Maximal Recovery Network Coding under Topology Constraint

Kiran Misra, Shirish Karande, Hayder Radha
Department of Electrical and Computer Engineering, Michigan State University
East Lansing, MI 48824-1226 USA
{misrakir, karandes, radha}@egr.msu.edu

Abstract—Recent advances have shown that channel codes can be mapped onto networks to realize efficient Network Coding (NC); this has led to the emergence of Code-on-Network-Graphs (CNG). Traditional CNG approaches (e.g Decentralized Erasure Codes) focus on a generating a sequence of encoded symbols from a given input source (of size K), such that the original symbols can be recovered from any subset of the encoded symbols of size equal to or slightly larger than K. However in all cases the number of source symbols recovered falls rapidly if the number of encoded symbols received falls below K. In this paper we determine the CNG code-ensembles (under statistical topology constraint) which result in maximal recovery of WSN source data (for different erasure-rates), thereby minimizing the deterioration in data recovery. We also perform fixed point stability analysis on the underlying LDPC code ensemble. We then propose a distributed algorithm for generating a sequence of encoded symbols adhering to the designed code ensemble. Optimal solutions for a sensor network with 1000 nodes is determined using the Differential Evolution algorithm, and the solution sensitivity to variance in number of sensor nodes and node-interconnectivity is evaluated.

Index Terms—Low Density Parity Check Codes, Wireless Sensor Networks, Network Coding

I. INTRODUCTION

Information theory and coding theory have increasingly been playing a significant role in providing viable solutions to networking problems. This resulted in an intriguing convergence between coding and networking for data delivery/persistence [9] [8] [10]. In this paper, we propose a distributed code-over-network framework for self-configuring Wireless Sensor Networks (WSNs), with data streams which adapt to the dynamic network configuration and to the fluctuating erasure rates within the network. We first describe the mapping of sensed-data to (a subset of) variable nodes of a Low Density Parity Check (LDPC) code [1] which gives erasure-resilient Network Coding (NC). We then outline a framework which: (a) Determine the code ensemble which maximizes the data recovered at the receiver (for different erasure-rates), (b) Analyze and determine fixed-point stability conditions for the “maximal recovery code ensemble”, (c) Develop a Distributed Code over Network (DCoN) mapping algorithm. In [11] we proposed a centralized NC approach which uses channel coding techniques. However, here we develop a distributed coding algorithm (supported by underlying WSN topology), critical for developing a framework robust to failures.

The paper is organized as follows: Section II discusses related work. Section III gives problem formulation, followed by asymptotic analysis in section IV. Section V describes the network model followed by a description of distributed code-over-network (DCoN) algorithm and its analysis. In section VII we describe the Differential Evolution algorithm used to obtain a global optima under different transmission range settings. The experimental results and solution-sensitivity are discussed in section VIII. We conclude in section IX.

II. RELATED WORK

Random linear network coding [2] when employed in WSN generates a sequence of encoded symbols, such that the sink can recover the original source (with high probability) if it receives a subset of the encoded symbols the same size or slightly greater than the original source [8] [9]. The source data is typically recovered using Gaussian Elimination (GE). Although, efficient in terms of the number of encoded symbols required for decoding, the GE algorithm has cubic complexity. Moreover, a significant drawback of GE based decoding is that if the number of received encoded symbols at the sink falls below a certain threshold, the amount of source data recovered quickly diminishes to zero. In [8] Dimakis et al. proposed GE decoding based, Decentralized Erasure Codes (DEC) which uses random linear NC for distributed storage in WSN. A slightly different approach was taken by Kamra et al in [10], where Growth Codes (GC) with dynamically changing degree of encoded symbols was employed in conjunction with Belief Propagation (BP) decoding [3]. The use of BP decoding permits partial data recovery. However, both DEC and GC generate a stream of encoded symbols which can be tens of times the length of embedded source data, resulting in significant energy consumption. In energy constrained WSN, we are often allowed to transmit only a fixed number of symbols towards the sink, and the receiver (through post-processing) tries to recover as much of the original data as possible. In this paper we design fixed-length LDPC codes (constrained by underlying network connectivity) which provide optimal recovery for different erasure-rates within the WSN.

