Low-Complexity Cooperative Coding for Sensor Networks using Rateless and LDGM Codes

Andrew W. Eckford, Josephine P. K. Chu, and Raviraj S. Adve
Dept. of Electrical and Computer Engineering, University of Toronto
10 King's College Road, Toronto, Ontario, Canada M5S 3G4
E-mail: andrew.eckford@utoronto.ca, {chuj,rsadve}@comm.utoronto.ca

***Abstract*— Given limitations with current technology, nodes in a sensor network have stringent energy and complexity constraints. This paper presents a scheme for cooperative error-control coding, using rateless and low-density generator-matrix codes, for sensor networks. Assuming knowledge of the source-relay channel quality, we use density evolution to show that the proposed scheme achieves good performance and a good energy tradeoff despite low computational complexity. The scheme exploits the flexibility of rateless codes to permit, depending on the channel conditions, independent, relay and cooperative modes of operation. As a motivating example, we analyze networks of two cooperating nodes communicating with a more sophisticated receiver. We also discuss the generalization of our framework to a multi-node system.**

I. Introduction

A *sensor network* is a system in which distributed sensors take local measurements of a phenomenon and form a network to share their information, or to transmit it to some central authority. Such networks have a wide variety of potential applications, from wildlife monitoring [1] to load monitoring in structures [2]. Many of these applications require the network to be unobtrusive and ubiquitous, and to function with little or no maintenance. Nodes, therefore, must be as small, inexpensive, and efficient as possible, though the data sink (receiver) may be quite sophisticated.

In the literature, an important strategy for efficient communication in a network uses the relay channel [3], in which a transmitter is assisted by intermediate transceiver in sending a message, where the transceiver has no message of its own to send. This idea can be generalized to cooperative diversity [4], in which two (or more) transmitters assist each other in sending their messages to a common receiver. This idea has been developed for wireless ad-hoc and sensor networks. In [5], cooperative diversity was combined with error-control coding as a more flexible strategy than merely repetition by the partner while in [6], the two component codes of a Turbo code were split up between two relaying nodes, and used to implement a distributed Turbo code.

It is notable that much current research in sensor network communication, including research cited above, ignores the computational limitations of the deployed sensor nodes. For instance, many proposed schemes for sensor networking, such as [7], rely on the capability of the sensor node to decode complicated error-control codes, such as LDPC codes. Even the encoding of such codes requires relatively high complexity, large amounts of memory, or both. If the true gains of error-control coding are to be achieved in practical sensor networks, it will be necessary to find powerful codes that are simple to encode, and relay schemes that can operate without intermediate decoding. In this regard, the proposal in [8], though using a simplistic code, is interesting for its simplicity of encoding.

Another challenge for sensor nodes is to ensure reliable communication in channels with widely varying qualities. In order for a sensor's signal to be discerned successfully, a sufficient amount of energy must arrive at a receiver. With a traditional error-correcting code, since the block length (and hence the transmission time, for fixed symbol rate) is fixed, this may be done by changing the transmitted power. However, there has recently been much interest in *rateless codes*, such as Luby transform (LT) codes [9], which can vary their block length to adapt to any channel condition. Rateless codes also have the appealing property the encoding process is extremely simple. Rateless codes have recently been proposed for relay channels [7], though in a manner that greatly increases the complexity and requires intermediate decoding at the relay. These codes are very similar to *low-density generator-matrix* (LDGM) codes, which are easy to encode, at the expense of some performance as compared to LDPC codes [10]. In fact, one may think of an LDGM code as a rateless code where the rate is fixed in advance.

The main contribution of this paper is a low-complexity error-control coding and cooperation scheme based on rateless and LDGM codes. The proposed system is unique in the literature in that it has been expressly designed with flexibility and simplicity in mind, and should be usable on contemporary sensor networking hardware. We are most interested in using rateless codes in the relay as a tool to improve efficiency – that is, using the extra information provided by the code and the relay to reduce the computational or energy burden at each of the sensors. Of work in the literature, our approach is most similar to [6]–[8]. However, unlike [6], [8], we provide the flexibility of an inherently variable rate to match a wide range of possible channel conditions, and the use of the relay is not mandatory to gain the benefit of the full code. Furthermore, our approach has a much lower computational burden than the one proposed in [7]. We point out that Turbo encoding, such as suggested in [6], is relatively simple, and that much work has been done on reducing the complexity of LDPC encoding (such as [11]), but LDGM encoding is extremely simple, and neither of these existing codes can be used ratelessly.

