Progressive Coding and Iterative Source-Channel Decoding in Wireless Data Gathering Networks

Congduan Li, Paul G. Flikkema and Sheryl L. Howard
EECS Department
Northern Arizona University
Flagstaff, AZ 86001

***Abstract*— Wireless data gathering networks are often tasked to gather correlated data under severe energy constraints. The use of simple channel codes with source-channel decoding can potentially provide good performance with low energy consumption. Here we consider progressive coding in multi-hop networks, where an intermediate node decodes its received noisy codewords. The estimated information is concatenated with the node's own information word and encoded; the resulting progressively-encoded codeword is then transmitted to the next node. In non-progressive coding, the node simply forwards the received noisy codewords along with its own encoded data. Here we compare the performance of two codes with low decoding complexity, Repeat-Accumulate (RA) and Low-Density Parity-Check (LDPC) codes, in combination with two progressive coding schemes. Progressive channel coding uses only channel decoding at the intermediate node, while progressive source-channel coding uses source-channel decoding, exploiting the probabilistic dependency of the information words (caused by the correlation structure of the data) jointly with the deterministic dependency induced by channel coding. Two decoding schemes are considered at the data center: channel decoding only and iterative source-channel decoding. In simulation experiments, we consider a line network topology with systematic RA and LDPC coding. Results show that progressive coding performs better than non-progressive coding, and RA codes perform better with lower computational complexity than LDPC codes, both for channel-decoding-only and iterative source-channel decoding.**

I. Introduction

Many wireless sensor network (WSN) applications involve data gathering, where sensors periodically sample their embedding environments and forward measurements to a sink or data center. Sensor nodes tend to have severely limited energy supplies, and communication normally dominates their energy-use budgets, especially since communication channels in sensor networks are often quite unreliable [1], [2]. Thus ensuring reliable communication is a significant challenge in wireless data gathering network design. While the use of forward error-correcting codes (ECC) can improve performance, practical applications have used ARQ techniques to avoid the computational complexity and energy cost of coding and, in the case of multi-hop topologies, in-network decoding.

However, ECC remains attractive, since the energy cost of retransmissions in ARQ can be extremely high when there are only a handful of bit errors per packet. Thus researchers have continued to explore the use of ECC in WSN [3], [4], [5], [6]. In [3], the effect of route diversity was examined and turbo codes [7] were applied. A comparison of codes for WSN was performed in [4] that included measurements of energy costs on hardware. It was found that BCH coding [8],[9] implemented using an application-specific integrated circuit (ASIC) can be a highly energy-efficient solution. In [5], the energy efficiencies of low-density parity-check (LDPC), convolutional and BCH codes were compared, with LDPC codes proving the most energy-efficient. In [6], repeat-accumulate (RA) [10] coding in a single-hop WSN was shown to reduce energy consumption by 24% relative to convolutional coding.

In multi-hop networks, the gathered data may be correlated in space or time, motivating the study of joint source-channel coding. For example, [5] used randomly-punctured LDPC-based joint source-channel coding. An alternative when information payloads are small is to perform no source coding and simple channel coding, but employ iterative source-channel decoding at the data center where energy is plentiful [11].

This paper explores the use of progressive coding and decoding to protect data as it is forwarded from node to node in multi-hop networks. A previous decode-and-forward strategy separately encodes parent and child data [12]. In progressive coding, parent nodes decode received messages from child nodes and then re-encode them together with their own data for forwarding. We consider two approaches: Progressive *channel* coding uses only channel decoding at intermediate (forwarding) nodes; we term this technique encode - channel decode - encode (ECDE). We also evaluate a more sophisticated (and hence energy-hungry) *source-channel* decoding technique that we call encode - source-channel decode - encode (ESCDE); it exploits the probabilistic dependency of the information bits (caused by the correlation structure of the data) jointly with the deterministic dependency induced by channel coding. We compare both ECDE and ESCDE to the non-progressive scheme where intermediate nodes forward received noisy codewords along with their own encoded data. At the data center, we employ iterative source-channel decoding (ISCD): channel and source decoders exchange estimated messages, with the source decoder favoring more probable messages based on a second-order probabilistic data model. RA [10] and LDPC [13] codes are compared in this setting in terms of performance and energy cost of encoding and decoding. RA codes are attractive in data gathering networks as they have low encoding and decoding complexity, yet achieve remarkable performance with iterative decoding [14].

