Collaborative decoding of a broadcast message in bandwidth-constrained environments

Arun 'Nayagam, John M. Shea, and Tan F. Wong

Wireless Information Networking Group (WING), University of Florida

Email: {arun@dsp, jshea@ece, twong@ece}.ufl.edu

Abstract

We present a cooperative communication scheme in which a group of receivers can collaborate to decode a message that none of the receivers can individually decode. The proposed approach is targeted at systems in which the cooperative exchanges must utilize digital modulation. In such systems, previously proposed schemes such as amplify-and-forward require a large amount of information be exchanged when there are many collaborating nodes. The approach presented in this paper provides a higher level of adaptation than previously proposed cooperative schemes. The approach presented in this paper is called improved least-reliable bits (I-LRB) collaborative decoding. 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 a constrained-overhead, incremental form maximal ratio combining (MRC).

Keywords: cooperative communications, SISO decoding, user cooperation, cooperative diversity

I. Introduction

Several network-based approaches to achieve spatial diversity have been studied in recent years [1]–[4]. In these schemes multiple nodes take advantage of spatial diversity in wireless communications by collaborating to efficiently relay and combine the different received copies of a message. This type of collaboration is useful when the physical sizes of the radios do

This work was supported by the Office of Naval Research under Grant N00014-02-1-0554, by the National Science Foundation under Grant ANI- 0220287, and by the DoD Multidisciplinary University Research Initiative administered by the Office of Naval Research under Grant N00014-00-1-0565.

not permit the use of multiple antennas. The diversity achieved through collaboration has been termed *cooperative* diversity.

The communication channel resulting from the use of cooperative diversity is a *relay channel*. The *relay channel*, first studied by Van der Meulen in 1968 [5], can be considered as the incipient stages of user cooperation. In 1979, Cover and El Gamal put forward several different approaches to achieve user cooperation in a relay channel [6]. In *cooperation*, the relay decodes the transmission from 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. Variants of these two relaying schemes, called amplify-and-forward (A-F) and decoded-and forward (D-F), respectively, were considered for fading channels in [1], [7]. Sendonaris *et al.* [2] study relaying-based user cooperation in a cellular CDMA system to increase the sum capacity of the network. The schemes in [2] are based on simple repetition coding. Several relaying techniques that use better error correction codes have been studied recently in [8], [9], [10]. These coded cooperation schemes do not easily scale to cooperation with more than one relay. Also note that all these schemes rely on correct decoding at the relay, and thus are limited by the rate between the source and the relay.

In many digital communication systems, it is not possible to transmit an arbitrary analog signal, and thus amplifying and forwarding the analog received signals is not possible. For instance, many mobile devices operate their power amplifiers in the saturation region and utilize signals with a low peak-to-average power ratio. In such systems, the received symbols have to be quantized and relayed using digital modulation. The messages exchanged by cooperating nodes contribute to overhead in the system, and will henceforth be referred to as *cooperation overhead*. A-F results in high overhead since demodulator outputs must be quantized and broadcast for each received symbol at each collaborating node. Thus the overhead become particularly large as the number of collaborating nodes increases. This high overhead will not be acceptable in systems that are constrained in bandwidth or that require a certain minimum throughput guarantee.

Collaborative decoding schemes were proposed in [11], [3], [4], [12] as a technique to allow a group of receivers to collaborate to recover a message transmitted from a distant transmitter. One of the primary features that distinguishes collaborative decoding from other cooperative communication schemes is that collaborative decoding utilizes reliability information generated in soft-input soft-output decoding to determine which information to exchange. In this article, we present a collaborative decoding scheme that makes better use of reliability information that is temporally correlated. We develop a collaborative decoding scheme for convolutional codes that can achieve better performance than other collaborative approaches, while still achieving performance close to that of MRC.

In this paper we present a collaborative decoding scheme that provides better utilization of soft information in the soft-input and soft-output decoder to provide good performance with low collaborative overhead for convolutionally encoded communication. We show that reliability information can be temporally correlated because it comes from the same set of competing paths in the code trellis. We then develop a collaborative decoding scheme that utilizes this fact. We show that competing paths in the code trellis can be explicitly computed using calculations that are already performed in the max-log-MAP implementation of the BCJR [13] algorithm. We also show how the amount of information can be adapted for each unreliable trellis section. The scheme we develop offers the following advantages over previous cooperative schemes:

- Unlike D-F, correct decoding is not required at any of the cooperating nodes.
- The cooperating overhead is smaller than comparable A-F schemes.
- The schemes scale easily to multiple relays.

II. System model

The system model studied in this work is shown in Figure 1. A distant transmitter broadcasts a packet to a cluster of receiving nodes. Typical scenarios could be military applications in which a battleship broadcasts a message to a platoon of soldiers on the mainland or commercial applications wherein a base station communicates with a cluster of mobile users. 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

$$r_{i,j} = a_j x_i + n_{i,j}, \tag{1}$$

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 schemes that we propose, 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 [14]. The sample mean of the reliabilities at node i, μ_i, is a measurement of the overall reliability of the decoder's decision. We assume that the nodes exchange the μ_is after the first decoder iteration and that combining occurs at the node with the largest μ_i, 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.

We next describe the two main cooperative schemes that will be compared in this paper.

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 only provide an overview of it in this section. A detailed description of I-LRB is given in Section IV after we develop some necessary decoding techniques.

A. 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 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 $N_{rx}-1$ iterations, so in each iteration, $pNqN_{rx}/(RN_{iter})$ bits are exchanged, where N_{iter} is the total number of iterations allowed. The information in each iteration represents a set of $pNN_{rx}/(RN_{iter})$ received symbols from the

[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.

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.

B. *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 proposed in [11], [4]. 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 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.
4) 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.

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 IV after we develop several decoder techniques in Section III.

III. The Decoder

As previously mentioned, in this paper we consider the implementation of the I-LRB collaborative decoding scheme for convolutionally encoded communications. We utilize features of the max-log-MAP implementation of the BCJR algorithm to identify which information to exchange and how much information to exchange. We begin by defining the terminology and notation used in this section.