The remainder of the paper is organized as follows. In Section II, we introduce our system model and motivating example. In Section III, we discuss the use of rateless and low-density generator-matrix codes, and show the modifications required to operate in our framework. In Section IV we analyze our system with density evolution, and show how our framework can be generalized beyond two nodes. Finally, in Section V, we present some preliminary results found using density evolution.

II. System Model

We are concerned with sensor networks that transmit their measurements to a central authority, i.e., the sensor network is composed of several sensors and one information sink. The sensors are equipped with simple two-way digital radios, and are capable of computational tasks of limited complexity. The energy resources of the sensors are also limited, and the most energy-intensive task of the sensor is assumed to be wireless transmission; all other tasks are assumed to have negligible energy cost. On the other hand, the information sink is considered to have effectively unlimited power and computational resources. The task of the sensors is to communicate their measurements as accurately as possible to the sink.

Since reception is far more energy efficient than transmission, it makes sense for the sensors to cooperate with each other in transmitting their measurements. However, we assume that the constrained computing resources of the sensors will be fully occupied with *encoding* an error-correcting code. Decoding the relayed transmission is assumed to be beyond the sensor's abilities.

As the motivating example of this paper, consider the system in Fig. 1, with two sensors, numbered 1 and 2. Let $\mathbf{x}_1 \in \{0,1\}^k$ represent the k-bit binary information sequence observed by sensor 1, and $\mathbf{x}_2 \in \{0,1\}^k$ the k-bit information sequence observed by sensor 2. Because it observes part of sensor 1's transmission, sensor 2 may use part of its transmission to act as a relay for channel 2. The vector of transmitted bits sent by sensors 1 and 2 are $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively; the binary vectors received at the sink from sensors 1 and 2 are $\mathbf{y}_1$ and $\mathbf{y}_2$, respectively. The transmission received by sensor 2 from sensor 1 is $\mathbf{y}_c$. The links from sensors 1 and 2 to the information sink are binary symmetric channels (BSCs) with crossover probabilities p_1 and p_2, respectively; while the cross-link from sensor 1 to 2 is a BSC with crossover probability p_c. This scenario is depicted in Fig. 1. Our scheme need not be restricted to two sensors; in Section IV-B we briefly discuss the generalization of our scheme to larger sensor networks.

For ease of analysis, we assume that only sensor 2 acts as a relay, but this assumption may be relaxed. The assumption of BSCs from the sensors to the destination simplifies the analysis - in the final version of the paper we will show results where these channels are Gaussian. However, the assumption that the cross-link is a BSC is appropriate for our relay strategy, since

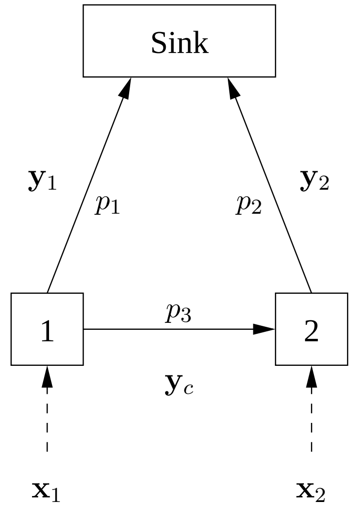

Fig. 1. Schematic of the system model. Each link is a BSC with the indicated crossover probability.

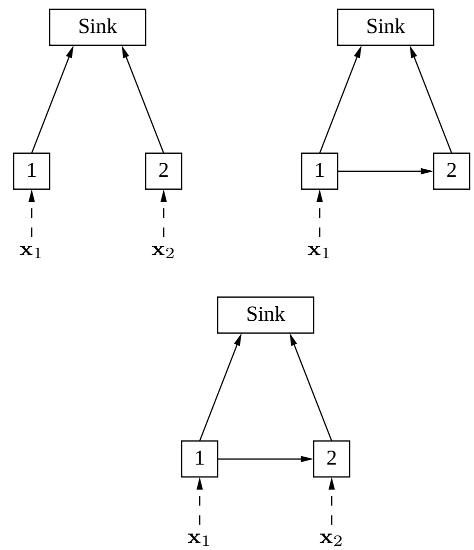

Fig. 2. System modes of operation. Clockwise from top left: independent; relay; cooperative.

the relay would make a hard decision rather than attempting to encode a continuous Gaussian value.

