On Low-Complexity Network Coding for Data Collection in Wireless Sensor Networks

Dejan Vukobratović, Čedomir Stefanović, Vladimir Crnojević

***Abstract* — In this paper, we investigate the possibilities of developing and applying reduced complexity network coding schemes. The proposed schemes are motivated by recent developments in the field of sparse-graph erasure codes and the iterative decoding algorithms. Performances of low-complexity network coding schemes deployed for the purpose of data collection in wireless sensor networks are analyzed by means of the simulation experiments.**

***Keywords* — Data Collection, Network Coding, Wireless Sensor Networks.**

I. INTRODUCTION

IN the landmark paper [1], Ahlswede et al. established the multicast capacity of a network of lossless links, and introduced a technique named network coding, as a generalization of routing, necessary to achieve the multicast capacity. Introduction of network coding paradigm into communication networks created a wave of theoretical research on this topic, followed by a wave of research that analyzes possibilities for its applications in the real-world scenarios. In this paper, we investigate the possibilities of developing and applying reduced complexity network coding schemes, motivated by a well developed sparse-graph erasure codes theory. Performances of low-complexity network coding schemes deployed for the purpose of data collection in wireless sensor networks (WSN) are analyzed by means of the simulation experiments.

Practical randomized network coding framework is first described in [2]. In a certain sense, it shares a common ground with the recently popular idea of fountain coding developed in coding theory [3][4]. Both are developed for single-source multicast setting over the packet based networks, where the data generated at the source node should be available at the destination nodes. In fountain coding scenario, the data encoding process is performed only at the source node and data multicast can be understood as a set of erasure channels with different erasure statistics between the source node and each destination node. In the network coding scenario, the whole network between the source and the set of destination nodes is used for the encoding process. Another fundamental difference between the two is that fountain codes employ sparse linear equations with binary coefficients as their encoded data, decoded by simple linear-time iterative BP decoder at the destination, whereas as a result of network coding operation, the receivers are supplied by a standard, non-sparse, linear equations with coefficients from a selected finite field, decoded by the cubic complexity Gaussian elimination decoding. The simplicity and complexity advantages of the fountain approach, and network capacity utilization advantages of network coding, motivates the idea of combining the two schemes, particularly in reduced functionality environments such as wireless sensor networks, which is the direction we follow in this paper.

The work presented in this paper is supported in part by the applied research project No. 11022/08, awarded by the Ministry of Science and Technology, Rep. of Serbia.

D. Vukobratović, Č. Stefanović and V. Crnojević are with the Dept. of Power, Electronics and Communication Engineering, University of Novi Sad, Serbia (phone: 381-21-4750015; fax: 381-21-4752997; e-mail: {dejanv, cex, crnojevic}@uns.ns.ac.yu.

II. PRACTICAL NETWORK CODING SCHEMES

In this subsection, we describe the standard for practical implementation of network coding schemes, as described in [2]. The source data, residing in the source node memory, is divided into k data blocks $\boldsymbol{b} = [b_1, b_2, \ldots, b_k]$ of length l bits, where l is typically a whole number of bytes. The base finite field $\mathbf{F}(2^q)$ of size 2^q is selected; typically q equal to 8 or 16 in practical scenarios. It is assumed that q divides l so we can segment each data block into a sequence of $L=l/q$ elements from $\mathbf{F}(2^q)$. Each field element can be represented as q-bit sequence. Multiplication of q-bit field element with l-bit block is performed as $\mathbf{F}(2^q)$ field multiplication of the field element with each of L consecutive, non-overlapping, q-bit segments of the l-bit block. Each node performs distributed randomized linear network coding on the set of M coded blocks $\boldsymbol{c} = [c_1, c_2, \ldots, c_M]$ contained in its memory (the blocks received so far), or on the set of source blocks $\boldsymbol{b}$ in case of the source node. Each encoded block c_i is of the same length as the source blocks and is transmitted together with the vector $\boldsymbol{g}$ of its *global encoding coefficients* $\boldsymbol{g} = [g_1, g_2, \ldots, g_k]$, $g_i \in \mathbf{F}(2^q)$, such that $c_i = \boldsymbol{g}\boldsymbol{b}^{\mathbf{T}}$. It is assumed that $l << kq$, which makes the size of the overhead $\boldsymbol{g}$ acceptable. Every node in the network sends encoded blocks c^{out} concatenated with its global encoding vectors $\boldsymbol{g}^{\text{out}}$ on its outgoing links whenever transmission opportunity occurs, using the following encoding rule:

- Creating encoded block c^{out}: Randomly select $m \leq M$ field elements $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$ and form a random linear combination with m randomly chosen encoded blocks $\boldsymbol{c'} = [c'_1, c'_2, \ldots, c'_m]$ contained in memory, $c^{\text{out}} = \boldsymbol{y}\,\boldsymbol{c'}^{\mathbf{T}}$.

