COMPRESSIVE LINEAR NETWORK CODING FOR EFFICIENT DATA COLLECTION IN WIRELESS SENSOR NETWORKS

Francesca Bassi[†], *Chao Liu*[†], *Lana Iwaza*[†], *Michel Kieffer*[†,*,¯]

[†]LSS – CNRS – SUPELEC – Univ Paris-Sud, 91192 Gif-sur-Yvette, France
[*] On sabbatical leave at LTCI – CNRS Télécom ParisTech, 75013 Paris, France
[¯] Institut Universitaire de France

ABSTRACT

We address the problem of data collection in a wireless sensor network. Network coding is used for data delivery. The correlation between the measurements is exploited to recover the data at the sink, even in case of rank-deficient network matrix. The network coding operations are seen as lossy source compression, achieved by a finite-field random code generated during transmission. Decoding is performed using belief propagation on a factor graph which accounts for the correlation between the sensor measurements. Experimental results illustrate the performance of this technique for various field sizes and correlation levels.

Index Terms— Belief Propagation, Network Coding, Finite fields, Wireless Sensor Networks

1. INTRODUCTION

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors, and realizes a low-cost and massive sensing platform to monitor a physical quantity (*e.g.* temperature, pressure, sound). The task of the network is to collect measurements at the nodes and to convey them to a remote sink. A simple and computationally light transmission strategy is packet routing within the network [1], which fails, however, to exploit the fact that the data is broadcast to enhance the throughput. An effective utilization of the wireless feature is obtained by network coding [2], *i.e.* by allowing the nodes to combine all received packets before forwarding them, thus obtaining data delivery in a distributed and cooperative fashion. Network coding exhibits an all-or-nothing characteristic in performance: to be able to reconstruct the sensor measurements, the sink requires a number of independent linear combinations equal to the number n of the nodes. Depending on the routing algorithm, it might be necessary to wait until the delivery of much more than n coded packets before being able to perform the decoding operation. This is costly in terms of transmission delays and global throughput.

In this work we address the problem of the collection of spatially correlated measurements, characteristic of the sensing of physical quantities. We aim to exploit the statistical structure of the quantized measurement vector $\boldsymbol{X}$ to perform reconstruction even for rank-deficient transfer matrix A. Because its element are dependent, $\boldsymbol{X}$ is regarded as a compressible source. The observation vector $\boldsymbol{Y} = \mathrm{A}\boldsymbol{X}$ at the sink represents a collection of m random projections of $\boldsymbol{X}$, *i.e.* a compressed version of $\boldsymbol{X}$ computed as a by-product of the linear network coding operation. The decoding is performed using the belief propagation algorithm, which allows, when the rank of A is $m < n$, to exploit the statistical structure of $\boldsymbol{X}$ to compensate for the missing linear constraints. This strategy is inspired by [3], where compressible sources are encoded using the parity-check matrix of nonbinary LDPC codes. It is to be remarked, however, that in our scenario compression is achieved in a distributed fashion, jointly with network coding.

The literature already reports several solutions exploiting the correlation in the measurements to enhance performance. Network coding of compressed packets has been considered in [4] in the context of file sharing. Distributed lossless compression [5] within the network has been proposed in [6]. These techniques rely on the separation of the compression and the network coding stages. A joint approach has been explored in connection with the use of compressed sensing tools [7]. Layered network organization have been proposed in [8]: measurements from sensors in the lower layer are gathered at intermediate collection nodes, which perform random linear combinations to be transmitted to the upper layer, in charge to deliver them to the sink, see also [9, 10]. Real-field network coding [11–13] as well relies on compressed sensing. Real-field codes facilitate the recovery of data samples allowing a progressive improvement of the quality of the reconstructed samples with the number of network-coded packets received.

The techniques presented in this paper are as well motivated by concepts similar to the compressed sensing principle. The main difference with respect to known results is the proposition of a coding scheme on a finite (as opposed to the real) field. This is motivated by the way effective network codes are implemented, and by the fact that the measures are usually quantized before transmission, factors precluding the exploitation of compressive sensing methods as such.

