CStorage: Decentralized compressive data storage in wireless sensor networks

Ali Talari\textsuperscript{a,}*, Nazanin Rahnavard\textsuperscript{b}

\textsuperscript{a} Oklahoma State University, Stillwater, OK 74078, United States
\textsuperscript{b} University of Central Florida, Orlando, FL 32806, United States

\textbf{A B S T R A C T}

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.

\textsuperscript{*} 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).

\url{http://dx.doi.org/10.1016/j.adhoc.2015.09.009} 1570-8705/© 2015 Elsevier B.V. All rights reserved.

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
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 \( \theta = [\theta_1, \theta_2, \ldots, \theta_K]^{T} \) \( (\theta_i \in \mathbb{R}) \) be the transform of a signal \( x = [x_1, x_2, \ldots, x_n]^{T} \) \( (x_i \in \mathbb{R}) \) in transform domain \( \Psi \in \mathbb{R}^{N \times M} \), i.e., \( x = \Psi \theta \). \( x \) is said to be compressible in \( \Psi \) if \( \theta \) 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 \( x \) is \( K \)-sparse in \( \Psi \), only \( M \ll N (M \geq O(K \log N)) \) measurements \( y = [y_1, y_2, \ldots, y_M]^{T} \) of \( x \) can reproduce an estimate \( \hat{x} \) 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 \( y = \Phi x \), where \( \Phi \) is a well-chosen \( M \times N \) random matrix called projection matrix.

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

\[
\hat{\theta} = \arg \min ||\theta||_1 \quad \text{s.t.} \quad y = \Phi \Psi \theta,
\]

where \( ||\theta||_1 = \sum_{i=1}^{N} |\theta_i| \). 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 \( \mathcal{O}(0, 1) \), where \( \mathcal{O}(0, 1) \) 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

\[
\tau^2 = \frac{A (\ln n + \omega(n))}{\pi n},
\]

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 \( T_{PB}(p) \) 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 \( p = 0.24 \) a large fraction (about 70%) of nodes receives the reading. Moreover, we can see that although increasing \( p \) beyond \( p = 0.24 \) 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
collected at a sink. These algorithms in contrast to CStorage require routing tables or full topology knowledge, which are not always possible in practical WSNs.

Authors in [26] proposed to employ gossiping to disseminate the reading in the network. In gossiping, each node iteratively exchanges their reading with a random neighbor. After many iterations all network nodes obtain the value of the reading and a measurement is formed at nodes. We will compare CStorage with Gossiping since the both algorithms require the same type of information, which makes a fare comparison possible.

Finally, authors in [2,27,28] proposed data storage algorithms for sensor networks based on error correction codes. Although these algorithms are efficiently designed, they have not exploited the compressibility of the signals in a WSN to reduce the number of transmissions.

On the other hand, existing work in [29–33] are comparable with CStorage-B, which will be discussed in Section 4.1.

3. Compressive data storage employing PB

In CStorage, node \(n_j, j \in \{1,2,\ldots,N\}\), maintains a CS measurement \(y_j \in \mathbb{R}\), where \(y_j = \phi_j x\) and \(\phi_j\) is an \(N\)-dimensional row vector and \(x = [x_1,x_2,\ldots,x_N]^T\) is sensors’ reading (\(x_i\) is the reading of \(n_i\)). Let \(\Phi_{tot} = [\phi_1^T\phi_2^T\ldots\phi_N^T]^T\) and \(y_{tot} = [y_1y_2\ldots y_N]^T\). Further, let \(\varphi_{ij}\) be the element at the \(i\)th row and the \(j\)th column of \(\Phi_{tot}\). The matrix \(\Phi_{tot}\) is formed when nodes receive various readings employing an underlying data dissemination algorithm. We will propose two dissemination algorithms for this purpose. We first employ PB and refer to the compressed storage scheme as CStorage-P. We also propose another dissemination scheme called alternating branching, and refer to the corresponding compressed storage scheme as CStorage-B.

