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 $\underline{x} = [x_1 x_2 \ldots x_N]^T$ ($x_i \in \mathbb{R}$) 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 Ψ 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 Ψ, only $M \ll N$ ($M \geq O(K \log N)$) measurements $\underline{y} = [y_1 y_2 \ldots y_M]^T$ 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 Φ 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

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

where $\|\underline{\theta}\|_1 = \sum_{i=1}^{N} |\theta_i|$. The 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 Ψ is *dense* and *orthonormal*, e.g., Fourier transform basis, a *sparse* Φ also satisfies CS requirements on Φ [9,10]. Therefore, in this paper we employ sparse Φ matrices, since as we later see they can be formed with a much smaller number of transmissions. Further, the selection of Ψ 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 Ψ 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

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

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* Φ matrices can satisfy CS requirements and designed a data storage algorithm based on these sparse Φ matrices. Further, authors in [6,7,22–25] proposed *centralized* data collection algorithms where measurements are formed enroute and are

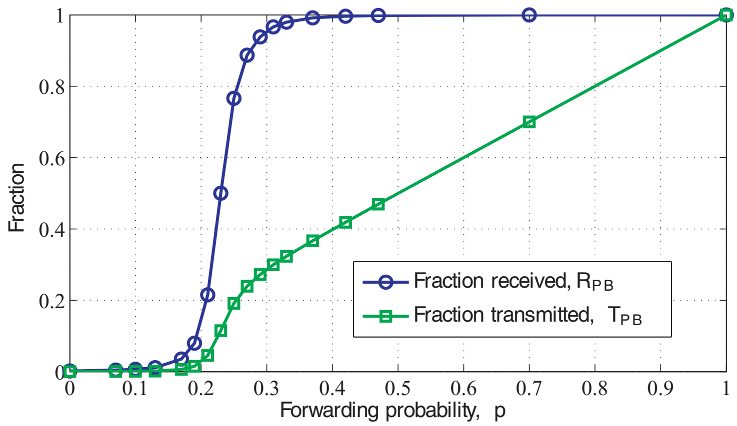

Fig. 1. The fraction of nodes receiving a transmission $R_{PB}(p)$ and fraction of nodes the perform the transmissions $T_{PB}(p)$ versus forwarding probability p in PB.

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 $y_j = \phi_j \underline{x}$ and ϕ_j is an N-dimensional row vector and $\underline{x} = [x_1 x_2 \ldots x_N]^T$ 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 $\underline{y}_{tot} = [y_1 y_2 \ldots y_N]^T$. Further, let $\varphi_{j,\,i}$ be the element at the jth row and the ith column of Φ_{tot}. The matrix Φ_{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, Φ_{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 ϕ_j maintained at each node, and forms $\underline{y} \in \mathbb{R}^M$ and $\Phi \in \mathbb{R}^{M \times N}$. Next, the data collector obtains $\hat{\underline{x}}$, an estimate of $\underline{x}$, employing basis pursuit by solving (1).

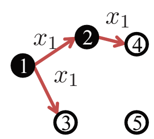

Fig. 2. Network with $N = 5$ and n_1 transmitting x_1 employing PB. The transmitting nodes are depicted with a dark color.

3.1. CStorage-P design

The CStorage-P to form Φ_{tot} in the network is described in the following.

1. All nodes choose $\varphi_{j,\,j}$ from $\mathfrak{N}(0, 1)$ and initialize their measurement to $y_j = \varphi_{j,\,j} x_j$, where $\mathfrak{N}(0, 1)$ is the zero mean and unit variance Gaussian distribution.
2. N_s nodes randomly select themselves as a source node and broadcast their reading to their neighbors.
3. Upon reception of x_i for the *first time* by node l, n_l, performs the following:
   (a) Chooses $\varphi_{l,\,i}$ from $\mathfrak{N}(0, 1)$ and adds $\varphi_{l,\,i} x_i$ to y_l.
   (b) *Broadcasts* x_i with probability p (PB).

