A Practical Scheme for Relaying in Sensor Networks Using Repeat-Accumulate Codes

Andrew W. Eckford 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, rsadve@comm.utoronto.ca

Submitted as a *regular paper* for CISS 2006

Abstract— **Given limitations with current technology, nodes in a sensor network have stringent energy and complexity constraints. This paper presents a scheme for cooperation via relaying and error-control coding in sensor networks, using punctured systematic repeat-accumulate codes. 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. An interesting additional benefit of such a scheme is the notion of fractional cooperation by a relay node. As a motivating example, we analyze networks of up to ten cooperating nodes in Rayleigh fading, which communicate with a more sophisticated receiver.**

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. However, the data sink may have access to substantial energy and computational resources, within the limits of reasonable expense and contemporary technology.

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. 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 networks, including research cited above, do not address the computational limitations of the sensor nodes. For instance, many proposed schemes for sensor networking, such as [7], rely on the sensor node decoding 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 error control decoding. In this regard, the proposal in [8], though using a simplistic code, is interesting for the gains achieved in spite of the low complexity required of the relay nodes.

Another challenge for sensor nodes is to ensure reliable communication in fading channels. For successful reception, 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, this is not the only option – a code with variable block length may be used as well. Example of such codes are the Luby transform (LT) codes [9], and the related Raptor codes [10]. Unfortunately, it is known that LT codes, while simple to encode, suffer from high error floors outside of erasure channels [11]. Raptor codes, which have high encoding complexity, were recently been proposed for relay channels [7], though in a manner that requires intermediate decoding at the relay.

In this paper we use a particular variety of repeat-accumulate (RA) codes [12] known as systematic punctured RA codes. RA codes are LDPC-like with low encoding complexity, are iteratively decodable with the sum-product algorithm [13], and have no error floors. Indeed, in the limit of long block length, they exhibit the same thresholding behavior as LDPC codes, with arbitrarily low probability of bit error for sufficiently high signal-to-noise ratio (SNR). Puncturing these codes gives them a multi-rate quality, in that the rate can be easily changed by varying the number of bits punctured or transmitted. Furthermore, like all codes decoded using the sum-product algorithm, it is easy to modify the decoding procedure to account for the relay. To the knowledge of the authors, this work represents the first use of RA codes in a sensor networking or relaying context.

The main contribution of this paper is a low-complexity error-control coding and relaying scheme based on RA 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 interested in using RA codes to improve efficiency – that is, using the flexibility provided by the code at the source and relay to reduce the computational or energy burden at each of the sensors. The ability to puncture the code allows *fractional cooperation* by the relay, such that it need to contribute only as much information as necessary for successful decoding. This is in contrast to the currently proposed all-or-nothing cooperation schemes [14].

In our setup, relays are used to enhance utility – that is, the total number of information bits returned to the sink per possible channel use. Of the available 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, compared to [7], our approach significantly reduces the computational burden on the sensor nodes. 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 [15]), but RA encoding is extremely simple, and LDPC codes in particular are difficult to puncture.

The remainder of the paper is organized as follows. In Section II, we introduce our system model and underlying assumptions. In Section III, we discuss the use of repeat-accumulate codes, and indicate how they can be used in a relaying framework. In Section IV, we analyze our system with density evolution. Finally, in Section V, we present some preliminary results found using density evolution.

II. System Model

Our scenario imagines a sensor network with multiple sensors and one information sink, and the task of the sensors is to convey their information as efficiently and accurately as possible to the sink. The sensors possess simple two-way radios, processors of limited complexity, and limited power resources; while the data sink possesses virtually unlimited radio, computational, and power resources, within the limits of contemporary technology. The sensors' limited capabilities, and their ability to communicate with each other, imply that they should co-operate in conveying information to the sink. Furthermore, it is assumed that data traffic is light in this network, which is a reasonable assumption for a sensor network measuring a physical phenomenon changing slowly with time.

Figure 1 depicts our system, containing n sensor nodes. To simplify the analysis, we allow single relays only; although multiple relays are possible in principle. We assume that the transmitting node is sufficiently aware of the system state that it can choose the best available relay. Since we assumed that network traffic is light, we can also assume that this desired relay is always available when needed.

