Ad Hoc Networks 37 (2016) 475–485

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

# CStorage: Decentralized compressive data storage in wireless sensor networks

Ali Talari [a,*], Nazanin Rahnavard [b]

[a] Oklahoma State University, Stillwater, OK 74078, United States
[b] University of Central Florida, Orlando, FL 32806, United States

ARTICLE INFO

*Article history:*
Received 27 May 2014
Revised 25 August 2015
Accepted 21 September 2015
Available online 30 September 2015

*Keywords:*
Compressive sensing/sampling
Distributed data storage
Wireless sensor networks
Probabilistic broadcasting
Data dissemination

ABSTRACT

In this paper, we employ *compressive sensing* (CS) to design a distributed *compressive data storage* (CStorage) algorithm for *wireless sensor networks* (WSNs). First, we assume that no neighbor information or routing table is available at nodes and employ the well-known *probabilistic broadcasting* (PB) to *disseminate* sensors reading throughout the network to form *compressed samples* (measurements) of the network readings at each node. After the dissemination phase, a data collector may query *any arbitrary* set of $M \ll N$ nodes for their measurement and reconstruct all $N$ readings using CS. We refer to the first implementation of CStorage by *CStorage-P*.

Next, we assume that nodes collect two-hop neighbor information and design a novel *parameterless* and *scalable* data dissemination algorithm referred to by *alternating branches* (ABs), and design *CStorage-B*. We discuss the advantages of CStorage-P and CStorage-B and show that they considerably decrease the total number of required transmissions for data storage in WSNs compared to existing work.



## 1. Introduction

To increase the data persistence in *wireless sensor networks* (WSNs) with $N$ nodes, distributed *data storage* algorithms have been proposed to disseminate sensors reading *throughout* the network so that a data collector can query an *arbitrary small subset* of nodes to obtain all $N$ readings [1,2].

Recently, *compressive sensing* (CS) techniques [3,4] have shown that a *compressible* signal with length $N$ can be reconstructed from only $M \ll N$ *random projections* of the signal (also called as measurements or compressed samples). Since natural signals are known to be compressible due to strong *spatial* correlation of sensor readings [5–7], CS may be exploited to design efficient data storage algorithms. Consequently, we design a decentralized *compressive data storage* algorithm (CStorage) that exploits the *spatial correlation* of the nodes reading along with CS to considerably reduce the total number of transmissions for data storage.

In CStorage, we propose to form a CS *measurement* at each node by disseminating *enough* number of readings throughout the network. First, we employ the well-known *probabilistic broadcasting* (PB) for data dissemination and propose *CStorage-P*. In PB, no neighbor information or routing table is required for data dissemination. Nevertheless, PB has a parameter called *forwarding probability* that needs to be tuned at all nodes when the network changes, which is not always possible.

Therefore, we assume that nodes can obtain two-hop neighbor information and design a *parameterless* and efficient data dissemination algorithm referred to by *alternating branches* (ABs), and design *CStorage-B*. Since AB has no parameter to tune, CStorage-B is *scalable* and can automatically adapt to drastic network topology changes. We will show both CStorage-P and CStorage-B reduce the total number of transmissions compared to existing algorithms for data

* Corresponding author. Tel.: +1 405 744 4669.
*E-mail addresses:* ali.talari@okstate.edu, alitalari@gmail.com (A. Talari), nazanin@eecs.ucf.edu (N. Rahnavard).

http://dx.doi.org/10.1016/j.adhoc.2015.09.009
1570-8705/

storage in WSNs *without* routing tables, while CStorage-B surpasses CStorage-P in the number of transmissions. The initial results of this paper on CStorage-P have appeared in [8]. In this paper, we design AB and introduce CStorage-B. Further, we employ real readings from a WSN to evaluate the performance of our proposed schemes.

The paper is organized as follows. Section 2 provides the required background. In Section 3, we propose CStorage-P. In Section 4, we design and analyze AB and CStorage-B. In Section 5, we evaluate the performance of CStorage-P and CStorage-B. Finally, Section 6 concludes the paper.

## 2. Background

In this section, we review CS, PB, and the related work.

### 2.1. Compressive sensing

Let $\underline{\theta} = [\theta_1\theta_2 \ldots \theta_N]^T$ ($\theta_i \in \mathbb{R}$) be the transform of a signal $\underline{x} = [x_1x_2 \ldots x_N]^T$ ($x_i \in \mathbb{R}$) in transform domain $\Psi \in \mathbb{R}^{N\times N}$, i.e., $\underline{x} = \Psi\underline{\theta}$. $\underline{x}$ is said to be compressible in $\Psi$ if $\underline{\theta}$ has only $K$ significant coefficients (the rest $N-K$ coefficients can be set to zero). Such a signal is referred to by $K$-sparse signal.