© EURASIP, 2012 - ISSN 2076-1465

The rest of the paper is organized as follows. Section 2.1 presents the considered signal model and the data transmission paradigm. Section 3 describes the reconstruction algorithm at the sink. Section 4 introduces a routing algorithm whose application allows to obtain sparse transfer matrices, well suited for the convergence of the decoding algorithm. Section 5 concludes the paper, presenting simulation results.

2. MEASUREMENT COLLECTION AND TRANSMISSION

2.1. Signal model

We consider a network composed by n sensors, randomly spread across a designated area $\mathcal{A}$. Each sensor locally measures some physical quantity. All the measurements need to be recovered at the sink node, k. The n measurements are modeled as the realization of a random vector $\boldsymbol{S} \sim \mathcal{N}(\boldsymbol{0}, \Sigma)$, with $\sigma_i^2 = 1, \forall i \in \{1, \cdots, n\}$. The statistical dependence between S_i and S_j is completely described by the correlation coefficient $\rho_{ij} = \mathbb{E}[S_i S_j]$. Upon sensing, each node applies to the measure the same q-level (with q prime) scalar quantizer, $Q : \mathbb{R} \to \mathrm{GF}(q)$. Let $\mathcal{I}_a$ denote the interval in $\mathbb{R}$ mapped to the index a by Q. The probability mass function (pmf) of the quantized measure $X_i = Q(S_i)$ is easily obtained as $p_{X_i}(a) = \int_{\mathcal{I}_a} f_{S_i}(s)\,\mathrm{d}s$. The joint and conditional pmfs are obtained similarly. In the following, the conditional pmf $p_{X_j|X_i}(b|a)$ will be often represented by the transition matrix $\mathbb{P}^{(j|i)}_{[q\times q]}$.

2.2. Data transmission

The nodes collaborate to rely the packets containing quantized measurements to the sink. This is achieved via linear network coding [14]. At each transmission instant, the i-th node computes the linear combination of its source packet and of the incoming coded packets via random coefficients drawn on $\mathrm{GF}(q)$. The i-th coded packet is hence broadcast, along with the header containing the coefficients of the linear combination. The sink is equipped with a buffer, which stores, at each time instant, the m coded packets received since the beginning of the transmission, along with their headers. Equivalently, the sink node k observes the vector $\boldsymbol{y} = [y_1, \cdots, y_m]^T$ of coded packets, evaluated as the projection of the measurement vector $\boldsymbol{x} = [x_1, \cdots, x_n]^T$ on the random matrix $\mathrm{A} \in \mathrm{GF}(q)^{m\times n}$ representing the network coding operation

$$\boldsymbol{y} = \mathrm{A}\boldsymbol{x}, \qquad y_h = \sum_{j=1}^{n} \alpha_{hj}\, x_j. \tag{1}$$

3. RECONSTRUCTION OF THE SOURCE PACKETS

The sink aims to reconstruct the vector $\boldsymbol{x}$ of the source measures, upon observation of $\boldsymbol{y}$ and A. Perfect recovery of $\boldsymbol{x}$ is possible whenever the transfer matrix A in (1) has rank n [15]. We exploit the knowledge of the correlation structure of $\boldsymbol{X}$ to devise a reconstruction algorithm for $\mathrm{rank}(\mathrm{A}) < n$, which provide an *estimate* $\widehat{\boldsymbol{x}}$ of the measurement vector.

3.1. MAP estimation

The sink observes the received vector $\boldsymbol{y}$ and the rank-deficient transfer matrix A, and has knowledge of the joint pmf $p_{\boldsymbol{X}}(\boldsymbol{x})$, which we assume estimated during the establishment of the connection between the nodes in the WSN. The reconstruction $\widehat{\boldsymbol{x}}$ is computed as the maximum *a posteriori* (MAP) estimate of $\boldsymbol{X}$ upon observation of $\boldsymbol{y}$ in (1)

$$\widehat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}}\ p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \arg\max_{\boldsymbol{x}}\ p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}), \tag{2}$$

where $p_{\boldsymbol{X}}(\boldsymbol{x})$ and $p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})$ are, respectively, the marginal and the conditional pmfs of the vector $\boldsymbol{X}$. One has

$$\mathbb{P}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{y} = A\boldsymbol{x} \\ 0 & \text{else} \end{cases}. \tag{3}$$