Once a node picks a relay, the system may be reduced (without loss of generality) to the sensor, its relay, and the destination, as in Fig. 2. Let $\mathbf{w}_s \in \{0,1\}^k$ represent the k-bit binary information sequence observed by the sensor, and let $\mathbf{x}_s$ represent the data modulated symbol sequence representing m-bit codeword ($m > k$) generated and transmitted by the sensor. Define $\rho : \{0,1\} \rightarrow \{+1,-1\}$ as the function translating between the two binary alphabets, where $\rho(0) = +1$, and $\rho(1) = -1$, and when ρ takes a vector argument, the translation is applied to each element in the vector. Using BPSK modulation, the relay observes

$$\mathbf{y}_{sr} = A_{sr}\rho(\mathbf{x}_s) + \mathbf{n}_{sr}, \tag{1}$$

and the sink observes

$$\mathbf{y}_{sd} = A_{sd}\rho(\mathbf{x}_s) + \mathbf{n}_{sd}, \tag{2}$$

where A_{sr} and A_{sd}, incorporating the relevant channel, are the scalar amplitudes of the source-to-relay and source-to-sink links, respectively; and $\mathbf{n}_{sr}$ and $\mathbf{n}_{sd}$ are white Gaussian noise vectors with unit variance.

In the second, cooperative, phase, the relay transmits data on behalf of the source. To minimize complexity, the relay (which is itself a sensor node) does not decode the received

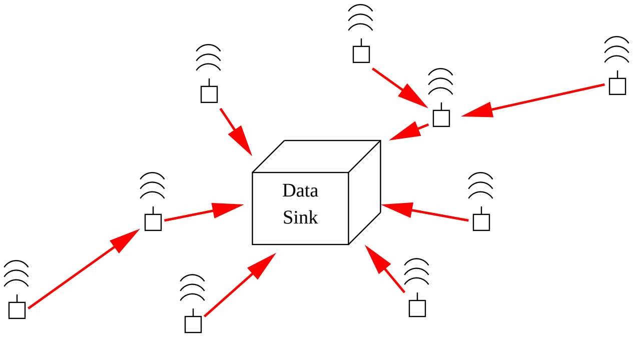

Fig. 1. An example sensor network with several nodes. As shown, either direct links or single relays to the sink are permitted.

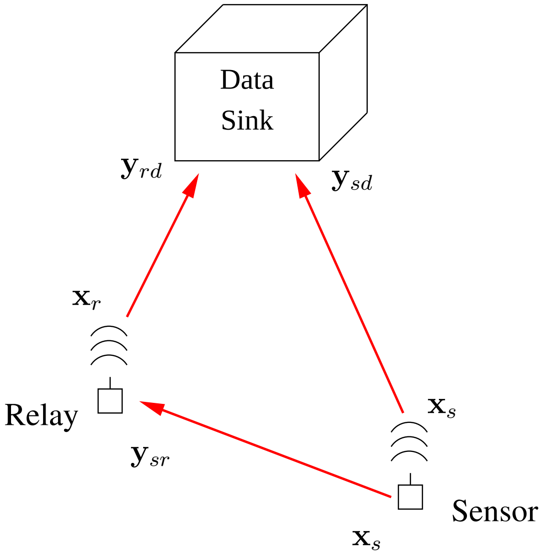

Fig. 2. The relaying scheme once a relay has been selected.

transmission. Instead, it simply makes hard decisions on the channel observations. Let the relay's estimate of the transmitted codeword be $\hat{\mathbf{x}}_{sr} = \sigma(\mathbf{y}_{sr})$, where $\sigma(\cdot)$ is a hard decision function yielding a length-m binary sequence,. The relay may only wish to retransmit a fraction of $\hat{\mathbf{x}}_{sr}$. Let $\mathbf{w}_r \in \{0,1\}^q$, $q \leq m$, represent the symbols taken from $\hat{\mathbf{x}}_{sr}$ to be retransmitted by the relay. Analogously to the source, let $\mathbf{x}_r$ represent the transmitted n-bit codeword, i.e., the relay uses a (q,n) code. In this phase, the sink observes

$$\mathbf{y}_{rd} = A_{sd}\rho(\mathbf{x}_r) + \mathbf{n}_{rd}, \tag{3}$$

where A_{sd} is the amplitude of the relay-to-sink link, and $\mathbf{n}_{rd}$ is unit-variance white Gaussian noise. The sink is assumed to know the channel state from the source, the relay and between the source and relay.