When the transmissions are over, \(\Phi_{tot}\) is formed distributively (as described in detail later) in the network. The data collector queries \(M\) nodes for their measurements \(y_j\) and the corresponding \(\phi_j\) maintained at each node, and forms \(y \in \mathbb{R}^M\) and \(\Phi \in \mathbb{R}^{M \times N}\). Next, the data collector obtains \(\hat{x}\), an estimate of \(x\), employing basis pursuit by solving (1).
\( \Phi_{\text{tot}} \) corresponds to the measurements formed at node \( i \).

\[
\Phi_{\text{tot}} = \begin{pmatrix}
\psi_{1,1} & 0 & 0 & 0 & 0 \\
\psi_{2,1} & \psi_{2,2} & 0 & 0 & 0 \\
\psi_{3,1} & 0 & \psi_{3,3} & 0 & 0 \\
\psi_{4,1} & 0 & 0 & \psi_{4,4} & 0 \\
0 & 0 & 0 & 0 & \psi_{5,5}
\end{pmatrix}
\] (3)

3.2. Suitable values of \( N_i \) and \( p \)

As shown in [9], a sparse \( \Phi \) matrix can be used to recover a signal with same order of number of measurements as a dense measurement matrix if \( \Phi \) has \( M \) independent rows (is full rank). In other words, any \( M \) rows of \( \Phi_{\text{tot}} \) that correspond to collecting any \( M \) measurements by a data collector need to be independent. We need to find the suitable values of \( N_i \) and \( p \) such that the collected \( M \) rows of \( \Phi_{\text{tot}} \) form a sparse \( \Phi \) with the aforementioned properties while \( N_{\text{tot}} \), the total number of transmissions for the \( N_i \) disseminations, is minimized. If \( \text{TPB}(p) \) denotes the fraction of network nodes that perform the retransmission in a PB with forwarding probability \( p \) (see Section 2.2), each PB requires \( \text{TPB}(p)N \) transmissions. Therefore, we have \( N_{\text{tot}} = \text{TPB}(p)NN_i \). In the following theorem from [8], we find the expected number of independent rows of \( \Phi \) as a function of \( N_i \) and \( p \).

**Theorem 1** ([8]). Let an \( M \times N \) matrix \( \Phi \) be the measurement matrix obtained from \( \Phi_{\text{tot}} \) in CStorage-P. Further, let \( \text{R}_{\text{PB}}(p) \) be the fraction of nodes that receive a transmission using PB with forwarding probability \( p \) (see Section 2.2). \( r(j) \), the expected number of independent rows of \( \Phi \) after the \( j \)th transmission (out of \( N_i \) transmissions), is given by the following:

\[
r(0) = 0,
\]

\[
r(j) = 1 - (1 - \text{R}_{\text{PB}}(p))^{M - r(j-1)} + r(j-1), j \in \{1, 2, \ldots, N_i\}.
\] (4)

Using Theorem 1, we can show that for \( N_i \geq M \) the number of independent rows of \( \Phi \) approaches \( M \) for a large enough \( p \), and for \( N_i < M \) the number of independent rows of \( \Phi \) never reaches \( M \) [8]. Further, we can show that as \( N_i \) increases a suitable matrix can be generated with a smaller value of \( p \). Consequently, we see an interesting trade-off since increasing \( N_i \) increases \( N_{\text{tot}} = \text{TPB}(p)NN_i \), while it reduces the required \( p \) and consequently \( \text{TPB}(p) \). It can be shown that the optimal value of \( p \) and \( N_i \) that minimizes \( N_{\text{tot}} \) is when \( N_i \) is set slightly larger than \( M \) and \( p = p^* \) [8]. The values of \( p^* \) for various \( r_i \)'s and the respective average number of neighbors for \( A = 1 \times 1 \) and \( N = 10^4 \) are given in Table 1.

<table>
<thead>
<tr>
<th>( r_i )</th>
<th>No. of neighbors</th>
<th>( p^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.021</td>
<td>13</td>
<td>0.38</td>
</tr>
<tr>
<td>0.022</td>
<td>16</td>
<td>0.32</td>
</tr>
<tr>
<td>0.024</td>
<td>18</td>
<td>0.28</td>
</tr>
<tr>
<td>0.026</td>
<td>21</td>
<td>0.25</td>
</tr>
<tr>
<td>0.027</td>
<td>24</td>
<td>0.22</td>
</tr>
<tr>
<td>0.029</td>
<td>27</td>
<td>0.19</td>
</tr>
<tr>
<td>0.031</td>
<td>30</td>
<td>0.17</td>
</tr>
<tr>
<td>0.033</td>
<td>33</td>
<td>0.16</td>
</tr>
<tr>
<td>0.034</td>
<td>37</td>
<td>0.14</td>
</tr>
</tbody>
</table>

4. Compressive data storage in WSNs employing CStorage-B

In this section, we propose a novel data dissemination algorithm referred to by alternating branching (AB) that is independent of network topology (has no parameter to tune). We will then employ AB for data dissemination in CStorage and propose CStorage-B in Section 4.6.
tune $p$. In [33], nodes need to know the optimal number of next transmitters, which is network dependent.