As illustrated in Fig. 2, three possible system architectures are considered in this paper: *independent transmission*, in which relays are not used; *relay transmission*, in which sensor 2 has no information of its own, and only acts as a relay for sensor 1; and *cooperative transmission*, in which sensor 2 splits its transmission between relaying for sensor 1 and transmitting its own information.

III. LDGM and Rateless Codes

A. Single rateless and LDGM codes

A rateless code is a code in which an information sequence $\mathbf{x}$ is mapped into a semi-infinite sequence $\mathbf{w}$, so that any prefix of $\mathbf{w}$ is a codeword of a good error-correcting code. Such codes are called rateless because they have no pre-determined rate, and the prefix property implies that the code can be terminated at will (for example, when an acknowledgment is received). An LDGM code is a linear code in which the generator matrix, $\mathbf{G}$, is sparse. The block length, and hence the rate, of an LDGM code is fixed. In this paper, we restrict ourselves to systematic LDGM codes, so that the generator matrix has the form

$$\mathbf{G} = [\mathbf{I}\ \mathbf{P}],$$

where $\mathbf{P}$ is called the *parity generator matrix*, and $\mathbf{P}$ must be sparse.

The rateless code that we use is related to the LT code [9]; codes of this type are also related to LDGM codes. In a LT code, for $1 \leq i \leq \infty$, the encoded bit w_i is obtained by forming an even parity of a random subset of $\mathbf{x}$ (of size u_i), chosen uniformly from all possibilities. More formally, let $\mathcal{N} = \{1, 2, \ldots, k\}$ represent an index set, let $\mathcal{N}_i \subseteq \mathcal{N}$ represent a subset of $\mathcal{N}$ such that $|\mathcal{N}_i| = u_i$, and let $\sigma_i : \{1, 2, \ldots, u_i\} \to \mathcal{N}_i$ be an arbitrary bijective mapping. Let $\mathbf{p}_i$ be a binary row vector of length k such that $p_{i,j} = 1$ if $j \in \mathcal{N}_i$ and $p_{i,j} = 0$ otherwise. Then

$$w_i = \mathbf{x}\mathbf{p}_i^T,$$

where the superscript ^{T} represents transposition. For any fixed length n, the rateless codeword may thus be represented as $\mathbf{w}^{(n)} = \mathbf{x}\mathbf{P}^{(n)}$, where $\mathbf{w}^{(n)}$ represents the first n bits of $\mathbf{w}$, and the parity generator matrix $\mathbf{P}^{(n)}$ is given by

$$\mathbf{P}^{(n)} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n \end{bmatrix}^T.$$

If u_i is usually small, then $\mathbf{P}^{(i)}$ is sparse. Thus, we define a systematic rateless code as a sequence of systematic LDGM codes with parity generator matrices $\{\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots\}$.

With simplicity of the encoder in mind, we propose using a single value for every check degree, i.e., $u_i = u$ a constant, $\forall i$. It is known that LT codes achieve the Shannon capacity of every erasure channel, though no degree distribution exists to achieve the capacity of every channel with symmetric noise (such as the AWGN channel) [12].

Consider some properties of this LT code. First, the parity check matrix is given by

$$\mathbf{H}^{(n)} = \left[(\mathbf{P}^{(n)})^T\ \mathbf{I}\right], \tag{1}$$

which is a low-density matrix since $\mathbf{P}^{(n)}$ is sparse. As with a standard low-density parity-check matrix, this code can be represented on a factor graph, as in Fig. 3. Second, consider the column Hamming weight of $\mathbf{P}^{(n)}$, which gives the degrees of the information variable nodes in the factor graph. Since the subsets $\mathcal{N}_i$ are selected uniformly at random from all possible subsets of size m, the variable node degrees are also random.

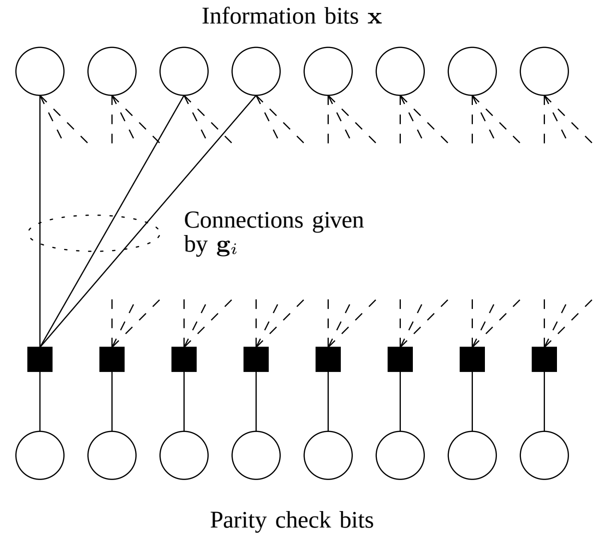

Fig. 3. Factor graph representation of a rateless code.