III. Repeat-Accumulate Codes

A. Code structure

Regular RA codes have a structure similar to that of an LDPC code. To encode an information sequence $\mathbf{w} \in \{0,1\}^k$, we start by forming the sequence $\mathbf{a} \in \{0,1\}^{kr}$, which contains the elements of $\mathbf{w}$, repeated r times each. The sequence $\mathbf{a}$ is then permuted by some random permutation $\pi(\cdot)$, known to both the encoder and decoder, to form the new sequence $\mathbf{b} = \pi(\mathbf{a})$. Finally, the codeword $\mathbf{x}$ is formed by taking the cyclic mod-2 accumulated sum of all symbols in $\mathbf{b}$:

$$x_i = \begin{cases} b_i, & i = 1, \\ x_{i-1} \oplus b_i, & 1 < i \leq kr. \end{cases} \tag{4}$$

A systematic RA code can be easily generated. In (4), $\mathbf{x}$ is of length kr, the same as $\mathbf{b}$; instead, suppose $\mathbf{x}$ is of length $(k+1)r$, and

$$x_i = \begin{cases} w_i, & 1 \leq i \leq r, \\ b_{i-r}, & i = r+1, \\ x_{i-r-i} \oplus b_{i-r}, & r+1 < i \leq (k+1)r, \end{cases} \tag{5}$$

so the systematic bits are at the beginning of the codeword. A factor graph representing a regular RA code is shown in Fig. 3.

The codeword of a punctured systematic RA code is formed by transmitting all the information bits, and a randomly selected fraction of the remaining bits. Let $\hat{\mathbf{x}}$ be a systematic RA codeword satisfying (5), and let $\mathcal{P}$ be the index set of elements from $\hat{\mathbf{x}}$ that are in the punctured codeword (i.e., for some integer j, if $j \in \mathcal{P}$, then $\hat{x}_j$ is in the punctured codeword). Since all systematic bits are transmitted, all the integers up to r are in $\mathcal{P}$, but integers from $r+1$ to $(k+1)r$ are added randomly. Thus, letting P_i represent the ith smallest element of $\mathcal{P}$, the transmitted codeword $\mathbf{x}$ has elements given by

$$x_i = \hat{x}_{P_i}.$$

The factor graph for this code is the same as for the non-punctured code, but the channel messages for punctured symbols are set to zero.

If all the symbols are transmitted, the rate of the non-systematic RA code is $1/r$, and the rate of the systematic code is $1/(1+r)$. The rate of the RA code can be crudely controlled by varying r, but it is easily seen that the maximum practical rate is $1/2$ for a non-systematic code, or $1/3$ for a systematic code. Furthermore, it is known that codes with $r = 2$ have poor performance, although the performance can approach the Shannon limit when $r \geq 3$. Alternatively, the rate can be varied by using a systematic code, fixing r, and randomly puncturing (i.e., not transmitting) some of the code symbols. This method retains good performance over a wide spectrum of rates, and has the advantage of being incremental. That is, to decrease the rate, the code can remain the same while more punctured symbols are transmitted.

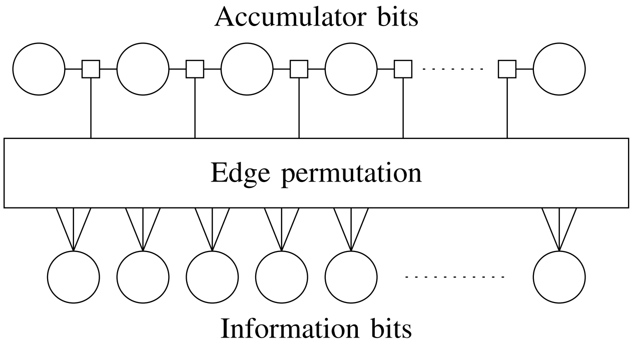

Fig. 3. Factor graph representation of a Repeat-Accumulate code. The value of r is equal to the degree of the information nodes, so in this case, $r = 3$.