A. 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[3]. Note that the metric is a maximum-likelihood (ML) decision statistic for additive white Gaussian noise (AWGN) channels.

B. Max-log-MAP decoding of convolutional codes

The BCJR algorithm is a bitwise maximum *a posteriori* (MAP) decoder [13], which minimizes the bit error probability. When implemented in the log domain, the inputs to a BCJR MAP

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

decoder are *a priori* probability LLRs and LLRs for the received symbols, and the output consists of APP LLRs. For each information bit u_i, the Log-MAP decoder computes the APP LLR as

$$L(u_i|\mathbf{r}) = \ln \frac{\mathrm{P}(u_i = 0|\mathbf{r})}{\mathrm{P}(u_i = 1|\mathbf{r})} = \ln \frac{\sum_{\mathbf{c}\in C_+^i} \mathrm{P}(\mathbf{c}|\mathbf{r})}{\sum_{\mathbf{c}\in C_-^i} \mathrm{P}(\mathbf{c}|\mathbf{r})}. \tag{2}$$

In what follows, we consider a nonfading (i.e., $a_j = 1$ in (1)) additive white Gaussian noise (AWGN) channel. The results extend easily to the case of quasi-static fading by premultiplying the codeword by the channel coefficient a_j. 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 log-APP in (2). Using this approximation, and assuming that all the codewords are equally likely, the soft-output for codewords transmitted on an AWGN channel with noise variance σ^2 can be written as [15]

$$L(u_i|\mathbf{r}) = \min_{\mathbf{c}\in C_+^i} \left(\frac{\|\mathbf{r}-\mathbf{c}\|^2}{2\sigma^2} \right) - \min_{\mathbf{c}\in C_-^i} \left(\frac{\|\mathbf{r}-\mathbf{c}\|^2}{2\sigma^2} \right). \tag{3}$$

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

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

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.

◊ *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,

$$\mathbf{c}^i_{\mathrm{comp}} = \underset{\{\mathbf{c}\in\mathcal{C}: u^i(\mathbf{c}) \neq u^i(\mathbf{c}_{\mathrm{ML}})\}}{\operatorname{argmin}} \|\mathbf{r}-\mathbf{c}\|^2. \tag{4}$$

As in the case of the ML codeword, there may be more than one codeword that satisfies (4), in which case the tie is broken by choosing one of the codewords arbitrarily. Note that although there is only one $\mathbf{c}_{\mathrm{ML}}$, there may be many different $\mathbf{c}^i_{\mathrm{comp}}$ for different values of i.

Then the reliability for bit i, which is the magnitude of the soft information in (3), can be expressed as

$$\Lambda_i \triangleq |L(u_i|\mathbf{r})| = \frac{1}{2\sigma^2} \left\{ \|\mathbf{r}-\mathbf{c}^i_{\mathrm{comp}}\|^2 - \|\mathbf{r}-\mathbf{c}_{ML}\|^2 \right\}. \tag{5}$$

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) 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 [16], [17]. 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 IV utilizes both the bit reliabilities and knowledge of $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}_{\text{comp}}^i$ in determining which information should be exchanged in the collaborative decoding process. In the next section, we detail how $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}_{\text{comp}}^i$ can be determined for a particular trellis section.

C. Obtaining the ML and competing path using the BCJR algorithm

Following the development in [18], the soft information in (3) can be expressed as

$$L(u_i|\mathbf{r}) = \max_{C_+^i}\Big(\alpha_{i-1}(s') + \gamma_i(s',s) + \beta_i(s)\Big) - \max_{C_-^i}\Big(\alpha_{i-1}(s') + \gamma_i(s',s) + \beta_i(s)\Big), \qquad (6)$$

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

It can also be shown that (see [18])

$$\alpha_i(s) = \max_{s' \in \mathcal{S}(\rightarrow s)} (\alpha_{i-1}(s') + \gamma_i(s',s)) \qquad (7)$$

$$\beta_{i-1}(s) = \max_{s' \in \mathcal{S}(s \rightarrow)} (\beta_i(s') + \gamma_i(s,s')) \qquad (8)$$

$$\gamma_i(s',s) \propto -\|\underline{\mathrm{r}}_i - \underline{\mathrm{c}}_i\|^2, \qquad (9)$$

where $s' \in S(\rightarrow s)$ and $s' \in S(s \rightarrow)$ are defined in Table I, $\alpha_0(0) = 0$ and $\beta_N(0) = 0$. Thus, it is seen from (9) that $\gamma_i(s',s)$ is proportional to the branch metric (cf. [19]), $P(\underline{\mathrm{r}}_i|\underline{\mathrm{c}}_i)$, 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 (6) 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 (3) and (6), 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 $\mathbf{c}_{\text{comp}}^i$. For example, if

$$\max_{C_+^i}\Big(\alpha_{i-1}(s') + \gamma_i(s',s) + \beta_i(s)\Big) > \max_{C_-^i}\Big(\alpha_{i-1}(s') + \gamma_i(s',s) + \beta_i(s)\Big),$$

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

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

Proof: See Appendix.

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

$$s_{k-i}' = \underset{s \in \mathcal{S}(\rightarrow s_{k-i+1}')}{\text{argmax}} \ \{\alpha_{k-i}(s) + \gamma_{k-i+1}(s, s_{k-i+1}')\}, \quad i = 2, 3, \ldots, k \tag{10}$$

$$s_{k+i}^* = \underset{s \in \mathcal{S}(s_{k+i-1}^*)}{\text{argmax}} \ \{\beta_{k+i}(s) + \gamma_{k+i}(s_{k+i-1}^*, s)\}, \quad i = 1, 2, \ldots, N-k \tag{11}$$

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

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 $s_i(\mathbf{c}^i_{\text{comp}})$. By recording information about the states that lead to the maximum values in (7) and (8) during the BCJR algorithm, $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ can be computed with no additional computations.

