Compressive Data Persistence in Large-Scale Wireless Sensor Networks

Mu Lin*, Chong Luo†‡, Feng Liu* and Feng Wu‡
*School of Electronic and Information Engineering, Beihang University, Beijing, P.R.China.
†Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, P.R.China
‡Microsoft Research Asia, Beijing, P.R.China. Email: {cluo, fengwu}@microsoft.com

***Abstract*—This paper considers a large-scale wireless sensor network where sensor readings are occasionally collected by a mobile sink, and sensor nodes are responsible for temporarily storing their own readings in an energy-efficient and storage-efficient way. Existing data persistence schemes based on erasure codes do not utilize the correlation between sensor data, and their decoding ratio is always larger than one. Motivated by the emerging compressive sensing theory, we propose compressive data persistence which simultaneously achieves data compression and data persistence. In the development of compressive data persistence scheme, we design a distributed compressive sensing encoding approach based on Metropolis-Hastings random walk. When the maximum step of random walk is 400, our proposed scheme can achieve a decoding ratio of 0.36 for 10%-sparse data. We also compare our scheme with a state-of-the-art Fountain code based scheme. Simulation shows that our scheme can significantly reduce the decoding ratio by up to 63%.**

I. Introduction

This paper considers a large-scale wireless sensor network where sensors are deployed in harsh environment and there is no static powerful sink deployed. Sensors periodically generate readings, but these readings have to be saved within the network until a mobile sink visits and gathers them. Since sensors are energy-constrained and prone to failures, it is desired that sensor readings are stored with redundancy, so that the sink is able to reconstruct the readings even if a large portion of sensor nodes cease to function. This problem is known as the data persistence problem.

The data persistence problem is essentially equivalent to the reliable data transmission problem in an erasure channel, and therefore can be addressed through erasure codes. For a source message comprising of m symbols, an erasure code generates $n(n > m)$ encoded symbols, such that the original symbols can be reconstructed from a subset of the encoded symbols. The ratio of m to n is defined as the code rate. Denote m' as the number of encoded symbols required for successful decoding, then the ratio m'/m indicates decoding efficiency.

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 60872011 and 60933012) , the National Science and Technology Major Project of China (Grant No. 2009ZX03006-001-003), the State Key Development Program for Basic Research of China (Grant No. 2010CB731800), the Program for New Century Excellent Talents in University, and the Fundamental Research Funds for the Central Universities.

Reed-Solomon codes and low-density parity-check (LDPC) codes have been widely used for data persistence in distributed data storage [1][2]. Usually, source data are encoded in a centralized location and then the encoded symbols are distributed to different machines for storage. However, in a wireless sensor network, data sources are distributed. It is not practical to designate an energy-constrained sensor node to perform centralized encoding, not to mention about the communication cost to transmit all sensor readings to the rendezvous point. In order to achieve data persistence in a wireless sensor network, it is desired that encoding of the erasure code can be implemented in a distributed way.

Random linear network code, as the name suggests, encodes the source data with linear operations while they flow in the network. It has been successfully applied in wireless sensor networks to improve the degree of fault tolerance [3][4][5]. However, the decoding complexity of random linear network code is $O(m^3)$, which consumes a huge amount of computing resources when the scale of sensor network is large. Digital Fountain codes [6], a.k.a. rateless erasure codes, then arise as a low-complexity alternative [7][8]. The decoding complexity of Fountain codes is only $O(m \log m)$, and the encoding can be achieved distributedly because the encoded symbols are independent from each other. In particular, Dimakis et al. [7] assume that each node knows its location and uses geographic routing to disseminate the source data. Lin et al. [8] relax this assumption and significantly reduce the routing control overhead by using random walk to disseminate source data.

These previous works based on erasure codes all neglect an important fact that sensor readings are correlated data instead of independent ones [9]. If data correlation is taken into account, the original m symbols can be potentially reconstructed from less number of encoded symbols, i.e. the decoding ratio m'/m could be less than one. Motivated by the compressive sensing (CS) theory [10][11] and its recent development in Bayesian CS [12], we propose compressive data persistence (CDP) which simultaneously achieves sensor data compression and data persistence. The original sensor readings are routed to the storage nodes through Metropolis-Hastings random walk. Then, each encoded symbol, or so called *CS measurements* in our scheme, is distributedly computed at storage nodes. With our proposed scheme, sensor data reconstruction can be achieved at a decoding ratio far less than one.

