Optimal Progressive Error Recovery for Wireless Sensor Networks using Irregular LDPC Codes

Saad B. Qaisar and Hayder Radha,, *Senior Member, IEEE*

***Abstract*—We study the problem of providing reliable data transmission in energy constrained Wireless Sensor Networks (WSNs). Low rate channel coding can increase reliability and eliminate the need of costly retransmissions for sensor data. However, low rate channel coding on end to end basis puts a considerable burden in terms of transmit energy on resource constrained sensor nodes. We propose a scheme that progressively provides error resilience as information reaches the final destination. Precisely, we present a novel framework for processing within the network (in-network) using irregular low density parity check (LDPC) codes for channel coding, in which nodes progressively decode the information at intermediate nodes. We not only present the in-network processing setup, but also an Optimal Progressive Error Recovery Algorithm (OPERA) that optimally maps the decoding iterations to attain maximum throughput at the destination node. We use density evolution algorithm for belief propagation decoding of LDPC codes to cast the optimization problem and use dynamic programming to reach the solution. We compare the performance of our scheme with end to end channel coding and establish the efficiency of proposed solution for a given energy budget. Finally, we give a comparison between our scheme and random iteration assignment for decoding at intermediate nodes, and show that our scheme performs considerably better.**

***Index Terms*—Density evolution, LDPC, partial processing, sensor networks**

I. INTRODUCTION

WIRELESS Sensor Networks require resilience against channel induced errors. Retransmission based schemes prove too costly for energy constrained sensor devices. This has prompted various research studies (see [8][9][10]). Most of these works are restricted to *end to end* error protection catering for worst case scenarios over the end to end path between source and destination. *Full processing* and *forwarding* represent two extremes of processing the information within the network (in-network processing) before it arrives at the final destination. Complete encoding and decoding of information bits at each intermediate node corresponds to *Full processing* whereas in *forwarding*, the information is forwarded to next node without any further processing. *Full processing*, though optimal in throughput sense, puts a considerable computational burden on energy constrained sensor devices. On the contrary, *forwarding* does not provide any additional throughput benefits over the *end to end* error recovery mechanism already in place. As an alternative, *Partial processing* of information bits at intermediate nodes [4][5] can provide considerable throughput benefits without overburdening the sensor nodes. In fact, it can help increase the overall life expectancy of the sensor network for a given information content delivery.

In this paper, we propose a novel scheme that optimally enhances the throughput for wireless sensor networks using finite complexity processing at intermediate nodes. Specifically, we formulate a setup in which intermediate nodes over the end to end path between source and destination, partially decode the incoming packets prior to forwarding them to the next node. We propose an Optimal Progressive Error Recovery Algorithm that optimally assigns decoding iterations to the intermediate nodes. We use irregular low density parity check codes to show the efficiency of our scheme. We use density evolution algorithm for message passing decoding (discussed later) of LDPC codes and the relevant LDPC theorems to formulate the optimization problem, and use dynamic programming for arriving at the solution. We show that our scheme performs significantly better than end to end channel coding for a given energy budget.

Resource limitation of sensor nodes have restricted studies on channel coding in sensor networks. Few works exist, and they as well are confined to end to end error protection. Sankarasubramaniam et al [10] use BCH codes for Forward Error Correction (FEC) in sensor networks, whereas, Shih et al [9] base their work on convolutional codes. Mina et al [8] propose joint source and channel coding in sensor networks using LDPC codes for end to end error protection catering for worst case channel conditions. These schemes put an additional burden in terms of transmit energy on the nodes experiencing better link quality down the network than the nodes transmitting on noisy links. Transmitting an additional single bit of data is much costlier than instructions used for

Manuscript received January 2, 2007.

Saad B. Qaisar is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (phone: 517-355-3769; fax: 517-353-1980; e-mail: qaisarsa@ egr.msu.edu).

Hayder Radha is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: radha@ egr. msu.edu).

computation in the individual sensor nodes [11][12]. In a recent study, a single bit transmission is shown equivalent to 2000 computational instructions executed in the sensor node [11]. This makes a strong case in favour of in-network processing for increased reliability in data transmission to the collector node. We achieve these processing benefits with proposed OPERA scheme using irregular low density parity check codes.

