# Distributed Data Aggregation for Sparse Recovery in Wireless Sensor Networks

Shuangjiang Li, Hairong Qi
Electrical Engineering and Computer Science
University of Tennessee, Knoxville, TN 37996
{shuangjiang, hqi}@utk.edu

***Abstract*—We consider the approximate sparse recovery problem in multi-hop Wireless Sensor Networks (WSNs) using Compressed Sensing/Compressive Sampling (CS). The goal is to recover the $n$-dimensional data values by querying only $m \ll n$ sensors based on some linear projection of sensor readings. To solve this problem, a distributed compressive sparse sampling (DCSS) algorithm is proposed based on sparse binary CS measurement matrix. Each sensor first samples the environment independently, then the fusion center (FC), acting as a pseudo-sensor, samples the sensor network to select a subset of sensors ($m$ out of $n$) that respond to the FC through shortest path for data recovery purpose. The sparse binary matrix is designed using the unbalanced expander graph which achieves the state-of-the-art performance for CS schemes. This binary matrix can be interpreted as a sensor selection matrix whose fairness is analyzed. Extensive experiments on both synthetic and real data sets show that by querying only the minimum amount of $m$ sensors using the DCSS algorithm, the CS recovery accuracy outperforms existing sparse random matrices and can be as good as those using random dense measurement matrices but using much less number of sensors. We also show that the sparse binary measurement matrix works well on compressible data which has the closest recovery result to the known best $k$-term approximation. The recovery is robust against noisy measurements and does not require regular WSN deployments (e.g., grids). The sparsity and binary properties of the measurement matrix contribute, to a great extent, the reduction of the in-network communication cost as well as the computational burden.**

***Keywords*—Data Aggregation; Distributed Compressed Sensing; Sparse Binary Matrix; Sparse Recovery; Expander Graph;**

## I. Introduction

Wireless Sensor Networks (WSNs) consist of a number of nodes which measure the environmental changes. In many applications, aggregate functions of the sensor data are more important than individual node data. When the scale of the network is large, the value collaboratively collected from the whole network is more important than the value collected by one single node, as stated in [1]. One would expect more data communications as well as heavy computation to achieve this goal, while in a typically resource-limited wireless sensor network, it is imperative that we develop effective data aggregation techniques, that accommodate both the communication and computation constraints in the network.

Traditional approaches for data acquisition first collect the entire signal and then process it for compression and transmission, or storage. In comparison, the *Compressed Sensing/Compressive Sampling (CS)*-based approaches obtain directly a nonadaptive linear measurement of the entire signal. A $k$-sparse signal $x \in \mathbb{R}^n$ (i.e., $x$ contains at most $k$ non-zero elements) can then be accurately recovered from the linear measurement vector $y = \Phi x \in \mathbb{R}^m$ [2], [3] with $m \ll n$, where $\Phi$ is often referred to as the *measurement matrix*. Current CS-based approaches have several problems that prevent them from practical deployment in WSNs. First, in CS, $\Phi$ is usually a random dense matrix (e.g., Gaussian, Scrambled Fourier matrices, etc.) that induces the dense measurement problem (i.e., each measurement in CS is a linear combination of many, if not all, samples of the signal to be reconstructed). This would require a lot of inter-communications between sensors [4]. Second, most of the data routing algorithms rely on the measurement matrix $\Phi$ where each row of $\Phi$ is treated as a routing path. In theory, CS requires the measurement matrix $\Phi$ to be incoherent with the sparsifying basis (e.g., DCT, wavelet), building the measurement matrix based on the routing path lacks theoretic foundation for CS reconstruction and causing inefficiency. Finally, these CS-based data routing for aggregation usually requires regular WSN deployments (e.g., grids), which makes them less practical.

In this paper, we study the CS-based sparse recovery problem in multi-hop WSNs on arbitrary sensor deployment, where each individual sensor first samples the environment to obtain sensor readings and then the Fusion Center (FC), acting as a pseudo-sensor, samples the sensor network based on compressive sampling (e.g., how many sensor measurements are needed), to obtain a subset of sensors ($m$ out of $n$) for sparse recovery. Thus the traditional problem that sensing will cause data deluge is well alleviated.

In summary, the main contributions of this paper are as follows:

Firstly, a Distributed Compressive Sparse Sampling (DCSS) algorithm is developed that efficiently selects and aggregates the sensor reading. The FC chooses $m$ designated sensors within the network based on the CS measurement matrix, which provides a necessary number of $m$ measurements for accurate data recovery. The distributed algorithm enjoys the sparsity of the measurement matrix that reduces the communication cost and relieves the computational burden of the network while also solving the dense measurement problem.