During the trace-back (or trace-forward) procedure, if $s_{i-k}(\mathbf{c}_{\text{ML}}) = s_{i-k}(\mathbf{c}^i_{\text{comp}})$ for some k, then the sequence of state-transitions obtained for any time before k will be the same for $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$. Similarly, if or $s_{i+k}(\mathbf{c}_{\text{ML}}) = s_{i+k}(\mathbf{c}^i_{\text{comp}})$, then the sequence of state-transitions will be the same for $\mathbf{c}_{\text{ML}}$ and $\mathbf{c}^i_{\text{comp}}$ for any time after k. Thus, it is sufficient to execute the trace-back and trace-forward procedures until $s_{i\pm k}(\mathbf{c}_{\text{ML}}) = s_{i\pm k}(\mathbf{c}^i_{\text{comp}})$.

IV. Improved Least Reliable Bits Collaborative Decoding

In this section we describe the Improved LRB (I-LRB) collaborative decoding scheme for convolutionally encoded communications. It is well known that errors at the output of a convolutional code are bursty, and similarly the soft-output/reliabilities are temporally correlated [20], [17]. One 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). We have verified this occurrence through simulation.

Recall that in I-LRB, the best receiver sorts the trellis section 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 section, 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.

A. 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) can be further simplified as

$$\Lambda_i = \frac{1}{\sigma^2}\mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}). \tag{12}$$

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

$$\Lambda_{i,j} = \frac{1}{\sigma^2} a_j^* \mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}), \tag{13}$$

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.

$$\mathbf{S}_i = \{k : \mathbf{c}_{\text{ML}}(k) \neq \mathbf{c}^i_{\text{comp}}(k)\} = \{k : \mathbf{e}^i(k) = 1\} \tag{14}$$

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[4], 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

$$\|\mathbf{r}^* - \mathbf{c}^i_{\text{comp}}\|^2 < \|\mathbf{r}^* - \mathbf{c}_{ML}\|^2 \tag{15}$$

$$\Longrightarrow \quad 2\mathbf{r}^{*\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}) < 0 \tag{16}$$

$$\Longrightarrow \quad 2\mathbf{r}^{\text{T}} \cdot (\mathbf{c}_{\text{ML}} - \mathbf{c}^i_{\text{comp}}) + \sum_{\substack{l \in \eta \\ \eta \subset \mathbf{S}_i, |\eta| = \kappa}} 2\mathbf{r}'(l)(\mathbf{c}_{\text{ML}}(l) - \mathbf{c}^i_{\text{comp}}(l)) < 0, \tag{17}$$

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 (13), 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 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

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

another receiver as follows,

$$\min_{\kappa} P(X_i < -2\sigma^2\Lambda_i) \geq \Theta \tag{18}$$

$$\min_{\kappa} Q\left(\frac{\sigma^2\Lambda_i - 2a_2^2\kappa}{2\sqrt{a_2^2\kappa\sigma^2}}\right) \geq \Theta, \tag{19}$$

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. $P(X_i < -\Lambda_i)$ will be referred to as the *correction probability after combining* (P_c).

B. 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 [21] for use in selecting which symbols should be retransmitted in an ARQ scenario. Several different schemes were considered in [21], 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 II. 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 II.

C. 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 Section III-C. Then for each LRB,

1) RX_1 estimates κ using (19).
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 2 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 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. As previously described, coded symbols for a particular trellis section from a particular collaborating nodes are only ever 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^1_{i+2} 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^1_i 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.

V. RESULTS

In this section, we present the performance of our collaborative decoding scheme. For all the results in this paper, 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 3 for different number of collaborating nodes. For these results, a 5% overhead constraint with respect to 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 COI-MRC 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 4. 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.

The block error rate of COI-MRC and I-LRB is compared in Figure 5 for different overhead constraints when there are eight collaborating nodes. The corresponding average cooperation overhead is shown in Figure 6. 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 7. 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 5), 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 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 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 8. 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). 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 that the overhead of I-LRB is indeed around 50% of COI-MRC at 0 dB (for the 5% constraint).

VI. Conclusions

In this paper, we present a novel approach called *improved least-reliable-bit* (I-LRB) collaborative decoding for user-cooperation in bandwidth-limited scenarios. We also present a new constrained-overhead incremental MRC (COI-MRC) scheme that offers good performance with lower overhead than full MRC. In the I-LRB collaborative decoding scheme, the cooperating nodes iterate between a process of information exchange and decoding. The I-LRB scheme has the advantage over COI-MRC and other previously proposed cooperation strategies 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 and the bits that distinguish between the MRC and competing paths in the BCJR decoder. Second, for each chosen trellis section, I-LRB adapts the amount of coded-symbols combined based on the reliability. I-LRB reduces the overhead by not combining coded symbols for all the trellis sections that have decoded with correlated reliabilities.

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, the I-LRB offers an efficient approach for collaboration when the maximum collaborating overhead is constrained.

Appendix

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

Lemma: $\alpha_k(s) \propto \min_{\mathbf{c}\in\mathcal{C}:s_k(\mathbf{c})=s} \|\mathbf{r}_1^k - \mathbf{c}_1^k\|^2$ 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 (7) as

$$\alpha_1(0) = 0 + \gamma_1(0,0) \tag{20}$$

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 (9), the lemma holds for $k = 1$.

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

$$\begin{aligned} \alpha_k(s^*) &= \max_{s \in \mathcal{S}(\rightarrow s^*)} \left(\alpha_{k-1}(s) + \gamma_k\left(s, s^*\right)\right) && (21) \\ &\propto \max_{s \in \mathcal{S}(\rightarrow s^*)} \left(\max_{\mathbf{c} \in \mathcal{C}: s_{k-1}(\mathbf{c}) = s} - \|\mathbf{r}_1^n - \mathbf{c}_1^{k-1}\|^2 + \gamma_k(s, s^*)\right) && (22) \\ &\propto \min_{\mathbf{c} \in \mathcal{C}: s_k(\mathbf{c}) = s^*} \|\mathbf{r}_1^k - \mathbf{c}_1^k\|^2, && (23) \end{aligned}$$

where (22) follows from the assumption about the claim, and the last equation follows from (9). 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 $(\log P(\mathbf{r}_1^k|\mathbf{c}_1^k))$ [19] 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 (7). 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 $s_k = s$. ∎

REFERENCES

[1] N. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, Accepted for publication.