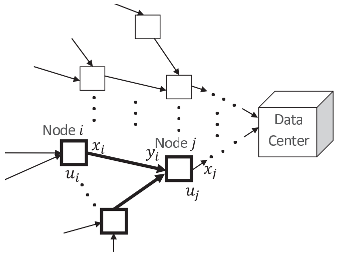

Fig. 1. Model of portion of a multi-hop wireless data gathering network, with BSC communication links shown as directed edges. Dotted edges indicate (possibly multiple) intermediate nodes and links.

The remainder of the paper is organized as follows. In Section II, we introduce the system model used in this paper. Section III reviews the structure and encoding/decoding algorithms of RA and LDPC codes. Iterative source-channel decoding and progressive coding strategies are described in Section IV and Section V respectively. In Section VI, the computational complexities of RA and LDPC codes are discussed. Section VII presents simulation results and discussion for progressive coding and decoding, followed by conclusions in Section VIII.

II. System Model

This paper assumes a multi-hop network model with a tree topology rooted at a single data center (Figure 1). Data flows toward the root, with each node connected to one or more preceding (child) nodes and a subsequent (parent) node. All communication links are binary symmetric channels (BSCs), reflecting the use of binary modulation and hard-decision detection in typical WSN node transceivers [15]. In our numerical results we assume a common crossover or bit error probability ρ for all links, but our algorithms can be trivially extended to the general case where links have variable reliabilities.

Each node i obtains a Gaussian-distributed analog measurement m_i with mean μ_m and variance σ_m^2. The measurements are correlated with correlation coefficient r, so that the $(i,\ j)$-th element of the measurement-vector covariance matrix Σ is $\Sigma_{i,j} = \sigma_m^2 r,\ i \neq j$, with $\Sigma_{i,i} = \sigma_m^2$. Each node i quantizes m_i into a k-bit information word $u_i \in \{0,1\}^k$. All information words collected by node i are termed as U_i, which is u_i for nodes without any child nodes, or a combination of u_i and decoded information words from child nodes in the case of progressive coding (Section V). The information word U_i is then channel-encoded into an l-bit codeword x_i. In our numerical results, we illustrate system performance using rate 1/4 systematic RA and LDPC channel codes.

Noise in the BSC link $i \rightarrow j$ is represented by $n_i \in \{0,1\}^l$; node j receives the noisy encoded sequence y_i from node i as $y_i = x_i \oplus n_i$, where $\oplus$ represents bitwise mod-2 addition. For progressive coding, node j decodes y_i to obtain an estimate $\hat{U}_i$ from each child node. These estimates are combined with u_j and encoded to form x_j, which is then transmitted to its parent node. In non-progressive coding, node j simply encodes u_j to x'_j and concatenates x'_j with the received codewords from its child nodes to form the transmitted sequence x_j.

Channel decoding at the sensor nodes and the data center uses the sum-product algorithm (see, e.g., [17]). Joint source-channel decoding at the data center depends on the transmission scheme utilized, and is discussed in Section IV. For the non-progressive scheme, channel decoding consists of separate decoders for each node, so that channel decoding occurs independently, with each decoder i producing an estimate $\hat{u}_i$. Both the nodes and the data center are assumed to know the crossover probabilities of their incoming links.

III. RA and LDPC Codes

First presented by Gallager in his 1962 thesis [13], LDPC codes are parity-check codes with a sparse, or low-density, parity-check matrix H, such that the density of 1s in H is very low. LDPC codewords x are generated from the information word u as $x = G^T u$; all codewords x satisfy $Hx = \mathbf{0}$, as $HG^T = \mathbf{0}$ [18]. Decoding occurs on a bipartite graph consisting of variable nodes (bits) and parity-check nodes formed from the parity-check matrix H [16], [17]. LDPCs are described by their node degree, or number of connections (edges) between variable and check nodes. Regular (d_v, d_c) LDPCs have constant variable degree d_v and check degree d_c. Irregular LDPCs have varying column and/or row weight.

RA codes [10], [14], [18] are linear block codes which have become popular for their low encoding and decoding complexity (with sum-product decoding), good performance and ease of implementation. An RA code combines two simple encoding operations, a repeater and an accumulator. These two operations are separated by an interleaver that permutes the order of the repeated bits. As RA codes have an H matrix that does not have constant column (or row) weight, RA codes may be viewed as irregular LDPC codes.

Both RA and LDPC codes are usually decoded by the iterative message-passing decoding algorithm known as sum-product decoding. The sum-product decoding algorithm uses probabilities as messages to calculate marginal distributions on graphical models. LDPC sum-product decoding operates on the factor graph of the code's parity-check matrix H [16], often using log-likelihood ratios (LLRs) $\lambda_x(i)$ as probabilistic messages, with $\lambda_x(i) = \log(p(x_i = 0)/p(x_i = 1))$.