To describe CStorage-P, let us consider a small network with $N = 5$ nodes as shown in Fig. 2 and investigate one PB of CStorage-P. At the beginning, we have $\varphi_{i,\,j} = 0$ for all $i \neq j, i, j \in \{1, \ldots, 5\}$. Assume n_1 broadcasts x_1. Since n_2 and n_3 are in the transmission range of n_1, they would receive x_1. Nodes n_2 and n_3 multiply x_1 by $\varphi_{2,\,1}$ and $\varphi_{3,\,1}$ and add them to y_2 and y_3, respectively. Next, 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 PB of x_1 is over and the matrix Φ_{tot} obtains the form of (3). As we can also see from Φ_{tot} that x_1 (corresponds to the 1st column of Φ_{tot}) contributes to CS measurements y_1, y_2, y_3, and y_4. The same procedure is performed for N_s source nodes selected *uniformly at random* to form Φ_{tot}. Therefore, column j of Φ_{tot} corresponds to dissemination of x_j, sensor reading of node j, and row i of

Φ_{tot} corresponds to the measurements formed at node i.

$$\Phi_{tot} = \begin{pmatrix} \varphi_{1,1} & 0 & 0 & 0 & 0 \\ \varphi_{2,1} & \varphi_{2,2} & 0 & 0 & 0 \\ \varphi_{3,1} & 0 & \varphi_{3,3} & 0 & 0 \\ \varphi_{4,1} & 0 & 0 & \varphi_{4,4} & 0 \\ 0 & 0 & 0 & 0 & \varphi_{5,5} \end{pmatrix} \quad (3)$$

3.2. Suitable values of N_s and p

As shown in [9], a sparse Φ matrix can be used to recover a signal with same order of number of measurements as a dense measurement matrix if Φ has M independent rows (is full rank). In other words, any M rows of Φ_{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_s and p such that the collected M rows of Φ_{tot} form a sparse Φ with the aforementioned properties while N_{tot}, the total number of transmissions for the N_s disseminations, is minimized. If $T_{PB}(p)$ denotes the fraction of network nodes that perform the retransmission in a PB with forwarding probability p (see Section 2.2), each PB requires $T_{PB}(p)N$ transmissions. Therefore, we have $N_{tot} = T_{PB}(p)NN_s$. In the following theorem from [8], we find the expected number of independent rows of Φ as a function of N_s and p.

Theorem 1 ([8]). *Let an $M \times N$ matrix Φ be the measurement matrix obtained from Φ_{tot} in CStorage-P. Further, let $R_{PB}(p)$ 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 Φ after the jth transmission (out of N_s transmissions), is given by the following*:

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

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

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.

Table 1
Transmission range r_t, and the corresponding average number of neighbors and p^* in random WSNs.

r_t	No. of neighbors	p^*
0.021	13	0.38
0.022	16	0.32
0.024	18	0.28
0.026	21	0.25
0.027	24	0.22
0.029	27	0.19
0.031	30	0.17
0.033	33	0.16
0.034	37	0.14

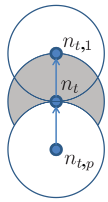

Fig. 3. Structure of AB algorithm, where the current transmitter, n_t, selects one next transmitter, $n_{t,1}$.

4.1. Issues with PB algorithm

Consider the nodes in Fig. 3, where n_t is about to rebroadcast a reading x_i (for instance using PB). Let $n_{t,p}$ be the *parent* of n_t, from which n_t has received x_i. Clearly, all nodes in $\mathcal{N}(n_{t,p})$ have received x_i, where $\mathcal{N}(n_{t,p})$ denotes the set of one-hop neighbors of $n_{t,p}$. When n_t performs the transmission, nodes in the gray shaded area of Fig. 3 receive x_i potentially for the first time.

Clearly, to greedily minimize the total number of transmissions, the distance of n_t to $n_{t,p}$ (hence the size of gray area in Fig. 3) should be maximized [29]. However, since in PB, n_t *blindly* makes the forwarding decision regardless of its distance to $n_{t,p}$, n_t may be positioned close to $n_{t,p}$ and its transmission may be redundant. This is the first issue in PB that results in redundant transmissions.

Authors in [29] proposed to employ the location of nodes obtained by GPS to find a node n_t that has the maximum distance with $n_{t,p}$. However, GPS may be unavailable in many WSNs, while *two-hop neighbor* information can be easily obtained at nodes. Therefore, we design *alternating branching* dissemination algorithm that takes advantage of the two-hop neighbor information to find nodes that are *possibly the farthest* from the current transmitter exploiting their neighbor information. This resolves the first issue of PB.