Using (3) with (2) gives

$$\widehat{\boldsymbol{x}} = \arg \max_{\boldsymbol{x}\in\mathcal{X}(\boldsymbol{y})} p_{\boldsymbol{X}}(\boldsymbol{x}), \tag{4}$$

$$\text{with} \quad \mathcal{X}\left(\boldsymbol{y}\right) = \{\boldsymbol{x} \in \mathrm{GF}(q)^n \mid \boldsymbol{y} = A\boldsymbol{x}\}\,. \tag{5}$$

Example 1 illustrates the behavior of the MAP estimator (4).

Example 1. *Consider a WSN of $n = 3$ nodes. Data are quantized on $q = 5$ levels. Assume that the joint pmf factorizes as $p_{\boldsymbol{X}} = p_{X_2|X_1}(x_2|x_1)p_{X_3|X_2}(x_3|x_2)p_{X_1}(x_1)$, where $p_{X_1}(x_1)$ is uniform and*

$$\mathbb{P}^{(2|1)} = \mathbb{P}^{(3|2)} = \mathrm{toeplitz}(0.8, 0.1, 0, 0, 0.1).$$

Assume further that

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \ \textit{and}\ \mathbf{y} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

The set $\mathcal{X}\left(\boldsymbol{y}\right)$ of compatible measurement vectors is given by

$$\mathcal{X}(\boldsymbol{y}) = \{(0,1,2)^T, (1,0,3)^T, (2,4,4)^T, (3,3,0)^T, (4,2,1)^T\}.$$

The probability $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$ is maximum for $\widehat{\boldsymbol{x}} = (0, 1, 2)^T$.

The MAP estimation (4) performed via explicit enumeration of all elements of $\mathcal{X}\left(\boldsymbol{y}\right)$ is only tractable when n, m, and q are very small, which is not verified in large sensor networks.

3.2. Approximate MAP estimation via belief propagation

We apply the belief propagation (BP) algorithm on factor graphs to help solve the estimation problem described in the previous section. First we represent the joint distribution

$f_{\boldsymbol{S}}(\boldsymbol{s})$ with a directed acyclic graph (DAG) [16]. To avoid directed cycles, we define an increasing well-numbering[1] of the variables, *i.e.* we choose to represent the directed graphical model associated to the following factorization

$$f_{\boldsymbol{S}}(\boldsymbol{s}) = f_{S_1}(s_1)\prod_{i=1}^{n} f_{S_i|S_{i-1},\cdots,S_1}(s_i|s_{i-1},\cdots,s_1). \qquad (6)$$

Define the partial correlation coefficient $\rho_{ij|\boldsymbol{v}}$, with $\boldsymbol{v} = \{1,\cdots,n\} \setminus \{i,j\}$, as the correlation coefficient associated to $f_{S_i,S_j|\boldsymbol{V}}(s_i,s_j|\boldsymbol{v})$. Since the vector $\boldsymbol{S}$ reflects a spatial correlation structure, several partial correlation coefficients are vanishing. This expresses the fact that measurements at nodes which are physically distant in the network can be considered independent, provided that measurements at intermediate nodes are known. This, for the directed Markov property [16], results in missing edges in the DAG associated to (6), which exhibits a sparse structure. We assume that the selection of the DAG representation of (6) is performed by the sink during the establishment of the connection in the WSN, *e.g.* using the techniques presented in [17].