The rest of this paper is organized as follows: Section II provides background on compressive sensing theory and reviews related work on distributed compressive sensing. Section III describes the proposed compressive data persistence scheme. Section IV validates the design choices in CDP and demonstrates its effectiveness through simulations. Finally, section V concludes the paper.

II. Background and Related Work

A. Background of compressive sensing

Compressive sensing theory concerns the representation and reconstruction of sparse signals. An m-dimensional signal $\mathbf{x}$ is called an s-sparse signal if it has only s non-zero entries, or can be represented by s non-zero coefficients in a transform domain Ψ. The intuition behind compressive sensing is that a small number of linear projections (or so called measurements) of a sparse signal contains adequate information for its reconstruction. Mathematically, let $\mathbf{x} = [x_1 x_2 \ldots x_m]^T$ be an s-sparse signal. Let us take m' measurements of $\mathbf{x}$ through linear projection:

$$\mathbf{y} = \Phi \mathbf{x} \tag{1}$$

CS theory states that the m-dimensional signal $\mathbf{x}$ can be perfectly reconstructed from m'-dimensional ($m' < m$) measurement $\mathbf{y}$ under certain conditions [10][11]. The central problem in CS is how measurement matrix Φ should be designed and what algorithm should be used to recover $\mathbf{x}$ from the underdetermined linear system defined in (1).

In the early stage of CS theory development, a dense matrix whose entries are random variables drawn from i.i.d. Gaussian distribution $\mathcal{N}(0, \frac{1}{m'})$ is often used as the measurement matrix. Reconstruction of $\mathbf{x}$ from measurement $\mathbf{y}$ can be achieved through solving the following l_1-minimization problem by linear programming:

$$(P1) \quad \min \|\theta\|_{l_1}, \quad s.t. \quad \mathbf{y} = \Phi \mathbf{x},\ \mathbf{x} = \Psi \theta \tag{2}$$

Recently, several papers have reported the consideration of CS reconstruction from Bayesian inference if the statistical characterization of the signal is available [12][13]. This essentially bridges CS with LDPC codes, although the operation in CS is arithmetic plus instead of logical exclusive or (XOR). Due to the complexity in belief propagation, CS reconstruction by Bayesian inference can be practically implemented only if Φ is a low-density matrix. Baron et al. [12] propose to use a low-density measurement matrix whose elements are drawn from $\{0, 1, -1\}$. It is reported that the decoding complexity is on the order of $O(m \log m)$.

B. Distributed compressive sensing

Compressive sensing has the potential to be used for data persistence in wireless sensor network because sensor data are usually correlated. In particular, spatially correlated sensor data have been shown to be sparse in wavelet domain [14][15]. Baron et al. [16], Bajwa et al. [17], and Luo et al. [18] have reported the efficiency of distributed CS in wireless sensor data gathering when a sink is constantly available. In this paper, we are interested in the case where sensors continuously generate data but the sink visits and collects data only occasionally. In such a network, the main challenge is how to preserve sensor data by sensor nodes in an energy-efficient and storage-efficient way.

Rabbat et al. [19] realize distributed CS for data persistence through randomized gossiping. In particular, in order to generate the i^{th} measurement y_i, each sensor node multiplies its reading x_j with a random coefficient ϕ_{ij} and generates its initial message $u_{ij}^{(0)}$. Then, sensor nodes start random gossiping. Suppose, at time step $t+1$, node j receives a message from its neighbor k, denoted as $u_{ik}^{(t)}$, it then updates its own message as $u_{ij}^{(t+1)} = \frac{1}{2}(u_{ij}^{(t)} + u_{ik}^{(t)})$. It has been proved that, after sufficient time steps T, messages at all sensor nodes will converge to the same value which is equivalent to the average of all the initial values. Mathematically,

$$\begin{aligned} & u_{i1}^{(T)} = u_{i2}^{(T)} = ... = u_{im}^{(T)} \\ = \ & \frac{1}{m}(\phi_{i1}x_1 + \phi_{i2}x_2 + ...\phi_{im}x_m) = \frac{1}{m}y_i \end{aligned} \tag{3}$$

Repeat this process for m' times, then all the sensors will have a copy of each of the CS measurements. Therefore, by visiting any of the sensor nodes, the sink is able to obtain m' CS measurements and reconstruct the original data. In order to save the storage, each sensor can choose to store only a subset of the measurements, and the sink can collect all the measurements by visiting a few sensors. This approach consumes huge computation and communication resources for two reasons. First, because a dense measurement matrix is employed, every sensor node is involved in generating all the m' measurements. Second, random gossiping requires considerable amount of message exchange and takes a long time to converge.

