

CStorage: Distributed Data Storage in Wireless Sensor Networks Employing Compressive Sensing

Ali Talari and Nazanin Rahnavard
Oklahoma State University, Stillwater, OK 74078,
Emails: {ali.talari, nazanin.rahnavard}@okstate.edu

Abstract—In this paper, we propose CStorage a fully distributed and efficient data storage scheme for wireless sensor networks (WSNs) based on *compressive sensing* (CS) techniques. CStorage requires much smaller number of transmissions compared to the existing algorithms by exploiting the *compressibility* of the natural signals along with the *broadcast property* of wireless channels. In CStorage, after a probabilistic readings dissemination phase, each node obtains one *compressed sample* (measurement) of the network’s readings, which are later queried in part by a data collector to recover all readings. We find the optimal parameters of CStorage and show that it considerably decreases the total number of required transmissions for distributed data storage.

I. INTRODUCTION

To increase the data persistence in wireless sensor networks (WSNs), it has been proposed to disseminate sensors readings *throughout* the network such that a data collector can query *any small subset* of nodes to obtain all sensors’ readings [1, 2]. However, existing algorithms either ignore the *correlation* and *compressibility* of the readings, require routing tables and cannot be distributively implemented, or incur a large number of transmissions.

It has been shown that signals collected in WSNs from natural phenomena are highly compressible due to their strong spacial correlation [3–5]. Therefore, such compressible signals including N readings from N network nodes can be reconstructed from only $M \ll N$ *compressed samples* (measurements) of the signal employing *compressive sensing* (CS) techniques [6, 7]. On the other hand, wireless channels have the inherent *broadcast* property, hence *probabilistic broadcasting* (PBcast) [8, 9] can be effectively utilized to disseminate nodes’ readings in the networks with minimal transmissions.

Utilizing the two aforementioned properties, we propose an efficient and flexible *cross-layer* data storage algorithm referred to as *compressive sensing data storage* (CStorage), which considerably reduces the number of required transmissions for distributed data storage in WSNs. In CStorage, when the data dissemination phase using PBcast is finished a data collector can query $M \ll N$ measurements from *any* set of M nodes and recover all N readings. CStorage is fully scalable and distributed since nodes independently make decisions without employing any routing table.

The paper is organized as follows. Section II provides a brief review on compressive sensing, PBcast, and the related

work. In Section III, we propose CStorage. In Section IV we evaluate the performance of CStorage and compare it to existing algorithms. Finally, Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

In this section, we provide the necessary background to design CStorage.

A. Compressive Sensing in WSNs

Let us consider a WSN with N nodes collecting a natural signal $\underline{x} = [x_1 x_2 \dots x_N]^T$, where x_i represents the reading of the i^{th} sensor. Due to spacial correlation of sensor readings \underline{x} may be represented by a K -sparse signal $\underline{\theta}$ with only $K \ll N$ large coefficients in some *appropriate* transform basis Ψ , where $\underline{x} = \Psi \underline{\theta}$ [4, 5, 10]. More precisely, when the coefficients of $\underline{\theta}$ are sorted based on their absolute magnitude, they decay faster than $Ci^{-\frac{1}{\beta}}$ for $0 < \beta \leq 1$ and a constant C [4, 5, 10].

Consider a natural signal \underline{x} collected from a WSN that is K -sparse in some proper basis Ψ . CS techniques are able to recover \underline{x} from only $M \ll N$ *random projections* (also called measurements or compressed samples) of \underline{x} , where $M \geq O(K \log N)$ [6, 7]. Generally, CS is composed of two following key components.

Signal Sampling: The random projections are generated by $\underline{y} = \Phi \underline{x}$, where Φ is a well-chosen $M \times N$ random matrix, called projection matrix.

Signal Recovery: Signal reconstruction can be done by finding the estimate $\hat{\underline{\theta}}$ (and accordingly $\hat{\underline{x}} = \Psi \hat{\underline{\theta}}$) from the system of linear equations $\underline{y} = \Phi \Psi \underline{\theta}$. This is an underdetermined system with infinitely many solutions. However, the knowledge of $\underline{\theta}$ being a sparse signal allows us to have a successful reconstruction w.h.p. It is shown that $\hat{\underline{\theta}}$ can be estimated via solving the ℓ_1 optimization problem given by [6, 7, 11, 12]

$$\hat{\underline{\theta}} = \underset{\underline{\theta}}{\operatorname{argmin}} \|\underline{\theta}\|_1, \text{ s.t. } \underline{y} = \Phi \Psi \underline{\theta}, \quad (1)$$

where $\|\underline{\theta}\|_1 = \sum_{i=1}^N |\theta_i|$. The ℓ_1 optimization problem (1) can be solved with linear programming techniques called *basis pursuit* (BP) [12]. Later, we employ CS in CStorage design.