We start with brief discussion in section II on LDPC codes, Density Evolution, and relevant LDPC theorems. We formulate the constrained optimization problem for OPERA in section III. Section IV provides the dynamic programming solution to the optimization problem, whereas section V presents the simulation setup, results and related discussion. We summarize the key conclusions in section VI.

II. LDPC Codes, Density Evolution and Relevant Theorems

Low Density Parity Check codes are systematic block codes renowned for their near capacity performance using low complexity decoders. Gallager provided an algorithm for near optimal decoding of LDPC codes [6]. The algorithm comes under different names including Belief Propagation Algorithm (BP), Sum Product Algorithm (SPA) or in general, Message Passing Algorithm (MPA). We use Belief Propagation algorithm for LDPC decoding. Specifically, we use log domain belief propagation decoding which has benefits in terms of computational complexity.

We consider a low density parity check code specified by edge degree distributions $\sum_i \lambda_i x^{i-1}$ and $\sum_i \rho_i x^{i-1}$, where, λ_i (ρ_i) denotes the fraction of all edges connected to degree-i variable (check) node [6].

A. *Decoding Thresholds and Density Evolution*

Urbanke et al. {[1][2][3][20]} provide upper bounds on the fraction of errors that a belief propagation decoder can correct based upon the evolution of message densities. We assume a memoryless channel with a symmetry property i.e. $\mathrm{p}(\mathrm{y} \mid \mathrm{x}=1)=\mathrm{p}(\mathrm{-y} \mid \mathrm{x}=-1)$. Further, we assume that the LDPC decoder satisfies sign symmetries for messages passed over the bipartite graph [2]. Under these assumptions, there exists a threshold ε^* such that for any $\varepsilon < \varepsilon^*$ there exists a block length $n(\varepsilon)$ such that almost every code in LDPC ensemble with $n > n(\varepsilon)$, has probability of error smaller than a non negative constant, assuming that transmissions take place over a channel with parameter ε. corresponding author only.

Density evolution is a dynamic system in a space of probability distributions representing the progress of iterative decoders. Empirically, LDPC performance is excellent – LDPC codes have been shown to be the best known block codes [2]. Density Evolution provides an analytical tool to evaluate the LDPC code performance. The messages passed in the factor graph in LDPC message passing decoding are random variables. Based upon the probability density function of messages passed at the lth iteration, the density of messages passed at iteration $l+1$ can be found under certain symmetry assumptions.

The performance of Belief Propagation Algorithm (BPA) for decoding of LDPC codes can be analyzed by determining the fraction of incorrect messages passed at the lth decoding iteration with the assumption that the underlying graph does not have cycles of length $2l$ or less. Cycle free assumption provides independence between the incoming messages. Using the following notation:

P_0: Distribution of the received messages from the symmetric channel

P_l: Distribution of messages sent from the variable nodes to the check nodes in the lth iteration

R_l: Distribution of messages sent from the check nodes to the variable nodes in the lth iteration

$\otimes$: Convolution operator

Γ^{-1}: Change of variable operation associated with the function $\psi(x)=\log\left((e^x+1)/(e^x-1)\right)$

Assuming linearity and all 1s code transmitted over a memoryless BIAWGN channel, we have distribution of messages passed from variable node to check nodes as:

$$\mathrm{R}_l = \Gamma^{-1}\rho(\Gamma(\mathrm{P}_0 \otimes \lambda(\mathrm{R}_{l-1}))) \qquad (1)$$

The distribution of messages passed from a variable node to a check node in the lth iteration is given by the convolution sum

$$\mathrm{P}_l = \mathrm{P}_0 \otimes \lambda(\mathrm{R}_{(l-1)}) \qquad (2)$$

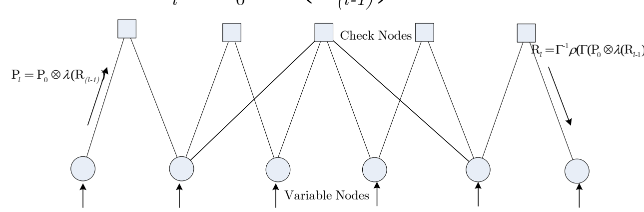

Fig. 1. Density Evolution for message passing algorithm on a Tanner graph

Then the conditional error probability conditioned on transmission of all ones codeword is given as :

$$\mathrm{P}_e^{\,l} = \int_{-\infty}^{0} \mathrm{P}_l(\mathrm{x})\mathrm{dx} \qquad (3)$$

Figure 1 shows the evolution of message densities over a Tanner graph [19] for LDPC codes. (See [1][2][3] for detailed discussion on density evolution.)