Since the vector $\boldsymbol{X}$ is derived by quantization of $\boldsymbol{S}$, the statistical structure of $p_{\boldsymbol{X}}(\boldsymbol{x})$ is described by the DAG underlying $f_{\boldsymbol{S}}(\boldsymbol{s})$. Before the decoding operation the sink expands the graphical model incorporating the observed variables $\boldsymbol{Y}$. The observation Y_i is independent on all the other variables in the graph, conditionally to the measurements involved in the linear combination (indicated by the i-th line of A), which are connected to it with outgoing edges. The resulting DAG corresponds to the factorization of the joint pmf $p_{\boldsymbol{Z}}(\boldsymbol{z})$

$$p_{\boldsymbol{Z}}(x_1,\cdots,y_m) = \prod_{i=1}^{n}\prod_{j=1}^{m} p(x_i|\mathrm{pa}(X_i))\; p(y_j|\mathrm{pa}(Y_j)), \quad (7)$$

where $\mathrm{pa}(X_i)$ is the set of realizations of the parent nodes of the variable X_i.

The factor graph associated to the DAG is derived, as described in [18], by introducing *factor nodes* connecting *variable nodes*. The j-th factor node is associated with a function $g_j(\overline{\boldsymbol{z}}_j)$, whose vector of arguments $\overline{\boldsymbol{z}}_j$ is composed by the variables connected by incident edges, which represents one of the local conditional distributions in (7). The factors accounting for the linear constraints in A are evaluated as $p_{Y_j|\mathrm{pa}(Y_j)}(y|\mathrm{A}_{[j,:]}\boldsymbol{x}) = \delta(y - \mathrm{A}_{[j,:]}\boldsymbol{x}) = \delta(y - y_j)$, where y_j is the j-th packet received by the sink.

Example 2. *A very simple example of the factor graph obtained for the linear mixing estimation problem is depicted in Figure 1. The DAG of the system of random variables is depicted on the left. The factor graph, on the right, connects the factor nodes (depicted by a square) corresponding to the pmf factorization* (7) *with the correspondent variable nodes.*

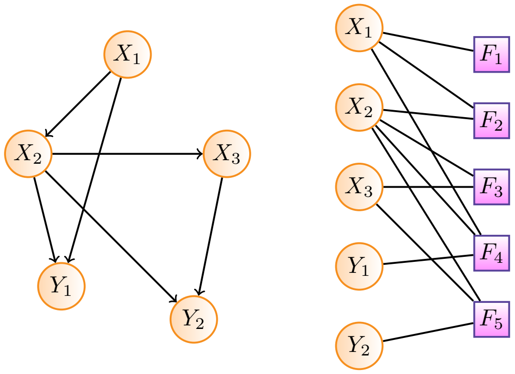

Fig. 1. The DAG and the factor graph associated to the simple network in Example 1.

The BP algorithm allows to marginalize the joint pmf (7): we are interested, in particular, to evaluate $p_{\boldsymbol{X}}(\boldsymbol{x})$, in order to solve the optimization problem (4). The detailed algorithm can be found in [18]. BP provides only an approximate solution when the factor graph contains cycles: nevertheless, loopy BP proves very effective in many cases, *e.g.* in the decoding of LDPC and turbo codes [19]. In order to achieve convergence, high sparsity of the factor graph is required, which in general is not true for typical transfer matrices resulting from linear network coding. The following section describes an appropriate routing strategy yielding sparse transfer matrices A, and hence sparse factor graphs.

4. A SPARSE-TRANSFER ROUTING ALGORITHM

The density η of the network is given by the ratio between the number n of the devices and the surface of $\mathcal{A}$. In order to account for the effect of signal attenuation and interference, we assume that the message broadcast by node i can be received only by all nodes lying in the circle of radius $\theta(\eta)$ centered in i. The parameter $\theta(\eta)$ is chosen to guarantee that the probability of one node to be disconnected from the rest of the network is below 0.05.

The nodes exchange linear network-coded packets, as described in Section 2.1. The routing algorithm starts at instant t_0, when each node broadcasts their coded packet (at this moment containing only the source measurement) to all the nodes within distance $\theta(\eta)$. After a period has elapsed, at instant t_1, all nodes have computed the adjourned linear combinations, and the new packets are again broadcast. The packets are transmitted using non-orthogonal radio resources. The parameter $p_{\mathrm{loss}}(\eta)$ models the probability that a packet is not received by the destination node in the range due to radio collision. The algorithm fixes at T the maximum number of time periods during which coded packets relative to the same set of measurements are circulating in the network.