B. Probabilistic Broadcasting: PBcast

Consider a WSN with N nodes *randomly* deployed over a field of size $A = 1 \times 1$. To ensure the connectivity of the network we assume all nodes have identical transmission range

of $r_t^2 > \frac{A \ln(N)}{\pi N}$ [13], and two nodes can communicate if their Euclidian distance is less than r_t .

In PBcast, a node n_i *broadcasts* its reading x_i instead of routing it to a specific neighbor. Therefore, all neighbors of the n_i receive x_i and would be able to store it (in CStorage a compressed measurements is stored instead of readings as will be discussed later). Each node that has received x_i for the *first time* will *rebroadcast* it with probability p and this probabilistic forwarding continues. Figure 1 shows the average fraction of network nodes $R(p)$ that receive a particular reading after the PBcast and the total number of transmissions $N_t(p)$ versus the forwarding probability p for a connected WSN with $N = 10^4$ and $r_t = 0.02568$.

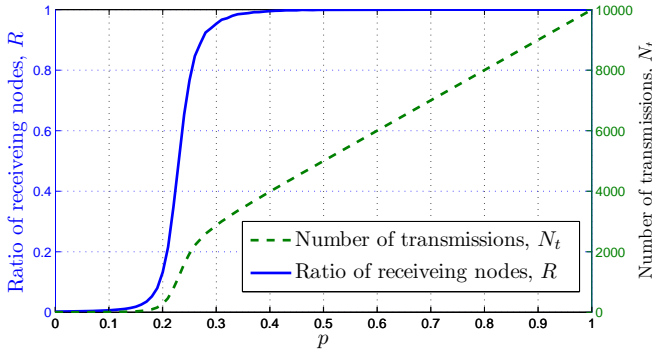


Fig. 1. The fraction of nodes receiving a reading R and the number of transmissions N_t versus forwarding probability p for PBcast.

As we see in Figure 1, at $p \approx 0.24$ a large fraction of nodes receive the broadcast of reading x_i . Moreover, we can observe that although increasing p beyond $p \approx 0.24$ does not significantly contribute to the delivery of x_i , it considerably increases the number of transmissions (almost linear in p). Therefore a well-chosen small forwarding probability $p^* = 0.24$ would be sufficient to ensure a large fraction of nodes in a network have received a reading.

It has been shown that p^* is close to the probability p^G that a *giant component* appears in the network, where asymptotically $p^G \approx \frac{1.44}{Nr_t^2}$ [8, 9]. For our given network topology with $N = 10^4$ and $r_t = 0.02568$ we have $p^G = 0.23$. Therefore, p^* can be approximated with p^G when N is large enough.

C. Related Work

Authors in [14] and [15] have proposed to employ *gossiping* and *random walks*, respectively, to disseminated reading in a large-scale WSN and form measurements at nodes. As we later show, CStorage outperforms these algorithms in the of measurements due to efficient utilization of PBcast.

Authors in [2, 16, 17] have proposed data storage algorithms for sensor networks based on *error correction* codes. Although these algorithms are efficiently designed, they have not exploit the compressibility of the signals in a sensor network to reduce the number of transmissions.

Finally, authors in [3–5, 10, 18] have considered the compressibility of the data collected. However, they have assumed that either routing tables are available or the measurements are

collected at a central sink. Therefore, these schemes cannot be implemented in a distributed large-scale WSN.

III. PROPOSED ALGORITHM: CStorage

In this section, we propose and discuss CStorage.