The second issue with PB is that the *local density* of neighbors is not included in the calculation of p. Therefore, nodes in network corners, close to borders, and in sparse regions of network receive less number of transmissions. Although, there have been several work that propose to locally tune p, they still have a parameter that needs tuning based on network wide information. Authors in [30] propose *SmartGossip*, which has several parameters (γ, T, μ_1, μ_2, σ, and δ) that are tuned based on network parameters. In algorithms proposed in [31,32], nodes need to be aware of the *network diameter* to 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_i 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_i 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_i, the next transmitter of n_t is selected from $\mathcal{N}(n_t)\backslash\mathcal{N}(n_{t,p})$, where \ 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

$$n_{t,1} = \text{argmin}_{n_{t,l}} |\mathcal{N}(n_{t,l}) \bigcap \mathcal{N}(n_{t,p})|, \quad (5)$$

where $\bigcap$ 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_i that initiates the broadcast of x_i and assume all its neighbors rebroadcast x_i. 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_i 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_i 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.

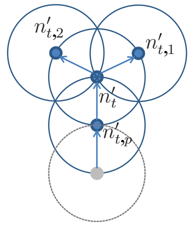

Fig. 4. Structure of AB algorithm, where the current transmitter, n_t, selects two next transmitters, $n_{t,1}$ and $n_{t,2}$.

Let $n'_{t,1}, n'_{t,2} \in \mathcal{N}(n'_t)\backslash\mathcal{N}(n_{t,p})$ 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

$$n'_{t,1}, n'_{t,2} = \text{argmin}_{n'_{t,1},n'_{t,2}} |\mathcal{N}(n'_{t,1}) \bigcap \mathcal{N}(n'_{t,2}) \bigcap \mathcal{N}(n_{t,p})|, \quad (6)$$

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. To control the branching, we propose to include a *single-bit binary counter* as *branching flag* along with x_i. 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_i 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_i but it has received it before, the branch has been chosen from an area where x_i 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_i initiates the broadcast of x_i 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 toward 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*.

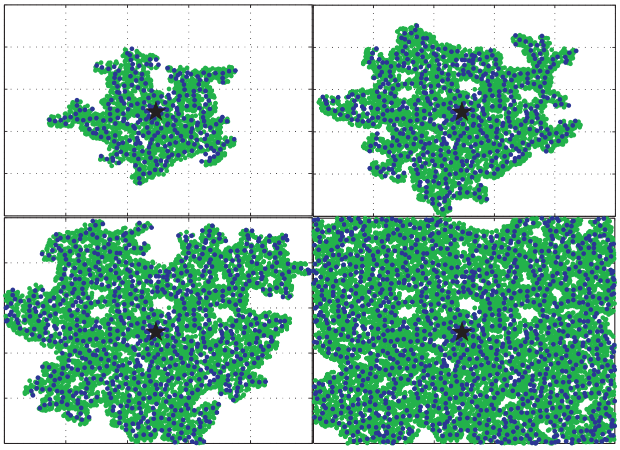

Fig. 5. Dissemination of a reading from the source node at the center (shown by a star) using AB. The dark colored nodes are the transmitters forming branches, the light colored nodes are the nodes that receive the reading, the white areas are the nodes that do not receive the transmission. Figures belong to the same dissemination in progressive snap times from left to right and up to down, until the dissemination is complete.

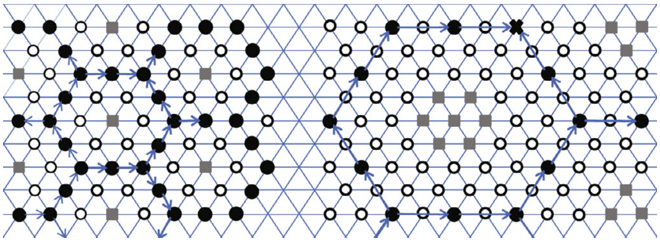

Fig. 6. Ideal implementation of AB that results in isometric grid. The transmitters are shown with filled black circles, nodes that receive the transmission but do not retransmit are shown by hollow circle, the nodes that do not receive the transmission are shown by gray square (in the center of hexagons 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