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

[3] 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.

[4] A. Avudainayagam, J. M. Shea, T. F. Wong, and X. Li, "Reliability exchange schemes for iterative packet combining in distributed arrays," in *Proc. 2003 Wireless Communications and Networking Conference*, vol. 1, New Orleans, LA, Mar. 2003, pp. 832–837.

[5] 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.

[6] 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.

[7] J. N. Laneman, "Cooperative Diversity in Wireless Networks: Algorithms and architechtures," Ph.D. dissertation, M.I.T, September 2002.

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

[9] T. Hunter, S. Sanayei, and A. Nosratinia, "Outage analysis of coded cooperation," *IEEE Trans. Info. Theory*, pp. 375–391, Feb. 2006.

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

[11] 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)*, vol. 2, Anaheim, CA, Oct. 2002, pp. 704.2.1–5.

[12] A. Avudainayagam, J. M. Shea, T. F. Wong, and Y. Fang, *Design and Analysis of Wireless Networks*. Nova Science Publishers, 2004, ch. Cooperative techniques in wireless communications.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] W. E. Ryan, *"Concatenated Codes and Iterative Decoding"* in *Wiley Encyclopedia of Telecommunications* (J. G. Proakis ed.). New York: Wiley and Sons, 2003.

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

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

[21] 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.

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[5] 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.

TABLE I

NOTATION USED IN THIS PAPER

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

TABLE II

INSTANTANEOUS SNR ESTIMATION FOR TRELLIS SECTIONS BASED ON THE AVERAGE OF THE INSTANTANEOUS SNRS OF THE PARITY BITS IN THE CANDIDATE SET

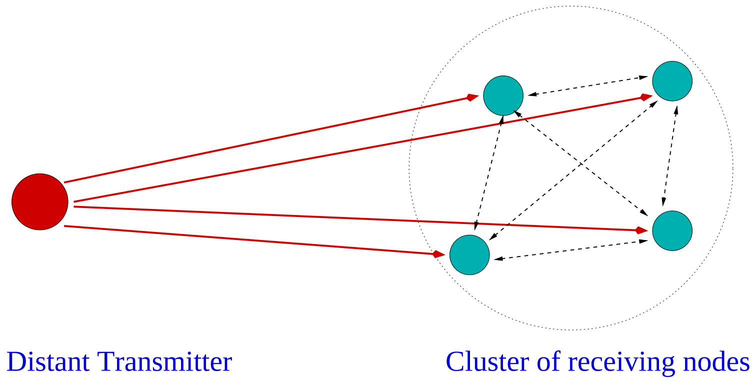

Fig. 1. System Topology for Collaborative Decoding.

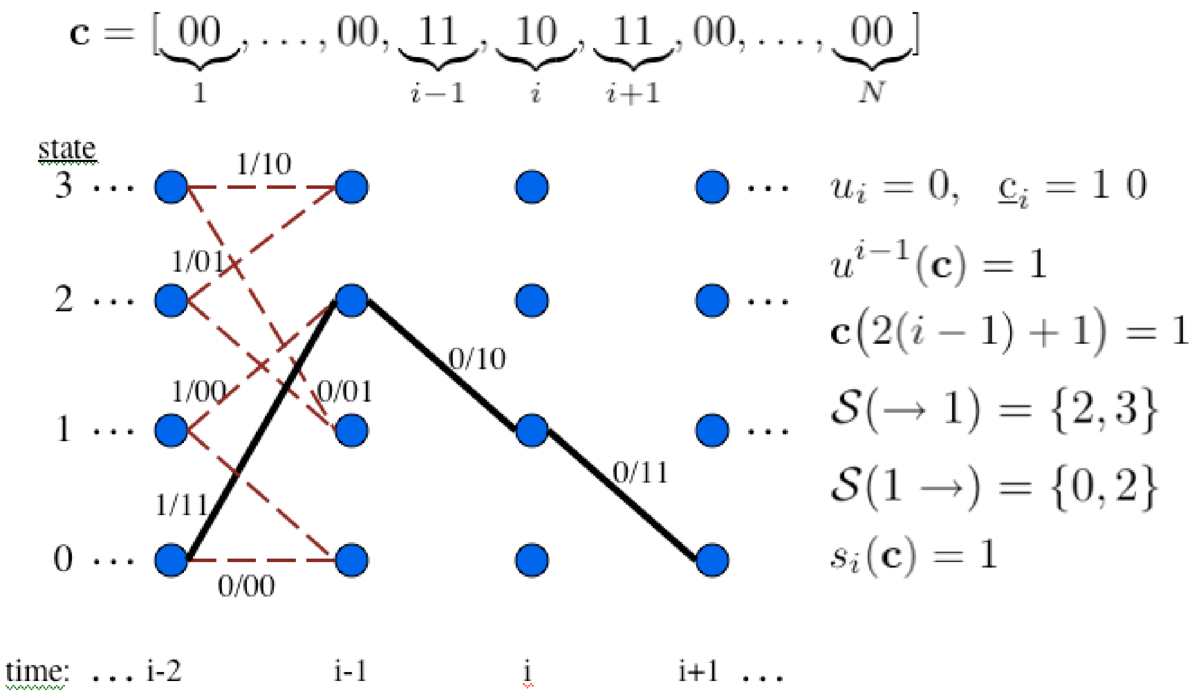

Fig. 2. The code-trellis for the $(5, 7)$ convolutional code with examples of the notation used in this paper.