Wang et al. [20] propose to employ a low-density measurement matrix for CS encoding. In addition, the measurements are generated in a controlled instead of randomized manner. To compute one CS measurement, every sensor j locally generates a random variable ϕ_{ij} . If the variable is zero, the sensor does nothing. If it's nonzero, the sensor sends the product of $\phi_{ij} x_j$ to storage node i. The complexity in this approach mainly resides in precise routing in multi-hop ad hoc network. As we know, there is usually no stable routing structure in a large-scale sensor network. Transmitting data to a specified node needs sensor coordination and incurs considerable amount of control overhead.

III. Compressive Data Persistence

We propose compressive data persistence (CDP) based on Bayesian CS and random walk. We consider a wireless sensor network with m sensing nodes and n storage nodes. For simplicity, the sensing nodes are only responsible for generating sensor readings, and the storage nodes are responsible for distributed data encoding and storage. Our proposed CDP is composed of three phases. In the data distribution phase, sensor readings are sent out by sensor nodes, and being distributed among storage nodes through random walk. Then, sensor readings are encoded and the generated CS measurement is stored at the corresponding storage node. In the last phase, the sink visits a subset of the storage nodes to collect m' measurements which are sufficient for reconstructing original sensor data.

A. Data distribution through random walk

CDP intends to generate CS measurements using a low-density matrix Φ. If each storage node generates one CS measurement, Φ is an $n \times m$ matrix. We follow the matrix construction used by Baron et al. [12]. In each row of Φ matrix, there are only l non-zero entries uniformly drawn from $\{-1, +1\}$. The value of l is also referred to as row weight of Φ. Correspondingly, the column weight of Φ is denoted by r. When the row and column weights are both constant, they have the following relationship:

$$mr = nl \tag{4}$$

The column weight r indicates the number of measurements that a particular sensor reading contributes to. Therefore, in the data distribution stage, every sensor node injects r copies of its reading to the network. This is achieved by transmitting its reading to a randomly picked storage node within its communication range, and repeat this process for r times. Then, sensor readings are disseminated among storage nodes through random walk.

Random walk proceeds in steps. For a particular copy of a sensor reading, denote $X(t) = n_i$ as its position after t steps of random walk. In $(t+1)^{th}$ step, node n_i randomly picks one neighbor storage node and forwards the data. A random walk can be modeled as a time-reversible Markov chain, and be characterized by a transition matrix P. Each entry P_{ij} in the matrix indicates the probability that node n_i forwards the data to node n_j. The design of the transition matrix P is associated with the desired equilibrium distribution $\pi = (\pi_1, \pi_2 ... \pi_n)$. In CDP, since every storage node is expected to receive the same number of sensor readings, the equilibrium distribution is a uniform distribution. Then, the entry P_{ij} in the transition matrix can be written into the following simplified form when the Metropolis-Hastings algorithm is adopted [21].

$$P_{ij} = \begin{cases} \min\{1/d_i, 1/d_j\} & \text{if } (i,j) \in \mathcal{E}, i \neq j \\ 1 - \sum_{(i,k)\in\mathcal{E}} \min\{1/d_i, 1/d_k\} & \text{if } i = j \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases} \tag{5}$$

where d_i is the degree of node i in the connectivity graph, and $\mathcal{E}$ is the set of communication links.

Let $\pi(t)$ be the probability distribution of the state after step t, the state distribution satisfies $\pi(t+1) = \pi(t)P$. It has been shown [21] that the total variation distance of $\pi(t)$ and the uniform distribution is bounded by:

$$\sup \left\| \pi(t) - \frac{1}{n} \cdot \mathbf{1} \right\|_{tv} = \frac{1}{2} \max_i \sum_j \left| P_{ij}^t - \frac{1}{n} \right| \leq \frac{1}{2} \sqrt{n} \mu^t \tag{6}$$

where μ is the second largest eigenvalue modulus of P.

In CDP, the data distribution phase stops after T random walk steps. The selection of T is an engineering choice, which should strike a tradeoff between communication cost and the quality of state distribution. Denote l_i as the number of sensor readings received by storage node n_i at the end of data distribution phase. Then the mean of l_i, $i = 1, 2...n$ is l, but each individual value may not be equal to l due to the randomization in random walks.

B. CS encoding and reconstruction

In CS encoding phase, each storage node generates one CS measurement. Let $\mathbf{x}^i$ denote the measurements received at node i. Then, measurement y_i is generated by:

$$y_i = \langle \phi_i, \mathbf{x}^i \rangle \tag{7}$$

where elements of ϕ_i are drawn from set $\{-1, +1\}$ with equal probability.