A. CStorage

In CStorage, node $n_j, j \in \{1, 2, \dots, N\}$, maintains a CS measurement y_j , where y_j is formed as $y_j = \phi_j \underline{x}$ and ϕ_j is an N -dimensional row vector. Let $\Phi_{tot} = [\phi_1^T \phi_2^T \dots \phi_N^T]^T$ and $\underline{y}_{tot} = [y_1 y_2 \dots y_N]^T$. Further, let $\varphi_{j,i}$ be the element at the j^{th} row and the i^{th} column of Φ_{tot} . CStorage steps are as follows:

- 1) $N_s \geq M$ nodes randomly select themselves as a source node and broadcast their readings to their neighbors.
- 2) Upon the reception of reading i (x_i) for the *first time*, node l performs the following:
 - a) Chooses $\varphi_{l,i} = +1$ or -1 with equal probabilities and adds $\varphi_{l,i} x_i$ to y_l .
 - b) *Broadcasts* x_i with probability p (PBcast).

When the transmissions are finished, a Φ_{tot} with approximately N_s non-zero entries per row (for a large enough p) and $\underline{y}_{tot} = \Phi_{tot} \underline{x}$ have been distributed in the network nodes. Therefore, a data collector may gather M measurements $\underline{y} \in \mathbb{R}^M$ and the corresponding ϕ_j 's from an arbitrary set of M nodes and obtain the measurement matrix $\Phi_c \in \mathbb{R}^{M \times N}$ (by putting together the M collected rows of Φ_{tot}). Next, the data collector obtains $\hat{\underline{x}}$ an estimate of \underline{x} employing BP via solving (1).

Note that in CStorage the only information the network nodes need to know is the value of N_s (or equivalently the probability $\frac{N_s}{N}$ where a node selects itself as a source node) and the value of p , which may be preprogrammed into nodes before the network deployment.

B. Forming Φ_{tot} Employing CStorage

Let us consider a small network with $N = 5$ nodes as shown in Figure 2 and investigate one PBcast. Clearly, at the beginning, Φ_{tot} is an all-zero 5×5 matrix. Assume that node n_1 broadcasts its reading x_1 (see Figure 2). Since n_2 and n_3 are in the transmission range of n_1 , they would receive x_1 . Node n_2 multiplies x_1 by $\varphi_{2,1}$ and adds $\varphi_{2,1} x_1$ to y_2 . Similarly, n_3 multiplies x_1 with $\varphi_{3,1}$ and adds $\varphi_{3,1} x_1$ to y_3 . The resulting Φ_{tot} matrix at this step is given by (2).

$$\Phi_{tot} = \begin{pmatrix} \varphi_{1,1} & 0 & 0 & 0 & 0 \\ \varphi_{2,1} & 0 & 0 & 0 & 0 \\ \varphi_{3,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

At this point, n_2 and n_3 independently decide whether to broadcast x_1 with probability p or not. Assume that n_2 decides to broadcast x_1 . Node n_4 would receive x_1 and adds $\varphi_{4,1} x_1$ to y_4 . However, we assume that n_3 and n_4 decide not to rebroadcast x_1 . Thus, the PBcast of x_1 is over and the matrix Φ_{tot} obtains the form of (3). Note that in Figure 2, we have shown the transmitting nodes with a dark color, while the rest of the nodes are shown by white color.

$$\Phi_{tot} = \begin{pmatrix} \varphi_{1,1} & 0 & 0 & 0 & 0 \\ \varphi_{2,1} & 0 & 0 & 0 & 0 \\ \varphi_{3,1} & 0 & 0 & 0 & 0 \\ \varphi_{4,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

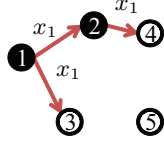


Fig. 2. Network with $N = 5$ and n_1 transmitting x_1 employing PBcast.

The same procedure is performed for N_s source nodes selected *uniformly at random*. Therefore, for $p \approx p^* \approx p^G$ the resulting Φ_{tot} and Φ_c from CStorage have approximately N_s non-zero entries placed uniformly at random in each row. Further, generating a dense Φ_c with N non-zero entries per row imposes a large communication load, i.e., $N_t(p) \times N$ transmissions. As a result, generating a *sparse* Φ_c is of interest since the number of transmissions would significantly reduce.

C. Selection of Transform Basis Ψ

In [11, 19] authors showed that for any dense orthonormal Ψ , e.g., Fourier transform basis, the number of required measurements for the recovery of a K -sparse signal can be obtained from $M \geq C' K \log^4 N$, where C' is a constant. This implies that if the signal is compressible in some dense orthonormal basis we may employ a sparse Φ with *at least one non-zero* placed independently and randomly per row and satisfy CS conditions [11]. Therefore, we may also employ a *sparse* Φ_c to decrease the number of transmissions in CStorage without deteriorating the CS reconstruction performance as long as the rows of Φ_c are randomly selected, or equivalently are linearly independent.