The sparsity of the transfer matrix is obtained trying to

[1] It can be proven that the definition of the directed graphical model for jointly Gaussian variables does not depend on the choice of the well-numbering, which is not unique, see [17] and references therein.

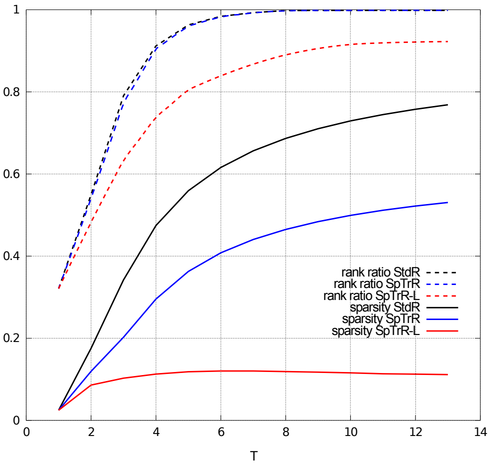

Fig. 2. Performance of the sparse-transfer routing algorithm.

reduce the degree (*i.e.* the number of measurements involved in each linear combination) of the coded packets by partial decoding at the intermediate nodes. Partial decoding works as follows. If the value of a measurement can be deduced by the linear combinations received by a node (for example because packets are formed by a single element, as it is frequent during the first steps of the routing algorithm, or because all but one measures in one combination have already been locally decoded) the node stores its value. Incoming packets are inspected for known elements, which are purged from the outgoing packet. Local decoding happens only after the computation of the outgoing packet, to ensure that locally decoded measures have been forwarded at least once by the node. Since partial decoding locally removes packets already circulating in the network, it allows, after T periods have elapsed, to increase the sparsity of the transfer matrix A with respect to standard (wired) network coding, without affecting its rank.

The algorithm, moreover, fixes at L the maximum degree of coded packets. If the degree of the combination of incoming packets (after removal of known measures) exceeds L, incoming packets are randomly discarded until the degree constraint is met. Discarded packets are not locally decoded, so that they will be forwarded upon later reception. Appropriate tuning of the parameter L allows to attain the target level of sparsity in the transfer matrix A. The rank of the matrix, however, is in general decreased with respect to standard routing. This effect is more severe as the parameter L gets smaller.

Figure 2 shows the simulation performance of the sparse-transfer routing algorithm without (SpTrR) and with limitation L of the degree (SpTrR-L), in comparison to the performance of the standard (wired) network coding algorithm (StdR). The results are obtained for parameters $n = 40$, $q = 61$, $\eta = 19.417$, $\theta(\eta) = 0.679$, $L = 18$ and $p_{\text{loss}} = 0.3$. The dashed lines in Figure 2 depict the normalized rank $\text{rank}(\mathrm{A})/n$ of the transfer matrix as a function of the maximum number of periods T. As expected, the SpTrR algorithm performs very closely to the StdR algorithm, with a very small loss due to the effects of packet collisions, while the SpTrR-L algorithm produces rank-deficient matrix A. The solid lines compare the sparsity of A as a function of T, obtained in the three cases. The SpTrR algorithm allows an appreciable increase of the sparsity with respect to the StdR algorithm, without reduction of the rank. The SpTrR-L algorithm, however, allows to match the targeted sparsity 0.1, at the expense of moderate reduction of the rank of A.

5. SIMULATION RESULTS

In what follows, a WSN consisting of $n = 40$ sensor nodes is considered. Each sensor i generates samples which are realizations of zero-mean unit-variance Gaussian variables S_i, $i = 1, \dots, n$. A simple correlation model is considered, namely

$$\mathbb{E}\left[S_i S_j\right] = \begin{cases} \rho & \text{if } |i-j| = 1 \\ 0 & \text{else.} \end{cases} \qquad (8)$$

The S_i are quantized using a q-level scalar quantizer.