III. PROBLEM FORMULATION

In [1], Bao et al. mapped XOR-based NC operations of the incoming data flows (variable nodes of LDPC) at intermediate nodes of a WSN to the check equations of an LDPC code. This enabled Network Coding (NC) to be
represented with a belief network and equated the NC design problem to that of LDPC code design. Fig. 1, depicts a simple example where the XOR operations of NC can be converted to LDPC check equations (belief network):

\[
\begin{align*}
&v_1 + v_2 + v_3 = s_1 \
&v_2 + v_3 = s_2 \
&v_1 + v_2 + v_3 + s_1 = 0 \
&v_2 + v_3 + s_2 = 0
\end{align*}
\]

The subset of XOR-ed symbols which reach the sink is subjected to BP decoding to retrieve as much of the original source data as possible. In the rest of the paper we refer to source as “data” bits, and the encoded symbols generated by the intermediate sensor nodes as “parity” bits. Additionally, nodes corresponding to the XOR equations in the belief network are termed as “check” nodes and the collection of data and parity bits which appear in the equation are called “variable” nodes. The amount of original data recovered at the sink is a function of the number of encoded symbols received and the node degree distribution of the received parity symbols. In the rest of the paper we refer to source data as possible. In the rest of the paper we refer to source as “data” bits, and the encoded symbols generated by the intermediate sensor nodes as “parity” bits. Additionally, nodes corresponding to the XOR equations in the belief network are termed as “check” nodes and the collection of data and parity bits which appear in the equation are called “variable” nodes. The amount of original data recovered at the sink is a function of the number of encoded symbols received and the node degree distribution of the received parity subset [7]. The precise definition of “node” and “edge” degree distributions, along with their notational representation is given in Table I. The “node” and “edge” parity degree distribution can be inter-converted using:

\[
\phi_i = \sum_{j \neq j} \phi_j, \quad \phi_i = \sum_{j \neq j} \phi_j
\]

The inter-conversion formulae for data, check and variable degree distributions are identical to (1). Let us now take a closer look at the BP decoding algorithm.

TABLE I

| $\phi$ | \{\phi_1, \phi_2, ..., \phi_q\}, where $\phi_i$ is the fraction of edges connected to degree i parity nodes (q is maximum parity node degree). |
| $\varphi$ | \{\varphi_1, \varphi_2, ..., \varphi_q\}, where $\varphi_i$ is the fraction of nodes connected to degree i parity nodes. |
| $\theta$ | \{\theta_1, \theta_2, ..., \theta_n\}, where $\theta_i$ is the fraction of edges connected to degree i data nodes (n is maximum data node degree). |
| $\rho$ | \{\rho_1, \rho_2, ..., \rho_{q+1}\}, where $\rho_i$ is the fraction of edges connected to degree i check nodes. |
| $\beta$ | \{\beta_1, \beta_2, ..., \beta_n\}, where $\beta_i$ is the fraction of nodes connected to degree i check nodes. |
| $\lambda$ | \{\lambda_1, \lambda_2, ..., \lambda_n\}, where $\lambda_i$ is the fraction of edges connected to degree i variable nodes. |
| $\theta$, $\rho$ | $\approx \{\theta, \rho\} \approx \{\lambda, \beta\}$: Code Degree Distribution |

If there is at least one check equation that has exactly one variable node erased then the erased variable node can be recovered immediately by XOR-ing the remaining known variable nodes participating in the check equation. The previously erased variable nodes can now be flagged as known and the process is repeated till no check equations with exactly one variable node erased remains.