The basis Ψ , under which the signal is sparse or compressible, depends on the nature of the physical phenomenon. For example, in [5] it was shown that the temperature readings of a sensor network are compressible in *discrete cosine transform* (DCT) basis. Therefore, without loss of generality in the rest of our paper we assume that Ψ is the DCT transform basis (while it also may be any other dense orthonormal basis).

D. Optimum Values of N_s and p

On one hand, p and N_s affect the properties of the Φ_{tot} generated in the network, and on the other hand, they directly determine the number of transmissions performed in CStorage, i.e., $n_{tot} = N_t(p) \times N_s$. Hence, we need to find the optimum values of p and N_s for which the minimum n_{tot} is obtained while Φ_c matrix has M independent rows.

Assume that CStorage has been performed with parameters N_s and p . Clearly, we would have non-zero entries only in the columns of Φ_c that correspond to N_s source nodes that broadcast their own reading. Further, the rest of $N - N_s$ columns are all zero and may not contribute in forming independent rows for Φ_c . As shown in Figure 3, consider the submatrix $\Phi_s \in \mathbb{R}^{M \times N_s}$ of Φ_{tot} formed by selecting the N_s

columns of Φ_{tot} corresponding to the N_s nodes broadcasting their readings, and M rows corresponding to M measurements obtained by data collector. Clearly, if Φ_s has *rank* at least M , then the rows of Φ_s and consequently Φ_c are independent.

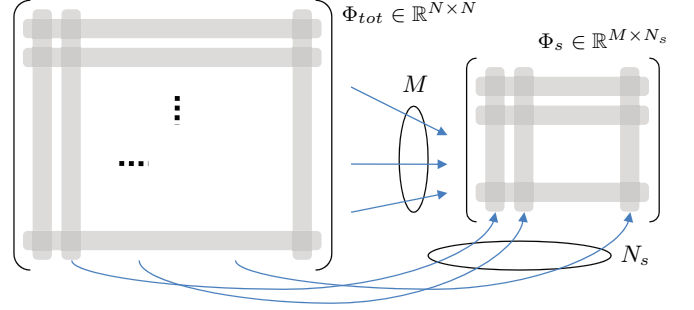


Fig. 3. The submatrix $\Phi_s \in \mathbb{R}^{M \times N_s}$ derived from Φ_{tot} .

We formulate the rank of Φ_s as a function of the ratio of nodes receiving a particular transmission $R(p)$ (see Figure 1) and N_s in the following theorem.

Theorem 1: Let Φ_{tot} be the measurement matrix generated one row per node employing CStorage. Further, let Φ_s be the submatrix from Φ_{tot} by selecting any desired M rows and N_s columns corresponding to N_s source nodes. $r(j)$ the *expected* rank of the matrix Φ_s after j^{th} transmission out of N_s transmissions of CStorage is given by the following recursive equation:

$$\begin{aligned} r(0) &= 0, \\ r(j) &= 1 - (1 - R(p))^{M-r(j-1)} \\ &\quad + r(j-1), j \in \{1, 2, \dots, N_s\}. \end{aligned} \quad (4)$$

Proof:

Clearly, if the network nodes uniformly receive a reading PBcast then $R(p)$ would also be the probability that each node receives the reading. However, generally the dissemination is not uniform, e.g., nodes on the border of network would receive fewer readings. Let $R_{N_s}(p)$ denote the probability that a node receives all N_s transmissions. Clearly, in a uniform distribution $R_{N_s}(p) = R(p)^{N_s}$. However, authors in [8, 9] found the bounds on $R_{N_s}(p)$ in a network where a ratio of $R(p)$ nodes non-uniformly receive a PBcast as follow:

$$R(p)^{N_s} \leq R_{N_s}(p) \leq R(p).$$

It is worth noting that the bounds become tighter as $N \rightarrow \infty$. Therefore, we consider the worst case and assume that all nodes of the network uniformly receive each PBcast with probability $R(p)$.