Consequently, in AB we propose each transmitter to select a fixed number of next transmitter(s) regardless of any of network parameter to have a uniform dissemination throughout the network. This ensures that there are enough number of transmitters even in sparse areas of network, and results in uniform dissemination of $x_t$ regardless of the density of nodes as we later see.

4.2. The alternating branching design

Although, in large scale WSNs global routing tables may not be obtained, obtaining one-hop and two-hop neighbor information is simple. If all nodes broadcast a hello message, every node obtains one-hop neighbor information. If all nodes broadcast the list of their neighbors following all hello messages of the first round, every node obtains two-hop neighbor information. Clearly, this results in 2N transmissions in total.

Based on our discussions, in AB we propose a node $n_t$ that is retransmitting $x_t$ to be responsible to choose the next transmitter(s). Thus, $n_t$ has been selected to be a transmitter by $n_{t,p}$. Assume only one next transmitter $n_{t,1}$ is chosen by $n_t$ in Fig. 3. Because all nodes in $\mathcal{N}(n_{t,p})$ have already received $x_t$, the next transmitter of $n_{t,1}$ is selected from $\mathcal{N}(n_t) \setminus \mathcal{N}(n_{t,p})$. Where $\setminus$ denotes the subtraction of two sets (nodes in the gray area of Fig. 3).

Clearly, a neighbor of $n_t$ that has the minimum number of common neighbors with $n_{t,p}$ is probably (and not necessarily) farthest node whose transmission potentially results in the largest new covered area. This is a greedy selection at $n_t$ and is not by any means an optimal farthest node selection when only two-hop neighbor information is available. Consequently, $n_t$ chooses the next transmitter $n_{t,1}$ such that

$$n_{t,1} = \arg\min_{n_{t,1}} |\mathcal{N}(n_{t,1}) \setminus \mathcal{N}(n_{t,p})|,$$

where $\setminus$ denotes the intersection of two sets.

Ideally, $n_{t,1}$ is placed on the transmission border of $n_t$ and on the straight line connecting $n_t$ and $n_{t,p}$. We emphasize that we have shown the ideal setup for the sake of simplicity and in our actual implementation next hop is not necessarily on the edge of transmission range nor is on a straight line with $n_{t,p}$ (it is selected based on Eq. (5)).

Consider a source node $n_t$ that initiates the broadcast of $x_t$ and assume all its neighbors rebroadcast $x_t$. If we allow these nodes to choose only one next transmitter, and those transmitters to choose one transmitter again and so on, they will (ideally) form straight lines of transmitters that emanate from $n_t$ and travel toward borders. Clearly, such dissemination will be incomplete in the network. Therefore, some nodes should choose more than one next transmitter so that the transmitters branch and multiply (as the branches of a tree multiply) and $x_t$ is well disseminated by an increase in the number of transmitters.

Consider selecting two next transmitters by the current transmitter $n_t'$, as depicted in Fig. 4. We can see that as the number of next transmitters increases, the overlapping area of their coverage also increases, hence their transmissions become less efficient. Consequently, we propose to choose only two next transmitters when branching occurs.

Let $n_{t,1}', n_{t,2}' \in \mathcal{N}(n_t') \setminus \mathcal{N}(n_{t,p})$ denote the two next transmitters. Similar to choosing one next transmitter $n_{t,1}$, we can possibly provide the largest new covered area by the transmission of $n_{t,1}', n_{t,2}'$ when they have minimum number of common neighbors with each other and with $n_{t,p}$. Therefore, $n_{t,1}', n_{t,2}'$ are selected such that

$$n_{t,1}', n_{t,2}' = \arg\min_{n_{t,1}', n_{t,2}'} |\mathcal{N}(n_{t,1}') \setminus \mathcal{N}(n_{t,2}') \setminus \mathcal{N}(n_{t,p})|,$$

as shown in Fig. 4.

