decoding to help develop cooperation strategies that are efficient in terms of the cooperation overhead. In any collaborative decoding scheme, the cooperating nodes iterate between a process of information exchange and decoding. The information exchange portion of collaboration provides information about decoding at a particular node to other nodes. The other nodes use this information to decide what information to transmit. Collaborative decoding relies on the use of soft-input soft-output (SISO) decoders. The magnitude of the output of the SISO decoder is called the *reliability* and is an indication of the correctness of the decoded bit. In collaborative decoding, the nodes use reliability information from the SISO decoder to adapt the messages that are exchanged among the cooperating nodes. This is the biggest difference between collaborative decoding and conventional cooperation strategies wherein the information exchanged during collaboration is predetermined and fixed. Unlike previous cooperation strategies, collaborative decoding provides a convenient approach to trade performance for overhead, and collaborative decoding scales easily to multiple cooperating nodes.

We also develop guidelines for the design of collaborative decoding strategies. We use these guidelines and design a novel approach called *improved least-reliable-bit* (I-LRB) collaborative decoding for user-cooperation in bandwidth-limited scenarios. The I-LRB scheme has the advantage over previously proposed cooperation strategies in that it adapts the information exchanged in collaborative process based on the *a posteriori* probabilities at the decoding node. There are two levels of adaptation in I-LRB. First, I-LRB adapts the set of bits for which information is requested based on the reliabilities. Second, for each chosen trellis section, I-LRB adapts the number of coded-symbols exchanged based on the reliability. I-LRB reduces the overhead by not combining coded symbols for all of the trellis sections that correspond to a single error event.

The advantages of the I-LRB scheme come from exploiting information generated in the BCJR decoder. We show that temporal correlation in reliabilities arise due to the same choice of competing paths for different trellis sections. We show that the competing paths can be explicitly calculated using computations that are already performed in the decoder. By observing competing paths that occur in the decoder, I-LRB can request for the minimum number of coded symbols that can correct all the trellis sections that choose that competing path in their reliability computation. Simulation results show that I-LRB achieves a lower probability of block error with a lower average collaborative information exchange than the COI-MRC scheme. The results show that I-LRB can provide a 30%-60% improvement in throughput with respect to traditional cooperation schemes. The overhead required for this improvement is less than 5% of the overhead of traditional combining schemes like MRC. Thus, I-LRB offers an efficient approach for collaboration when the maximum collaborating overhead is constrained.

7.2 Directions for Future Research

We now present areas of potential research that can be pursued using the ideas present in this dissertation.

- The performance of collaborative decoding can be studied in multiple access wireless networks by abstracting the results in this dissertation into a network simulator. Since collaboration among a group of nodes introduces interference in the network, it is not clear if collaborative decoding can actually improve the throughput of the entire network.

- So far we have used fixed convolutional codes that are guaranteed to achieve the minimum bit error rate among all the convolutional codes with the same constraint length. We have not considered the issue of code design. It is not clear if a convolutional code that provides the best error-performance in a single-user scenario will also provide the best error performance in a cooperative setting. For example, we found that a recursive systematic convolutional (RSC) code provides a lower probability of block error than an equivalent non-recursive convolutional code for the same cooperation overhead. It will be interesting to study the reason behind this observation.
- We have studied user-cooperation in a bandwidth-constrained setting. User-cooperation can also be studied in an error-costrained system. In these systems, the nodes should achieve a certain bit/block error rate through cooperation. We can compare collaborative decoding to conventional cooperation schemes to see which technique achieves the required error rate with the lowest overhead. One way to design a collaborative decoding scheme for this system is compute the number of trellis sections for which information is to be requested in order to achieve the target error rate. The closed form expression for the bit error rate can be used to compute this. Using the closed form approximation it is seen that a target bit error rate translates to a target mean of the reliabilities. Thus, after a round of decoding, each node can compute its mean of the reliabilities and then estimate how much information to request in order to achieve the target mean.

REFERENCES

[1] J. G. Proakis, *Digital Communications,* 4th ed., McGraw-Hill, New York, 2000.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, "Increasing uplink capacity via user cooperation diversity," in *Proc. 1998 IEEE Int. Symp. Inform. Theory*, Boston, Aug. 1998, p. 156.

[3] J. N. Laneman, *Cooperative Diversity in Wireless Networks: Algorithms and Architechtures,* Ph.D. dissertation, M.I.T, September 2002.

[4] N. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[5] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–part I: System description," *IEEE Trans. Commun*, vol. 51, pp. 1927–1938, Nov. 2003.

[6] 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.

[7] T. E. Hunter and A. Nosratinia, "Cooperative diversity through coding," in *Proc. 2002 IEEE ISIT*, Laussane, Switzerland, July 2002, p. 220.

[8] B. Zhao and M. Valenti, "Distributed turbo coded diversity for the relay channel," *IEE Electronics Letters*, vol. 39, pp. 786–787, May 2003.

[9] T. F. Wong, X. Li, and J. M. Shea, "Iterative decoding in a two-node distributed array," in *Proc. 2002 IEEE Military Communications Conference (MILCOM)*, Anaheim, CA, Oct. 2002, vol. 2, pp. 704.2.1–5.

[10] J. Wieselthier, G. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in ad hoc wireless networks," in *Proc. IEEE Military Communications Conference*, 1999, pp. 1414–1418.

[11] E. C. van der Meulen, *Transmission of Information in a T-Terminal Discrete Memoryless channel*, Ph.D. dissertation, University of California, Berkeley, CA, September 1968.