IV. Iterative Source-Channel Decoding (ISCD)

Source-channel decoding (SCD) is a joint decoding process combining both a source and a channel decoder. The channel decoder contains knowledge of the channel code and minimizes the bit error rate (BER) of $\hat{u}$. The source decoder contains prior knowledge of the data's statistical parameters (i.e., data covariance matrix Σ and mean μ) and reduces the mean squared error (MSE) between sampled and estimated data. Iterative source-channel decoding (ISCD) [19] sends probabilistic information back and forth between source and channel decoder in an iterative fashion as for turbo decoding [7].

This section describes ISCD as applied at an intermediate node j in the network, shown in bold in Figure 1. Node j has a set of children $C_j = \{C_{j,1}, C_{j,2}, \ldots C_{j,|C_j|}\}$, where $|\cdot|$ is the cardinality operator. A child i of node j is denoted as $C_{j,i}$.

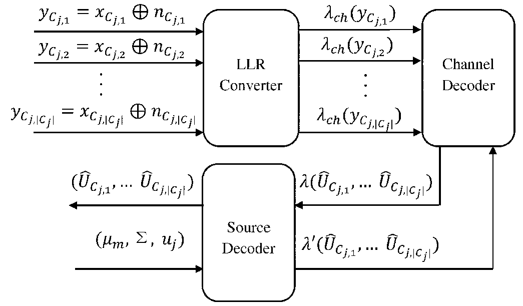
$y_{C_{j,1}} = x_{C_{j,1}} \oplus n_{C_{j,1}}$
$y_{C_{j,2}} = x_{C_{j,2}} \oplus n_{C_{j,2}}$
$y_{C_{j,|C_j|}} = x_{C_{j,|C_j|}} \oplus n_{C_{j,|C_j|}}$
LLR Converter
$\lambda_{ch}(y_{C_{j,1}})$
$\lambda_{ch}(y_{C_{j,2}})$
$\lambda_{ch}(y_{C_{j,|C_j|}})$
Channel Decoder
$(\hat{U}_{C_{j,1}}, \dots \hat{U}_{C_{j,|C_j|}})$
(μ_m, Σ, u_j)
Source Decoder
$\lambda(\hat{U}_{C_{j,1}}, \dots \hat{U}_{C_{j,|C_j|}})$
$\lambda'(\hat{U}_{C_{j,1}}, \dots \hat{U}_{C_{j,|C_j|}})$

Fig. 2. Block diagram of ISCD at intermediate node j.

The collection of information words from $C_{j,i}$ is termed $U_{C_{j,i}}$, which includes $u_{C_{j,i}}$ as well as information words from child nodes of i, if any.

A block diagram of the source-channel decoder used at node j is shown in Figure 2. The parent node j receives the noisy codewords $y_{C_{j,1}}, y_{C_{j,2}} \dots y_{C_{j,|C_j|}}$ and converts them to LLRs, sending them to its channel decoder. LLRs on the estimated information words $\hat{U}_{C_{j,1}}, \hat{U}_{C_{j,2}}, \dots \hat{U}_{C_{j,|C_j|}}$ output from the channel decoder are then sent to the source decoder. The source decoder combines this information with its knowledge of the network's statistical parameters and its own data to output an updated estimate of the estimated information words $\hat{U}_{C_{j,1}}, \hat{U}_{C_{j,2}}, \dots \hat{U}_{C_{j,|C_j|}}$ and their updated LLRs. Decoding may stop there. Alternately, decoding could continue in an iterative fashion, if the source decoder's LLR output is sent back to the channel decoder for use as *a priori* input, for another round of decoding. Iterative source-channel decoding (ISCD) may continue for a fixed number of iterations, or stop when a valid codeword is reached.

The source decoder used at node j contains conditional probabilities $p(u_i|U_{j\setminus i})$ for each information word u_i at node i, given the other information words $U_{j\setminus i}$ for all child nodes prior to node j, excluding node i. It calculates $p(u_i = A)$ by marginalizing over the joint probability $p(u_i = A, U_{j\setminus i})$, as

$$p(u_i = A) = \sum_{U_{j\setminus i}} p(u_i = A|U_{j\setminus i} = Q)p(U_{j\setminus i} = Q). \quad (1)$$

By combining probabilities on the information words from other nodes, the source decoder takes the inter-node correlation in account. Further details on the source decoder and iterative source-channel decoding (ISCD) are presented in [11]. Other types of source decoding could be used instead.