In the third phase of CDP, the sink visits the network and collects CS measurements from a subset of storage nodes. The number of storage nodes that the sink should visit, denoted by m' is determined by the sparsity of sensor data. The more sparse the data are, the less number of measurements are required for perfect reconstruction. In addition, the size of m' is also affected by the state distribution of random walk. All these factors will be evaluated in the next section.

When the sink has collected sufficient number of measurements, it can reconstruct the original sensor readings through belief propagation (BP) algorithm introduced by Baron et al. [12]. The BP reconstruction resembles the BP decoding for LDPC codes. As CS measurements are obtained by arithmetic plus instead of XOR, the belief messages are represented by a probability distribution function instead of a single log likelihood ratio (LLR). Despite of this additional complexity, the BP process for CS reconstruction also has the potential to be sped up through parallelization.

IV. Evaluation

We validate the design choices of the proposed CDP and evaluate its performance through Matlab simulations. A total of 3000 nodes are uniformly distributed in a unit disc, in which 1000 are sensor nodes and 2000 are storage nodes ($m = 1000$ and $n = 2000$). Two nodes whose Euclidean distance are within 0.1 are able to directly communicate with each other. Therefore, the diameter of the network is around 10 hops. Without loss of generality, we assume binary sensor readings in our simulation.

A. Design choices of CDP

1) Low-density CS measurement matrix: In the proposed CDP, we adopt low-density measurement matrix Φ for CS encoding. The row weight l is an important design parameter. According to (4), when the sizes of m and n are fixed, the number of duplicate copies of each sensor reading is proportional to the size of l. Therefore, selecting a small l can reduce the overall communication cost. However, a too small l may not be sufficient to capture source information and may result in low reconstruction accuracy.

Fig. 1 plots the bit error ratio (BER) of CS reconstruction when different measurement matrices are used. In this simulation, a 10%-sparse binary source with length $m = 1000$ is considered, i.e. the probability of each bit being a "1" is 10%. When a dense random Gaussian matrix is used, CS decoding is performed through l_1-minimization. For all the three low-density measurement matrices, we use belief propagation algorithm for decoding. Surprisingly, using low-density measurement matrix does not cause performance loss, in the case of binary sources. The reconstruction BER decreases as the row weight l increases, but it shows asymptotic behavior when l is sufficiently large. Therefore, in the rest of the simulation, we set $l = 30$. Correspondingly, in the data distribution phase of CDP, each sensor node distributes 60 copies of its reading.

2) Metropolis-Hastings chain: The efficiency of a random walk algorithm can be characterized by the *mixing time*, which is defined as the minimal length of a random walk to approximate the steady-state distribution within a certain error. Several simple algorithms have been proposed to achieve fast mixing [21]. Lin et al. [8] adopt the maximum-degree chain by assigning the following transition probabilities:

$$P_{ij}^{md} = \begin{cases} \min\{1, \pi_j/\pi_i\}/D_m & \text{if } (i,j) \in \mathcal{E}, i \neq j \\ 1 - \sum_{(i,k)\in\mathcal{E}} P_{ik} & \text{if } i = j \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases} \quad (8)$$

where D_m is the maximum node degree in the entire network.

It has been shown in (6) that the total variation distance between state distribution and uniform distribution after t steps of random walk can be bounded by $\frac{1}{2}\sqrt{n}\mu^t$. We compare the bounds of the maximum-degree chain used by Lin et al. [8] and the Metropolis-Hastings chain adopted in CDP. Fig. 2 clearly shows that the state distribution of Metropolis-Hastings chain converges to uniform distribution more quickly than the maximum-degree chain. In addition, using Metropolis-Hastings chain does not require the global information of D_m and will simplify the implementation.

B. Performance

1) Results for 10%-sparse data: In the first set of experiments, the sensor readings are 10%-sparse. In the first phase of CDP, they are injected into the network through T steps of random walk. Then storage nodes independently encode their received sensor readings. After that, m' randomly selected measurements are collected for CS decoding. In the case that certain sensor readings are not encoded in any of the m' measurements, we randomly pick more measurements until the entire set of sensor readings are covered.