Lemma 1. *In ideal AB on an isometric grid, for transmission range $r_g \in \{1, 2, \ldots\}$, 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*

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

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*

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

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

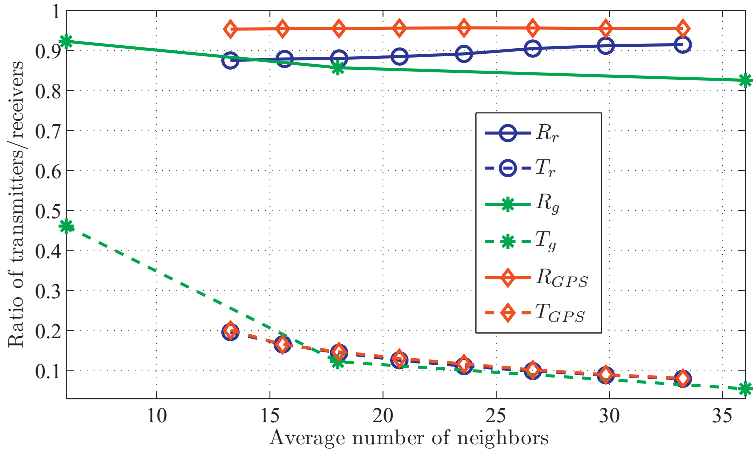

Fig. 7. The average fraction of nodes receiving and transmitting in AB dissemination. R_r, R_g, and R_{GPS} denote the fraction of receivers in AB, AB on isometric grid, and AB_{GPS}, respectively. Further, T_r, T_g, and T_{GPS} denote the fraction of transmitters in AB, AB on isometric grid, and AB_{GPS}, respectively.

by $6\sum_{i=1}^{r_g} i \in \{6, 18, 36, \ldots\}$. 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 AB provides almost *constant* fraction of receivers and transmitters despite drastic changes in network topology. Therefore, if the network changes over time AB *automatically* adapts to changes. This is in contrast to PB where R_{PB} and T_{PB} are greatly affected by p. In addition, from Fig. 7 we can observe that although AB performs very close to AB_{GPS} (ideal setup), while it eliminates the need for GPS information.

We can also observe that as the transmission range increases, AB becomes more efficient and T_r reduces while R_r increases, and its performance approaches that of AB_{GPS}. However, in ideal grid network the fraction of receivers drops as the networks becomes denser. In addition, Fig. 7 shows that isometric grid analysis of AB can provide close estimates for R_r and T_r. This shows that AB performs close to grid model on random networks although the neat hexagon shaped cells may not appear due to random placement of nodes.

4.4. Distance between transmitters in random networks

It is of interest to find the *expected* distance of n_t with the next hops, $\bar{d}$, in AB since next transmitters are not necessarily placed on the border of n_t. Clearly, we expect $\bar{d}$ to be as close as possible to r_t. Let us first compare $\bar{d}$ in AB and AB_{GPS} in Fig. 8. We remind that AB_{GPS} maximizes $\bar{d}$ using exact nodes locations while AB uses two-hop neighbor information (which is a very limited information compared to exact nodes location) to perform the same task.

Fig. 8 confirms that AB can perform very close to ideal case, AB_{GPS}. Therefore, we may assume AB is maximizing the distance of the next transmitter to n_t. Using this result, we can analytically find $\bar{d}$. For the sake of simplicity, let us assume the transmission range of n_t is unit, i.e., $r_t = 1$, and find $\bar{d}$. Let X_i be a random variable indicating the Euclidian distance of n_t with a neighbor in $\mathcal{N}(n_t)$. The pdf and cdf of X_i are given by $f_{X_i}(d) = 2d, 0 \le d \le 1$ and $F_{X_i}(d) = d^2$ [29]. The following lemma gives the pdf of a random variable defined as maximum of several random variables.