V. Progressive Coding

Progressive coding at a node j incorporates estimates of received data $\hat{U}_{C_j}$ from child nodes C_j together with u_j into the transmitted codeword x_j. This technique better protects data sent from child nodes further from the data center than does the non-progressive technique of simply forwarding noisy codewords received from child nodes in the network.

This paper considers two forms of progressive coding: 1) encode - channel decode - encode (ECDE), which uses channel decoding alone, ignoring data correlation; and 2) encode - source-channel decode - encode (ESCDE), which uses source-channel decoding with prior knowledge of the data correlation.

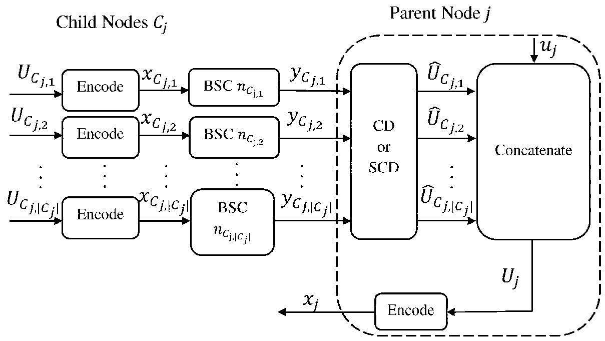

Fig. 3. Block diagram of progressive coding at intermediate node j.

Figure 3 shows a block diagram of progressive coding for the portion of a network bolded in Figure 1. The parent node j receives the encoded data $y_{C_{j,1}}, y_{C_{j,2}} \dots y_{C_{j,|C_j|}}$ from its $|C_j|$ child nodes. The two forms of progressive coding at node j are described below:

1) *ECDE Progressive Coding:* Using channel decoding, node j decodes $y_{C_{j,1}}, y_{C_{j,2}} \dots y_{C_{j,|C_j|}}$ to obtain estimates $\hat{U}_{C_{j,1}}, \hat{U}_{C_{j,2}}, \dots \hat{U}_{C_{j,|C_j|}}$. It then concatenates $\hat{U}_{C_{j,1}}, \hat{U}_{C_{j,2}}, \dots \hat{U}_{C_{j,|C_j|}}$ with its own data u_j as $U_j = [\hat{U}_{C_{j,1}}, \hat{U}_{C_{j,2}}, \dots \hat{U}_{C_{j,|C_j|}}, u_j]$, and encodes U_j into the codeword x_j.
2) *ESCDE Progressive Coding:* Same as ECDE progressive coding, except iterative source-channel decoding (ISCD) is used at node j instead of channel decoding alone to decode $y_{C_{j,1}}, y_{C_{j,2}} \dots y_{C_{j,|C_j|}}$. The estimates are concatenated with u_j to form U_j, which is encoded into x_j as for ECDE. To minimize the complexity for progressive coding at the intermediate nodes, only a single iteration of ISCD may be used.

ESCDE progressive coding uses ISCD at the intermediate nodes. For ECDE progressive coding, decoding at the intermediate nodes uses only channel decoding, without source decoding, to reduce complexity. Either form of progressive coding, or non-progressive coding, could be used at the intermediate nodes, combined with either form of decoding at the data center.

VI. Complexity Analysis of RA versus LDPC codes

In this section, the computational complexity of RA versus LDPC encoding and decoding algorithms are compared. All cases assume an information word u of length k, codeword of length n, and code rate $R_c = k/n$.

A. Channel Coding

RA encoding involves repetition, an interleaver and an accumulator (mod-2 addition). All these operations are linear in k. Thus the computational complexity for RA encoding is $O(k)$.

LDPC encoding involves matrix multiplication of the information word u of length k with G, the $k \times \frac{k}{R_c}$, or $k \times n$, generator matrix. Thus the computational complexity is $O(k^2)$. LDPC encoding is thus more complex and energy-consuming than RA encoding for the same data length and code rate.

B. Channel Decoding

As RA and LDPC codes are decoded using the same sum-product graph-based decoding algorithm, we can determine their relative computational complexities by analyzing their H matrices or factor graphs. A larger number of edges in the factor graph (number of 1s in the H matrix) results in more computations at both parity-check and variable nodes.

A systematic RA code of rate R_c and repetition rate R (with $R_c = 1/(R+1)$) has k systematic bits of variable degree $d_v = R$, $n-k-1$ non-systematic bits with $d_v = 2$ and 1 non-systematic bit with $d_v = 1$. The systematic RA code graph has $Rk + 2(n-k-1) + 1 = 3k((1/R_c) - 1) - 1$ edges.