B. *LDPC Theorems*

Urbanke et al.[1][2] give monotonicity and concentration theorems for LDPC codes.

Monotonicity Theorem

Let P_l be the message distribution at the lth decoding iteration and let g be a consistent distribution on $\overline{\mathbb{R}}$. Then

$$P_e^l(P_l \otimes g)$$

is a non increasing function of l.

Infact, the probability of error decreases exponentially fast within few decoding rounds for any $\varepsilon < \varepsilon^*$ ([1] for proof and further discussion on consistency).Therefore, motivated by monotonicity theorem, for a cascade of channels, a partial processing framework can be formulated in which intermediate nodes undertake some fixed number of decoding rounds prior to forwarding the information if channel error probability $\varepsilon < \varepsilon^*$. Otherwise, nodes just do forwarding.

Concentration Theorem

Let $P_e^n(l)$ be the expected fraction of incorrect messages which are passed in the lth iteration of LDPC decoding, where expectation is over all instances of the code, the choice of the message and realization of the noise. For any $\delta > 0$, the probability that the actual fraction of incorrect messages which are passed in the lth iteration for any particular such instance lies outside the range $(P_e^n(l) - \delta, P_e^n(l) + \delta)$ converges to zero exponentially fast in n ([1] for proof).

The concentration theorem asserts that (almost) all LDPC codes behave alike and so the determination of the average behavior of the ensemble suffices to characterize the individual behavior of (almost) all codes [1]. Any code from the LDPC ensemble would have performance approaching exponentially fast in n to the mean of the ensemble. Thus, for a sufficiently large n, any LDPC code from the ensemble would have little deviation in performance from the mean of ensemble.

III. Optimization Problem

We consider a line network with N nodes in cascade. Each node is capable of decoding the received messages. In order to obtain the channel error probability over the cascade, let $\{n_t; t \in T\}$ be the subset of nodes simultaneously transmitting over a certain subchannel, with transmit power P_t for each node. Then the SINR at node n_j for a transmission from node n_i, $i \in T$ can be calculated as in [13]:

$$SINR = \frac{\frac{P_i}{d(n_i, n_j)^\alpha}}{P_A + \sum_{\substack{t \in T \\ t \neq i}} \frac{P_t}{d(n_t, n_j)^\alpha}} \tag{4}$$

Where $d(n_i, n_j)$ is the separation between n_i and n_j, P_A is the ambient noise power and $\alpha > 2$.

Then the channel error probability ε is:

$$\varepsilon(n_i) = Q(\sqrt{2(SINR(n_i))}) \tag{5}$$

Where, we have: $Q(a) = \frac{1}{\sqrt{2\pi}} \int_a^\infty e^{-\frac{u}{2}} du$

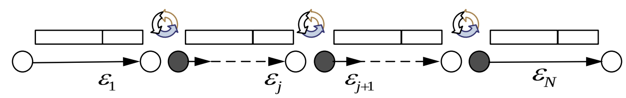

Fig. 2. A Line network with In-Network processing.

Node n_0 generates k message bits which are LDPC encoded using rate R LDPC code prior to transmission to n_1. On arrival at n_1, l_1 decoding rounds are performed on each message bit. Similarly, l_2 decoding iterations are carried out at n_2 and so on, till the message arrives at the final destination. At destination, decoding takes place using l^* decoding rounds with $l^* >> l_i$.