The idea behind CS is that when $\underline{x}$ is $K$-sparse in $\Psi$, only $M \ll N$ ($M \geq O(K \log N)$) measurements $\underline{y} = [y_1y_2 \ldots y_M]^T$ of $\underline{x}$ can reproduce an estimate $\hat{\underline{x}}$ using CS *reconstruction* with a comparable error to the best approximation error using $K$ largest transform coefficients [3,4,9]. CS is composed of the two following key components.

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

*Decoding:* Signal reconstruction can be performed by finding the estimate $\hat{\underline{\theta}}$ (and consequently $\hat{\underline{x}} = \Psi\hat{\underline{\theta}}$) via solving

$$\hat{\underline{\theta}} = \text{argmin}\|\underline{\theta}\|_1, \text{ s.t. } \underline{y} = \Phi\Psi\underline{\theta}, \tag{1}$$

where $\|\underline{\theta}\|_1 = \sum_{i=1}^{N} |\theta_i|$. The problem (1) is an underdetermined system of equations. In this paper, we employ the well-known *basis pursuit* technique to solve (1) [3,4].

Initially, measurement matrices were *dense* random matrices with entries selected from $\{-1, +1\}$ or $\mathfrak{N}(0, 1)$, where $\mathfrak{N}(0, 1)$ is the zero mean and unit variance Gaussian distribution [3,4]. Later, it was shown that when $\Psi$ is *dense* and *orthonormal*, e.g., Fourier transform basis, a *sparse* $\Phi$ also satisfies CS requirements on $\Phi$ [9,10]. Therefore, in this paper we employ sparse $\Phi$ matrices, since as we later see they can be formed with a much smaller number of transmissions. Further, the selection of $\Psi$ depends on the nature of the signal. For instance, temperature signals are shown to be sparse in *discrete cosine transform* (DCT) basis [7]. Therefore, without loss of generality in the rest of this paper we assume that $\Psi$ is the DCT transform basis, while we could have chosen any other dense and orthonormal basis.

### 2.2. Probabilistic broadcasting

Consider a WSN with $N$ nodes having identical transmission range $r_t$ deployed uniformly and randomly in an area $A = 1 \times 1$, where two nodes can communicate if their Euclidian distance is less than $r_t$. The network is asymptotically connected with

$$r_t^2 = \frac{A(\ln n + \omega(n))}{\pi n}, \tag{2}$$

if and only if $\omega(n) \to \infty$ [11]. In PB, a node $n_i$ *broadcasts* its reading $x_i$ to all its neighbors. Any node in the network that receives $x_i$ for the *first time* rebroadcasts $x_i$ with *forwarding probability* $p$ [12] (with $p = 1$, PB boils down to simple Flooding [13]). The fractions of nodes that receive a particular transmission $R_{PB}(p)$ and the fraction of nodes that perform the transmission $T_{PB}(p)$ are depicted in Fig. 1 for $N = 10^4$ and $r_t = 0.025$.

Fig. 1 shows that at $p \approx 0.24$ a large fraction (about 70%) of nodes receives the reading. Moreover, we can see that although increasing $p$ beyond $p \approx 0.24$ does not improve the delivery of the reading, it considerably increases the number of transmissions. Therefore, a well-chosen small forwarding probability $p^* = 0.24$ would be sufficient to ensure that a large fraction of nodes in a network has received a transmission [14,15]. Using a few simple calculations, for $N = 10^4$ and $r_t = 0.025$ we can see that a node has on average $n_{neighbor} = 20$ neighbors, and receives $n_{neighbor} \times p^* \approx 5$ copies of each transmission on average.

### 2.3. Related work

Authors in [16] propose LORD Scalable and a Mobility-Resilient Data Search System. The LORD maps sensor reads to a geographical region and stores it in multiple nodes in the region, thus enhancing mobility-resilience. In contrast to CStorage, LORD does not take advantage of compressibility of the readings due to the spatial correlation of the readings. In [17], authors discuss that the real sensor readings may not be compressible in DCT nor in other orthogonal transformations. To achieve a sparse representation for spatiotemporal readings in real WSNs, they develop a novel two-dimensional dictionary training method.

Authors in [18] fit the power-law decaying data model to the real data collected in WSNs due to its strong compressibility, and propose CDC. CDC performs on-the-fly compression of sensor readings to reduce communication overhead and energy consumption. Authors in [19] propose to employ *random walks* to form the random measurements in a WSN. We will compare CStorage with such algorithms later and show that CStorage outperforms in the number of required transmissions.

Authors in [20] proposed ICStorage, which is built on top of our initial results on CStorage-P [8]. They propose to merge the received measurements from neighbors into the measurements maintained at nodes, and forwarding the new packets. Further, authors in [21] propose STCNC that exploits both spatial and temporal (spatiotemporal) correlations among sensor readings that further increases the energy efficiency. These algorithms consider a different problem compared to CStorage.

Previously, Wang et al. in [9] showed that *sparse* $\Phi$ matrices can satisfy CS requirements and designed a data storage algorithm based on these sparse $\Phi$ matrices. Further, authors in [6,7,22–25] proposed *centralized* data collection algorithms where measurements are formed enroute and are