A regular (d_v, d_c) LDPC of rate R_c has all n codeword bits of degree d_v, and has $d_v n = d_v k/R_c$ edges. A regular $(d_v = 3, d_c)$ LDPC has $3k/R_c$ edges. Compared to the systematic RA code with $3k((1/R_c) - 1) - 1$ edges, regular $d_v = 3$ LDPC code graphs have $3k + 1$ more edges than RA code graphs. This results in $3k+1$ more sum-product computations at both parity-check and variable nodes for LDPCs compared to RA codes.

From both an encoding and decoding standpoint, RA codes have lower complexity than LDPCs, and thus make better sense in an energy-constrained network. As shown in Section VII, short RA codes of the same code rate R_c and codeword length n (data length k) also display better performance due to fewer short cycles in the code graph. For a wireless data gathering network with short data sequences and energy-constrained nodes, RA codes provide a better channel coding option than LDPCs from both energy and performance viewpoints.

C. Source Decoding

Source decoding depends on the correlation between data at different nodes and is independent of channel coding. Source decoding complexity is affected by the number of nodes and the joint distribution of the data. The relative complexity of source-channel decoding with RA codes compared to that with LDPC codes is the same as the relative complexity for channel decoding alone, showing a complexity savings with RA codes in decoding as well as encoding.

D. Progressive and Non-progressive Coding: Encoding

RA codes provide reduced complexity in comparison to LDPCs in both types of progressive coding (ESCDE and ECDE) as well as non-progressive coding. Assuming N nodes in the network, non-progressive coding would see an encoding complexity savings when using RA encoding of $N \times O(k)$ compared to $N \times O(k^2)$ for LDPCs, as each node encodes its data sequence of length k. However, in progressive coding, the concatenated data sequence increases in length when approaching the data center. Assuming a final node N prior to the data center, which concatenates the previous $N-1$ decoded data sequences, the final data sequence to be encoded has length Nk rather than k. The encoding complexity savings for RA encoding with progressive coding would be $O(Nk)$ compared to $O(N^2k^2)$ for LDPC encoding with progressive coding. When using progressive coding, the complexity savings of RA codes are amplified.

E. Progressive and Non-progressive Coding: Decoding

RA codes also provide reduced complexity in comparison to LDPCs in the decoding process for both types of progressive coding. For non-progressive coding, decoding only occurs at the data center. Since the data center is assumed to have unlimited energy resources, the relative complexities of RA and LDPC codes are unimportant for non-progressive coding.

In progressive coding, assuming a final node N prior to the data center which concatenates the previous $N-1$ decoded data sequences, the final data sequence to be encoded has length Nk. From Section VI-B, the complexity savings of RA decoding increases as the number of sensor nodes increases.

There is no complexity savings for RA codes in the source decoding portion of ISCD.

VII. Simulation Results and Discussion

This section presents simulation results for the following coding and decoding schemes described in Section V:

1) ECDE progressive coding with channel decoding (CD) at the data center;
2) ESCDE progressive coding with ISCD at the data center;
3) Non-progressive coding with channel decoding (CD) at the data center; and
4) Non-progressive coding with ISCD at the data center.

A two-node line network is used, with identical BSC links with crossover probability ρ between nodes. According to our model (Section II), the data samples m from different sensors are correlated with correlation coefficient r and identically Gaussian-distributed with mean μ_m and variance σ_m^2. The data sequence u at each sensor node has length $k = 8$ bits.

The performance measure used is the average mean squared error (MSE) between the analog values of $m_1,\ m_2$ and the estimates $\hat{m}_1$, $\hat{m}_2$ found from $\hat{u}_1$ and $\hat{u}_2$ after decoding at the data center. In data gathering networks, MSE is often preferable to BER, since the error between the estimated measurement $\hat{m}$ and actual measurement m is the quantity of importance. For example, compare $\hat{m} = 0$ to $m = 8$; only 1 bit of $\hat{u}$ is incorrect, but the measurement error $|m - \hat{m}|$ is quite large.

Rate 1/4 RA and LDPC codes are used for comparison in all four cases, with a 4-cycle-free regular (3,4) LDPC and an 8-cycle-free RA code. Sum-product decoding is used for all channel decoding. with 12 decoding iterations used in the channel decoder at the data center, and 5 source-channel iterations between source and channel decoders for ISCD at the data center. Both types of progressive coding use 12 channel decoding iterations at node 2. To reduce complexity at the energy-constrained intermediate node, only one source-channel decoding iteration is used at node 2 for ESCDE progressive coding. Each MSE point is calculated when 200 codeword errors have occurred after data center decoding.