Finally, decoding is accomplished using the sum-product algorithm over the factor graph in Fig. 3. From (5), the symbols b_i, x_i, and x_{i-1} satisfy

$$b_i \oplus x_i \oplus x_{i-1} = 0.$$

In other words, the three symbols satisfy a parity check equation. Since the code is systematic, b_i, x_i, and x_{i-1} are all symbols included in the transmitted codeword, and from Fig. 3, all codeword symbols are represented as nodes in the factor graph. Thus, all the constraints in this code are simple parity checks, and so the message calculations are all the same as in LDPC codes.

B. RA codes in relaying and cooperation

We now discuss the impact of relaying on the RA coding scheme. As mentioned previously, the relay does not decode, but directly incorporates noisy observations into its information string.

In our proposed scheme, the source sensor encodes its symbols in a systematic punctured RA code and transmits them. The relay observes $\mathbf{y}_{sr}$ from the sensor, then makes a hard decision on these symbols to form $\hat{\mathbf{x}}_{sr} = \sigma(\mathbf{y}_{sr})$, and forms a string $\mathbf{w}_r$ by selecting either some or all of the hard decisions to relay. The codeword formed by the relay, $\mathbf{x}_r$, is formed in the same manner as by the source.

We consider two possibilities in decoding this code. We first consider a *serial* decoder, in which the sink first decodes the relay's transmission, and uses the result to help decode the source's transmission. Assuming that $\mathbf{y}_{rd}$ is successfully decoded, the sink has knowledge of $\mathbf{w}_r$, which is equivalent to observing some (or all) of $\mathbf{x}_s$ through an independent binary symmetric channel. Suppose the ith sensor symbol, $x_{s,i}$, corresponds to the jth symbol in $\mathbf{w}_r$, $w_{r,j}$. Let p_{sr} represent the crossover probability of the equivalent source-to-relay binary symmetric channel, where

$$p_{sr} = \frac{1}{2}\,\mathrm{erfc}\left(\frac{A_{sr}}{\sqrt{2}}\right).$$

Using the same notation as in the previous section, the channel message for $x_{s,i}$ is then given by

$$c_i = \begin{cases} A_{sd} y_{sd,i} + \log\frac{1-p_{sr}}{p_{sr}}\rho\left(w_{r,j}\right), & i \in \mathcal{P}, \\ 0, & i \notin \mathcal{P}. \end{cases} \tag{6}$$

and the additive term on the right is deleted if $y_{sd,i}$ is not relayed. Notice that the additive term is the message for a pure binary symmetric channel. Decoding proceeds in exactly the same manner as in the non-relay case.

In practice, serial decoding can result in error propagation, as the decoding procedure fails if either the relay transmission or the sensor transmission result in a decoding failure. As an alternative, we propose *parallel* decoding, in which both the relay and sensor codes are decoded simultaneously. The sequences $\mathbf{x}_s$ and $\mathbf{w}_r$ are correlated random variables, and are therefore connected with a factor graph structure. The expression in (6) gives the message value where $w_{r,j}$ is perfectly known. In the sum-product algorithm, if $w_{r,j}$ is uncertain, the additive term on the right would be marginalized over the two binary values, which, for $i \in \mathcal{P}$, would result in

$$\begin{aligned} c_i &= A_{sd} y_{sd,i} \\ &\quad + \log\frac{1-p_{sr}}{p_{sr}}\left(\Pr(w_{r,j}=+1) - \Pr(w_{r,j}=-1)\right). \end{aligned} \tag{7}$$

and $c_i = 0$ for $i \notin \mathcal{P}$. The probabilities in this expression are *a posteriori* probabilities provided by the code attached to $w_{r,j}$. Thus, the two RA code factor graphs are connected with structures as depicted in Fig. 4. An analogous calculation is performed for the message being passed in the other direction.