If k and n are asymptotically large, it is easy to see that these degrees have a Poisson distribution with parameter ν, where

$$\nu = u\left(\frac{n-k}{k}\right).$$

As a result of the Poisson distribution, some of the rows of the generator matrix must have low weight, so such a code has poor minimum distance properties. Thus, it is known that LDGM codes suffer from high error floors, which can be mitigated by concatenation of two such codes [10]. Further, it is known that unmodified rateless codes in noisy channels suffer from error floors [13]. However, the concatenation strategy introduces extra complexity, and makes it difficult to use the codes without a fixed rate. In Section V, we will present results to show that, for practical code rates and channels, the error floors might not present a problem.

Since the parity check matrix $\mathbf{H}^{(n)}$ from (1) is sparse, decoding of the rateless code may be accomplished using the sum-product algorithm over the factor graph [14]. This algorithm is now well established; we omit the details and direct the reader to the reference.

B. Rateless codes in relaying and cooperation

We now discuss the impact of relaying on the rateless coding scheme. As mentioned previously, we do not wish the relaying node to make any attempt to decode the code. Instead, we wish the relay to directly incorporate the noisy observations of its neighbor into its information string.

In our proposed scheme, the first sensor encodes its symbols in a systematic code and transmits them. The second sensor observes the first sensor's transmission, or a fraction thereof. Let $\mathbf{w}_1$ represent the transmitted sequence from sensor 1, and let $\mathbf{y}_c$ represent the received sequence at sensor 2, so that

$$\mathbf{y}_c = \mathbf{w}_1 \oplus \mathbf{z}, \tag{2}$$

where $\oplus$ represents componentwise mod-2 addition, and $\mathbf{z}$ represents a binary noise sequence.

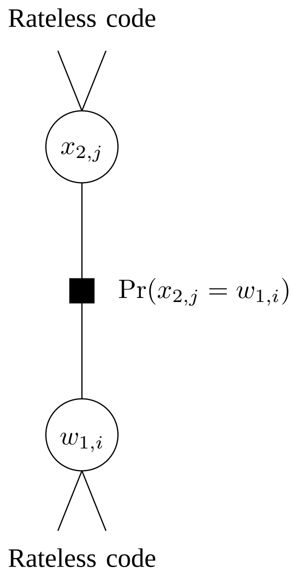

Fig. 4. Linkage between information symbols in the relay factor graph. These nodes connect the two rateless code factor graphs.

In relay mode, where sensor 2 has no information of its own to send, its information sequence $\mathbf{x}_2$ is formed by selecting k elements from $\mathbf{y}_c$, including both systematic and parity bits. In the cooperative, joint transmission mode, we more generally define ϵ as the fraction of $\mathbf{x}_2$ devoted to relaying as opposed to sending independent information. Thus, $\mathbf{x}_2$ is composed of $k(1-\epsilon)$ bits of independent information and $k\epsilon$ bits selected from $\mathbf{y}_c$ (again, either all systematic observations, or a mixture). A codeword is formed from the information vector $\mathbf{x}_2$ using a rateless or LDGM code, in the same manner as for the first sensor. Notice that setting $\epsilon = 0$ corresponds to independent transmission, and $\epsilon = 1$ corresponds to the case of relayed transmission, with all settings in between corresponding to cooperative transmission.

Decoding is slightly more complicated in the relay case, but still relies on passing messages with the sum-product algorithm. The information sequences $\mathbf{x}_1$ and $\mathbf{x}_2$ are encoded as described in the previous section, so these variables are connected to factor graph structures similar to Fig. 3. However, these information sequences are themselves correlated random variables, and are therefore connected with a factor graph structure. For example, sensor 1 transmits $w_{1,i}$ (which includes $\mathbf{x}_1$ since the code is systematic), and sensor 2 observes $y_{c,i}$. This symbol is then included in $\mathbf{x}_2$ as $x_{2,j}$. From (2), $x_{2,j} = w_{1,i} \oplus z_i$. Since we can write $\Pr(x_{2,j} = w_{1,i}) = \Pr(z_i = 0)$, $x_{2,j}$ and $w_{1,i}$ are correlated random variables. Thus, the two rateless code factor graphs are connected with structures as depicted in Fig. 4.