Then the distortion at the destination node is given as:

$$D = P_e(P_e(P_e(P_e(P_e(\varepsilon_1, l_1) * \varepsilon_2, l_2) * \ldots \varepsilon_j, l_j) * \ldots \varepsilon_{N-1}, l_{N-1}) * \varepsilon_N, l^*) \tag{6}$$

where $\varepsilon_1 * \varepsilon_2 = \varepsilon_2(1 - \varepsilon_1) + \varepsilon_1(1 - \varepsilon_2)$

For

$$\Im(\bar{l}) = \sum_{j=1}^{N-1} l_j \tag{7}$$

and $\Im(\bar{l}) \le \Im(\bar{l})_{budget}$, a simplistic approach can be to randomly assign iterations to the nodes remaining in budget constraints. That may not necessarily be an optimal assignment in terms of throughput at the final destination. Thus, within the budget constraints, we intend to find an iteration assignment vector $\bar{l}$ in such a way that the net throughput is maximized at the destination node.Thus we have the optimization problem for in-network error recovery as:

Minimize:

$$D = P_e(P_e(P_e(P_e(\varepsilon_1, l_1) * \varepsilon_2, l_2) * \ldots \varepsilon_j, l_j) * \ldots \varepsilon_{N-1}, l_{N-1}) * \varepsilon_N$$

Such that:

$$\Im(\bar{l}) \le \Im(\bar{l})_{budget}$$

IV. Dynamic Programming Algorithm

For the given optimization problem, we are interested in finding an optimal tuple $\left[D, \Im(\bar{l}), \bar{\varepsilon}\right]$ such that

$\Im(\bar{l}) \leq \Im(\bar{l})_{\text{budget}}$. The problem can be seen as minimizing a distortion measure D subject to rate constraints in rate distortion theory for optimal quantizer design [14]. The computational complexity to exhaustively look for optimal tuples $\left[\mathrm{D}, \Im(\bar{l}), \bar{\varepsilon}\right]$ is high as we may end up looking for all possible iteration assignments within the budget constraint. We adopt a dynamic programming algorithm to solve the problem. The algorithm progressively assigns iterations in a greedy fashion with an aim to maximize the throughput at the destination node, or, alternatively, minimize distortion.

Initially, no iterations are assigned to the intermediate nodes giving maximum D, and thus, the tuple $\left[\mathrm{D}^0, \Im(\bar{l})^0, \bar{\varepsilon}\right]$. The algorithm then adds a single iteration, one by one in a greedy manner to each node, and chooses the node where minimum distortion is achieved. This provides the next operating point $\left[\mathrm{D}^1, \Im(\bar{l})^1, \bar{\varepsilon}\right]$. This process is repeated till the minimization constraint $\Im(\bar{l}) \leq \Im(\bar{l})_{\text{budget}}$ and all the iterations are not assigned.

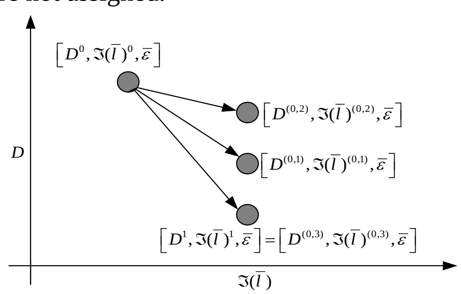

Fig. 3. Selection of Next Optimal Point $\left[\mathrm{D}^1, \Im(\bar{l})^1, \bar{\varepsilon}\right]$ for Dynamic Programming algorithm

Therefore, the algorithm maximizes the gradient between the optimal point achieved in the prior run D^- and the new points D^+ obtained in the current run for all intermediate nodes on the path. This corresponds to maximizing the gradient between the two points.

Mathematically, we have:

$$\bar{l} = \underset{\text{j=1:N-1}}{\operatorname{argmax}}\left[\frac{\mathrm{D}^- - \mathrm{D}^+}{\Im(\bar{l}^-) - \Im(\bar{l}_j^+)}\right] \tag{8}$$

where, we take $\Im(\bar{l}^-) - \Im(\bar{l}^+) = 1$.

For extending the algorithm to a multihop multi path network with all nodes transmitting to a central base station, we optimally assign iterations to individual paths from source to destination to obtain assignment vector $\bar{\mathrm{l}}$ and distortion D. The distortion is then averaged over all the paths, from sources to base station to obtain cumulative bit error rate.