C. A note on encoder complexity

Our primary motivation in proposing these codes is the low complexity with which they can be encoded. The computational tasks in encoding a punctured systematic RA code are repetition, permutation, accumulation, and puncturing. Repetition and accumulation are computationally simple, with linear complexity in the length of the code. There exist computationally simple algorithms for permutation (see, e.g., [16]), so long as a random number generator is available. Random puncturing can be accomplished with similar ease when a random number generator is available. Furthermore, a random number generator could be as simple as a pseudo-noise sequence generator with a known seed; certainly, algorithms exist to generate them with low complexity, or where storage is not an issue, a lookup table could be used. The overall complexity is therefore linear in the number of encoded symbols.

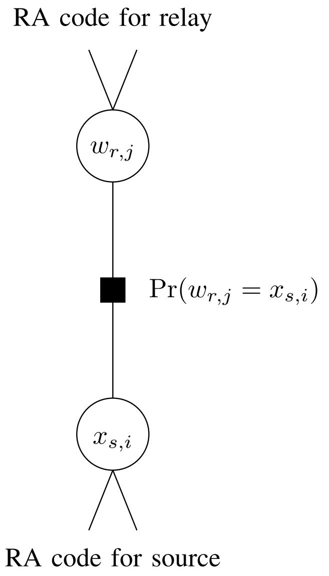

Fig. 4. Linkage between information symbols in the relay factor graph. These nodes connect the two RA code factor graphs.

IV. Density evolution

We analyze the proposed system using *density evolution* [17], 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 RA and other Turbo-like codes, it is easy to show that this assumption becomes asymptotically correct as the number of information symbols approaches infinity. Moreover, the performance at shorter block lengths is usually similar to the asymptotic performance (e.g., see the simulation results in [17]).

The message calculations performed at the parity check and variable nodes within the RA 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 [17] for the details.

V. Results

Our results are generated using density evolution, and under the assumption of serial decoding. In the final version of the paper, we plan to include simulation results using parallel decoding.

Our first result demonstrates the performance of punctured RA codes in a simple point-to-point setting, without relays. Puncturing changes the rate of the code, so using density evolution, the SNR threshold (i.e., the minimum SNR at which decoding is successful) is obtained for various punctured rates and repetition values r. ¿From Fig. 5, we see that the highest thresholds are obtained where $r = 3$. However, as we pointed out earlier, when $r = 3$, the minimum rate is $1/(1+r)$, or $1/4$, since this is the rate when every code symbol is transmitted, and none are punctured. If the channel is so weak that its SNR is below the threshold for rate $1/4$, it becomes necessary to use a higher value of r. Thus, for some maximum repetition value $r_{\max}$, we have the following encoding scheme:

- If the SNR is greater than the threshold for $r = 3$, use $r = 3$ with the appropriate puncturing;
- If the SNR is lower than the threshold for $r = 3$, but greater than the threshold for some integer r', where $r' < r_{\max}$, then encode with $r = r'$; and
- If the SNR is lower than the threshold for for $r = r_{\max}$, declare an outage.

The above encoding scheme can be simply modified for relaying, as follows.

- Using the above encoding scheme, find the minimum redundancy (in total channel uses per information bit) to transmit without relaying.
- For relaying, assume both the sensor and the relay use the above encoding scheme. For all potential relays, find the relay link with minimum overall redundancy, including transmissions from both the sensor and the relay. An outage occurs if the sensor is using $r = r_{\max}$ and decoding is unsuccessful.
- The system uses the link with the lowest overall redundancy, either with or without relaying, and an outage is declared if outages exist on every possible link.

In all the relaying results in this section, we assume that every symbol transmitted by the sensor is relayed to the sink. For this relaying scheme, outage probability is plotted as a function of SNR in Fig. 6, assuming all links have the same average SNR. This figure shows that the outage probability nearly achieves the benefit of full diversity for a given number of available relays, even at relatively low SNR.