Messages passed through the connecting node are obtained simply from the sum-product algorithm. The messages that arrive at the connecting node represent the *a posteriori* probabilities of each symbol connected to it. For example, the message from the connecting node to the node representing $x_{2,j}$ in Fig. 4 would be given as a log-likelihood ratio by

$$\ell = \log \frac{\sum_{w_{1,i} \in \{0,1\}} p_{X_{2,j},W_{1,i}}(x_{2,j} = 0, w_{1,i}) p_{W_{1,i}}(w_{1,i})}{\sum_{w_{1,i} \in \{0,1\}} p_{X_{2,j},W_{1,i}}(x_{2,j} = 1, w_{1,i}) p_{W_{1,i}}(w_{1,i})}, \quad (3)$$

where $p_{W_{1,i}}(w_{1,i})$ represents the *a posteriori* probability of $w_{1,i}$. A similar calculation is performed for the message in the reverse direction.

C. A note on encoder complexity

Our primary motivation in proposing these codes is the low complexity with which they can be encoded. In this section, we justify our assertion that the encoder is computationally simple.

There are two computational tasks involved in encoding a linear code:

1) Storing the information string; and
2) For each column of the generator matrix $\mathbf{G}$, calculating $w_i = \mathbf{x}\mathbf{g}_i^T$.

The first task requires random-access memory equal to the length of the information string. For the second task, we deliberately formulated the sequence of parity generator matrices $\{\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots\}$ so that each parity check was generated by selecting variables at random. As a result, instead of storing a generator matrix, we can implement a pseudo-random number generator with a known seed, selecting elements of $\mathbf{x}$ pseudo-randomly and taking their mod-2 sum. A pseudo-random number generator can be implemented simply using a finite-state machine. To reduce the complexity further, we maintain the same check degree u for all parity checks, so the length of every mod-2 sum is the same. Furthermore, note that the bits w_i need not be stored beyond time i, as they can transmitted immediately.

Clearly, these hardware requirements are not particularly strenuous; or alternatively, on a general-purpose microcontroller, these tasks could be implemented in a straightforward manner in assembly language.

IV. Analysis and Discussion

A. Density evolution

We analyze the proposed system using *density evolution* [15], which is a method for analyzing message-passing decoders for linear codes with sparse parity-check matrices. Messages passed within the factor graph are functions of random observations of a codeword, and thus are themselves random variables. Density evolution tracks the probability density functions (PDFs) of all the messages passed within the factor graph, and assumes (for the sake of tractability) that all the previous messages passed through the graph that contribute to the calculation of a current message are statistically independent. For LDGM and rateless (as well as LDPC) codes, it is easy to show that this assumption becomes asymptotically correct as the number of information symbols approaches ∞. Moreover, the performance at shorter block lengths is usually similar to the asymptotic performance.

The message calculations performed at the parity check and variable nodes within the rateless code's factor graph are identical to those performed in an LDPC factor graph, for which density evolution is well known. For example, at a variable node, the outgoing message ℓ along a particular edge is the sum of all the incoming messages m_i along the other edges, i.e.,

$$\ell = \sum_{i=1}^{d_v - 1} m_i,$$

where d_v is the degree of the variable node. Under the assumption that the incident messages are all independent, the message PDF is given by

$$f_L(\ell) = f_{M_1}(m_1) \star f_{M_2}(m_2) \star \ldots \star f_{M_{d_v-1}}(m_{d_v-1}),$$

where $\star$ represents convolution. The PDF transformation at parity check nodes is tedious to describe, though not conceptually difficult, and the reader is directed to [15] for the details.

As we noted in the previous section, when the system is operating in relay mode, there exist connecting nodes in the factor graph that depict the relationship between information variables and relay variables. Since such nodes are not generally found in LDPC decoding, we need a new PDF transformation in order to implement density evolution for the relay case. The message calculation is given in (3), and is a simple mapping from a scalar to a scalar. There are many possible ways to calculate the resulting PDF transformation, but in our case, we quantize the PDF, then perform the message calculation in (3) for each quantized point, and form a histogram of the results.