Let $t(j)$ denote the probability that at least one independent row is added to Φ_s after j^{th} PBcast. Further, let $r(j)$ be the expected value of the rank of Φ_s . At the beginning, Φ_s is an all-zero matrix; hence it has rank 0, i.e., $r(0) = 0$. When the first broadcast is performed, if at least one node out of M nodes of interest receives this broadcast the rank of the matrix Φ_s increases to 1. There is also a possibility that none of these nodes receive this broadcast. For the first transmission, we would have $t(1) = 1 - (1 - R(p))^M$. Therefore, regardless

of the value assigned to all $\varphi_{i,j}$'s in the first transmission, *one independent row* is added to Φ_s by this transmission with probability $t(1)$. Hence the expected number of independent rows of Φ_s becomes $r(1) = 1 \times t(1)$.

If at least one node out of M nodes of interest has received the first transmission, the next broadcast should be received by one node out of $M - 1$ nodes so that the rank of Φ_s increases to 2. Therefore, we need to receive the second transmission by $M - 1$ nodes given that the first transmission was successful. Consequently, the second transmission should be received by at least one of the $M - r(1)$ nodes in expectation so that a new independent row *can* be added to Φ_s .

However, a new independent row would not be added to Φ_s if and only if the following event occurs. Assume an arbitrary set of nodes have received the first transmission. In addition, assume *exactly* the same set of nodes have received the second transmission. Further, assume that all these nodes have selected the same random $\varphi_{i,j}$ in the second reception as they selected in the first reception. In this case, all the rows of Φ_s with non-zero entries would look alike, hence the rank remains 1. However, it is not hard to show that such a rare event happens with probability

$$\sum_{l=1}^N \left[\binom{N}{l} R(p)^l (1 - R(p))^{N-l} \right]^2 \left[\binom{N}{l} 2^l \right]^{-1},$$

which is almost equal to zero for practical values of N .

Therefore, a new independent row is indeed added to Φ_s if one of the $M - r(1)$ nodes in expectation receive the new transmission. Therefore, we have $t(2) = 1 - (1 - R(p))^{M-r(1)}$ and $r(2) = r(1) + 1 \times t(2)$. Similarly, $r(j)$ can be found recursively as $r(j) = r(j-1) + 1 \times t(j)$ with $t(j) = 1 - (1 - R(p))^{M-r(j-1)}$ as given in the Theorem 1. ■

As an example, we set $N = 10^4$ and $M = 700$ and employ Theorem 1 along with the values of $R(p)$ given in Figure 1 to find the rank of Φ_s versus p and N_s as shown in Figure 4.

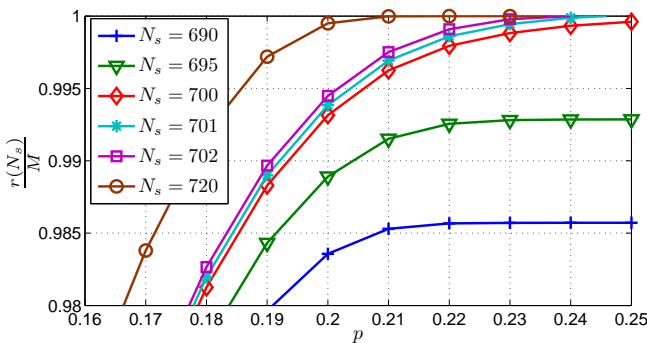


Fig. 4. $\frac{r(N_s)}{M}$ versus p and N_s .

Figure 4 shows that for $N_s \geq M$ the rank of matrix Φ_s approaches to M for a large enough p . More importantly, it shows that as N_s increases a suitable matrix can be generate with a *smaller* value of p . Consequently, we see an interesting *trade-off* since increasing N_s *linearly* increases the total number of transmissions, i.e., $N_s \times N_t(p)$, while it *non-linearly* reduces the required p and consequently $N_t(p)$ and n_{tot} .

Based on the results observed in Figure 4, we find the total number of transmissions $n_{tot} = N_s \times N_t(p)$ for the values of p and N_s for which $\frac{r(N_s)}{M} \geq 0.9999$ versus $N_s - M + 1$ in Figure 5 (we have plotted n_{tot} versus $N_s - M + 1$ to have a better view on the values close to $N_s = M$ on a log-scale axis). Note that we have not fixed p to obtain the curve in Figure 4, but rather relaxed the value of p and searched for the minimum n_{tot} .