The branching should occur frequently in random networks to ensure enough new branches are produced to explore new uncovered areas especially when nodes are sparse. Therefore, we propose to branch at every other transmitter.

Therefore, we propose to branch at every other transmitter. To control the branching, we propose to include a single-bit binary counter as branching flag along with $x_t$. When $n_{t}$ wants to broadcast, it first checks the branching flag. If the flag is 0, $n_t$ chooses one next transmitter and two otherwise. Next, it flips the flag, and rebroadcasts $x_t$ along with the ID of the next transmitter(s) and the branching flag. In addition, if a node is selected as the next transmitter of $x_t$ but it has received it before, the branch has been chosen from an area where $x_t$ has already been disseminated. Therefore, this transmission is redundant and is ignored. With such a scheme nodes alternatively select one and two next transmitters. Therefore, we refer to our algorithm by alternating branching (AB). Note that $n_t$ initiates the broadcast of $x_t$ with branching flag of 0. In Fig. 5, we have shown the dissemination of one reading using AB with the source node located in the center of a $A = 1 \times 1$ network with $N = 10^4$ nodes at four different progressive time snaps until AB is completed.

In Fig. 5, we can see that branches emanate from the source and are spread towards borders. However, due to random placement of nodes they may not move straightly towards edges. Further, we can see that branches may arrive at the same node after a few steps and terminate. Moreover, we can see that the nodes that have not received the transmission are well distributed throughout the network.

4.3. Analysis of AB on grids

Let us first investigate AB in an ideal grid setup. If we repeat the ideal pattern of transmitting nodes shown in Figs. 3 and 4, they form an isometric grid network shown in Fig. 6. We should note that isometric grids have been previously considered in WSNs [34]. It is easy to see that the transmitters form hexagon cells.
Fig. 5. Dissemination of a reading from the source node at the center (shown by a star) using AB. The dark colored nodes are the transmitters forming branches, the light colored nodes are the nodes that receive the reading, the white areas are the nodes that do not receive the transmission. Figures belong to the same dissemination in progressive snap times from left to right and up to down, until the dissemination is complete.

Fig. 6. Ideal implementation of AB that results in isometric grid. The transmitters are shown with filled black circles, nodes that receive the transmission but do not retransmit are shown by hollow circle, the nodes that do not receive the transmission are shown by gray square (in the center of hexagons formed by transmitters), and arrows show the progress direction of the branch. Clearly, transmitters form hexagon shaped cells. The left and the right panels show the grid when the transmission range is one and two grid size, respectively.

Let \( r_g \in \{1, 2, \ldots \} \) denote the transmission range of nodes on the isometric grid as multiples of grid-size (in Fig. 6, we have \( r_g = 1 \) and \( r_g = 2 \) on left and right, respectively). We may simply formulate the fraction of nodes that receive and transmit in AB on isometric grid. Since the whole network has the same hexagon shaped cells, the fraction of nodes that transmit and receive are equal for a cell and the whole network. The transmitters around a hexagon also belong to its neighboring hexagons too, while the nodes inside a hexagon only belong to one cell. Using Fig. 6 and the discussion provided, the number of nodes that solely belong to one hexagon \( N_H \) and the number of nodes that do not receive a transmission in one hexagon \( N_{NR} \) are given by the following lemma

\[
N_H = 1 - 6r_g + 6 \sum_{i=1}^{2r_g} i \quad \text{and} \quad N_{NR} = 1 + 6 \sum_{i=1}^{r_g-1} i. \quad (7)
\]

Using Lemma 1, we find the fraction of nodes that receive a transmission, \( R_g \), and the fraction of nodes that perform the transmission, \( T_g \), in a grid network in the following Theorem.

**Theorem 2.** In ideal AB on an isometric grid, for transmission range \( r_g \in \{1, 2, \ldots \} \), the fraction of nodes that receive the transmission, \( R_g \), and the fraction of nodes that perform the transmission, \( T_g \), is the same for one hexagon and the whole network. Therefore, we have

\[
R_g = \frac{N_H - N_R}{N_H} \quad \text{and} \quad T_g = \frac{6}{N_H}. \quad (8)
\]

We also employ Monte-Carlo numerical simulations to find the average fraction of receivers, \( R_r \), and transmitters, \( T_r \), when the deployment of nodes is random with \( N = 10^4 \). Further, to perform a comparison with existing work, we assume nodes in the random network are equipped with GPS [29], and also propose a second implementation of alternating branching, where nodes are equipped with GPS and the farthest nodes are selected based on their actual position referred to by AB\(_{GPS} \). We denote the average fraction of receivers and transmitters in AB\(_{GPS} \) by \( R_{GPS} \) and \( T_{GPS} \), respectively.