Lemma 2. *Let $X_i, i = \{1, 2, \ldots, k\}$ be i.i.d. random variables with the same cdf $F_X(d)$, and let the random variable $X_{max} = \max\{X_1, \ldots, X_k\}$. $F_{X_{max}}(d)$ the cdf of X_{max} is $F_{X_{max}}(d) = F_X^k(d)$.*

Proof. $F_{X_{max}}(d) = P(X_{max} \le d) = P(X_1 \le d, \ldots, X_k \le d) = P(X_1 \le d) \ldots P(X_k \le d) = F_X^k(d)$. □

Node n_t maximizes the distance of the next forwarders from the set $\mathcal{N}(n_t)\backslash\mathcal{N}(n_{t,p})$ located in the gray area in Fig. 3. The size of the shaded region is $A_{sel} = \frac{\bar{d}}{2}\sqrt{4-\bar{d}^2} + 2\arcsin\frac{\bar{d}}{2}$ [29].

The number of nodes in the shaded area is given by $N_{sel} = \frac{N}{A}A_{sel} = \rho A_{sel}$, where $\rho = \frac{N}{A}$ is the density of nodes. Let $X_{max} = \max\{X_1, \ldots, X_k\}$ be the random variable denoting the distance of next transmitters to n_t. Using Lemma 2, we have $F_{X_{max}}(d) \approx d^{2N_{sel}}$. Consequently, the expected distance $\bar{d}$ is simply obtained by $\bar{d} = E[X_{max}]$, where $E[.]$ 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 $E[Z] = \int_0^\infty (1 - F_Z(x))dx - \int_{-\infty}^0 F_Z(x)dx$. This gives

$$\begin{aligned}\bar{d} &= E[X_{max}] = \int_0^1 (1 - z^{2N_{sel}})dz,\\ &= 1 - \frac{1}{2N_{sel}+1}\\ &= 1 - \frac{1}{2\rho\left[\frac{\bar{d}}{2}\sqrt{4-\bar{d}^2} + 2\arcsin\frac{\bar{d}}{2}\right] + 1}\end{aligned}$$

After a few simple mathematical operations, we obtain

$$\rho = \frac{\bar{d}}{(1-\bar{d})\left(\bar{d}\sqrt{4-\bar{d}^2} + 4\arcsin\frac{\bar{d}}{2}\right)}. \quad (9)$$

The value of $\bar{d}$ may be obtained from (9) for any ρ. 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.

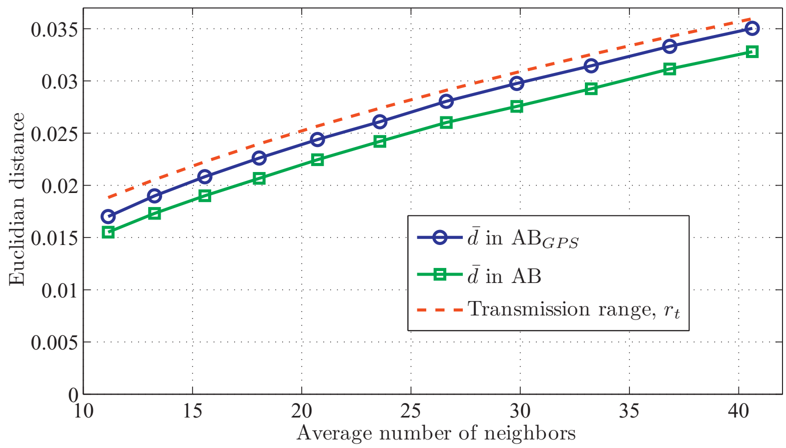

Fig. 8. The expected farthest node distance $\bar{d}$ to n_t in AB and ideal case using full GPS information, shown along with the transmission range r_t.

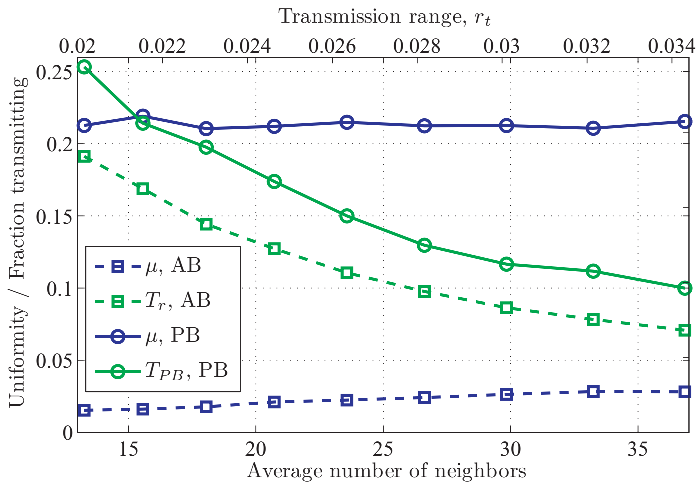

Fig. 9. Dissemination uniformity, μ, and the fraction of nodes that transmit in PB, T_{PB}, and in AB, T_r, versus r_t and average number of neighbors in a random network.