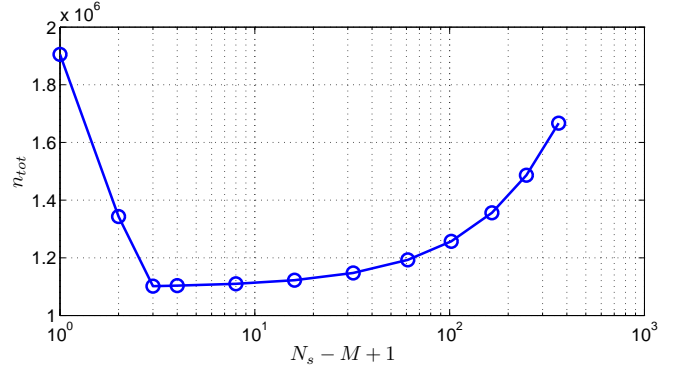


Fig. 5. The total number of transmissions n_{tot} required to generate a suitable Φ_s and Φ_c with $\frac{r(N_s)}{M} \geq 0.9999$ versus $N_s - M + 1$.

The aforementioned trade-off between N_s and p with n_{tot} can be observed in Figure 5, and we can see that the number of transmissions is minimized when N_s is *slightly larger* than M . Figure 5 shows that n_{tot} is minimized for $N_s = 702$, which is obtained for $p = 0.24 = p^*$.

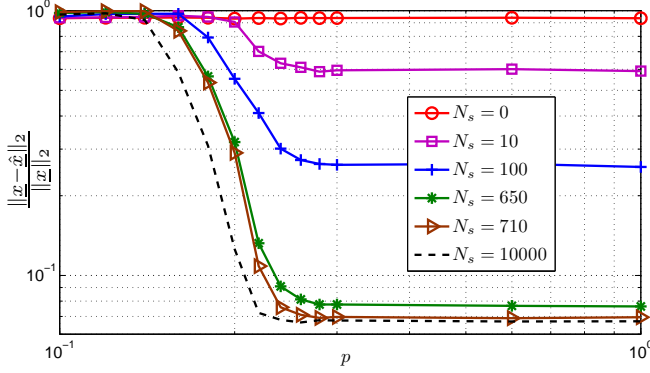
IV. PERFORMANCE EVALUATION OF CSTORE

We generate a compressible signal with $\beta = \frac{7}{8}$ similar to [4] and obtain a compressible signal by $\underline{x} = \Psi \underline{\theta}$. We run our simulation for a randomly deployed WSN as described in Section II-B with $N = 10^4$, $M = 700$, and $r_t = 0.02568$. Figures 6(a) and 6(b) show the normalized reconstruction error $\frac{\|\underline{x} - \hat{\underline{x}}\|_2}{\|\underline{x}\|_2}$ and n_{tot} employing CStorage versus p for various N_s 's, where $\hat{\underline{x}}$ is the reconstructed estimate of \underline{x} at data collector.

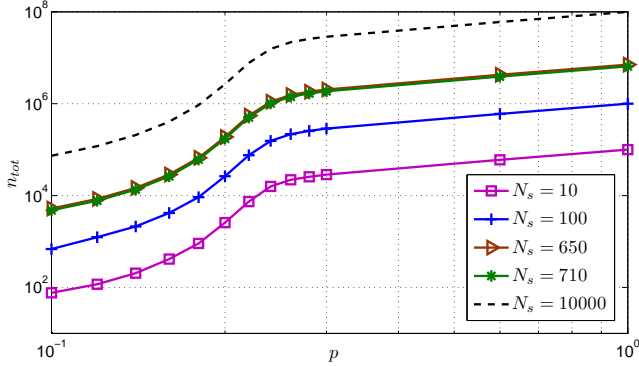
Figure 6(a) confirms that the minimum required p is about $p^* = 0.24$. Further, we can see that for N_s slightly larger than $M = 700$ the reconstruction of \underline{x} is almost as good as the ideal case with a dense Φ_c . Therefore, increasing N_s further does not reduce the reconstruction error while it considerably increases the total number of transmissions n_{tot} (Figure 6(b)). Moreover, we can see that due to the correlation of data in our network with only $M \approx 0.07N$ measurements all N readings have been reconstructed.

Further, we compare the total number of transmissions n_{tot} employing the same WSN employing various data storage algorithms that do not need routing tables in Table I. Note that in the last row of Table I we have evaluated the number of required transmissions if CStorage was implemented utilizing *random walks* as employed in [2, 15, 17] instead of PBcast. Note that the signal reconstruction quality in all algorithms is equal. Table I shows that we have decreased the number of

transmissions more than *one order* of magnitude compared to existing protocols. This is an excellent reduction in the number of transmissions that results in energy saving to a great extent and enhances the lifetime of the WSN while it does not add any more memory requirement or computational complexity to sensor nodes.