For a deployed sensor network, let R_n represent the average redundancy for links that are not in outage, and let p_{out} represent the probability of outage. We assume that the sensors are stationary, so the links are static, and a node in outage remains in outage forever. To model wireless fading, we assume that the link amplitudes A_{sr}, A_{rd}, and A_{sd} are Rayleigh distributed random variables. We also assume that the network is usually idle, so that the desired relay is always available when requested. Thus, for all the possible channel uses available to the network, the average number of information bits returned to the source is given by

$$u := \frac{1}{R_n} p_{\mathrm{out}}.$$

We use u as a simple utility metric to measure how efficiently the network transports data to the sink. In Fig. 7, we see the value of the utility metric for various values of average SNR and numbers of available relays, where all links have the same average SNR. The behavior in the figure is somewhat surprising at higher SNR, where the utility decreases as relays are added. This suggests that the network is expending too much effort to return information bits where both the direct link and relay link are weak.

We also provide results for a more realistic scenario where the nodes are spatially distributed. In this case, node locations were uniformly distributed on a square of 4 m × 4 m, where each node's location was independently distributed, and where the sink was located at the center of the square. All nodes were assumed to have the same transmit power, with appropriate path losses amongst the sensor, relay, and sink; for this example, we used a path loss exponent of 2 (i.e., free space propagation). For a given total number of nodes in the system, in Fig. 8 we plot the outage probability versus the SNR measured when the distance from transmitter to receiver is 1 m (called the SNR at 1 m). For given SNR at 1 m, we also plot the utility versus the total number of nodes in the system in Fig. 9. These figures show that significant gains can be expected in a realistic ad-hoc network.

VI. Conclusion

In this work, we have introduced systematic punctured RA codes for relaying and co-operative communication in wireless networks. These codes are relatively simple to encode, and provide good relaying performance and diversity over a wide range of SNR for a very simple relaying framework, which is within the capabilities of contemporary devices. Our results also show that relaying strategies need to be further studied in order to optimize the utility of such a system.

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.

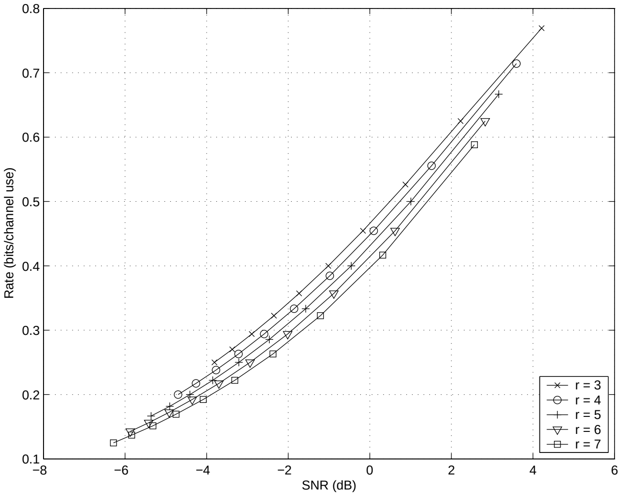

Fig. 5. Thresholds for punctured, systematic RA codes, without relaying

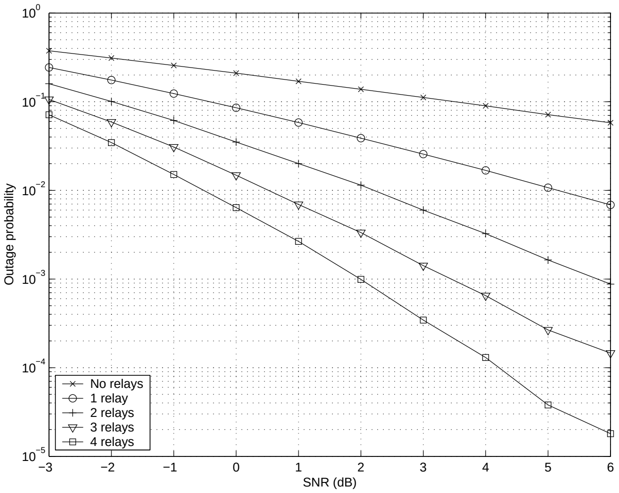

Fig. 6. Outage probability versus SNR for various numbers of relays.