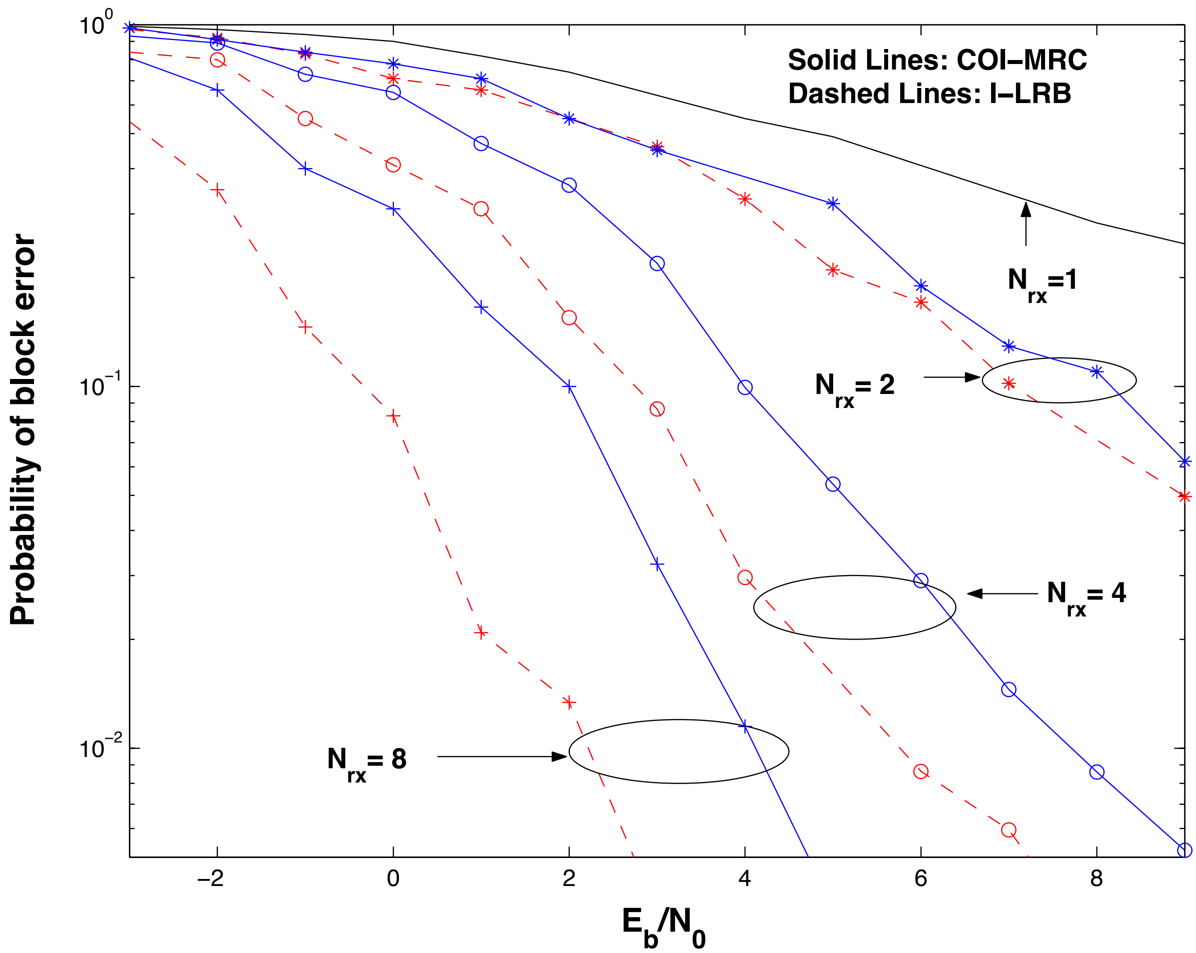

Fig. 3. Probability of block error for different number of collaborating nodes when the overhead constraint is fixed at 5% of the overhead for MRC.