Secondly, a sparse binary matrix based on the unbalanced expander graph is proposed. Although such measurement matrix has been developed for network streaming and only recently in CS [5], this paper represents the first attempt in designing the sparse binary matrix with a fixed number of 1's

in each column of the measurement matrix such that it can be applied in WSNs for data recovery purpose. Theoretical analysis shows the guaranteed sparse recovery result.

Lastly, based on the analysis of the average bit-hop cost for DCSS and comparison with several CS-based data routing and aggregation algorithms, we demonstrate that DCSS has a low intercommunication cost which paves the way for practical CS-based sparse recovery in WSNs.

The rest of this paper is organized as follows. In Section II, we discuss some related works on CS-based data aggregation. In Section III, we briefly review the background of CS, the sparse recovery problem, and the measurement matrix properties. Section IV formally describes the definition of expander graphs and the connection between the CS and the expander graph. Then we show how to design the proposed sparse binary matrix. Furthermore, we explain that the sparse binary matrix can be used as the sensor subset selection matrix with proved fairness. Section V introduces the DCSS algorithm. We conduct experiments in Section VI using both synthetic and real datasets tested on both exact $k$-sparse signal and compressible signals. Section VII concludes this paper.

## II. Related Work

One of the first papers studied CS for WSNs is [6], where all sensors transmit their readings directly to the FC through a single hop communication. However, this approach was not feasible for WSN platforms due to the limited resources. In [7], plain CS aggregation was applied on the large-scale WSN, but [8] demonstrated the disadvantage of plain CS aggregation in terms of throughput. The dense sampling problem had been tackled by many active researchers later on but only a few proposals applied CS to multi-hop networking. Interestingly, Wang et al. [9] showed that the remarkable results of CS could also be obtained using sparse random projection (SPR). The measurement matrix for SRP is defined as:

$$\Phi_{ij} = \sqrt{s} \begin{cases} +1 & \text{if} \quad p = \frac{1}{2s} \\ -1 & \text{if} \quad p = \frac{1}{2s} \\ 0 & \text{if} \quad p = 1 - \frac{1}{s}, \end{cases} \tag{1}$$

where $s$ is a parameter that determines the sparseness of the projection and $p$ is the probabilty.

Quer et al. in [10] investigated the routing cost for CS aggregation in multi-hop WSNs where the measurement matrix was defined according to the routing path. They also demonstrated the benefits of CS in realistic multi-hop WSNs. However, this method was only applicable to grid networks. Random routing based CS measurement matrix was studied in [11] under the same networking assumption. [12] utilized the spatially-localized sparse projections based on the observation that the measurement matrix had to take the sparse domain into account. The routing-dependent or domain-dependent design of the measurement matrix, unfortunately, contradicts to the nature of CS that measurement matrix can be random and easily generated. Neither of these measurement matrices in [7], [10]–[12] can recover CS aggregated data from arbitrarily deployed networks which is one of the key issues we address in this paper.

A remarkable recent work by Sartipi and Fletcher [4] proposed a compressive distributed sensing using random walk (CDS(RW)). Their measurement matrix is said to be independent of routing algorithm and network topologies using rateless coding, which shares the same design characteristic as our DCSS algorithm. In Section VI we will mainly compare the recovery results with SRP and the intercommunication cost with CDS(RW).

## III. Background on Compressive Sampling

The basic compressed sensing problem is to estimate a vector $x \in \mathbb{R}^n$ from a set of linear measurements $y = \Phi x$, where $y \in \mathbb{R}^m$ and $\Phi$ is a known $m \times n$ matrix. This method greatly reduces the number of digital samples required to reconstruct from highly incomplete information, typically well below the number expected from the requirements of the Shannon/Nyquist sampling theorem. The key idea in compressed sensing is that if the signal $x$ is constrained to be sparse or approximately sparse, then it is possible to recover $x$ even when $m \ll n$. More precisely, one of the basic results in compressed sensing is that there exist matrices $\Phi$ with only $m = \mathbb{O}(k \log(n/k))$ rows such that for all $k$-sparse $x$, i.e., all $x$ with at most $k$ nonzero components, we can recover $x$ exactly from $y = \Phi x$. Furthermore, it has been observed that recovery can be accomplished in polynomial time via linear programming (LP), provided that the measurement matrix $\Phi$ satisfies certain technical conditions [3], [13], [14]. In the following, we briefly describe the theories and properties related to compressive sampling.

### A. Sparse/Compressible Signal Model

We categorize the general signals used in this paper into two classes.

*1) $k$-sparse signals:* We say that an $n$-dimensional signal $x$ is $k$-sparse if it has $k$ or fewer non-zero components:

$$\|x\|_0 := |\text{supp}(x)| \leq k \ll m, x \in \mathbb{R}^n$$

where $|\text{supp}(x)|$ denotes the cardinality of the support set of $x$, and thus $\|x\|_0$, namely the number of non-zero components, is a quasi-norm.