Fig. 3 shows the reconstruction BER when different number of random walk steps are taken. Each data point shown in the figure is averaged over 200 test runs. It is clear that the reconstruction error decreases as the number of measurements increases. After 100 steps of random walk, the reconstruction BER drops below 10^{-4} when 360 measurements are used for decoding. The sensor data can be perfectly reconstructed from 360 measurements if 400 steps of random walk are taken during data distribution phase. This corresponds to a decoding ratio of 0.36, which is far below the decoding ratio that can be achieved by any erasure coding based schemes.

There is a tradeoff between the decoding ratio and the average energy consumption. Define unit energy as the energy consumed to transmit one sensor reading, then the energy consumption of each node equals to the number of sensor readings it transmits. For the proposed CDP, the average energy consumption is 2283.8, 4435.4, 9484.2 and 12959.0 after 100, 200, 400 and 600 steps of random walk. The average energy consumption scales not only with the number of random walk steps, but also with the number of storage nodes. Since CS based reconstruction only requires the measurements from a few hundred storage nodes, we may put a random set of storage nodes in sleep. This would further reduce the average energy consumption.

This figure also shows the reconstruction performance if every copy of sensor reading only takes 20 steps of random walk. When $T = 20$, the energy consumption is similar to Wang et al.'s scheme [20] which requires precise routing. As we have pointed out, precise routing in a large-scale ad

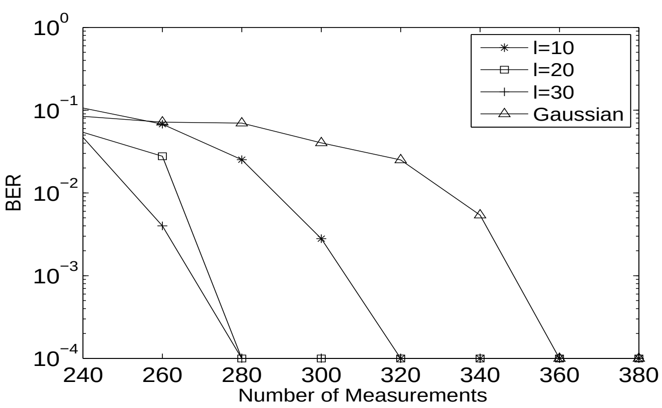

Fig. 1. Comparing CS reconstruction performance of dense measurement matrix and low-density measurement matrix with different row weights

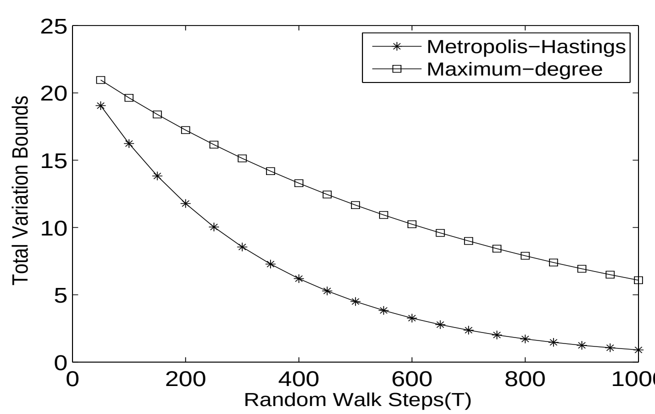

Fig. 2. Bounds of total variation distance between state distribution and uniform distribution after T steps of random walk