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 μ 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_{PB}, 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 $\Phi_{tot}^{N\times N}$ 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 $\varphi_{j,j}$ from $\mathfrak{N}(0,1)$ and initialize their measurement to $y_j = \varphi_{j,j}x_j$.
2. N_s 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 $\varphi_{l,i}$ from $\mathfrak{N}(0,1)$ and adds $\varphi_{l,i}x_i$ to y_l.

(b) Checks to see if it has been selected as a next forwarder or is a direct neighbor of source node, n_i. If either of aforementioned conditions is met, it checks x_i's single-bit flag. If the flag is 0, it chooses one next transmitter using (5), or otherwise chooses two next transmitters using (6). Finally, it flips the single-bit flag and *rebroadcasts* x_i along with the flag and the ID of the next forwarder(s) (AB).

After the transmissions are finished, N_s readings will be disseminated throughout the network. Similar to CStorage-P, a data collector queries M measurements $\underline{y}$ and the corresponding ϕ_j's from an arbitrary set of M nodes and obtains the measurement matrix Φ (which is subset of Φ_{tot}) and obtains $\hat{\underline{x}}$. We may rewrite Theorem 1 for CStorage-B to find the expected number of independent rows in Theorem 3 for N_s disseminations.

Theorem 3. *Let an $M \times N$ matrix Φ be the measurement matrix obtained from Φ_{tot} in CStorage-B. Further, let R_r 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 Φ' after the jth transmission (out of N_s transmissions), is given by the following*:

$$\begin{aligned} r(0) &= 0, \\ r(j) &= 1 - (1 - R_r)^{M - r(j-1)} \\ &\quad + r(j-1),\ j \in \{1, 2, \ldots, N_s\}. \end{aligned} \tag{10}$$

Employing Theorem 3 (similar to CStorage-P), it is easy to shows that N_s needs to be slightly larger than M to form a measurement matrix Φ 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_t. In PB, we set $p = p^*$ from Table 1 based on r_t.

We employ the normalized reconstruction error defined by $e = \frac{\|\underline{x} - \hat{\underline{x}}\|_2}{\|\hat{\underline{x}}\|_2}$ to evaluate the reconstruction accuracy, where $\|.\|_2$ denotes the *norm*-2 of the signal. The selection of M depends on the target reconstruction error of the signal $\underline{x}$. 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 Φ 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_s should be slightly larger than M using Theorems 1 and 3. In our simulations, we find the value of N_s for which the desired Φ 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 Φ corresponding to the N_s source nodes. Therefore, a larger N_s corresponds to a denser Φ, which has M independent rows with a higher probability.

Table 2
The temperature signal reconstruction error in CStorage-P and CStorage-B denoted by e_P and e_P, respectively, for *all* r_t.

N_s	e_P	e_B
10^2	0.371	0.367
500	0.205	0.208
10^3	0.138	0.138
1500	0.102	0.103
2100	0.084	0.083
10^4	0.084	0.083

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_t). Further, Table 2 confirms out theoretical results and shows that we need to set N_s slightly larger than M to achieve e_t, and increasing N_s 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 $N_s = 2100$, 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.

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 Φ, 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 Φ, with fixed $r_t = 0.025$.