*2) Compressible signals:* We consider a real data vector $x \in \mathbb{R}^n$, and fix an orthonormal transform $\Psi \in \mathbb{R}^{n \times n}$ consisting of a set of orthonormal basis vectors $\{\psi_1, \cdots, \psi_n\}$. $\Psi$ can be, for example, a Wavelet or a Fourier transform. The transform coefficients $\Theta = [\psi_1^T x, \cdots, \psi_n^T x]^T$ of the data can be ordered in magnitude, so that $|\theta_1| \geq |\theta_2| \geq \cdots \geq |\theta_n|$. The best $k$-term approximation keeps the largest $k$ transform coefficients and discards the remaining as zero. The approximation error is $\|x - \hat{x}\|_2^2 = \|\Theta - \hat{\Theta}\|_2^2 = \sum_{i=k+1}^n \|\theta_i\|^2$.

According to CS literature [2], [3], the data is compressible if the magnitude of its transform coefficients decay like a power law. That is, the $i$th largest transform coefficient satisfies $|\theta_i| \leq L i^{-\frac{1}{p}}$, where $L$ is some positive constant, and $0 < p \leq 1$. Note that $p$ controls the compressibility (or rate of decay) of the transform coefficients (i.e., smaller $p$ implies faster decay). The approximation error, is then

$$\|x - \hat{x}\|_2 = \|\Theta - \hat{\Theta}\|_2 \leq \alpha_p L k^{-1/p+1/2}$$

where $\alpha_p$ is a constant that only depends on $p$. Of course, sparse signals are special cases of compressible signals. Figure 1 shows the power-law decay curve and the $k$-term approximation with respect to the sorted index of the coefficients.

### B. Measurement Matrices and Sparse Recovery Problem

The measurement matrix, $\Phi$, is very important since it largely affects the process of sampling the system as well as determining how good the recovery is. We could also consider the measurement matrix as a *sampling operator* that preserves some suitable conditions under which the signal can be recovered exactly or with high probability. Recent years, a lot of good papers have already advanced the research on this topic [15], [16].

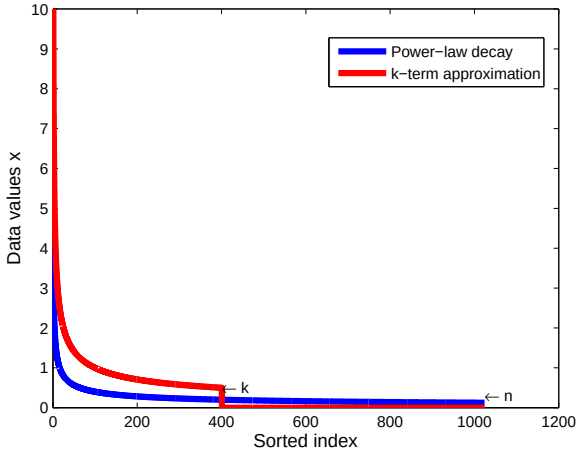


Fig. 1: The power-law decay curve and $k$-term approximation.

One of the most commonly used conditions for the measurement matrix is the *Restricted Isometry Property (RIP)* introduced by Candes and Tao [17]. RIP essentially requires that every subset of column of $\Phi$ with certain cardinality approximately behaves like an orthonormal system. For an $m \times n$ matrix $\Phi$ and an integer $k$, $1 \ll k \ll p$, the measurement matrix $\Phi$ should satisfy the restricted isometry property.

**Definition:** An $m \times n$ matrix $\Phi$ has the $k$-restricted isometry property ($k$-RIP) with constant $\delta_k > 0$ if, for all $x \in \mathbb{R}^n$,

$$(1 - \delta_k)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_k)\|x\|_2^2 \qquad (2)$$

In other words, the $k$-RIP ensures that all submatrices of $\Phi$ of size $m \times k$ are close to an isometry, and therefore preserving distance and information. Practical recovery algorithms typically require that $\Phi$ has a slightly stronger $2k$-RIP, $3k$-RIP, or higher-order RIP in order to preserve distances between $k$-sparse vectors (which are $2k$-sparse in general), three-way sums of $k$-sparse vectors (which are $3k$-sparse in general), and other higher-order structures, respectively.

It is important to note that RIP conditions are difficult to verify for a given matrix $\Phi$. A widely used technique for avoiding checking the RIP directly is to generate the matrix randomly and to show that the resulting random matrix satisfies the RIP with high probability using the well-known Johnson-Lindenstrauss Lemma. See, for example, Baraniuk, et al. [16]. This is typically done for conditions involving only the restricted isometry constant $\delta_k$.