- Calculating global encoding vector $\boldsymbol{g}^{out}$: Multiply vector $\boldsymbol{y}$ with m x K matrix containing as its rows the global encoding vectors $[\boldsymbol{g'}_1, \boldsymbol{g'}_2, \ldots, \boldsymbol{g'}_m]$ corresponding to the randomly selected blocks $\boldsymbol{c'}$.

Receiving nodes perform original file recovery using the following decoding rule:

- Upon collecting k linearly independent encoded blocks, a node decodes the source data by solving a system of linear equations using Gaussian elimination. In other words, k x k matrix $\mathbf{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k]$ of the received global encoding row-vectors is first inverted and than multiplied with the vector of the received encoded blocks $\boldsymbol{c} = [c_1, c_2, \ldots, c_k]$.

The complexity of distributed randomized linear network coding is as follows. The encoding process takes mL field multiplications for c^{out} and mk field multiplications for $\boldsymbol{g}^{out}$. The decoding takes an order of $O(k^3)$ field operations for the matrix inversion, and Lk^2 field multiplications for the vector-matrix multiplication. In the following, we focus on reducing this complexity by assuming binary field arithmetic (simple ex-or operation) and the iterative decoding techniques.

III. Low-Complexity Network Coding in WSN

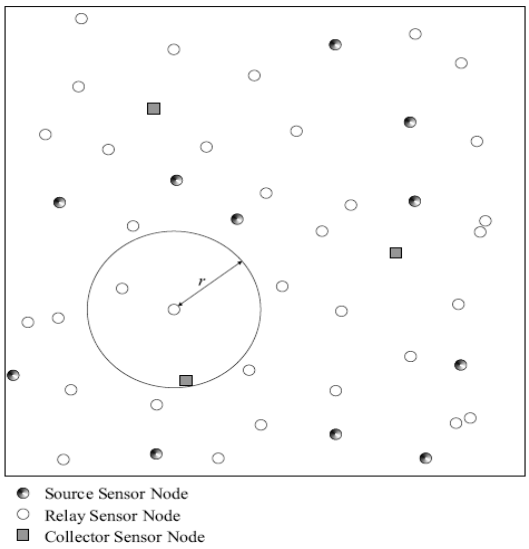

Figure 1. Wireless Sensor Network Setting.

In the following, we focus on the WSN environment as the communication network of interest. The possibilities of low-complexity data collection in WSN are discussed, where the goal is to transmit an occasionally generated data at the source sensor nodes to the small number of collector sensor nodes (gateways). We search for a mechanism that will be distributed, simple to implement, robust, and require as small as possible number of transmissions by each sensor node to make the source sensor nodes content available at the receiving collector nodes. Our development is based on both, the network coding principles and the fountain codes. To some extent, our setting is similar to the work presented in [7]-[9].

A. System Setting

We assume the WSN scenario with N sensor nodes randomly distributed over a unit square area. Each sensor is equipped with a wireless communication capabilities that enable reliable data transmission between the sensor and its neighbors within range r. This is a well established connectivity model of the random wireless network enviroment, called random geometric graphs $G(N,r)$ [5]. In our parameter settings, we use their fundamental critical radius of connectivity result that states that, in order for the network graph to be connected with high probability as $N \rightarrow \infty$, the following holds:

$$r > r_c = \sqrt{\frac{\log N + c}{\pi N}}, \qquad (1)$$

where $c \rightarrow \infty$.