Network coding is assumed to be performed in such a way that A contains in average about $\gamma n m$ non-zero random entries in $\mathrm{GF}(q)$, and that no column in A is the zero vector. This ensures that all measurements have been taken into account, either directly, or in a network-coded packet. Performance evaluations are done as a function of the number of received packets. Each point results from 300 realization of the source samples and of the network coding matrix. The maximum number of iterations of the BP algorithm is set to $N_{\max} = 20$.

The performance is evaluated in terms of *error rate*, corresponding to the proportion of erroneously estimated quantized samples x_i and in terms of reconstruction signal-to-noise ratio

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{\sum_{i=1}^{n} \mathbb{E}\left[S_i^2\right]}{\sum_{i=1}^{n} \mathbb{E}\left[\left(S_i - \widehat{S}_i\right)^2\right]} \right),$$

where $\widehat{S}_i$ obtained from $\widehat{\boldsymbol{x}}$ after inverse quantization.

In Figures 3 and 4, $\rho = 0.995$ and $\gamma = 0.05$. One sees that the probability of error gracefully decreases when m increases. Similarly, the SNR increases to reach a maximum when m is between 20 (when q is small) and 25 (for larger values of q). The reception of a number of packets about half of the number of sensors already allows a good reconstruction quality.

Figures 5 and 6 illustrate now the impact of the correlation between data measured by the sensors. The larger ρ, the more efficient the estimation. Here, $\gamma = 0.05$ and $q = 17$.

6. CONCLUSION

This paper shows that network coding of correlated measures may be used to perform lossy source compression, to effi-

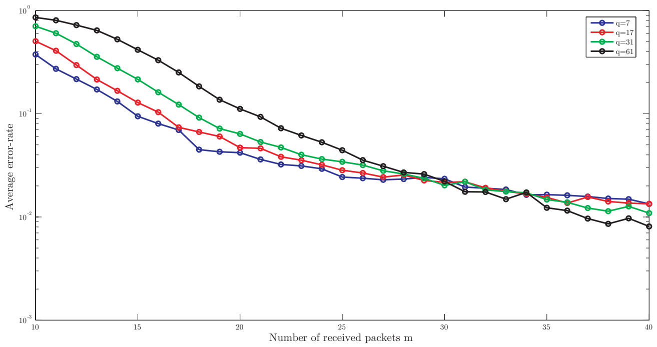

Fig. 3. Probability of reconstruction error as a function of the number m of received packets for various field size

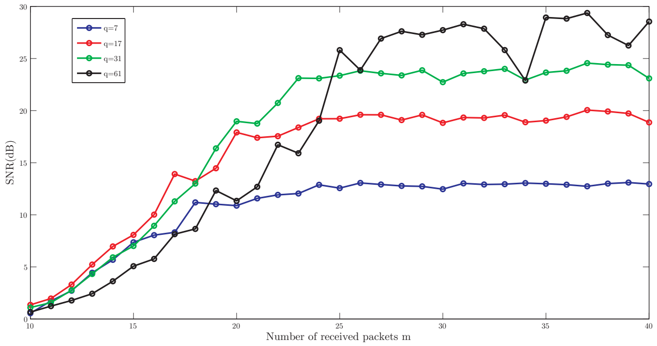

Fig. 4. SNR as a function of the number m of received packets for various field size

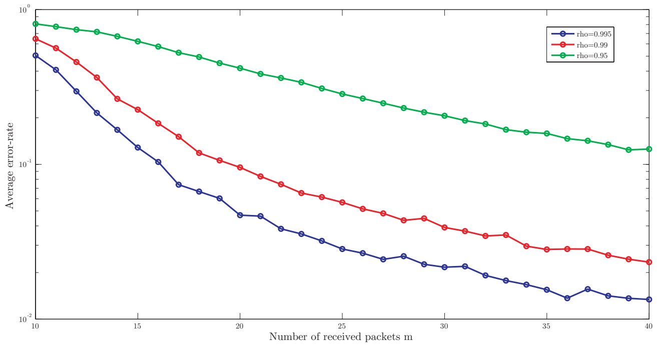

Fig. 5. Probability of reconstruction error as a function of the number m of received packets for different values of the correlation coefficient ρ