We consider the sparse recovery problem as the recovery of the $k$-sparse signal $x \in \mathbb{R}^n$ from its measurement vector $y = \Phi x \in \mathbb{R}^m$. If a matrix $\Phi$ satisfies certain RIP property, then the recovery process can be accomplished by finding a vector $x^*$ using the following linear program:

$$\min \|x^*\|_1 \quad \Phi x^* = \Phi x. \qquad \text{(Sparse Recovery)}$$

## IV. Design of the Compressive Sampling Matrix

In this section, we will first give the definition on the expander graph and the connection between the expander graph and the CS measurement matrix. We then describe the procedure on how to design the sparse binary matrix that resembles the expander graph with high probability. Lastly, some practical issues on using the sparse binary matrix as sensor subset selection matrix are discussed.

### A. Expander Graph

**Definition:** A $(k, \epsilon)$-*unbalanced expander* is a bipartite graph $G = (U, V, E)$, (i.e., edges only exist between nodes in set $U$ and set $V$; no edges exists between two nodes in set $U$ or two nodes in set $V$), $|U| = n$, $|V| = m$ with left degree $d$ (i.e., each node in $U$ is connected to $d$ nodes in $V$) such that for any $X \subset U$ with $|X| \leq k$, the set of neighbors $N(X)$ of $X$ (i.e., nodes in $V$ that are connected to nodes in $X$) has size $|N(X)| \geq (1 - \epsilon)d|X|$, for all $0 < \epsilon < 1$.

For example, the bipartite graph $G = (U, V, E)$ in Figure 2 is called a $(k, \epsilon)$-*expander* if for any subset of left nodes $X$, with cardinality $|X| \leq k$, they are connected to at least $(1-\epsilon)|E(X)|$ right-hand side nodes (namely the neighbors of $X$, denoted by $N(X)$), where $|E(X)| = d|X|$ is the set of links that go from $X$ to the right-hand side nodes. In other words, if we define the bi-adjacency matrix $T = [t_{ik}]$ where $t_{ik} = 1$ iff node $i$ in $U$ is connected to node $k$ in $V$, then $|E(X)|$ is the total number of nonzero elements in the columns corresponding to $X$ in the bi-adjacency matrix of the bipartite graph $G$, $|N(X)|$ is the number of nonzero elements in rows in the bi-adjacency matrix.

### B. Connection between Expander Graph and CS Matrices

We consider the CS matrices that are *binary* and *sparse*. They have only a small number (i.e., $d$) of 1's in each column, and all the other entries are equal to zero. It has been shown recently [18] that such matrices cannot satisfy the RIP property with parameters $k$ and $\delta$, unless the number of rows is $\Omega(k^2)$. Recent result in [19] demonstrated that such matrices satisfy a different form of the RIP property, namely RIP-$p$ property if for any $k$-sparse vector $x$, $\|\Phi x\|_p = (1 \pm \delta)\|x\|_p$ holds. In particular, it shows that this property holds for $1 \leq p \leq 1 + \mathbb{O}(1)/\log n$ if matrix $\Phi$ is an adjacency matrix of a high-quality *unbalanced expander graph*, where "high-quality" refers to an expander with $\epsilon$ as small as possible, creating an expansion as large as possible.

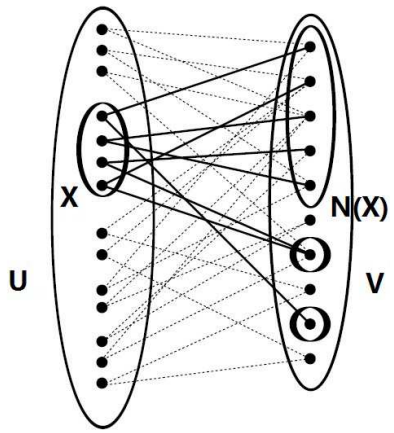


Fig. 2: An unbalanced expander graph: any sufficiently small subset $X$ on the left has a neightborhood $N(X)$ of size at least $(1-\epsilon)d|X|$.

The relationship between the adjacency matrix, the $(k,\epsilon)$-unbalanced expander graph, and the RIP-$p$ property of the adjacency matrix have been extensively studied by Berinde et al. in [19]. The main results (see Theorem $1, 2, 3$ in [19]) stated that: (1) The adjacency matrix of a $(k,\epsilon)$-unbalanced expander graph satisfies the RIP-$p$ property, which can be used for guaranteed CS recovery problem in (Sparse Recovery). (2) Any binary matrix $\Phi$ with each column having $\tau$ 1's and satisfying the RIP-$p$ property with proper parameters, must be an adjacency matrix of a good unbalanced expander. That is, a RIP-$p$ matrix and the adjacency matrix of an unbalanced expander are essentially equivalent. (3) If $\Phi$ is an adjacency matrix of an expander graph, then the Linear Programming (LP) decoding procedure can be used for recovering sparse approximations and provide recovery guarantees for the problem (Sparse Recovery).