B. Generalization to larger networks

It is quite easy to extend the method proposed here to a large sensor network, so long as the sensors were constrained to operate through at most one relay, and so long as all nodes in the network are able to communicate directly with the receiver. In that case, the complication would be to find a useful assignment of sensors and relays.

In a large network that disposed of these constraints, there are examples of interesting features that could be found. Here we describe two of them and indicate how they could be included in our framework, which involve minor changes to the factor graph at the receiver:

- **Multiple relays.** In a multiple relay, more than one relay node is used to convey information from source to destination. The multiple relays, numbered $r = \{2, 3, \ldots, r_{\max}\}$, would be represented by multiple variables $x_{r,j}$ correlated with $w_{1,i}$, obeying the relation (2). If each of the relays were protected by an LDGM or rateless code, the factor graphs for each code would be connected by structures such as in Fig. 4, with extra edges and variables representing the multiple relay symbols, appearing more like a "star" radiating from $w_{1,i}$ than a simple connection.
- **Compound relays.** In a compound relay, the path from source to destination includes more than one relay. We can represent a series of consecutive relays with a factor graph structure similar to Fig. 4, extending the line with more nodes and variables corresponding to each new relay in the path.

Since these behaviors can be described on a factor graph, the familiar sum-product algorithm may be used to decode the codes in a system containing them. A further challenge is the formulation of a protocol to allow them to be exploited optimally, but such a protocol is beyond the scope of this work.

V. Results

This section presents some proof-of-concept results on the proposed cooperation scheme. Density evolution is used to obtain performance results for the proposed system, providing a theoretical analysis rather than Monte Carlo simulation. For a given channel and rate, density evolution returns the average probability of symbol error at every decoding iteration for every symbol variable in the system. As we have noted above, the main drawback is that density evolution assumes the block length approaches ∞. In the final paper, we will present simulation results confirming the results from density evolution to demonstrate the proposed system's performance at practical block lengths.

Since the codes we use have error floors, and since we are making no effort to mitigate these error floors, density evolution will inevitably return some nonzero average probability of error for every possible rate. This may be surprising (and disappointing) to some rateless coding researchers, who are accustomed to rateless codes always being decoded correctly in any channel, if the rate is large enough. On the other hand, rateless codes exhibit *thresholds*: for channels worse than the threshold, the probability of symbol error is very poor (on the order of 10^{-1}), but for channels better than the threshold, the probability of symbol error is quite good (on the order of 10^{-5}). As we investigate the rateless properties of this coding system, we will generally investigate the effect of rate on the threshold, rather than attempting to determine where the probability of error vanishes.

A useful definition in this section is the *redundancy* of a relay coding system, ρ, which is the total number of channel uses per information bit. For a single code of rate R, clearly $\rho = 1/R$. However, in relay mode, it is easier to use redundancy since it is additive. For example, if $\rho = 2$ on the direct link and $\rho = 1$ on the relay link, then overall we have $\rho = 3$.

Our first result, in Fig. 5, indicates the performance of the system in relay mode, when operated as a pair of LDGM codes with fixed rates, with fixed check degree $u = 10$ in both codes. In this figure, the system is operating in relay mode, and both the direct link and the relay link use systematic LDGM codes. The redundancy in the direct link is 3, while the redundancy in the relay link is 1.5, for an overall redundancy of $\rho = 4.5$. The cross-link between the two sensors has a crossover probability p_c, fixed to the same value as p_2, and always less than p_1,, which is appropriate since the relay channel is not useful unless the cross-link is very good. The figure indicates the bit