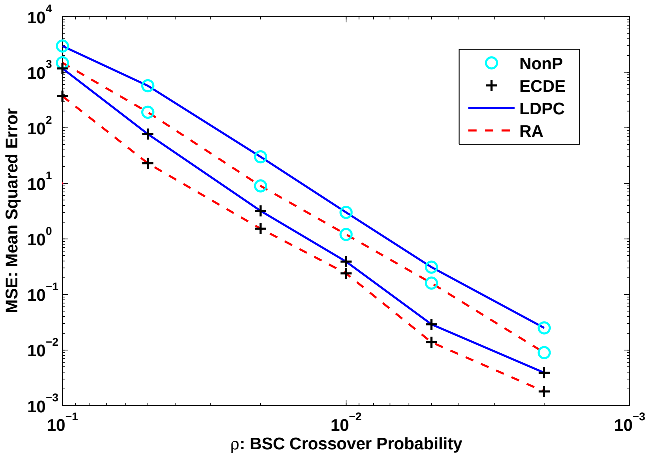

Fig. 4. MSE performance of ECDE progressive and non-progressive (NonP) coding, with channel decoding alone at the data center; $\mu_m = 70,\ \sigma_m^2 = 0.2,\ r = 0.9$.

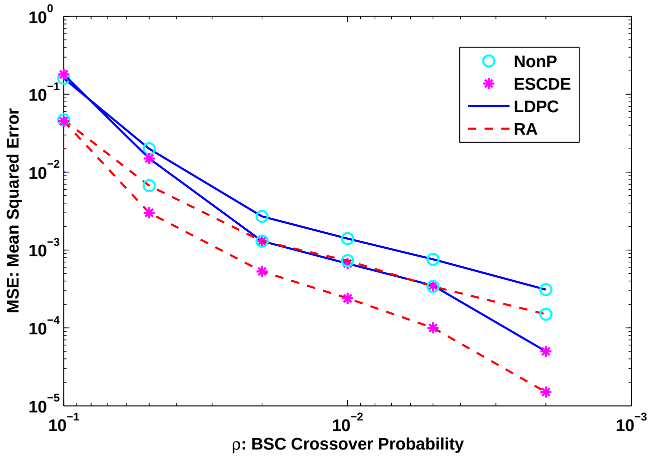

Fig. 5. MSE performance of ESCDE progressive and non-progressive (NonP) coding, with ISCD at the data center; $\mu_m = 70,\ \sigma_m^2 = 0.2,\ r = 0.9$.

A. Results

Figure 4 shows the MSE performance for ECDE and non-progressive coding using both RA and LDPC codes, with channel decoding alone at the data center (cases 1 and 3). The network data are identically Gaussian-distributed with low variance and high correlation: $\mu_m = 70,\ \sigma_m^2 = 0.2,\ r = 0.9$. From Figure 4, we see that progressive ECDE coding performs significantly better than non-progressive coding. As well, RA codes perform better than LDPC codes for short code lengths, for both ECDE and non-progressive coding.

In Figure 5, the MSE performance of ESCDE and non-progressive coding using RA and LDPC codes are compared using the statistical parameters of Figure 4, with ISCD used at the data center (cases 2 and 4). ESCDE progressive coding is shown to reduce MSE by more than an order of magnitude compared to non-progressive coding, Again, RA codes outperform LDPC codes, both for progressive and non-progressive coding. By comparing Figures 4 and 5, we observe that ESCDE with ISCD at the data center outperforms ECDE with channel decoding alone by two orders of magnitude or more, for both RA and LDPC codes, in a highly correlated network.

Figure 6 examines how the statistical parameters of the data affect MSE performance. The performance of ESCDE progressive coding with ISCD at the data center (case 2) is displayed for varying statistical parameters. Both RA and LDPC codes are considered. The results show that as data correlation decreases or data variance increases, MSE performance worsens. This is to be expected, as the source decoder provides less-reliable information to the channel decoder under these conditions.