### C. CS Matrix Design

We will now begin to give the procedure on how to design the binary sparse matrices to serve as the measurement matrix used in compressed sensing. A binary sparse matrix $\Phi$ of $m$ rows and $n$ columns is generated in the following steps:

- Step 1: For each column, randomly generate $\tau$ integers whose values are between 1 and $m$ and place 1's in those rows indexed by the $\tau$ numbers;
- Step 2: If the $\tau$ numbers are not distinct, repeat Step 1 until they are (this is not really an issue when $\tau \ll m$).

Based on the above facts and with some proper value of $\tau$ (e.g., $\tau = 8$ in our experiments), we see that such a matrix is the adjacency matrix of an expander graph of degree $\tau$ with high probability.

### D. Binary Matrix for Sensor Subset Selection

*1) Sensor Subset Selection Problem:* By definition, the sensor subset selection problem is to choose a set of $m$ sensor measurements, from a set of $n$ possible or potential sensor measurements. Solving this problem by evaluating the performance for each of the $C_n^m$ possible choices of sensor measurements is not practical unless $m$ and $n$ are small. Broadly speaking, this problem belongs to traditional feature selection problem.

A large class of algorithms have been developed that search for optimal solutions (e.g., exhaustive search, branch and bound, genetic algorithm) and deterministic suboptimal solutions (e.g., sequential forward selection, sequential backward selection). We now consider the zero-one matrix, $\Phi$, with elements $\{\phi_{ij}\}$ and the feature set $F$ with elements, $\{f_j\}$, $j = 1, 2, \cdots, n$. Defining $\phi_{ij} = 1$ if we select the set element $f_j$ and $\phi_{ij} = 0$ otherwise. In the sensor subset selection problem, we view $\sum_{i=1}^{m} \phi_{ij} f_j$, $j = 1, 2, \cdots, n$ as the selection of $m$ sensor measurements from totally $n$ number of sensors. Thus the sparse binary measurement matrix naturally severs as a sensor selection matrix.

*2) Fairness of the Subset Selection Process:* Since the sparse binary projection matrix we proposed can also be seen as a sensor subset selection matrix for choosing $m$ designated sensors to communicate with the FC, it is important to evaluate its fairness in sensor selection to avoid leaving "holes" in the network due to extensive energy consumption for computation and communication on only a few sensor nodes. In this case, a balanced energy consumption among all the sensor nodes is obvious since the proposed sparse binary projection matrix has exactly $\tau$ 1's along each column confirming that every sensor nodes in the WSN will be selected to be active $\tau$ times during the entire process of distributed data aggregation for recovery of $n$-dimensional data value $x$. The only imbalance energy consumption comes from these designated sensors which compute and report the projections to the FC through shortest path. However, since $m \ll n$ and the measurement matrix can be re-generated each time a query task is performed, in the long run, the energy consumption can still be balanced.

## V. Distributed Compressive Sparse Sampling Algorithm

In this section, we first define the network model for Distributed Compressive Sparse Sampling (DCSS) algorithm followed by the detailed algorithm description.

### A. Network Model

Consider a wireless network of $n$ sensors with diameter $d$ hops. The FC is placed at the center of the deployment area with average distance of nodes from the FC of $\mathcal{O}(d)$ hops. We consider that sensor nodes are time synchronized and are densely deployed in a homogeneous large-scale WSN, where sensor data tends to be correlated in both time and space. Assume each of the $n$ nodes acquires a sample $x_i$, $i = 1, 2, \cdots, n$, which is $k$-sparse or compressible. The ultimate goal for the sparse recovery problem in WSN is to gather sufficient information to recover the $n$-dimensional signal (i.e., sensor readings) $\mathbf{x} = [x_1, \cdots, x_n]^T$ at the FC while minimizing some other criteria (e.g., intercommunication cost, energy).

According to this network model, we assign each sensor node with an ID, $s_1, \cdots, s_n$. Based on the CS theory, the FC only needs to receive $m$ measurements to recover the readings of all sensors, we thus assign $m$ designated sensors based on the sparse binary matrix and these designated sensors marked as $D_1, \cdots, D_m$, will send there measurements to the FC.

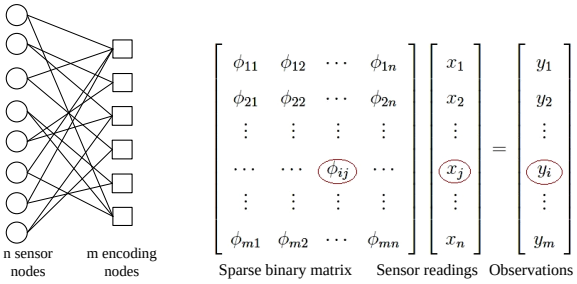


Fig. 3: (a) Sensor readings projection can be seen as generating a bipartite graph between the $n$ data nodes and the $m$ encoding nodes. (b) Example of observations from sensor data projection using sparse binary matrix.