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

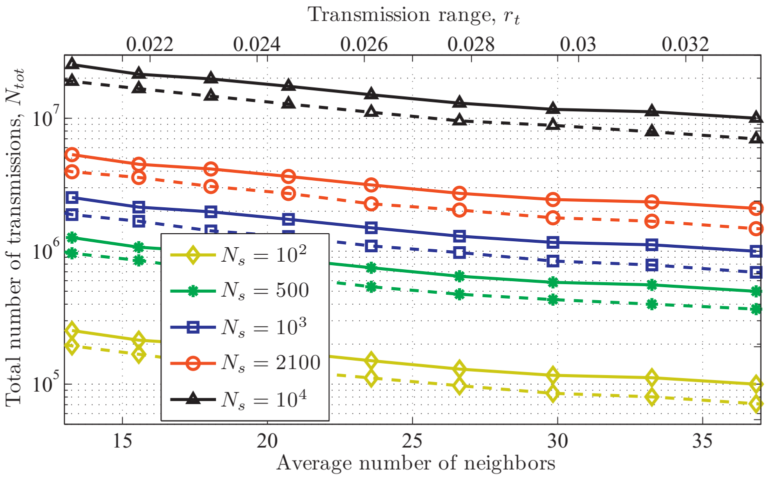

Fig. 10. The total number of transmissions N_{tot} versus average number of neighbors and r_t in CStorage-P and CStorage-B shown by solid and dashed lines, respectively.

[36] 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 $N_{tot} = 8.4 \times 10^7$.

The simplest dissemination algorithm is the simple Flooding [13], which results in $N_{tot} = N_s N = 2.1 \times 10^7$ transmissions when used along with CS. Clearly, if CS is not employed all N readings must be stored in all N nodes resulting in $N_{tot} = N^2 = 10^8$ transmissions. Therefore, employing CS reduces the number of transmissions from 10^8 to 2.1×10^7, and CStorage-P and CStorage-B further reduces N_{tot} to 3.27×10^6 and 2.83×10^6, 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

[1] A. Dimakis, K. Ramchandran, Y. Wu, C. Suh, A survey on network codes for distributed storage, Proc. IEEE 99 (3) (2011) 476–489.
[2] Z. Kong, S. Aly, E. Soljanin, Decentralized coding algorithms for distributed storage in wireless sensor networks, IEEE J. Sel. Areas Commun. 28 (2) (2010) 261–267.
[3] D. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[4] E. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2) (2006) 489–509.
[5] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, M. Zorzi, Data acquisition through joint compressive sensing and principal component analysis, Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM, 2009, pp. 1–6.
[6] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, A. Ortega, Compressed Sensing and Routing in Multi-Hop Networks, Technical Report, University of Southern California, 2009.
[7] C. Chou, R. Rana, W. Hu, Energy efficient information collection in wireless sensor networks using adaptive compressive sensing, in: Proceedings of the IEEE 34th Conference on Local Computer Networks (LCN 2009), 2009.
[8] A. Talari, N. Rahnavard, CStorage: distributed data storage in wireless sensor networks employing compressive sensing, Proceedings of the 2011 IEEE Global Telecommunications Conference (GLOBECOM 2011), 2011, pp. 1–5.
[9] W. Wang, M. Garofalakis, K. Ramchandran, Distributed sparse random projections for refinable approximation, in: Proceedings of International Conference on Information Processing in Sensor Networks, 2007, pp. 331–339.
[10] E. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory 52 (12) (2006) 5406–5425.
[11] P. Gupta, P. Kumar, Critical power for asymptotic connectivity, in: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 1, IEEE, 1998, pp. 1106–1110.
[12] Z. Haas, J. Halpern, L. Li, Gossip-based ad hoc routing, in: Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, INFOCOM 2002, vol. 3, 2002, pp. 1707–1716.
[13] C. Ho, K. Obraczka, G. Tsudik, K. Viswanath, Flooding for reliable multicast in multi-hop ad hoc networks, in: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999, pp. 64–71.