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

[10] M. A. Shokrollahi, "Raptor codes," in *Proc. IEEE International Symposium on Information Theory*, 2004.

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

[12] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for "turbo-like" codes," in *Proc. 36th Annual Allerton Conference on Communications, Control, and Computing*, pp. 201–210, 1998.

[13] 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.

[14] J. Laneman and G. W. Wornell, "Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks," in *Proceedings Global Telecomm. Conf.*, vol. 1, pp. 77–81, GLOBECOM, 2002.

[15] 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.

[16] D. E. Knuth, *The Art of Computer Programming, Volume 2*. Addison-Wesley, 1997.

[17] 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.

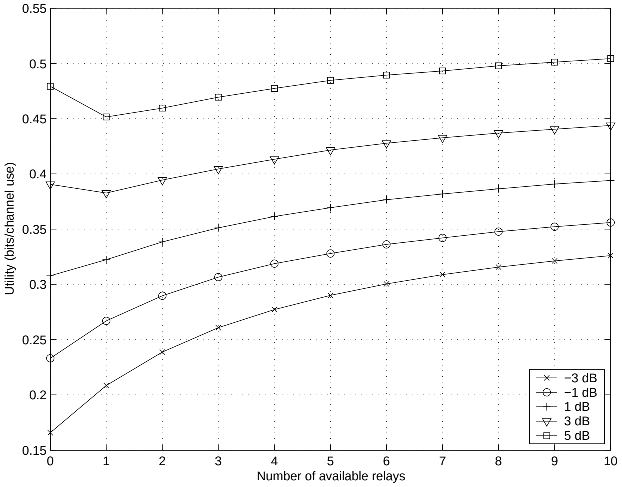

Fig. 7. Utility versus number of relays for various SNRs.

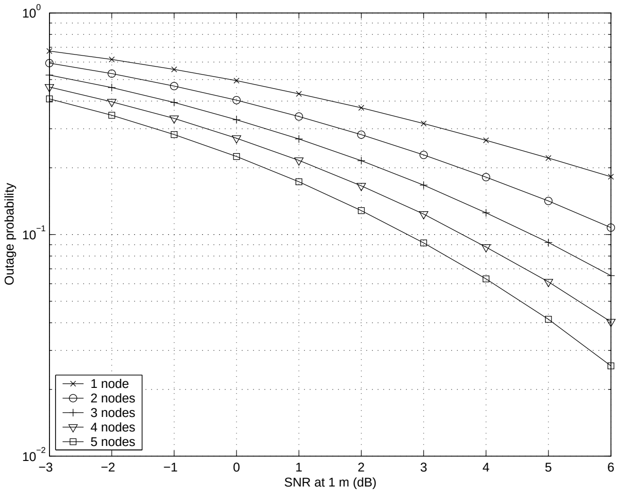

Fig. 8. Outage probability versus SNR for various numbers of system nodes for spatially distributed nodes.

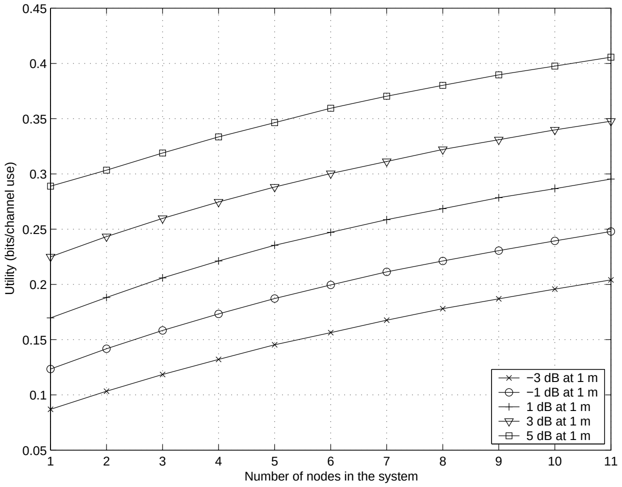

Fig. 9. Utility versus number of system nodes for various SNRs for spatially distributed nodes.