V. Simulation setup, Results and Discussion

For the proposed OPERA, we consider a density evolution optimized Progressive Edge Growth [15] based irregular LDPC code with $\lambda(x) = 0.207x^6 + 0.271x^2 + 0.522x^1$. We take $k = 1024$ bits for our simulations. We assume that if $\varepsilon < \varepsilon_{\min}$, on average, only one iteration is sufficient to completely decode the LDPC code, which is usually the case for the given choice of rates and codes here. We set $\varepsilon_{\min} = 0.01$ and $\varepsilon^* = 0.11$ [20]. For a single path, we fix $\Gamma(\bar{l})_{budget} = 25$ iterations. We take per bit transmit power $P_T = 1mW$, per second, for each node.

The total energy spent by sensor nodes per packet delivery to the destination node is the function of computational energy and the transmission energy. We have $\mathrm{E}_{\text{total}}$

$$\mathrm{E}_{\text{total}} = \mathrm{E}_{\text{Trans.}} + \mathrm{E}_{\text{Comp.}} \tag{9}$$

where, $\mathrm{E}_{\text{Comp.}}$ is the computational energy spent per packet for partial processing within the network and $\mathrm{E}_{\text{Trans.}}$ is the per packet transmission energy for end to end delivery.

Fossorier et al. [16] in their work on low complexity decoding of LDPC codes enlist the mathematical operations required per iteration for belief propagation decoding. Assuming each node a sensor mote equipped with Atmel Atmega128L processor, 2000:1 ratio between per bit transmit energy and computation energy spent per instruction [11] and average number of ones per column in a parity check matrix of a code from ensemble of $(\mathrm{n}, \lambda, \rho)$ LDPC codes as $\left(\int_0^1 \lambda(x)dx\right)^{-1}$ [18], we provide a performance comparison between proposed OPERA and end to end channel coding.

A. End to End vs OPERA for a Line Network

We consider 100 realizations of a line network with $N = 5$ and variation in separation d between the individual nodes with average end to end equivalent error probability $\varepsilon_{end_{eq}} = 0.0919$. For end to end channel coding, we take $R_{end_{\max}} = 0.731$ and $R_{end_{\min}} = 0.410$. For OPERA, we consider multiple rates $R_{OPERA} = 0.602, 0.569$.

We take $l^* = 200$. The results are averaged over 1000 runs. Figure 4 presents the comparison between both end to end and OPERA. With the expenditure of some computational energy in the network, significant enhancements in throughput is achieved as compared to end to end channel coding.

B. OPERA for Multiple Flows

We consider a wireless sensor network over a $10m$ x$10m$ square grid with N sensor nodes spread according to a random distribution. We place the base station at coordinates (5, 5) and limit the transmission range of individual nodes to a maximum transmission range $r = 2m$. As most of the WSN routing algorithms are different forms of

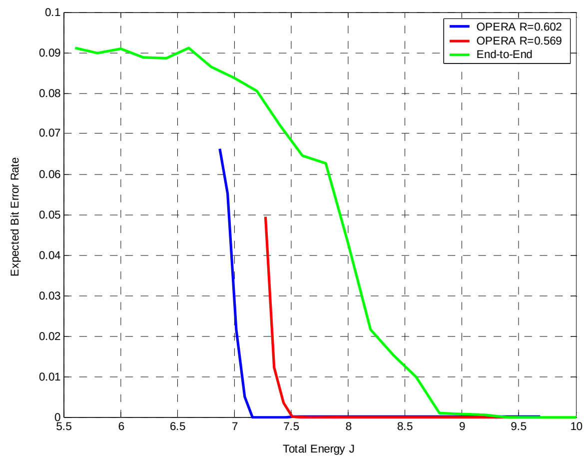

Fig. 4. End to End vs OPERA for a line network

shortest path routing, we assume that at any given time, the routes from individual nodes to the base station form a tree rooted at the base station [17].