Since AB has no parameter to tune, we vary the transmission range \( r_t \) from its minimum value, i.e., threshold of \( r_t \) for which network becomes disconnected (as discussed in Section 2.2), to large values where nodes are densely connected. The number of neighbors in isometric grid cannot take all values in contrast to random networks and is given
by 6 \sum_{i=1}^{5} i \in \{6, 18, 36, \ldots\}. Fig. 7, compares \( R_r, R_g, T_r, \) and \( T_g \). \( R_r \) and \( T_r \) are plotted versus the average number of neighbors.

Fig. 7 shows that \( \text{AB} \) provides almost constant fraction of receivers and transmitters despite drastic changes in network topology. Therefore, if the network changes over time \( \text{AB} \) automatically adapts to changes. This is in contrast to \( \text{PB} \) where \( R_g \) and \( T_g \) are greatly affected by \( \rho \). In addition, from Fig. 7 we can observe that although \( \text{AB} \) performs very close to ideal grid model, it is more efficient and its performance approaches that of \( \text{AB} \). In addition, Fig. 7 shows that \( \text{AB} \) provides almost as close as possible to \( \text{GPS} \) and \( \text{AB} \) on isometric grid, and \( \text{AB} \) on isometric grid, and \( \text{AB} \), respectively.

The average fraction of nodes receiving and transmitting in \( \text{AB} \) dissemination. \( R_r, R_g, \) and \( R_{\text{GPS}} \) denote the fraction of receivers in \( \text{AB}, \text{AB} \) on isometric grid, and \( \text{AB} \), respectively. Further, \( T_r, T_g, \) and \( T_{\text{GPS}} \) denote the fraction of transmitters in \( \text{AB}, \text{AB} \) on isometric grid, and \( \text{AB} \), respectively.

\[ \text{Fig. 7.} \text{ The average fraction of nodes receiving and transmitting in \( \text{AB} \) dissemination.} \]

Лемма 2. Пусть \( X_i = \{1, 2, \ldots, k\} \) - идентичные по значению случайные величины с одинаковым cdf \( F_{X_i}(d) \), и пусть случайная величина \( X_{\max} = \max\{X_1, \ldots, X_k\} \). \( F_{\max}(d) \) - это кумулятивная функция распределения \( X_{\max} \). \( F_{\max}(d) = F_{X_i}(d) \).

Доказательство. \( F_{\max}(d) = P(X_{\max} \leq d) = P(X_1 \leq d, \ldots, X_k \leq d) = P(X_1 \leq d) \ldots P(X_k \leq d) = F_{X_i}(d) \). \( \Box \)

Node \( n_t \) maximizes the distance of the next forwards from the set \( \mathcal{N}(n_t) \), located in the gray area in Fig. 3. The size of the shaded region is \( A_{\text{sel}} = \frac{d}{\sqrt{4 - d^2}} + 2 \arcsin \frac{d}{2} \).

The number of nodes in the shaded area is given by \( N_{\text{sel}} = N_{\mathcal{N}(n_t)} \rho A_{\text{sel}} \), where \( \rho = \frac{N}{\pi} \) is the density of nodes. Let \( X_{\max} = \max\{X_1, \ldots, X_k\} \) be the random variable denoting the distance of next forwarders to \( n_t \). Using Lemma 2, we have \( F_{\max}(d) \approx d^{2N_{\text{sel}}} \). Consequently, the expected distance \( \bar{d} \) is simply obtained by \( \bar{d} = E[X_{\max}] \), where \( E[\cdot] \) denotes the expected value of a random variable. The expected value of a random variable \( Z \) can be calculated from its cdf \( F_Z(x) \) by \( E[Z] = \int_0^\infty (1 - F_Z(x)) dx - \int_0^{\infty} F_Z(x) dx \). This gives

\[ \bar{d} = E[X_{\max}] = \int_0^1 (1 - 2^{2N_{\text{sel}}}) dz. \]

\[ = 1 - \frac{1}{2N_{\text{sel}} + 1} \]

\[ = 1 - \frac{1}{2\rho \left[ \frac{d}{\sqrt{4 - d^2}} + 2 \arcsin \frac{d}{2} \right] + 1} \]