For degree distributions with fixed maximum degree BP has complexity - linear in the number of received encoded symbols. We now define data-recovery as:

\[
\delta = \frac{\text{Number of source symbols recovered}}{K}
\]

where, $K$ is the number of source symbols encoded by the network. The optimal mixing rules ($\bar{\phi}^*, \bar{\theta}^*$) for maximal data recovery, can now be mathematically stated as:

\[
(\bar{\phi}^*, \bar{\theta}^*) = f(K, M, \text{P}_{\text{erasure}}, \bar{\tau}) = \arg \max_{\phi, \theta} \delta \quad (3)
\]

$M$ is number of parity symbols generated by network $P_r$, the rate of erasure as seen by the data-sink $\bar{\tau}$ is the network degree distribution having $\bar{\tau}$, fraction of sensor nodes with degree $i$ (h is maximum network node degree). The solution for the above problem determines the optimal $(\bar{\phi}, \bar{\theta})$, and equivalently the optimal $(\bar{\phi}, \bar{\theta})$ as shown below:

\[
\bar{\lambda} = \{\bar{\lambda}_1, \bar{\lambda}_2, ..., \bar{\lambda}_n\}
\]

\[
= \left(K/(K + M)\right) \ast \{\bar{\theta}_1 + K/M, \bar{\theta}_2, \bar{\theta}_3, ..., \bar{\theta}_n\} \quad (4)
\]

\[
\bar{\rho} = \{\bar{\rho}_1, \bar{\rho}_2, ..., \bar{\rho}_{q+1}\} = \{\bar{\rho}_1, \bar{\rho}_2, ..., \bar{\rho}_q\} \quad (5)
\]

IV. ASYMPTOTIC OPTIMALITY AND FIXED POINT STABILITY

We now look at some asymptotic results in literature to determine the nature of expected solution. We then determine a stability condition which needs to be satisfied by any $(\bar{\phi}^*, \bar{\theta}^*)$.

Random Walk based Analysis: In [7] extensive asymptotic analysis was carried out (using recent results in random-walks over hypergraphs [6]-Thm 2.1 for equation (3), where $\bar{\tau}$ was chosen to be a complete graph (and data size $K \rightarrow \infty$). We reproduce the key results of [7] below, so they may be compared to the results obtained in section VIII for more realistic settings of $\bar{\tau}$. Let us define $r$ as:

\[
r = (M(1 - P_{\text{erasure}})) / K
\]

Typically, $K$ is fixed and $M$ is determined based on the energy constraint within the WSN. Meanwhile, $P_r$ is a function of the packets drops resulting from congestion/corruption within the network. Equation (6) implies that a solution obtained for a single $r$ is applicable for a wide range of energy constraints ($M$) and erasure rates ($P_r$) within the WSN.

In [7] the Thm. 2.1 of [6] is restated as:

**Theorem 1.** Let $r \in \mathbb{R}, r > 0$, then we have:

\[
r \bar{\phi}(t) + ln(1 - t) > 0, \text{ for } 0 \leq t \leq s(r, \bar{\theta}) \text{ where } s(r, \bar{\theta}) = \inf \left\{ t \in [0, 1] \mid r \bar{\phi}(t) + ln(1 - t) \right\} \land 1
\]
If the limiting degree distribution \( \Phi \) for \( K \to \infty \) and number of received parity bits \( \sim Poi(rK) \), we have \( \delta \to s(r, \Phi) \).


In [7] a further analysis of Theorem 1 for \( r < 1 \) led to the following conclusions:

- For \( 0 < \delta \leq 1/2 \) \( \Rightarrow 0 \leq r \leq \ln 2 \), only degree 1 symbols are required at the receiver, implying that only uncoded data need be transmitted from the source.

  Since our problem formulation has a fixed \( M \), this corresponds to network conditions where erasure rate is: \( 1 - K \ln 2/M \leq P_{\text{erasure}} \leq 1 \).

- For \( 1/2 < \delta \leq 2/3 \) \( \Rightarrow \ln 2 \leq r \leq (3/4) \ln 3 \) any encoded symbols must be of degree 2. This translates to an erasure rate: \( 1 - (3K \ln 3)/(4M) \leq P_{\text{erasure}} < 1 - K \ln 2/M \).

- For \( \delta > 2/3 \) (i.e. \( P_{\text{erasure}} < 1 - (3K \ln 3)/(4M) \)) no analytic solution exists.

Above results imply that optimal \( (\Phi^*, \Phi^*) \) would change based on \( r \). We show that this is indeed the case in section VIII.