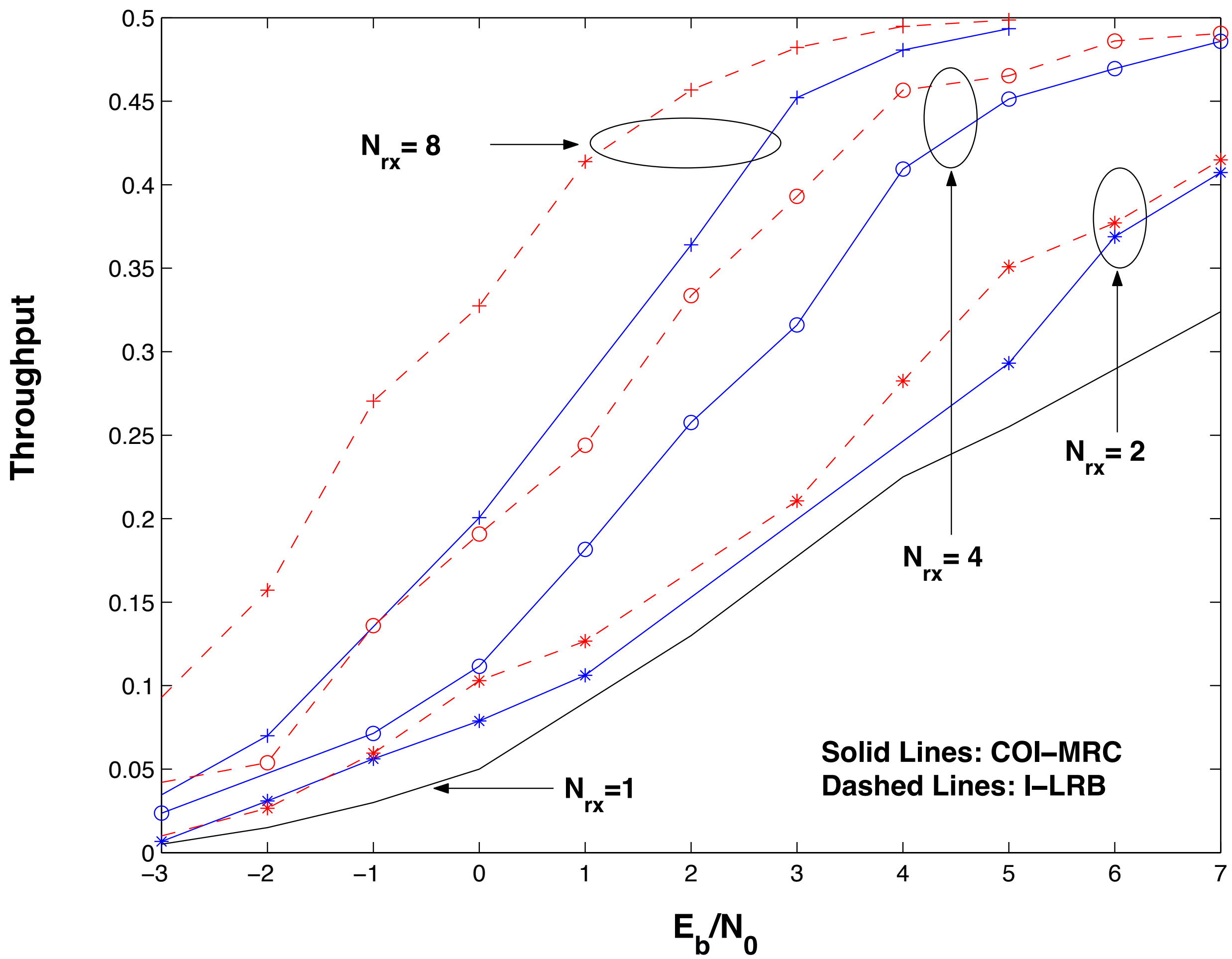

Fig. 4. Throughput for different number of collaborating nodes when the overhead constraint is fixed at 5% of the overhead for MRC

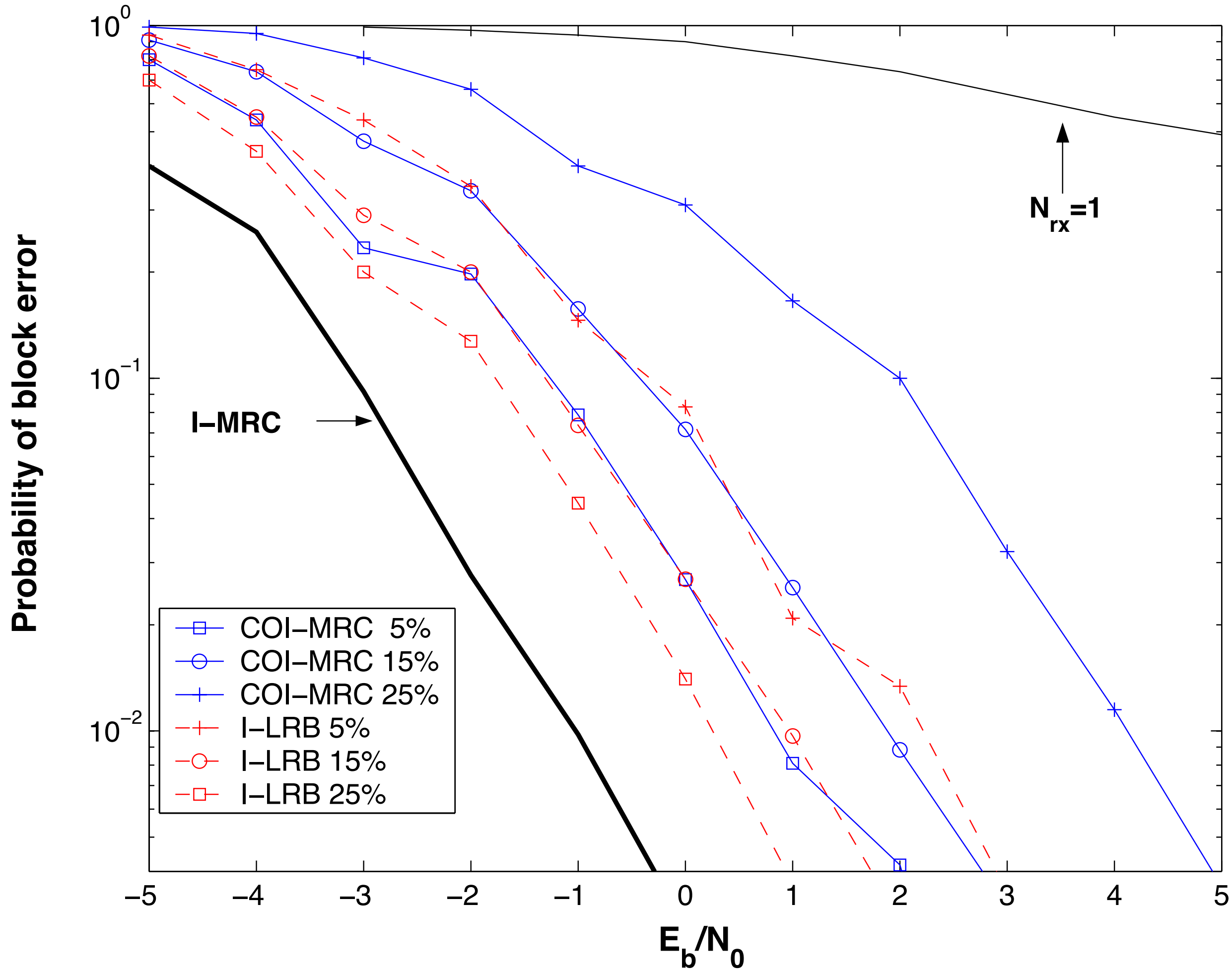

Fig. 5. Probability of block error overhead for COI-MRC and I-LRB with 8 cooperating nodes, and different constraints on the overhead.