(a) Normalized reconstruction error $\frac{\|x-\hat{x}\|_2}{\|x\|_2}$ for different values of p and N_s .



(b) Total number of transmissions n_{tot} for different values of p and N_s .

Fig. 6. Normalized signal reconstruction error and n_{tot} versus p and N_s for DCT basis.

TABLE I
COMPARISON OF n_{tot} IN CStorage WITH EXISTING ALGORITHMS.

Protocol	n_{tot}	Notes
CStorage	1.545×10^6	-
Flooding	10^8	$n_{tot} = N^2$
Gossiping	8.4×10^9	1200 gossips per measurement [14]
CStorage with random walks	1.93×10^8	$n_{tot} = C_1 N \log(N) \times N_s, C_1 = 3$ [2]

V. CONCLUSION

In this paper, we introduced a novel and fully distributed data storage scheme referred to as *CStorage* for wireless sensor networks (WSNs) employing novel *compressive sensing* (CS) techniques. In *CStorage*, N_s nodes from $N \gg N_s$ total nodes disseminated their readings employing *probabilistic broadcasting* throughout the network, hence all nodes acquire a random *compressed* sample of these readings. Next, a data collector needs to collect only $M \ll N$ measurements to restore all N readings utilizing CS. We found the optimal

parameters of *CStorage* and showed that N_s needs to be slightly large than M to achieve the minimum number of required transmissions, which is considerably lower than the required number of transmissions in existing protocols for distributed data storage in WSNs. In the future work, *CStorage* will be extended considering the spatio-temporal correlation of the readings.

REFERENCES

- [1] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, pp. 476–489, march 2011.
- [2] Z. Kong, S. Aly, and E. Soljanin, "Decentralized coding algorithms for distributed storage in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 261–267, february 2010.
- [3] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data acquisition through joint compressive sensing and principal component analysis," *IEEE Global Telecommunications Conference, GLOBECOM 2009.*, pp. 1–6, Dec 2009.
- [4] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega, "Compressed Sensing and Routing in Multi-Hop Networks," *Technical Report, University of Southern California.*, 2009.
- [5] C. Chou, R. Rana, and W. Hu, "Energy efficient information collection in wireless sensor networks using adaptive compressive sensing," *IEEE 34th Conference on Local Computer Networks (LCN 2009)*, 2009.
- [6] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [7] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, pp. 489–509, Feb. 2006.
- [8] N. Rahnavard, B. Vellambi, and F. Fekri, "CRBcast: a reliable and energy-efficient broadcast scheme for wireless sensor networks using rateless codes," *IEEE Trans. on Wireless Communications*, vol. 7, no. 12 Part 2, pp. 5390–5400, 2008.
- [9] N. Rahnavard and F. Fekri, "CRBcast: a collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks," *International conference on Information processing in sensor networks*, p. 283, 2006.
- [10] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," *International conference on Information processing in sensor networks*, pp. 331–339, 2007.
- [11] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [12] S. S. Chen, D. L. Donoho, Michael, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.
- [13] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity," *Proceedings of the 37th IEEE Conference on Decision and Control.*, vol. 1, pp. 1106–1110 vol.1, 1998.
- [14] M. Rabbat, J. Haupt, A. Singh, and R. Nowak, "Decentralized compression and predistribution via randomized gossiping," *International Symposium on Information Processing in Sensor Networks*, pp. 51–59, 2006.
- [15] M. Lin, C. Luo, F. Liu, and F. Wu, "Compressive data persistence in large-scale wireless sensor networks," *IEEE Global Telecommunications Conference, GLOBECOM*, pp. 1–5, 2010.
- [16] A. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized erasure codes for distributed networked storage," *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2809–2816, 2006.
- [17] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," *26th IEEE International Conference on Computer Communications*, pp. 1658–1666, may 2007.
- [18] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 145–156, 2009.
- [19] M. Rudelson and R. Vershynin, "Sparse reconstruction by convex relaxation: Fourier and gaussian measurements," *40th Annual Conference on Information Sciences and Systems*, pp. 207–212, march 2006.