Density Evolution based Stability Analysis: The BP algorithms performance can be determined by using Density Evolution [3] [4] an analytic tool used to track the evolution of log-likelihood ratios probability density function along the edges of a belief network. In Density Evolution, the fraction of incorrect messages passed from the variable node to the check nodes at the \( \ell \)-th iteration is calculated, by assuming that the graph does not contain cycles of length \( 2\ell \) or less. Message densities from check to variable-nodes can be represented as:

\[
R_\ell = \Gamma^{-1} \rho (\Gamma (P_0 \otimes \lambda (R_{\ell-1})))
\]

\( R_\ell \) represents probability distribution of messages from check nodes to variable nodes after \( \ell \) iterations.

\( P_0 \) represents rate adjusted probability distribution of messages received from the hypothetical channel (which is a combination of the actual relay channel and implicitly erased data bits).

\( \otimes \) represents convolution of message densities

\( \Gamma = \Gamma^{-1} \) is change of variable operation associated with the function \( \Psi(x) = \ln((e^x + 1)/(e^x - 1)) \)

\( \lambda(x) = \sum_{\psi_i} \lambda_i x^{\psi_i - 1} \)

\( \rho(x) = \sum_{\psi_i} \rho_i x^{\psi_i - 1} \)

For variable-nodes to check-nodes we have:

\[
P_\ell = P_0 \otimes \lambda (R_{\ell-1})
\]

where, \( P_\ell \) represents the probability distribution of messages from variable to check nodes after \( \ell \) iterations. We are now prepared to give the stability condition for any pair of \( (\lambda, \rho) \); given \( \delta \) fraction of source symbols were recovered.

Theorem 2. A \( (\Phi^*, \Phi^*) \) which requires \( \delta \) fraction of source symbols is stable if \( \lambda' (1 - \rho' (1 - \beta)) \rho' (1 - \beta) \leq \beta^{-1} P_{\text{erasure}} \), where \( (\lambda', \rho') \) is obtained by replacing \( M \) in (4), (5) with \( M' = (1 - P_{\text{erasure}})M \) (i.e. considering only the bipartite belief graph obtained after removing the erased parity nodes and the corresponding check nodes). Also, \( P_{\text{erasure}}^C = \left( \frac{K}{K+M} \right) P_{\text{erasure}} \) is the initial adjusted code erasure rate.

Proof: Let us represent the effective erasure-rate at iteration \( \ell \) and \((\ell + 1)\) with \( P_{\ell}^{C} \) and \( P_{\ell+1}^{C} \), respectively. For the Binary erasure channel \( P_{\ell+1} \) is said to be a fixed point of \((\lambda', \rho')\), if \( P_{\ell}^{C} \leq P_{\ell+1}^{C} \). For stability we require that there be no fixed points for \( P_{\ell}^{C} \in (\beta, P_{\text{erasure}}^{C}) \). This would ensure the BP decoder does not get stuck on its way to recovering \( \delta \) source symbols. We want the effective erasure in the codeword to reduce monotonicly, i.e. \( P_{\ell+1}^{C} < P_{\ell}^{C}; \forall \beta < P_{\ell}^{C} < P_{\text{erasure}}^{C} \). Using (7) and (8) we have: \( P_{\ell+1} = P_0 \otimes \lambda_r (\Gamma (-1 \rho_r (\Gamma (P_0)))) \) which implies \( P_{\ell+1}^{C} = P_{\text{erasure}}^{C} \lambda_r (1 - \rho_r (1 - P_{\ell}^{C})) \leq P_{\ell}^{C} \).

Let \( P_{\ell}^{C} = x \), the previous inequality can now be written as:

\[
P_{\ell}^{C} \lambda_r (1 - \rho_r (1 - x)) - x \leq 0 \quad \text{on linearizing the density evolution equation we obtain the following stability condition for the fixed point} \quad x = (1 - \delta) \frac{K+M}{K} = \beta:
\]

\[
\frac{d h}{dx} \bigg|_{x=\beta} \leq 0 \Leftrightarrow \lambda_r' (1 - \rho_r (1 - \beta)) \rho_r' (1 - \beta) \leq \frac{1}{P_{\text{erasure}}^{C}}
\]

This condition can be used to verify the asymptotic fixed-point stability of the optimal degree distributions \( (\Phi^*, \Phi^*) \).