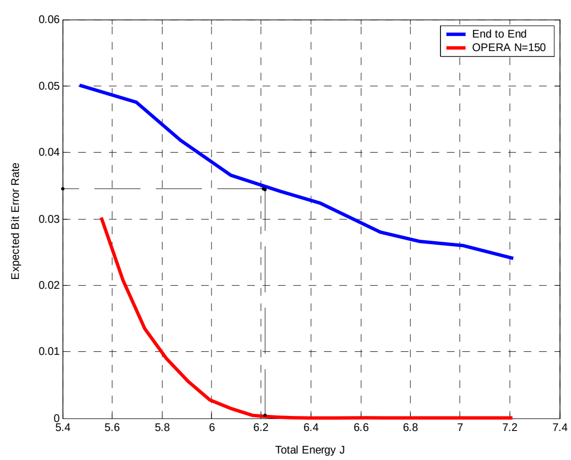

Fig. 5. End to End vs OPERA for $l^* = 200$ and $N = 150$

Figure 5 shows the performance of OPERA as compared to end to end channel coding. We use $R_{end_{min}} = 0.38$, $R_{end_{max}} = 0.50$ and $R_{OPERA} = 0.50$

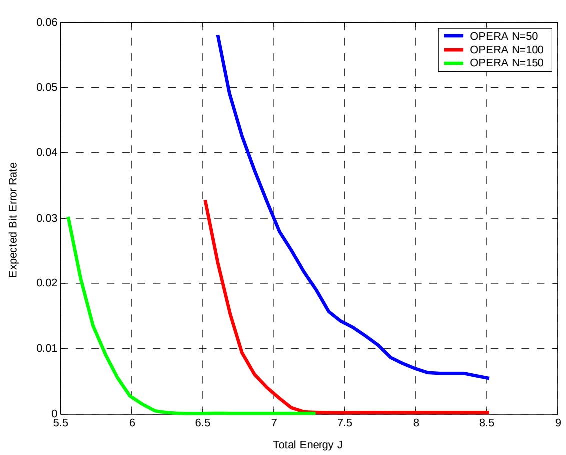

Fig. 6. OPERA for $N = 50, 100$ and 150

For a given energy budget, OPERA performs significantly better than end to end channel coding. For example, for an energy budget of 6.22J, OPERA provides absolute reliability as compared to end to end case which gives an expected bit error rate of 0.035 approx. Figure 6 shows the performance of OPERA with variation in number of nodes. As the number of nodes increases, lesser amount of in-network processing is required to achieve improvement in reliability. The above plots also show that it is feasible to trade off energy usage with reliability with variation in iteration assignment at intermediate nodes. At one extreme, we can get maximum reliability with maximum processing at intermediate nodes, whereas, other end represents the performance achieved with least energy spent for in-network processing. This gives us the ability to fine tune the operation of sensor network, with more iterations assigned if an important event is observed as compared to more passive state where we may prefer to save energy, despite getting noisy readings.

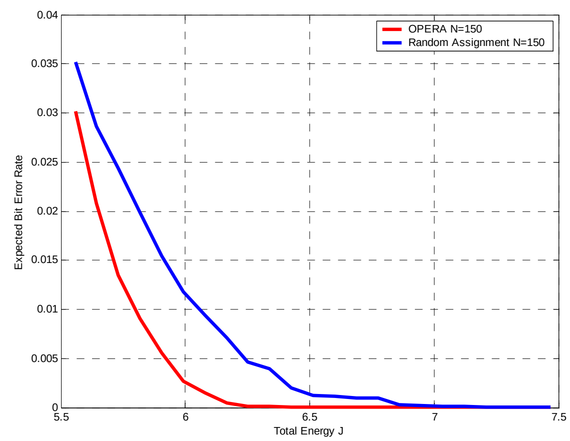

Fig. 7. OPERA vs Random Assignment for $N = 150$, $l^* = 200$

Figure 7 shows considerable performance benefits achieved using OPERA as compared to Random iteration assignment.

The above results further show that as the number of nodes increase the amount of in-network processing required to achieve improvement in reliability reduces due to decrease in number of noisy links. This observation has important implications about adapting the functional usage of a sensor network through its life-time. With time as sensors die the density of sensors reduces. At such point if the functional demand is not reduced the amount of energy that will have to be spent is infact increased. This increase in energy spending may set-off a chain reaction eventually leading to the death of a *network*. Thus, as the density reduces, it may be essential to adapt the functional demands from a network. The Energy-Distortion curves obtained by our analysis can provide important guidelines on how these demands should be reduced as sensors start dieing.