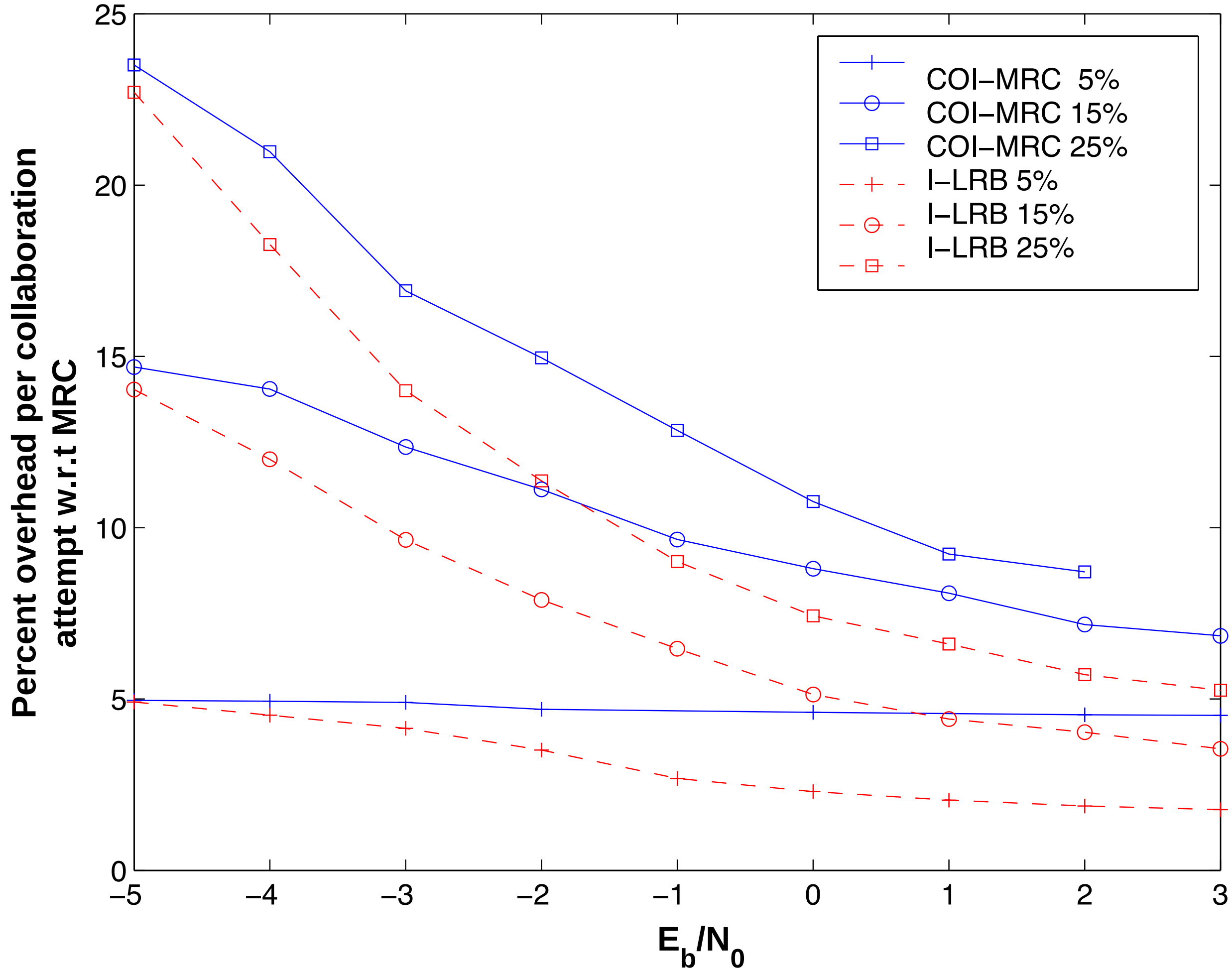

Fig. 6. Average cooperation overhead for COI-MRC and I-LRB with 8 cooperating nodes, and different constraints on the overhead.