### B. DCSS Algorithm

Consider an $m \times n$ sparse binary matrix $\Phi$ with entries defined in Section IV. Each sensor will compute and store the inner product $\sum_{j=1}^{n} \Phi_{ij} x_j$ between the aggregate data $x$ and one row of $\Phi$. See Figure 3 (a), we think of this as generating a bipartite graph between the $n$ data nodes and the $m$ encoding nodes. Figure 3 (b) shows that the $m$ encoding nodes can be used as the $m$ observations to reconstruct an approximation of $x$. The $m$ encoding nodes correspond to the $m$ designated sensors in the network. Below is the description of the DCSS algorithm.

**Algorithm 1** Distributed Compressive Sparse Sampling Algorithm (DCSS)

**Input:**
Sparse binary matrix $\Phi \in \mathbb{R}^{m \times n}$;
Environmental reading $\mathbf{x} \in \mathbb{R}^n$;
**Output:**
Recovered vector $\mathbf{x}^* \in \mathbb{R}^n$;

1: The fusion center (FC) generates a sparse binary matrix $\Phi$ and each sensor node in the network stores it locally;
2: For each individual sensor $s_j$, $(1 \leq j \leq n)$ , if $\phi_{ij} \neq 0$, then sensor node $s_j$ sends its reading $x_j$ to designated sensor ID $D_i$ through shortest path;
3: For the $m$ designated sensors $D_1, \cdots, D_m$, each computes and stores the sum of the reading it receives;
4: Finally, the $m$ designated sensors send their results to the FC through shortest path and the FC performs compressed sensing recovery for $\mathbf{x}^*$.

From the above DCSS algorithm, we see that for each individual sensor, $\phi_{ij} \cdot x_j$ means that sensor $s_j$ sends its readings $x_j$ to the designated sensor $D_i$, if $\phi_{ij} \neq 0$. Correspondingly, the designated sensor $D_i$ computes and stores the summation of the sensor reading it receives, and the $m$ designated sensors then report $y_1, y_2, \cdots, y_m$, as observations $y \in \mathbb{R}^m$ to the FC. This process is shown in Figure 4. In this figure three projections are initialized using the sparse binary matrix to aggregate data to the designated sensor nodes $D_1, D_2$, and $D_3$. For example, sensor node $s_2, s_4$, and $s_{22}$ send their readings through shortest path to designated sensor $D_2$ (marked by dashed lines). Node $D_2$ then adds the value of all the three sensor readings it received and sends the generated projection to the FC through the shortest path marked with solid line. The same procedure is applied to the designated sensors $D_1$ and $D_3$. After collecting all the $m$ observations, the FC will recover the original signal $\mathbf{x} \in \mathbb{R}^n$ using Linear Programming.

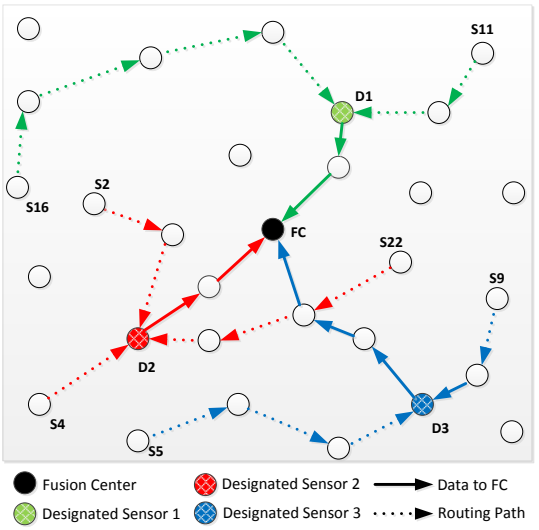


Fig. 4: Example of the DCSS algorithm using sparse binary matrix through shortest path.

TABLE I: Communication cost of different CS algorithms.

| Algorithm | Cost |
|---|---|
| DS | $\mathcal{O}(kn \log n)$ |
| SRP [9] | $\mathcal{O}(kd \log^2 n)$ |
| CDS(RW) [4] | $\mathcal{O}(k(t+d) \log n)$ |
| Sparse Binary | $\mathcal{O}(k(\tau\omega + d) \log n)$ |

### C. Communication Cost

It is proven that independent and identically distributed random Gaussian and Scrambled Fourier matrices can be used as measurement matrices for CS [2]. The projections generated by these measurement matrices are called Dense Sampling (DS), since a majority of the entries in $\Phi$ are non-zero. Assuming that the original signal can be recovered using $m = \mathcal{O}(k \log n)$ measurements, based on our network model (i.e., average bit-hop cost per reading), the communication cost for obtaining $\mathcal{O}(k \log n)$ projections using DS is $\mathcal{O}(kn \log n)$. Also, it was shown in [4] that the communication cost under the same network model for SRP is $\mathcal{O}(kd \log^2 n)$, and the cost of CDS(RW) is $\mathcal{O}(k(t+d) \log n)$, where $t$ is the mixing time of random walk and can be found through simulation.