After a few simple mathematical operations, we obtain

\[ \rho = \frac{\bar{d}}{(1 - \bar{d}) \left( d \sqrt{4 - d^2} + 4 \arcsin \frac{d}{2} \right)} \]  

The value of \( \bar{d} \) may be obtained from (9) for any \( \rho \). For instance, at average number of neighbors equal to 22 we have \( \bar{d} = 0.963 \) and in the worst case for almost disconnected network (average neighbor number of 12), we have \( \bar{d} = 0.93 \). Therefore, the assumption that next forwarders are placed on the transmission range border of \( n_t \) in grid networks is not far from reality in random networks. Therefore, \( R_g \) and \( T_g \) may provide close estimates of \( R_r \) and \( T_r \), as shown in Fig. 7.
4.5. Dissemination uniformity

Assume the data collector queries the $M$ nodes located in the center of the network to obtain $M$ measurements. These nodes will experience the best disseminations due to their centrality in the network. Let $R_{cen}$ denote the average fraction of these $M$ nodes that receive a particular transmission. Next, assume the data collector gathers $M$ measurements from $M$ nodes in a network corner, and let $R_{cor}$ denote the fraction of these $M$ nodes that receive the same transmission. In order to compare the dissemination uniformity of AB and PB, let us define dissemination uniformity $\mu = E[R_{cen} - R_{cor}]$. Clearly, we are interested in a uniform dissemination, which results in $\mu \approx 0$, i.e., nodes in the corner receive the disseminations with the same probability as the nodes in the center of the network.

We find $\mu$ for PB and AB using extensive numerical simulations in Fig. 9 for a network with $N = 10^4$ nodes and $M = 700$. In PB, for each transmission range we set $p = p^*$ from Table 1. To perform a comparison between these two algorithms, we have also depicted $T_r$ and $T_{pg}$, the fraction of nodes that perform the transmission in AB and PB, respectively.

Fig. 9 confirms that the dissemination in AB is well uniform and almost the same at the corners compared to the center of the network in contrast to PB, while the number of transmissions is even smaller.

4.6. CStorage-B design

Similar to CStorage-P, in CStorage-B node $n_j$, $j \in \{1, 2, \ldots, N\}$, maintains a CS measurement $y_j$ and after dissemination $\Phi_{N \times N}^{N \times N}$ is formed in the network, except that AB replaces PB for data dissemination purpose. Consequently, the steps of CStorage-B are as follows.

1. All nodes choose $\varphi_{ji}$ from $\mathfrak{N}(0, 1)$ and initialize their measurement to $y_j = \varphi_{ji}x_i$.
2. $N_i$ nodes randomly select themselves as a source node and broadcast their reading to their neighbors with the single-bit flag set to 0.
3. Upon the reception of $x_i$ for the first time by node $l$, $n_l$ it performs the following:
   (a) Chooses $\varphi_{li}$ from $\mathfrak{N}(0, 1)$ and adds $\varphi_{li}x_i$ to $y_l$. 

Fig. 8. The expected farthest node distance $\bar{d}$ to $n_i$ in AB and ideal case using full GPS information, shown along with the transmission range $r_t$. 

Fig. 9. Dissemination uniformity, $\mu$, and the fraction of nodes that transmit in PB, $T_{pg}$, and in AB, $T_r$, versus $r_t$ and average number of neighbors in a random network.
After the transmissions are finished, \( N_i \) readings will be disseminated throughout the network. Similar to CStorage-P, a data collector queries \( M \) measurements \( y \) and the corresponding \( \phi_i \)'s from an arbitrary set of \( M \) nodes and obtains the measurement matrix \( \Phi \) (which is subset of \( \Phi_{tot} \)) and obtains \( \hat{x} \). We may rewrite Theorem 1 for CStorage-B to find the expected number of independent rows in Theorem 3 for \( N_i \) disseminations.