We classify the set of sensors in WSN into the three sensor classes:

- Source sensors **S**: sensor from this class is able to perform measurments and generate packets of source data. We assume that WSN contains k source sensors S_i, $1 \leq i \leq k$, each of which produces a single block of data b_i to be communicated to the collector. We assume that the source data is generated periodically, in the discrete time instants t_{kT}, $k \geq 0$, with the relatively large period T. We focus on a single period $[t_{iT}, t_{(i+1)T}]$ of transmission of data blocks from the source nodes to the collector.
- Relay sensors **R**: relay sensors collect received data into their buffer memory and are able to produce encoded blocks based on their memory content. For simplicity, we will initially assume that their buffer capacity is infinite, and leave the finite buffer memory considerations for our future work. We assume that there are r relay sensors denoted as R_i, $1 \leq i \leq r$.
- Collector sensors **C**: or gateways, represent one or small number of WSN nodes that are connected with the outside network. The task of collector nodes is to collect data from their neighbors, recover the set of k source blocks $[b_1, b_2, \ldots, b_k]$ that originated at the source sensor nodes during a single time period, and forward this data to a database in the outside network.

All of the three classes of sensors are essentially simple wireless nodes with different modules. If the node has a sensing module, it is an **S** node, if it has a buffer memory and processing module, it is a **R** node, and if it has buffer memory, processing capabilities and network interface towards the outside network, it is a **C** node. It is possible in practice that a single node possess more than one of these functionalities, but for simplicity, we assume that these sets are disjoint in the following study. Also, we assume that there is considerably larger number of **S** and **R**, than **C** nodes (Figure 1).

The WSN network operation proceeds as follows. At the begining of the time period, in the time instant t_{iT}, all of the **S** nodes sense their environment and generate k source data blocks b_i. These data blocks are distributed across the **S** nodes of WSN and the goal is to collect this information at a single or small number of collecting points inside the

WSN. At the same initial time instant t_{iT}, the **S** nodes broadcast their data blocks to their neighbors, and their communication task is over until the beginning of the next time period $t_{(i+1)T}$. In the following period, during T discrete time intervals, the **R** nodes are responsible for disseminating this information. We assume that, at the beginning of each of T discrete time instant in the interval $[t_{iT}, t_{(i+1)T}]$, each **R** node produces a single encoded packet, and broadcasts this packet to all of its neighboring **R** and **C** nodes (**S** nodes do not have receiving buffer memory). The encoded packet is produced using linear network coding scheme, as described in the previous section, out of the existing packets in the node's memory. **C** nodes only collect blocks received from their neighbors, but do not produce and broadcast encoded data. Upon collecting sufficient number of packets inside its buffer memory, **C** node attempts to recover the **S** nodes data. The main goal of our (simulation) study is to make available all of the source data blocks $[b_1, b_2, \ldots, b_k]$ at any **C** node with high probability for as short time period T as possible, using low-complexity operations in **R** and **C** nodes.

B. Low-Complexity Network Coding for WSN

In this study, we take a different approach, based on a sparse graphs encoding and decoding techniques, to reduce the sensor node operation complexity. We deliberately trade content distribution system efficiency for its lower complexity. We minimize the encoding complexity in **R** nodes by reducing the base finite field to a binary field $\mathbf{F_2}$. In other words, new encoded blocks are formed by a bit-wise xor-ing of the encoded blocks contained in the node's memory. This way, we avoid field multiplications and substantially decrease the encoding complexity. However, by using binary field coefficients we deteriorate system efficiency, since distributed randomized linear network coding performance is tightly related with the base finite field size. On the other hand, linear combinations produced by each node are still dense and we are restrained from applying computationally efficient sparse-graph iterative decoding techniques at the collector nodes, to reduce the decoding complexity of Gaussian elimination decoding.

Our next step is to ensure that the exchanged coded blocks are described by sufficiently sparse binary global encoding vectors. At the **C** nodes we apply the iterative BP decoding applied on the incoming encoded blocks throughout the receiving process. This places any **C** node in the network in an equivalent setting as a digital fountain receiver. Ideally, in order to maximize efficiency (i.e. minimize ε), each **C** node should obtain from any of its neighbors an independent stream of fountain-like encoded blocks. In other words, we would like to ensure that binary global encoding vector of any received encoded block at any **C** node is as such as it is randomly and uniformly sampled at a fountain source using appropriately selected degree distribution $\Omega(x)$. In the most general setting, where any node in the network can be a **C** node, the previous requirement extends to any encoded block created in the network.

It is clear that in WSN scenario, where the source data is distributed across **S** nodes, no **S** or **R** node is able to produce independent fountain-like encoded blocks with binary global encoding vector drawn from the selected degree distribution $\Omega(x)$. The **R** nodes are limited by the content of their memory, and the requirement for low-complexity operation. The complexity of this problem is demonstrated in an interesting study performed in [6]. In our work, we would like to estimate, given the precise description of the low-complexity, sparse, binary-field data combining algorithm in **R** nodes, the basic properties of the encoded data received at the collector nodes, as the number of nodes in WSN increases. As this goal is part of our current research, in the following, we provide a results of simulation study, experimenting with several **R** nodes data combining algorithms.