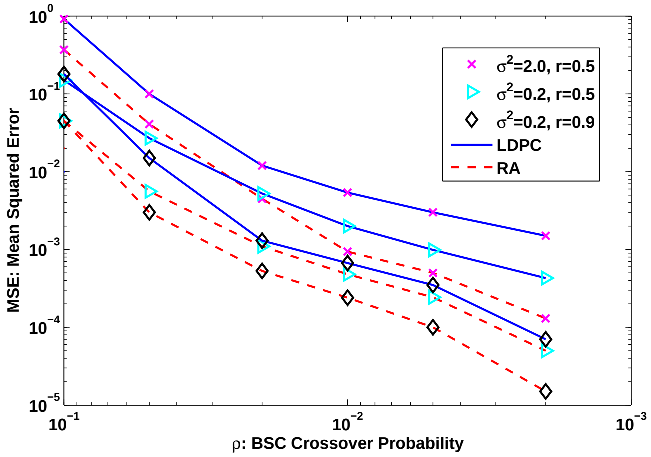

Fig. 6. MSE performance of ESCDE coding using RA and LDPC codes, with different statistical parameters and ISCD at the data center $\mu_m = 70$, $\sigma_m^2 = 0.2$ or 2.0, $r = 0.9$ or 0.5.

B. Analysis of Results and Discussion

As shown in Figures 4-6, better performance is achieved with RA codes of short length compared to LDPCs of the same rate and length. This result may seem surprising, given the lower complexity of RA codes. However, the very qualities that result in lower complexity for RA codes, namely, a highly-structured H matrix due to the simple repetition and accumulator encoding operations as well as fewer edges in the factor graph of H, also result in a graph with fewer loops or short cycles.

The effectiveness, and thus performance, of graph decoding algorithms such as sum-product decoding is reduced by loops or cycles in the graph, especially short cycles, such as a 4-cycle [20]. A 4-cycle connects two codeword bits to the same two parity-check nodes or equations, and is the shortest possible cycle. A 6-cycle connects each of three codeword bits to two of the three parity-check equations involved in the cycle. LDPC codes are typically constructed to be 4-cycle-free at a minimum, as shorter cycles degrade performance more than larger cycles.

RA codes of the same rate and length as regular LDPCs allow for easier construction of short-cycle-free H matrices due to their sparser and more constrained H matrix. As described in [18], the H matrix of systematic RA codes consists of a systematic (repetition/interleaver) portion H_u and a parity (accumulator) portion H_p, as $H = [H_u\ H_p]$. Each row of H_u contains a single 1; each column has R 1s, for repetition rate R. H_p is a double-diagonal matrix, indicating the accumulator operation. This special RA structure means that only one of the bits involved in a short cycle can be a systematic bit, so two of the 1s involved in the cycle are in the same column c of H_u. The double-diagonal structure enforces a cycle that runs from row r_1 of the first 1 to row r_2 of the second 1. The length of this cycle is determined by the row distance $r_2 - r_1$ between the two 1s in column c of H_u. When $r_2 - r_1 = 3$, an 8-cycle is formed. Increasing $r_2 - r_1$ to 4 results in a 10-cycle.

Due to the double-diagonal H_p, every systematic bit of an RA code is part of $R-1$ short cycles. However, these cycles can be lengthened by enforcing a minimum spreading distance S between all 1s in each column of H_u. $S = 3$ gives a minimum row distance of 4, and an 8-cycle-free H matrix. The maximum possible row distance for a rate $R_c = 1/(R+1) = k/n$ is $(n-k)/R = k$, obtained with 1s in column i located in rows i, $i+k$, and $i+2k$, $\forall i = 1, \ldots, k$. This provides maximum spreading $S_{\max} = k-1$. A random design of H_u with spreading constraint is unlikely to achieve $S_{\max}$, but a design with $S = \lfloor \frac{k}{2} \rfloor - 1$ can be found in reasonable time. Our RA design with $k = 8$ provided $S = 3$ for an 8-cycle-free interleaver.

Elimination of short cycles for an RA code of short length compared to an LDPC of the same length and code rate is responsible for the better performance of the RA code. Better short-cycle elimination for the RA code is primarily due to the specialized structure of its H matrix and secondarily to somewhat fewer edges in its factor graph (fewer 1s in H).

VIII. Conclusions

In this paper, we proposed two progressive coding techniques for wireless data gathering networks: ECDE (encode - channel decode - encode) and ESCDE (encode - source-channel decode - encode). These were compared to non-progressive coding (the standard encode-and-forward technique). The ESCDE technique utilizes knowledge of the source distribution via iterative source-channel decoding (ISCD). The performances of four different schemes were evaluated via simulation: 1) ECDE progressive coding with channel decoding at the data center; 2) ESCDE progressive coding with ISCD at the data center; 3) Non-progressive coding with channel decoding at the data center; and 4) Non-progressive coding with ISCD at the data center. For each scheme, the MSE performances of comparable (with respect to information word length and rate) RA and LDPC channel codes were measured for a line topology and specific statistical models of the network data.