VI. Conclusion

We presented a framework that progressively recovers errors over the network using irregular low density parity check codes. It provides significant performance enhancement at the cost of some computational energy. We give bounds on the performance of the algorithm for total energy spent within the network for partial decoding of ensembles of irregular LDPC codes and transmissions between the individual nodes. We give comparison between the proposed OPERA scheme and end to end channel coding and random iteration assignments. OPERA outperforms both these schemes by considerable margins. Additionally, we have based our work on belief propagation decoding of LDPC codes. Further reductions in computational energy can be achieved using optimizations of belief propagation such as Universally Most Powerful Belief Propagation algorithm (UMP-BP) studied by Fossorier et al.[16] at the cost of some decoding performance.

Acknowledgment

Shirish Karande and Kiran Misra for their valuable insights in the initial phase of the problem.

References

[1] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2000.

[2] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of provably good low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[3] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes, " *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb. 2001.

[4] D. Tuninetti, C. Fragouli, "On the Throughput Improvement Due to Limited Complexity Processing at Relay Nodes", in the Proceedings of *International Symposium on Information Theory*, 2005.

[5] D. Tuninetti, C. Fragouli, "Processing Along the Way: Forwarding vs. Coding", *ISITA 2005*, Parma (Italy) Oct. 2004.

[6] Robert G. Gallager , "Low Density Parity Check Codes", *PhD Thesis, MIT*, 1963.

[7] Xiao-Yu Hu, Dieter M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs", *IEEE Trans. on Information Theory*, Vol. 51, January 2005.

[8] M. Sartipi, F. Fekri, "Source and Channel Coding in Wireless Sensor Networks using LDPC Codes,'' *ICSACN*, pp. 309-316, October 2004.

[9] E. shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *ACM SIGMOBILE*, July 2001.

[10] Y. Sankarasubramaniam, I. F. Akyildiz, and S. W. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks," *IWSN*, 2003.

[11] Ramesh Govindan, IEEE COMSOC Wireless Sensor Networks Tutorial, 2006, http://www.comsoc.org/freetutorials/nsc/

[12] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein and Wei Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks", *5th Annual Symposium on Operating Systems Design and Implementation (OSDI)*, December 2002.

[13] P. Gupta and PR. Kumar, The Capacity of Wireless Networks, *IEEE Transactions on Information Theory*, 2000.

[14] G. M. Schuster, and A. K. Katsaggelos, "Rate-Distortion Based Video Compression, Optimal Video Frame Compression and Object Boundary Encoding," *Kluwer Academic Publishers*, 1997.

[15] Xiao-Yu Hu, Dieter M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs", *IEEE Trans. on Information Theory*, Vol. 51, January 2005.

[16] Marc P. C. Fossorier, Miodrag Mihaljevi´c, and Hideki Imai, "Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation", *IEEE Transactions on Communications*, Vol. 47, No. 5, May 1999.

[17] M. U. Ilyas and Hayder Radha, "Increasing Network Lifetime Of An IEEE 802.15.4 Wireless Sensor Network By Energy Efficient Routing", *Proceedings of IEEE International Conference on Communications (ICC)*, Istanbul, 2006.

[18] Igal Sason and R. Urbanke, "Parity-Check Density Versus Performance of Binary Linear Block Codes Over Memoryless Symmetric Channels", *IEEE Transactions on Information Theory*, Vol. 49, No. 7, July 2003.

[19] R. Michael Tanner, A Resursive Approach to Low Complexity Codes, *IEEE Transactions on Information Theory*, Vol. 27, No. 5, September, 1981.

[20] L. Bazzi, T. Richardson, and R. Urbanke, "Exact thresholds and optimal codes for the binary symmetric channel and Gallager's decoding Algorithm A," *IEEE Trans. Inf.Theory*, vol. 50, no. 9, pp. 2010–2021, Sep. 2004.