In order to analyze the communication cost for the proposed DCSS algorithm, we assume $\omega$ to be the average row weight of the sparse binary measurement matrix, $\tau\omega$ as the cost of gathering the sensor readings for each projection (i.e., $\sum_{j=1}^{n} \Phi_{ij} x_j$), and additionally, $d$ as the cost to send the projection to the FC. Thus, for the generation of $\mathcal{O}(k \log n)$ projections the total cost is $\mathcal{O}(k(\tau\omega + d) \log n)$. Table I summarizes the communication cost of the aforementioned algorithms.

## VI. Experiments

In this section, we analyze the performance of the various schemes on both synthetic and real datasets. The experimental design is described first.

### A. Experimental Design

(1) **Data aggregation schemes:** The random Gaussian or Scrambled Fourier measurement matrices are often used in the CS community as Dense Sampling measurement matrices (DS). Throughout the experiments, we use scrambled Fourier as the DS matrix, similar results have also been obtained on random Gaussian matrices. We also consider Sparse Random Projections (SRP) via shortest path. It has been shown that with $s = \frac{n}{\log n}$ and $\mathcal{O}(k \log n)$ projections, the reconstruction quality is as good as with obtaining the largest $k$-approximation [9] given that the data is bounded by the peak-to-total energy condition (i.e., for signal $\mathbf{u}$, $\frac{\|\mathbf{u}\|_\infty}{\|\mathbf{u}\|_2} \leq \mathcal{C}$, for some constant $\mathcal{C}$ ). We compare results from using the sparse binary matrix with SPR with $s = 64$ and $s = 128$, respectively.

For the synthetic data experiment part, we test the recovery of the exact $k$-sparse signals, noisy sparse signals with different noise levels, and compressible signals. In the real experiment, the real data are collected from a WSN deployed in the Intel Berkeley Research Laboratory. The network recorded temperature, humidity, light and voltage measurements at 31-sec intervals. In our experiments, we use the light intensity readings at node 19. The data values are represented by wavelet coefficients to obtain a compressible representation. Throughout the experiments, we assume the data value $\mathbf{x}$ we want to recover is in 1024-dimension. For the CS recovery algorithm processed in the FC, we use linear programming method in the popular $\ell_1$-magic package [20].

(2) **Evaluation metrics:** The reconstruction quality is measured using the relative reconstruction error defined as follows:

$$\varepsilon = \frac{\|\mathbf{x} - \mathbf{x}^*\|_2^2}{\|\mathbf{x}^*\|_2^2}, \tag{3}$$

where $\mathbf{x}$ is the value of the original signal, while $\mathbf{x}^*$ is the reconstructed signal.

### B. Exact Sparse Signal Recovery

We consider a sparse vector $\mathbf{x}$ with length $n = 1024$ contains only $\pm 1$, and set the sparsity level $k = 30$ (i.e., $\mathbf{x}$ only has 30 non-zero entries).

In Figure 5 we show the recovery error $\varepsilon$ as a function of the number of measurements $m$. We run the recovery process 100 times with the same sparse signal $\mathbf{x}$ each round but with different measurement matrices. The result demonstrates that the sparse binary matrix is as good as the dense sampling matrices in terms of recovery performance, while sparse random project with sparseness parameter $s = 128$ performs the worst.

### C. Noisy Sparse Signals Recovery

We consider the same sparse signal $\mathbf{x}$ but with different noise levels that occurred during the sensor data recording and data transmission stage. In our experiments we manually add

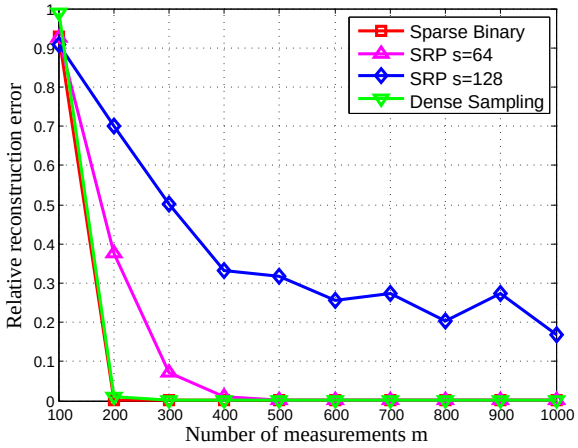


Fig. 5: Recovery result of an $n = 1024$, sparsity $k = 30$ sparse signal $\mathbf{x}$, with an average of 100 experiments using LP recovery method.