Using ISCD at the data center can provide a significant performance improvement over channel decoding alone, for highly correlated data. Progressive coding consistently outperformed non-progressive coding for both channel decoding only and ISCD. RA codes achieved better performance than LDPC codes; this was confirmed by analysis showing that for the short codes used here, RA codes had fewer short cycles. We also examined the schemes' computational complexities (as a predictor of energy efficiency) at the sensor nodes. Our results show that RA codes may have better energy efficiency than LDPC codes. Overall, progressive coding with RA codes and source-channel decoding in both intermediate nodes and at the data center provides the best MSE performance with moderate computational complexity. Source-channel decoding can exploit highly-correlated data, while ECDE progressive coding and channel decoding at the data center can be used to advantage in network environments with low data correlation.

Acknowledgments

The authors gratefully acknowledge NSF support of this research through grants BRIGE-0824322 and CNS-0540414.

References

[1] A. Woo, T. Tong and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," 1st *Int'l Conf. on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, USA, 2003, pp. 14-27.

[2] D. Puccinelli and M. Haenggi, "Reliable Data Delivery in Large-Scale Low-Power Sensor Networks," *ACM Trans. on Sensor Networks*, July 2010.

[3] T. Wada, K. Ohuchi, A. Jamalipour, H. Okada and M. Saito, "Performance Evaluation of Wireless Sensor Networks Using Turbo Codes with Multi-Route Transmission," *IEEE Int'l Conf. on Communication (ICC)*, June 2007, pp. 3859-3863.

[4] G. Balakrishnan, M. Yang, Y. Jiang and Y. Kim, "Performance Analysis of Error Control Codes for Wireless Sensor Networks," *Fourth Int'l Conf. on Information Technology*, Las Vegas, NV, Apr. 2007, pp. 876-879.

[5] M. Sartipi and F. Fekri, "Source and channel coding in wireless sensor networks using LDPC codes," *IEEE Conf. on Sensor and Ad Hoc Communications and Networks (SECON) 2004*, Oct. 2004, pp. 309-316.

[6] Y. Ye, X. Liu and H. Cho, "An energy-efficient single-hop wireless sensor network using Repeat-Accumulate codes," *Int'l Conf. on Communications, Circuits and Systems*, Fujian, China, May 2008, pp. 419-423.

[7] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *IEEE Int'l Conf. on Communication (ICC)*, May 1993, pp. 1064-1070.

[8] R.C. Bose and D.K. Ray-Chaudhuri, "On A Class of Error Correcting Binary Group Codes," *Information and Control*, 3:68-79, March 1960.

[9] A. Hocquenghem, "Codes correcteurs d'erreurs", *Chiffres* (Paris), 2:147-156, Sept. 1959.

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

[11] S.L. Howard and P.G. Flikkema, "Integrated Source-Channel Decoding for Correlated Data-Gathering Sensor Networks," *IEEE Wireless Communications and Networking Conf. (WCNC)*, 2008, pp. 1261-1266.

[12] A. Sendonaris, E. Erkip and B. Aazhang, "User cooperation diversity," *IEEE Trans. Commun.*, vol. 51, pp. 1927 - 1948, Nov. 2003.

[13] R.G. Gallager, "Low-density parity-check codes", *IRE Trans. Information Theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962.

[14] H. Jin and R.J. McEliece, "RA codes achieve AWGN channel capacity," *Proc.* 13th *Symp. Applied Algebraic Algorithms, Error Correcting Codes*, 1999, pp. 10-18.

[15] Y. He and P.G. Flikkema, "System-Level Characterization of Single-Chip Radios for Wireless Sensor Network Applications," 10th *Annual IEEE Wireless and Microwave Technology Conf.*, Apr. 2009.

[16] N. Wiberg, H.A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *Eur. Trans. on Telecommunications*, vol. 6, Sep. 1995, pp. 513-525.

[17] H.A. Loeliger and J. Dauwels, V. M. Koch, and S. Korl, "Signal processing with factor graphs: examples," *First Int'l Symp. on Control, Communications and Signal Processing*, 2004, pp. 571-574.

[18] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003, pp. 557-572, 582-587.

[19] J. Hagenauer and N. Görtz, "The turbo principle in joint source-channel coding", in *Proc. IEEE Infor. Theory Workshop*, 2003, pp. 275-278.

[20] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Infor. Theory*, vol. 45, no. 2, pp. 399-431, Mar. 1999.