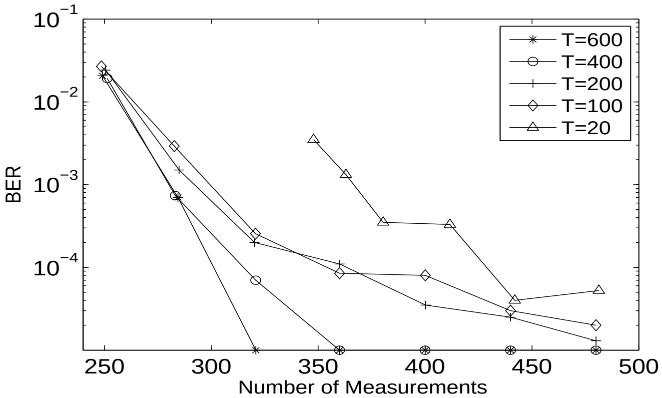

Fig. 3. BER of CS reconstruction vs. the number of storage nodes visited

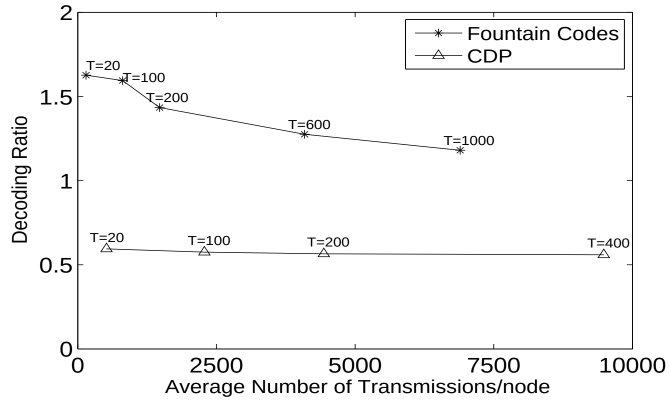

Fig. 4. Decoding ratio of CDP and Fountain code based scheme

hoc network incurs high control overhead. In contrast, our proposed CDP completely avoids this overhead. Although the reconstruction BER is less stable when $T = 20$, the performance loss can be compensated by letting the mobile sink visit a few more storage nodes.

2) Comparison with Fountain code: We compare CDP with Lin's scheme [8] which is based on digital Fountain code. We are mainly interested in the decoding ratio m'/m. The simulation is carried out on a set of non-compressible data, i.e. the probability of bit "1" in the sensor readings is 50%. Lin et al. distribute data through a maximum-degree chain, which has a longer convergence time but consumes less energy after the same number of random walk steps. In order to ensure a fair comparison, we use the average energy consumption instead of random walk steps as the x-axis in Fig. 4. As expected, the decoding ratio is always larger than 1 in Lin's scheme. Our proposed CDP can always achieve a decoding ratio below 0.6. When the random walk steps is small, the reduction in decoding ratio is up to 63%.

V. Conclusion

We have described in this paper a compressive sensing based scheme for data persistence in large-scale wireless sensor networks. The contributions are two fold. First, the proposed CDP scheme utilizes the correlation between sensor data, and therefore simultaneously achieves data compression and data persistence. Second, we implement distributed CS encoding through Metropolis-Hastings random walk. This avoids the high control overhead that a precise routing scheme may incur. We have validated the design choices and evaluated the CDP scheme through extensive simulations. Results show that CDP can reduce the decoding ratio of a state-of-the-art Fountain code based scheme by up to 63%.

References

[1] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells, and B. Zhao, "Oceanstore: an architecture for global-scale persistent storage," in *ASPLOS-IX*, (New York, NY, USA), pp. 190–201, ACM, 2000.

[2] J. Plank and M. Thomason, "A practical analysis of low-density parity-check erasure codes for wide-area storage applications," in *Proc. of Intl. Conf. on Dependable Systems and Networks*, pp. 101–110, Jan. 2004.

[3] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes," in *Proc. of IPSN*, pp. 111–117, Apr. 2005.

[4] D. Wang, Q. Zhang, and J. Liu, "Partial network coding: Theory and application for continuous sensor data collection," in *Proc. of IEEE IWQoS*, pp. 93–101, 2006.

[5] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: Maximizing sensor network data persistence," in *Proc. of ACM SIGCOMM*, pp. 255–266, Aug. 2006.

[6] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. of ACM SIGCOMM*, pp. 56–67, Oct. 1998.

[7] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Distributed fountain codes for networked storage," in *ICASSP*, vol. 5, May 2006.

[8] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," in *Proc. of IEEE Infocom*, pp. 1658–1666, May 2007.

[9] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, p. Article 9, Aug. 2007.

[10] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[11] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489 – 509, 2006.

[12] D. Baron, S. Sarvoham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Processing*, vol. 58, no. 1, pp. 269–280, 2010.

[13] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2346 –2356, 2008.

[14] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. of IPSN*, pp. 309–316, 2006.

[15] J. Aćimović, B. Beferull-Lozano, and R. Cristescu, "Adaptive distributed algorithms for power-efficient data gathering in sensor networks," in *Proc. of Intl. Conf. on Wireless Networks, Comm. and Mobile Computing*, pp. 946–951, Jun. 2005.

[16] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, *Distributed compressed sensing*. http://dsp.rice.edu/cs, 2005.

[17] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proc. of IPSN*, pp. 134–142, 2006.

[18] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *ACM Mobicom'09*, pp. 145–156, Sep. 2009.

[19] M. Rabbat, J. Haupt, A. Singh, and R. Nowak, "Decentralized compression and predistribution via random gossiping," in *Proc. of IPSN*, pp. 51–59, Apr. 2006.

[20] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *Proc. of IPSN*, pp. 331–339, April 2007.

[21] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," vol. 46, pp. 667–689, Dec. 2004.