**Theorem 3.** Let an \( M \times N \) matrix \( \Phi \) be the measurement matrix obtained from \( \Phi_{tot} \) in CStorage-B. Further, let \( R_i \) be the fraction of nodes that receive a transmission using AB on a random network (see Fig. 7), \( r(j) \), the expected number of independent rows of \( \Phi' \) after the \( j \)th transmission (out of \( N_i \) transmissions), is given by the following:

\[
r(0) = 0,
\]
\[
r(j) = 1 - (1 - R_i)^{M - r(j - 1)} + r(j - 1), j \in \{1, 2, \ldots, N_i\}.
\]

(10)

Employing Theorem 3 (similar to CStorage-P), it is easy to shows that \( N_i \) needs to be slightly larger than \( M \) to form a measurement matrix \( \Phi \) with \( M \) independent rows (becomes full rank).

5. Performance evaluation

To perform the numerical simulations we employ the real temperature readings data sets from EPFL's SensorScope project, LUCE deployment [35]. We capture a snapshot of the network temperature on 5/1/2007 at 12:1. We will have \( N = 10^4 \) nodes randomly deployed \( A = 1 \times 1 \) and vary \( r_i \). In PB, we set \( p = p^* \) from Table 1 based on \( r_i \).

We employ the normalized reconstruction error defined by \( e = \frac{\|x - \hat{x}\|_2}{\|x\|_2} \) to evaluate the reconstruction accuracy, where \( \|\cdot\|_2 \) denotes the norm-2 of the signal. The selection of \( M \) depends on the target reconstruction error of the signal \( x \). Clearly, \( e = 0 \) denotes perfect recovery. Without loss of generality, we set the target error to \( e_t = 0.09 \) (while any other \( e_t \) may be chosen). Employing dense \( \Phi \) matrices, we observe that \( M = 2 \times 10^3 \) results in average reconstruction error of \( e \approx 0.085 \). Therefore, we fix the number of measurements to \( M = 2 \times 10^3 \). Clearly, a smaller \( e_t \) necessitates choosing a larger \( M \).

5.1. Performance evaluation of CStorage-P and CStorage-B

We theoretically showed that \( N_i \) should be slightly larger than \( M \) using Theorems 1 and 3. In our simulations, we find the value of \( N_i \) for which the desired \( \Phi \) is constructed and \( e_t \) is achieved. We remind that the dissemination phase (employing PB and AB) forms non-zero entries in the columns of \( \Phi \) corresponding to the \( N_i \) source nodes. Therefore, a larger \( N_i \) corresponds to a denser \( \Phi \), which has \( M \) independent rows with a higher probability.

We implement CStorage-P and CStorage-B, and find their respective reconstruction errors \( e_P \) and \( e_B \) by running a large number of iterations of data dissemination on randomly deployed networks in Table 2. Further, we plot the total number of transmissions in Fig. 10.

Table 2 shows that CStorage-B performs as well as CStorage-P although it is absolutely parameterless (in CStorage-P we need to set \( p = p^* \) for each \( r_i \)). Further, Table 2 confirms out theoretical results and shows that we need to set \( N_i \) slightly larger than \( M \) to achieve \( e_t \), and increasing \( N_i \) further does not improve \( e_P \) and \( e_B \) while it considerably increases the number of transmissions as shown in Fig. 10. For performance evaluation of CStorage-P versus its parameter \( p \) see [8, Fig. 6].

We can see that in CStorage-P with \( N_i = 2100 \), for average number of neighbor of 13 (minimum number for connectivity) and 37 (densely connected), we have \( N_{tot} = 5.31 \times 10^6 \) and \( N_{tot} = 2.1 \times 10^6 \), respectively. For the same network structures CStorage-B requires \( N_{tot} = 4.68 \times 10^6 \) and \( N_{tot} = 1.19 \times 10^6 \), respectively. AB requires \( 2N = 2 \times 10^4 \) transmission for hello messages to obtain the two-hop neighbor information. This increases \( N_{tot} \) to \( N_{tot} = 4.7 \times 10^6 \) and \( N_{tot} = 1.21 \times 10^6 \). Therefore, CStorage-B decreases \( N_{tot} \) by at least 11.8%, while it can automatically match to network changes.