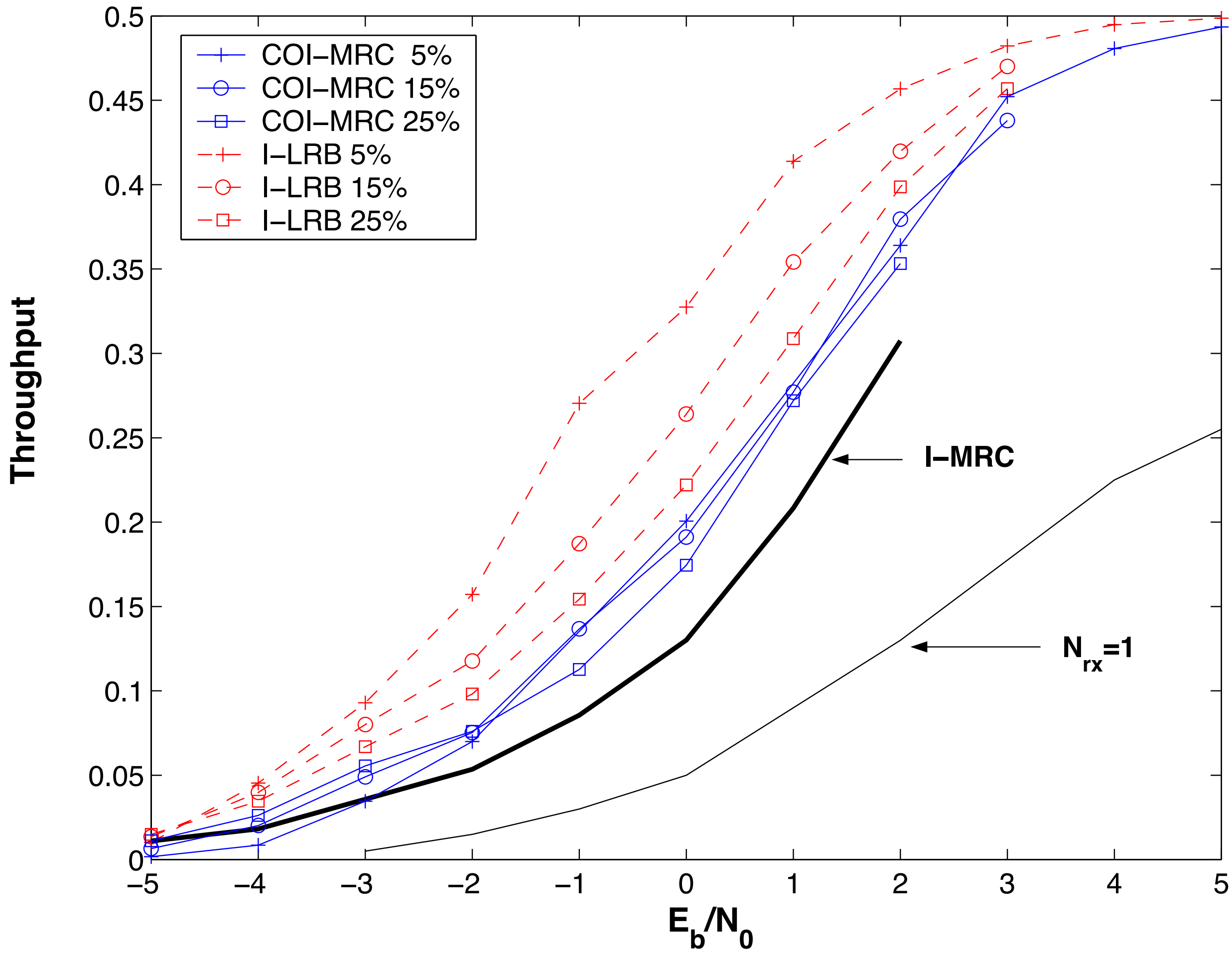

Fig. 7. Throughput for COI-MRC and I-LRB with 8 cooperating nodes, and different constraints on the overhead.

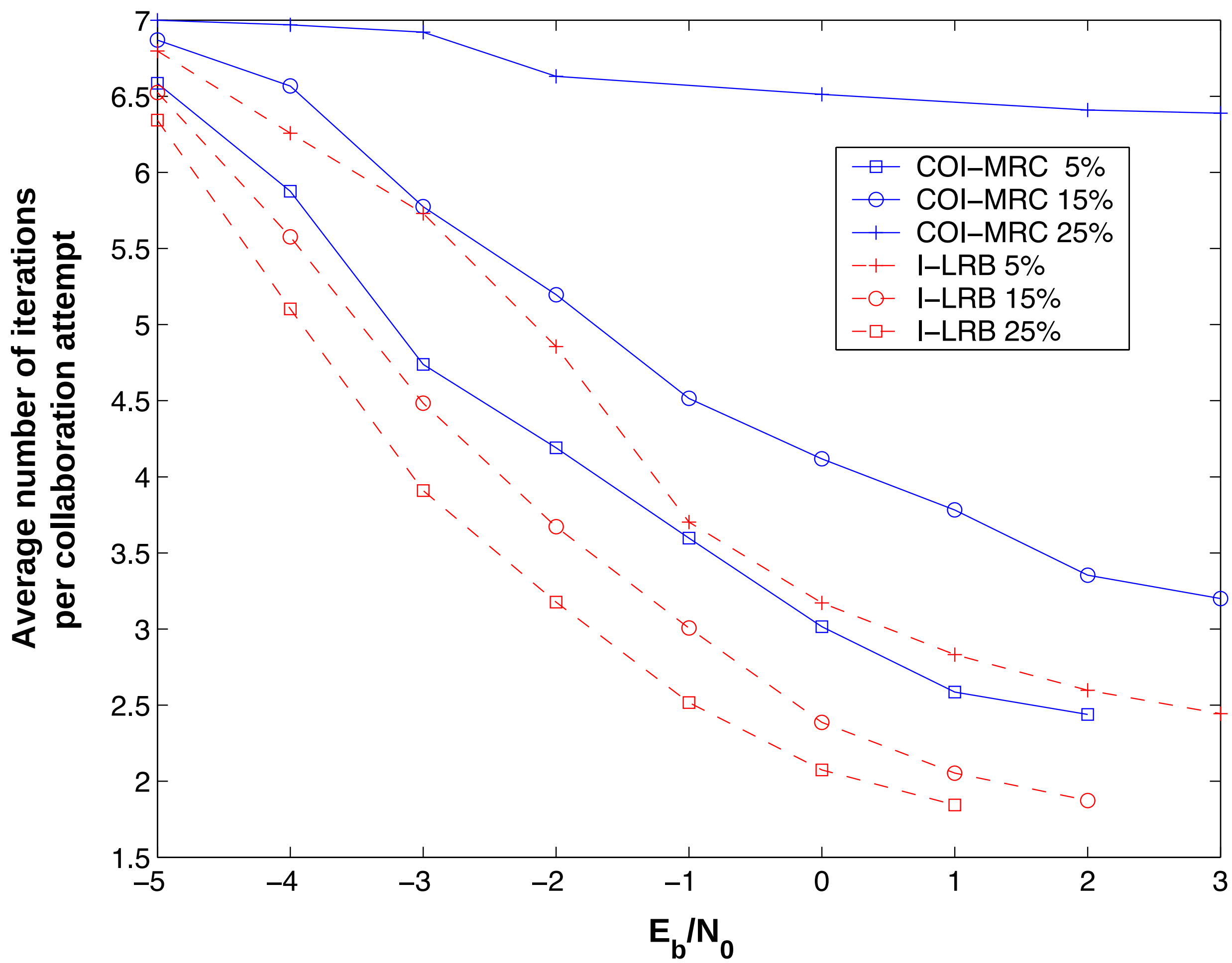

Fig. 8. Average number of iterations per collaborative decoding attempt required by 8 receivers.