IV. Simulation Results

In this section, we search for an intuition on behavior of a simple sparse binary data combining schemes in **R** nodes using simulation experiments. We set up a simulation where the total of $N = 101$ sensor nodes are uniformly distributed across the unit square surface. The transmission range of each sensor is set to $r = 0.35$. The set of sensors is divided into 50 **S** sensors, 50 **R** sensors, and a single **C** sensor. We set the time period to $T = 100$ discrete time instants. At each time instant, each **R** node combines the data in its memory and broadcasts a new encoded block to its neighbors. Also, at each time instant the **C** node attempts the decoding of its memory content and we are interested in the intermediate number of recovered **S** sensor blocks at each time instant. As our baseline scheme for comparison, we start with the performance of a simple forwarding scheme.

*A. Simple Random Forwarding in **R** Nodes*

In this scheme, the **R** nodes data combining algorithm is the simplest possible. The **R** node actually does not combine, but only forwards randomly selected data packet from its memory content. We are interested in the number of discrete time instants needed on average to collect the source data at the collector node.

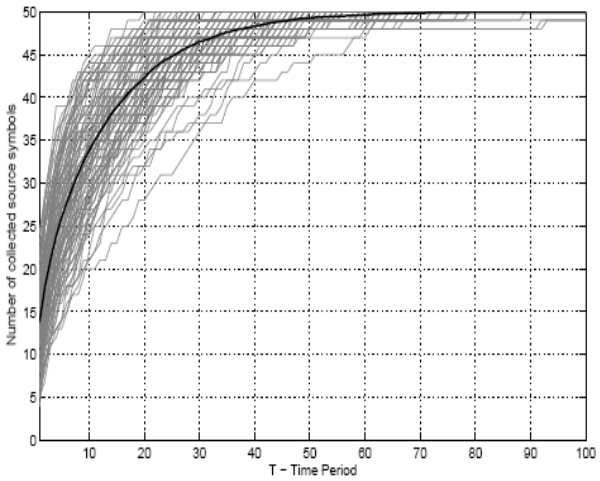

Figure 2. Data Collecting Performance with Simple Forwarding.

Figure 2 presents the simulation results of the simple random forwarding scenario. Total of 100 simulation runs are presented, and the average performance of the decoding process across simulation runs is given as the thick black line. If we change the WSN connectivity by changing the transmission range r, the data collection speed changes considerably. Figure 3 presents the average data collection performance over 100 simulation runs, where the simple random forwarding in **R** nodes is applied, for the transmission range r varying between 0.2 and 0.4. The simulation runs where any of **S** or **C** nodes are initially placed out of range of any other **R** or **C** nodes, i.e., there exist isolated **S** or **C** nodes, are repeated. In the following, we compare the results of the simple forwarding scheme with another simple data combining scheme in **R** nodes.

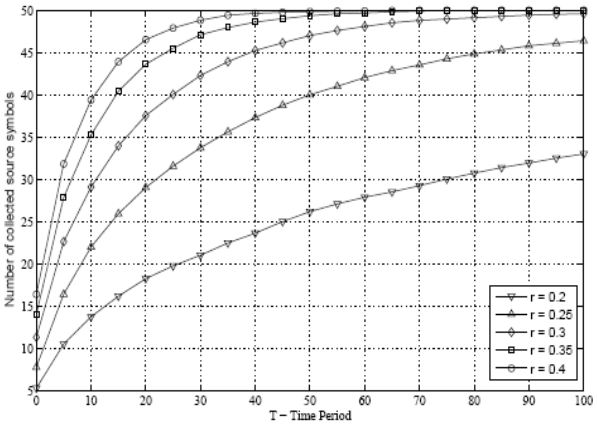

Figure 3. Simple Forwarding Performance for Different *r*.

*B. Simple Random Data Combining in **R** Nodes*

Unlike the previous **R** node behavior, where **R** nodes simply randomly forward the original source blocks from their memory, here we allow data combining in **R** nodes. As a very simple scheme, we select the one where **R** node either selects a single block from its memory and broadcasts it, with probability p_1, $0 \le p_1 \le 1$, or randomly selects two different blocks from its memory and performs their xor-ing and broadcasting. We represent the **R** node data combining using its data combining degree distribution $\Omega_R(x) = \sum_{i=1}^{D} p_i x^i$, where D is the maximum combining degree and p_1 is the probability of random combining of i packets from the **R** node memory.