V. NETWORK MODEL

The WSN is partitioned into two logical groups, the sensor field responsible for collecting data and generating the encoded stream, and the relay network which forwards the packets to the data-sink. A distributed encoding algorithm is employed at each sensor node to achieve designed code degree distribution (details are given in section VI). One of the factors determining the optimal code degree distribution is the global network degree distribution, which changes over time and needs to be periodically propagated within the WSN. However, as we shall observe, for random sensor placement it is possible to closely approximate the overall network degree distribution in a distributed fashion. Let us consider a network with sensor nodes placed randomly on a grid of dimensions \( N \times N \). Each cell within the grid has a probability \( p = K/N^2 \) of containing a sensor (with transmission range \( \Upsilon \)). The degree distribution of the nodes within the sensor field can then be approximated by the following Binomial Distribution: \( \pi \sim Bi(N_b, p_b) \) where, \( N_b = K = N^2p \) and \( p_b = \pi \Upsilon^2/N^2 \). For large \( N_b \) we can approximate \( \pi \) by normal distribution.

VI. DISTRIBUTED GENERATION OF ENCODED SYMBOLS

Assuming an optimal \( (\Phi^*, \Phi^*) \) is known, we now outline the details of a distributed code over network (DCoN) mapping algorithm. The DCoN algorithm needs \((M, K, \Phi^*, \pi)\) at the mixing nodes. The \( M \) encoded symbols along with the implicitly erased data-bits are collectively called a codeword. The complete DCoN mapping algorithm which generates the parity stream involves two-stages. In the first (information exchange) stage, each sensor node transmits the sensed information
bit to all its neighbors in the sensor-field. In the second stage, each node employs algorithm 1 outlined in the box to generate parity bits with the overall codeword adhering to the distribution $\bar{\theta}$. We shall see in section VIII that this localized single-hop mixing is often sufficient to achieve performance akin to that which would be obtained by mixing data beyond a single-hop.

Algorithm 1 leaves little control over the data degree distribution $\bar{\theta}$ of the codeword. We can however estimate the $\bar{\theta}$ achieved, by assuming that each parity degree $d$ contributes a small independent fraction to the overall $\bar{\theta}$. These contributions $\bar{\theta}^d$ is $\bar{\theta}^d \sim \mathcal{N}(n_d, p_d)$ where $n_d = M \bar{\theta}_d$ and $p_d = d/K \sum_i \bar{\pi}_i$. The sum of these independent $\bar{\theta}^d$ gives:

$$\bar{\theta} = \sum_{d} \bar{\theta}^d \sim \mathcal{N}(\mu_{\bar{\theta}}, \sigma^2_{\bar{\theta}})$$

(10)

where $\mu_{\bar{\theta}} = \sum_{i=1}^{q} \mu_i$, and $\sigma^2_{\bar{\theta}} = \sum_{i=1}^{q} \sigma^2_i$. Note, we can use the above approximation in the stability condition (9). The probability that a data-bit generated by a sensor participates in a codeword bit formation, is termed as the "probability of coverage" and can be calculated as $1 - P(\bar{\theta} < 1)$.

VII. DIFFERENTIAL EVOLUTION AND NETWORK CONSTRAINTS

The optimal degree distributions $\bar{\theta}^*$ for $(K = 1000, M = 1500)$ are obtained using Differential Evolution [12]. For fixed $(K, M)$ we can replace $P_{\text{crashure}}$ with $r$ as seen in equation (6). We determine optimal solutions for $r = (0.1, 0.2, 0.3, ..., 1.5)$. Infeasible solutions are avoided by using penalty functions identical to those defined in [5]. The initial population size is set to $10q$ and the cross-over parameter $CR = 1$.