[12] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Info. Theory*, vol. IT-25, pp. 572–584, Sept. 1979.

[13] T. Cover and J. Thomas, *Elements of Information Theory*, New York: Wiley and Sons, 1991.

[14] P. Gupta and P. R. Kumar, “Towards an information theory of large networks: An achievable rate region,” *IEEE Trans. Info. Theory*, vol. IT-49, pp. 1877–1894, Aug. 2003.

[15] G. Kramer, M. Gastpar, and P. Gupta, “Information-theoretic multi-hopping for relay networks,” in *2004 Int. Zurich Seminar*, Zurich, Switzerland, Feb. 2004, pp. 192–195.

[16] L-L.Xie and P. R. Kumar, “A network information theory for wireless communication: Scaling laws and optimal operation,” *IEEE Trans. Info. Theory*, vol. 50, no. 5, pp. 748–767, May 2004.

[17] L-L.Xie and P. R. Kumar, “An achievable rate for the multiple level relay channel,” *IEEE Trans. Info. Theory*, vol. 51, no. 4, pp. 1348–1358, Apr. 2005.

[18] G. Kramer, M. Gastpar, and P. Gupta, “Capacity theorems for wireless relay channels,” in *Proc. 41st Annual Allerton Conf. on Commun., Control and Comp.*, Monticello, IL, Oct. 2003.

[19] F. M. J. Willems, *Information Theoretical Results for the Discrete Memoryless Multiple Access Channel*, Ph.D. dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, October 1982.

[20] F. M. J. Willems, “The discrete memoryless channel with partially cooperating encoders,” *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 441–445, May 1983.

[21] F. M. J. Willems and E. C. van der Meulen, “The discrete memoryless channel with cribbing encoders,” *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 313–327, May 1985.

[22] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Space-time codes for high-data-rate wireless communications: Performance criterion and code construction,” *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[23] G. J. Foschini and M. Gans, “On the limits of wireless communication in a fading environment using multiple antennas,” *Wireless Personal Commun.*, vol. 6, pp. 311–335, 1998.

[24] A. Sendonaris, *Advanced Techniques for Next-Generation Wireless Systems*, Ph.D. dissertation, Rice University, May 1999.

[25] T. E. Hunter and A. Nosratinia, “Coded cooperation under slow fading, fast fading and power control,” in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2002.

[26] T. E. Hunter and A. Nosratinia, “Performance analysis of coded cooperation diversity,” in *Proc. 2003 Int. Conf. Commun.*, Anchorage, AK, May 2003, vol. 4, pp. 2688–2692.

[27] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[28] C. Berrou, A. Galvieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. 1993 IEEE Int. Conf. Commun.*, Geneva, Switzerland, 1993, vol. 2, pp. 1064–1070.

[29] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[30] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[31] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and max-log-MAP decodings," *IEEE Commun. Letters*, vol. 2, no. 5, pp. 137–139, May 1998.

[32] J. M. Shea, "Reliability-based hybrid ARQ," *IEE Electronics Letters*, vol. 38, no. 13, pp. 644–645, June 2002.

[33] A.Roongta and J. M. Shea, "Reliability based hybrid ARQ using convolutional codes," in *Proc. Int. Conf. Commun. (ICC)*, Anchorage, AK, May 2003, pp. 2889 –2893.

[34] L. Reggiani and G. Tartara, "Probability density functions of soft information," *IEEE Commun. Letters*, vol. 6, pp. 52–54, Feb. 2002.

[35] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 891–907, May 2001.

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

[37] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rates," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[38] T. F. Wong, X. Li, and J. M. Shea, "Distributed decoding of rectangular parity-check code," *IEE Electronics Letters*, vol. 38, pp. 1364–1365, Oct 2002.

[39] S. Draper, B. Frey, and F. Kschischang, "Interactive decoding of a broadcast message," in *Proc.* 41^{st} *Annual Allerton Conf. on Commun., Control and Computing*, Monticello, IL, Oct. 2003.

[40] B. Blanchard, "Quantization effects and implementation considerations for turbo decoders," M.S. thesis, University of Florida, 2002.

[41] G. Montorsi and S. Benedetto, "Design of fixed-point iterative decoders for concatenated codes with interleavers," *IEEE J.Select. Areas Commun.*, vol. 19, pp. 871–882, May 2001.

[42] W. E. Ryan, "Concatenated codes and iterative decoding" in *Wiley Encyclopedia of Telecommunications* (J. G. Proakis ed.), Wiley and Sons, New York, 2003.

[43] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[44] A. Avudainayagam, A. Roongta, and J. M. Shea, "Improving the efficiency of reliability-based hybrid-ARQ with convolutional codes," in *Proc. 2005 IEEE Military Communications Conference*, Atlantic City, NJ, Oct. 2005, pp. 1–7.

BIOGRAPHICAL SKETCH

Arun Avudainayagam received the B.E degree in electronics and communication engineering in 2000 from Anna University, India, and the M.S degree in electrical and computer engineering in 2001 from the University of Florida. Since January 2002, he has been working towards his Ph.D. degree at the University of Florida. He interned with the WCDMA systems test group at Qualcomm, Inc. from May 2005- Dec. 2005, where he was involved in access stratum performance evaluation of Qualcomm's WCDMA modem chips. His research interests include wireless communications, codes on graphs and belief propagation, iterative decoding techniques, cooperative communication, and applied error control coding.