<table>
<thead>
<tr>
<th>( N_i )</th>
<th>( e_P )</th>
<th>( e_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^2 )</td>
<td>0.371</td>
<td>0.367</td>
</tr>
<tr>
<td>500</td>
<td>0.205</td>
<td>0.208</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>0.138</td>
<td>0.138</td>
</tr>
<tr>
<td>( 1500 )</td>
<td>0.102</td>
<td>0.103</td>
</tr>
<tr>
<td>2100</td>
<td>0.084</td>
<td>0.083</td>
</tr>
<tr>
<td>( 10^4 )</td>
<td>0.084</td>
<td>0.083</td>
</tr>
</tbody>
</table>

5.2. Comparison with existing algorithms

To the best of our knowledge, there are three stateless and distributed data dissemination algorithms for large scale WSNs, simple Flooding [13], dissemination using random walks [2], and dissemination using gossiping [26]. Dissemination using gossiping has the advantage that it generates a dense \( \Phi \), which results in a large number of transmissions. Therefore, we compare the performance of CStorage with dissemination using flooding and random walks, which may generate a sparse \( \Phi \), with fixed \( r_i = 0.025 \).

Note that in dissemination using random walks, we continue the random walks until 70% of nodes have received the dissemination similar to CStorage-P. Further, we assume when a node performs the transmission in random walks all its neighbors may employ the received reading to form their measurement, although only one neighbor is selected as the next transmitter. To have a fast mixing-time and uniform dissemination we employ the Metropolis–Hastings algorithm.
with uniform equilibrium distribution to find the neighbor selection probabilities in random walk. With this setup, data dissemination using random walks in CStorage results in $N_{\text{tot}} = 8.4 \times 10^7$.

The simplest dissemination algorithm is the simple Flooding [13], which results in $N_{\text{tot}} = N^2 = 2.1 \times 10^7$ transmissions when used along with CS. Clearly, if CS is not employed all $N$ readings must be stored in all $N$ nodes resulting in $N_{\text{tot}} = N^3 = 10^8$ transmissions. Therefore, employing CS reduces the number of transmissions from $10^8$ to $2.1 \times 10^7$, and CStorage-P and CStorage-B further reduces $N_{\text{tot}}$ to $3.27 \times 10^6$ and $2.83 \times 10^6$, respectively. Therefore, we can see that CStorage-P and CStorage-B have reduced the number of transmissions about one order of magnitude, which can considerably increase the life time of the network.

6. Conclusion

In this paper, we proposed two distributed data storage algorithms using compressive sensing (CS) referred to by CStorage-P and CStorage-B. These algorithms are distributed and are suitable for WSNs where no routing tables may be obtained. In CStorage-P, the readings of randomly selected network nodes are disseminated throughout the networks using probabilistic broadcasting (PB) to form CS measurements at nodes. After the dissemination phase, a data collector may query a small arbitrary set of nodes to recover all readings.

CStorage-P has a parameter that needs to be tuned based on network parameters. Hence, it may not be scalable and flexible to network changes. Therefore, we designed a novel parameterless data dissemination algorithms referred to by alternating branching (AB) that requires two-hop neighbor information at nodes. AB can automatically tune to network changes and requires less number of transmissions compared to PB. We discussed the advantages of CStorage-P and CStorage-B and showed that they can greatly decrease the total number of transmissions for data storage compared to existing stateless algorithms.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. ECCS-1418710.

References


Ali Talari received his MSc degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2006. Ali was a Ph.D. student in the School of Electrical and Computer Engineering at Oklahoma State University since January 2006 to December 2012. His research interests are novel error control coding techniques, communications theory, signal processing in wireless sensor networks, and compressive sensing techniques.

Nazanin Rahnnavard (S97-M10) received her B.S. and M.S. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1999 and 2001, respectively. She then joined the Georgia Institute of Technology, Atlanta, GA, in 2002 where she received her Ph.D. degree in the School of Electrical and Computer Engineering in 2007. Dr. Rahnnavard joined the School of Electrical and Computer Engineering at Oklahoma State University as an Assistant Professor in August 2007. In 2014, Dr. Rahnnavard joined the Department of Electrical Engineering & Computer Science at University of Central Florida as an associate professor. Her current research interests lie in the area of telecommunications and signal processing with a special focus on wireless ad-hoc and sensor networks, cognitive radio networks, modern error-control coding schemes, and compressive sensing. She is the recipient of Outstanding Research Award from the Center for Signal and Image Processing at the Georgia Institute of Technology in 2007. She serves on the editorial board of the Elsevier Journal on Computer Networks (COMNET) and on the Technical Program Committee of several international IEEE conferences.