The $i$-th sample in next generation $(G+1)$ is determined using:

$$p_{i,G+1} = p_{\text{best},G} + 0.5 \left( p_{r_1,G} + p_{r_2,G} - p_{r_3,G} - p_{r_4,G} \right)$$

(11)

where, $r_1, r_2, r_3, r_4$ is chosen randomly. We present results for $q = 6$ in section VIII. Note, over a period of time the

\[\text{Fig. 2. (a) Recovery (d) as fraction of received symbols (r) increases, (b) Increase in Avg. Parity degree with increasing recovery(\delta)}\]

VIII. RESULTS

The data-recovery profile shown in Fig. 2a for different transmission ranges demonstrates that the loss in performance due to localized single-hop mixing ($\Upsilon = 6, 7, 8$) is small compared to mixing across the entire network with multiple hops ($\Upsilon = \infty$). A quick look at Table II however shows that the degree distributions which achieve the same amount of recovery for different transmission ranges are different. In Table II we can see the trend in the optimal degree distribution for $\Upsilon = \infty$ is similar to that predicted by the asymptotic analysis of section IV till $r = (3/4) \ln 3 \approx 0.8$. However there are some significant differences for the finite transmission range cases. The asymptotic analysis predicts a sharp change in degree distribution from all degree-one symbols to all degree-two symbols at $r = \ln(2)$. However, for ($\Upsilon = 6, 7, 8$) the change is gradual. Lower the transmission range, slower is the fall in fraction of degree-one nodes. This gradual fall in degree one nodes results in smaller average degree for finite transmission range for $\delta > 0.6$ (Fig. 2b). The degree two nodes sees a rapid rise for $\Upsilon = \infty$, and then higher degree nodes are introduced quickly. However, for finite transmission ranges the degree two node is dominant for a long time and therefore results in lower average degree (Fig. 2b). Next, we determine the sensitivity of this solution to change in network size and node-interconnections. To keep the same network degree distribution $\bar{\pi}$ for different number of sensors $K^*$ we determine new transmission ranges $\Upsilon^*$ for the sensors. We then analyze the sensitivity by measuring performance loss: $\Delta \delta = \delta(\bar{\pi}_{K^*}) - \delta(\bar{\pi}_{K^*})$ for $K^* = 500, 1000, 1500, 2000$.

Node-Interconnection Sensitivity: Fig. 3b shows the loss in performance for 1000 sensors, but with node-interconnections different from design. We observe $\Delta \delta < 0.02$, with most significant loss occurring in the range $0.7 \leq r \leq 1.3$. This shows that the degree distributions have some sensitivity to actual node-interconnections but not significant.

Sensor-Field Size Sensitivity: Fig. 3a,3c,3d illustrates that the loss in performance is more for full in $K$ then for a rise in $K$. This is consistent with results from channel coding where a
fall in codeword length typically leads to worse performance. Again most loss occurs for \(0.7 \leq r \leq 1.3\).

**IX. CONCLUSIONS**

Channel Coding formulation of NC can be used to design Network-Codes resilient to packet drops. We have shown how this formulation enables the design of Network-Codes tailored to the underlying network statistical topology. We also use this formulation to derive stability condition for the mixing rules (code degree distribution). We have demonstrated that the WSN can attain maximal recovery by choosing the appropriate mixing rules after considering the prevailing network conditions. A distributed code-over-network mapping algorithm was developed to achieve the designed degree distribution. It was shown that for networks with sufficient connectivity, single-hop mixing is adequate to achieve performance similar to mixing across the network with multiple-hops. Sensitivity of the “designed solution” to change in node-interconnections and sensor-field size was evaluated and performance loss was found to be small.

**ACKNOWLEDGEMENTS**

This work was supported in part by NSF Award CCF-0515253, NSF Award CNS-0453046, MEDC Grant GR-296, and unrestricted gift from Microsoft Research.

**REFERENCES**


**TABLE II**

<table>
<thead>
<tr>
<th>K</th>
<th>(\varphi_1)</th>
<th>(\varphi_2)</th>
<th>(\varphi_3)</th>
<th>(\varphi_4)</th>
<th>(\varphi_5)</th>
<th>(\varphi_6)</th>
<th>(\varphi_7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>1000</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>1500</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>2000</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

**Fig. 3.** Loss in performance for (a) \(K = 500\) (b) \(K = 1000\) (node interconnection sensitivity) (c) \(K = 1500\) (d) \(K = 2000\)