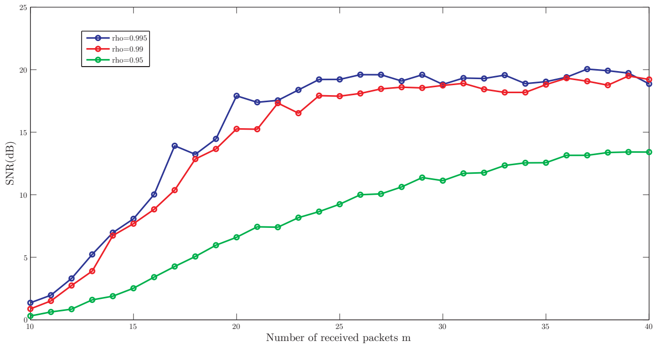

Fig. 6. SNR as a function of the number m of received packets for different values of the correlation coefficient ρ

ciently collect data at the sink in a WSN, even in case of rank deficient transfer matrix. Contrary to previous works, network codes over finite fields have been considered. The reconstruction of the source packets at the sink is obtained via approximate MAP estimation, performed using belief propagation on factor graphs. The presence of a statistical structure among the source measurements allows to compensate for the missing packets. A variant of random network coding is proposed, to allow the transfer matrix to achieve a target level of sparsity. This facilitates the convergence of the BP algorithm used for reconstruction.

7. REFERENCES

[1] J. N. Al-Karaki and A. E. Kamal, "Routing in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6 – 28, 2004.

[2] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. on Inf. Theory*, vol. 52, pp. 4413–4430, 2006.

[3] G. Caire, S. Shamai, and S. Verdu, *Noiseless Data Compression with Low-Density Parity-Check Codes*. DIMACS, Advances in network information theory, 2004.

[4] H. Chen, "Distributed file sharing: Network coding meets compressed sensing," in *Proc. ChinaCom '06*, pp. 1 –5.

[5] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inf. Theory*, vol. 19, pp. 471–480, 1973.

[6] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: theory, algorithms, and scaling laws," *IEEE Trans. on Inf. Theory,*, vol. 51, pp. 4057 – 4073, 2005.

[7] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 21, pp. 21 – 30, 2008.

[8] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *Signal Processing Magazine, IEEE*, vol. 25, pp. 92 –101, march 2008.

[9] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *Proc. MobiHoc*, 2005, pp. 402–413.

[10] A. Oka and L. Lampe, "Compressed sensing of Gauss-Markov random field with wireless sensor networks," in *Proc. SAM 2008*, pp. 257 –260.

[11] S. Shintre, S. Katti, and S. Jaggi, "Real and complex network codes: Promises and challenges," in *Proc. NetCod 2008*, pp. 1–6.

[12] N. Nguyen, D. Jones, and S. Krishnamurthy, "Netcompress: Coupling network coding and compressed sensing for efficient data communication in wireless sensor networks," in *Proc. IEEE Workshop on Signal Processing Systems*, vol. 6, 2010, pp. 356–361.

[13] C. Luo, J. Sun, and F. Wu, "Compressive network coding for approximate data gathering," in *Proc. GLOBECOM 2011*, 2011, pp. 1–6.

[14] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE Trans. on Networking*, vol. 2, no. 5, pp. 782 – 795, 2003.

[15] N. J. A. Harvey, D. A. Karger, and K. Murota, "Deterministic network coding by matrix completion," in *Proc. of SODA*, 2005.

[16] S. Lauritzen, *Graphical models*. Oxford, 1996.

[17] M. Drton and M. D. Perlman, "Multiple testing and error control in Gaussian graphical model selection," *Statistical Science*, vol. 22, no. 3, pp. 430 – 449, 2007.

[18] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Inf. Theoy*, vol. 47, no. 2, pp. 498 – 519, 2001.

[19] K. Murphy, Y. Weiss, and M. Jordan, "Loopy belief propagation for approximate inference: an empirical study," in *Proc. of UAI*, 1999.