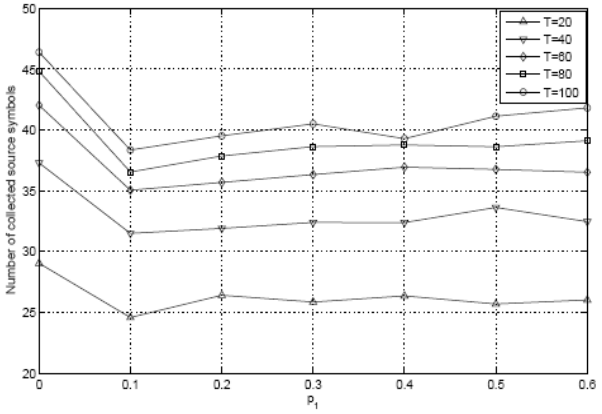

Figure 3. Simple Data Combining Performance for Different p_1.

In the simple combining scheme example, the data combining degree distribution is set to $\Omega_R(x) = p_1 + (1 - p_1)$. However, the major future goal of our study is to optimize the data combining degree distribution $\Omega_R(x)$ with respect to the duration of the data collection process, and the parameters of the randomly deployed WSN.

Figure 4 illustrates the results of simple data combining scheme using data combining degree distribution $\Omega_R(x) = p_1 + (1 - p_1)$, where p_1 is varied between $0 \le p_1 \le 1$. The average number of recovered source symbols at the **C** node after $T = \{20,40,60,80,100\}$ time instants, averaged over 100 simulation runs, is examined as a function of the probability p_1. The simulation results clearly demonstrate that obtaining a good $\Omega_R(x)$, which would significantly improve the simple random forwarding operation of **R** nodes, assuming the iterative decoder at the **C** node, is a complex task. Simple random forwarding beats in performance any version of simple random data combining parameterized for different values of p_1. Obviously, more involved design of $\Omega_R(x)$ is needed in order to make an improvement over the simple data forwarding scheme. We leave it for our future investigation.

V. CONCLUSION

In this paper, we examined some possibilities for robust, low-complexity, low-power consuming and efficient strategies for data collection in WSN. Our data collection strategy does not rely on data routing in WSN, but only on the broadcast nature of sensor communication. The goal of our study is to design a data combining scheme in **R** nodes that would maximize the speed of data collection process. This paper provides only a glimpse into the problem formulation, supported by simulation results of a few simple **R** node forwarding and combining schemes, and the further results on the topic are expected in the future.

REFERENCES

[1] Ahlswede, R., Cai, N., Lee, S.-Y.~R. and Yeung, R.~W.: "Network information flow," in IEEE Trans. on Info. Theory, 46(7),1204-1216, 2000.

[2] Chou, P.~A., Wu, Y. and Jain, K.: "Practical network coding," in Proc. of the 41th Allerton Conf., Allerton, USA, 2003.

[3] Luby, M.: "LT codes," in Proc. of the 43rd FOCS Symp., Vancouver, Canada, 271-282, 2002.

[4] Shokrollahi, A.: "Raptor codes," in IEEE Trans. on Info. Theory, 52(6), 2551-2567, 2006.

[5] Gupta, P. and Kumar, P.~R.: "The Capacity of Wireless Networks," in IEEE Trans. on Info. Theory, 46(2), 388-404, 2000.

[6] Puducheri, S., Kliewer, J. and Fuja, T.~E.: "The Design and Performance of Distributed LT Codes," in IEEE Trans. on Info. Theory, 53(10), 3740-3754, 2007.

[7] Acedanski, S., Deb, S., Medard, M. and Koetter, R.: "How Good is Random Linear Coding Based Distributed Networked Storage?," in Proc. of the 1st Netcod Conf., Riva del Garda, Italy, 2005.

[8] Dimakis, A., Godfrey, P., Wainwright, M. and Ramchandran, K.: "Network Coding for Distributed Storage Systems," in Proc. of the INFOCOM Conf., Anchorage, USA, 2007.

[9] Kamra, A., Feldman, J., Misra, V. and Rubenstein, D.: "Growth Codes: Maximizing Sensor Network Data Persistence," in Proc. of ACM SIGCOMM, Pisa, Italy, 2006.