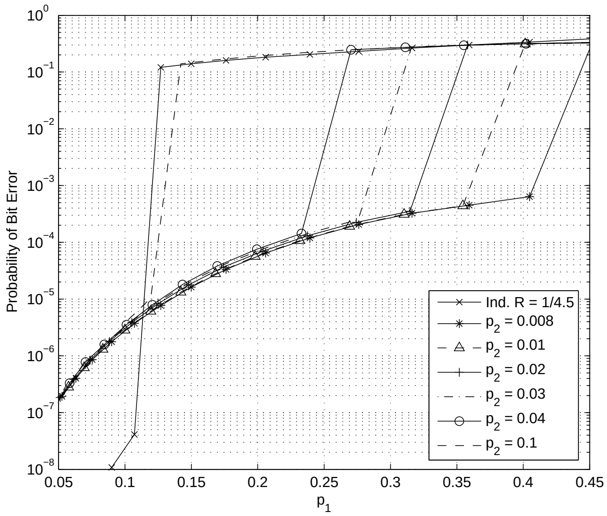

Fig. 5. Relay mode result for LDGM codes

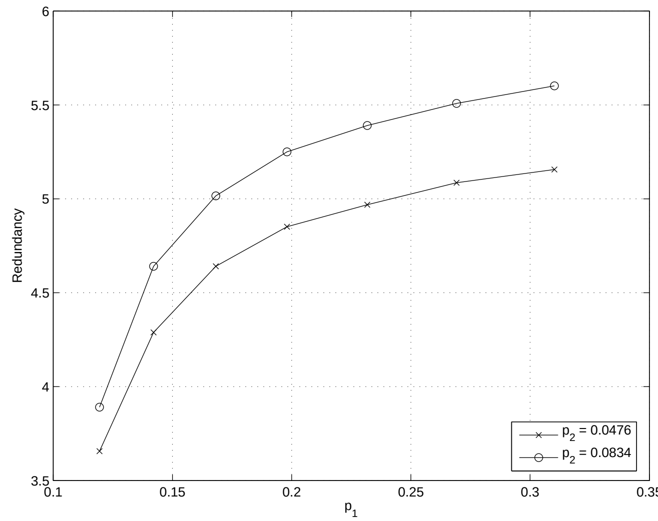

Fig. 6. Relay mode result for rateless codes

error performance of the system for various values of p_1 and p_2. For comparison, the dotted line indicates the performance of the system in independent mode with $\rho = 4.5$. Even for relatively poor p_2, the performance of the relay is superior to operating in independent mode from the perspective of the threshold. The error floor, which worsens as p_1 gets worse, is nonetheless low enough (at 10^{-4} to 10^{-5}) for worthwhile performance.

Our second result, in Fig. 6, indicates the performance of the system in relay mode, when operated ratelessly. In this curve, we indicate the minimum redundancy for which the threshold is passed. The probability of error after the threshold is not indicated, but is generally around 10^{-4} to 10^{-5}. Again, we use fixed check degree $u = 10$ in both codes, and again, $p_c = 0.01$. In this scheme, the relay waits until the redundancy of the original sender is $\rho = 3$ before starting to help, at which time the relay and the original sender transmit at the same rate. The relay's transmission is split evenly between systematic and parity bits until all the systematic bits have been sent. In the figure, we see that the redundancy flattens out as p_1 gets worse and p_2 is held constant, which indicates that the decoder is relying on the good information from the relay moreso than the bad information from the direct link, and indicates the effectiveness of the relay strategy.

The final version of the paper we will also present results illustrating the cooperative mode of the system, i.e., with both nodes transmitting their own data.

REFERENCES

[1] The Zebranet Wildlife Tracker, http://www.princeton.edu/∼mrm/zebranet.html.

[2] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Conf. on Embedded Networked Sensor Sys.*, 2004.

[3] T. Cover and A. E. Gamal, "Capacity Theorems for the Relay Channel," *IEEE Trans. Inform. Theory*, vol. 25, pp. 572–584, Sep. 1979.

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

[5] M. Janani, A. Hedayat, T. E. Hunter, and A. Nosratinia, "Coded cooperation in wireless communications: Space-time transmission and iterative decoding," *IEEE Trans. Signal Processing*, vol. 52, pp. 362–371, Feb. 2004.

[6] B. Zhao and M. C. Valenti, "Cooperative diversity using distributed turbo codes," *Proc. Virginia Tech Symp. on Wireless Personal Commun.*, 2003.

[7] J. Castura and Y. Mao, "Rateless coding for wireless relay channels," *Proc. IEEE International Symposium on Information Theory*, 2005.

[8] J. P. K. Chu and R. S. Adve, "Implementation of co-operative diversity using message-passing in wireless sensor networks," *Proc. 9th Canadian Workshop on Information Theory*, 2005.

[9] M. Luby, "LT codes," *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[10] J. Garcia-Frias and W. Zhong, "Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Communications Letters*, vol. 7, pp. 266–268, Jun. 2003.

[11] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 638–656, Feb. 2001.

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

[13] R. Palanki and J. S. Yedidia, "Rateless codes on noisy channels," *Proc. IEEE International Symposium on Information Theory*, 2004.

[14] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[15] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.