[14] N. Rahnavard, B. Vellambi, F. Fekri, CRBcast: a reliable and energy-efficient broadcast scheme for wireless sensor networks using rateless codes, IEEE Trans. Wirel. Commun. 7 (12 Part 2) (2008) 5390–5400.
[15] N. Rahnavard, F. Fekri, CRBcast: a collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks, in: Proceedings of International conference on Information processing in Sensor Networks, 2006, p. 283.
[16] H. Shen, Z. Li, K. Chen, A scalable and mobility-resilient data search system for large-scale mobile wireless networks, IEEE Trans. Parallel Distrib. Syst. 25 (5) (2014) 1124–1134.
[17] C. Wang, P. Cheng, Z. Chen, N. Liu, L. Gui, Practical spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks, in: Proceedings of the 2015 IEEE 81st Vehicular Technology Conference, VTC Spring, IEEE, 2015, pp. 1–6.
[18] X.-Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A.V. Vasilakos, M.-Y. Wu, CDC: Compressive data collection for wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 26 (8) (2015) 2188–2197.
[19] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, S. Xiao, Data gathering with compressive sensing in wireless sensor networks: a random walk based approach, IEEE Trans. Parallel Distrib. Syst. 26 (1) (2015) 35–44.
[20] X. Yang, X. Tao, E. Dutkiewicz, X. Huang, Y.J. Guo, Q. Cui, Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding, IEEE Trans. Wirel. Commun. 12 (10) (2013) 5087–5099.
[21] B. Gong, P. Cheng, Z. Chen, N. Liu, L. Gui, F. de Hoog, Spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks, IEEE Commun. Lett. 19 (5) (2015) 803–806.
[22] L. Xiang, J. Luo, A. Vasilakos, Compressed data aggregation for energy efficient wireless sensor networks, in: Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011, pp. 46–54.
[23] C. Luo, F. Wu, J. Sun, C. Chen, Efficient measurement generation and pervasive sparsity for compressive data gathering, IEEE Trans. Wirel. Commun. 9 (12) (2010) 3728–3738.
[24] C. Luo, F. Wu, J. Sun, C.W. Chen, Compressive data gathering for large-scale wireless sensor networks, in: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MobiCom'09, 2009, pp. 145–156.
[25] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, M. Zorzi, On the interplay between routing and signal representation for compressive sensing in wireless sensor networks, in: Proceedings of Information Theory and Applications Workshop, 2009, pp. 206–215.
[26] M. Rabbat, J. Haupt, A. Singh, R. Nowak, Decentralized compression and predistribution via randomized gossiping, in: Proceedings of International Symposium on Information Processing in Sensor Networks, 2006, pp. 51–59.
[27] A. Dimakis, V. Prabhakaran, K. Ramchandran, Decentralized erasure codes for distributed networked storage, IEEE Trans. Inf. Theory 52 (6) (2006) 2809–2816.
[28] Y. Lin, B. Liang, B. Li, Data persistence in large-scale sensor networks with decentralized fountain codes, in: Proceedings of the 26th IEEE International Conference on Computer Communications, INFOCOM, 2007, pp. 1658–1666.
[29] M. Heissenbuttel, T. Braun, M. Walchli, T. Bernoulli, Optimized stateless broadcasting in wireless multi-hop networks, in: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM, 2006, pp. 1–12.
[30] A. Kini, V. Veeraraghavan, N. Singhal, S. Weber, SmartGossip: an improved randomized broadcast protocol for sensor networks, in: Proceedings of the 5th International Conference on Information Processing in Sensor Networks, 2006, pp. 210–217.
[31] P. Kyasanur, R. Choudhury, I. Gupta, Smart gossip: an adaptive gossip-based broadcasting service for sensor networks, in: Proceedings of the 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems, MASS, 2006, pp. 91–100.
[32] B. Bako, F. Kargl, E. Schoch, M. Weber, Advanced adaptive gossiping using 2-hop neighborhood information, in: Proceedings of the IEEE GLOBECOM 2008 Global Telecommunications Conference, 2008, IEEE, 2008, pp. 1–6.
[33] L. Orecchia, A. Panconesi, C. Petrioli, A. Vitaletti, Localized techniques for broadcasting in wireless sensor networks, in: Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing, 2004, pp. 41–51.
[34] A. Durresi, V. Paruchuri, S. Iyengar, R. Kannan, Optimized broadcast protocol for sensor networks, IEEE Trans. Comput. 54 (8) (2005) 1013–1024.
[35] Available online: http://sensorscope.epfl.ch/index.php, Accessed May 2014.
[36] S. Boyd, P. Diaconis, L. Xiao, Fastest mixing Markov chain on a graph, SIAM Rev. 46 (4) (2004) 667–689.

Ali Talari received his MS 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 2008 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 Rahnavard (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. Rahnavard joined the School of Electrical and Computer Engineering at Oklahoma State University as an Assistant Professor in August 2007. In 2014, Dr. Rahnavard joined and 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.