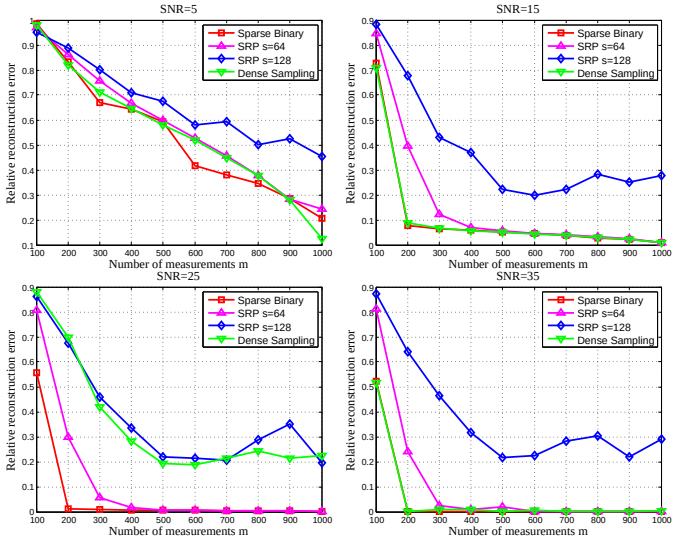


Fig. 6: Noisy recovery result of an $n = 1024$, sparsity $k = 30$ sparse signal $\mathbf{x}$, with different SNRs $(5, 15, 25,$ and $35)$ and an average of 100 experiments using LP recovery method evaluated by different measurements.

different noise levels to the sparse signal $\mathbf{x}$. We show in Figure 6 the recovery performance of noisy signals with different number of measurements $m$. We use four different noise levels with the signal to noise ratio (SNR) being $5, 15, 25,$ and $35$, respectively. The results are obtained by an average of 100 experiments using the same noisy signal $\mathbf{x}$ in each round. We observe that the higher the noise level, the worse the recovery accuracy for all data aggregation schemes. It is true because CS normally requires the sparsity feature of the signals while adding more noise causes the signal to be less sparse. SPR with larger sparseness parameter (i.e., $s = 128$) generally performs worse than with smaller $s$. Since large $s$ means less sensor reading projections transmitted to the FC which will affect the CS sparse recovery performances. For dense sampling, the recovery performance is not very stable on various SNRs, e.g., for the case when SNR $= 25$, DS performs much worse than

our sparse binary matrix, while in all other cases, DS can be as good as our proposed sparse binary matrix. We also see that the sparse binary matrix performs the best among all schemes in the most noisy case when SNR$= 5$.

### *D. Compressible Signal Recovery*

While a strict sparse signal is rare in reality, we conduct experiments on compressible signals which obey a decay signal model, $x = 4n^{-\frac{7}{10}}$. We then sample the signal $x$ to get $n = 1024$ data points. We compare the recovery result using different matrices and various measurements. The best $k$-term approximation is obtained by only keeping the first $k$ coefficients while setting the rest to zero. Based on the compressible signal model in Section III, it can serve as a baseline of the recovery performance evaluation. Figure 7 demonstrates that sparse binary matrices perform the best for compressible signals in the sense that it has the closest recovery result to the known best $k$-term approximation. Although dense matrices also achieve good performance, this only occurs when the number of the measurements arrives at certain amount (e.g., $m = 900$).

### *E. Experiment using Real Data*

In this experiment, we consider the dataset [21] for temperature, humidity, and light readings from a group of nodes deployed at the offices of Intel Research Labs in Berkeley, CA. The signals were recorded in an office environment and therefore exhibit periodic behavior caused by the activity levels during day and night. Therefore, we expect the signals to be compressible in the wavelet domain.

We consider the recovery from CS measurements for the light intensity signal at node 19. We obtain different number of CS measurements for the signal using the sparse binary matrix, sparse random projections, and dense sample matrices. Figure 8 plots the recovered signal using 100 coefficients for wavelet approximation and 400 coefficients for dense sampling matrix, sparse binary matrix, and sparse random projections. We observe that wavelet approximation gives us more smooth signal profile than the other measurement matrices. However, the recovered SNR is not the best. Dense sampling and sparse binary matrices all perform well on recovering the details around data point position 200 and 500 and sparse binary matrix has the recovered SNR equals to 25.5702, which is the highest among the three. SRP has the worst SNR since the peak-to-noise condition is violated for real data. Table II shows the detailed SNRs of the recovered signal.

We also evaluate the recovery performance using the relative approximation error metric. Figure 9 shows the result compared to various recovery methods. Since the proposed sparse binary matrix also serves as the sensor selection matrix, we conduct experiment to compare the sensor selection result versus random sensor selection. The experiment shows that the sparse binary matrix outperforms all other methods when

TABLE II: Recovered SNR of different measurement matrices.

| Methods | Wavelet approx. | Sparse binary | Dense sampling | SRP ($s = 64$) |
|---|---|---|---|---|
| SNR | 21.4735 | 25.5702 | 22.8528 | 10.9653 |

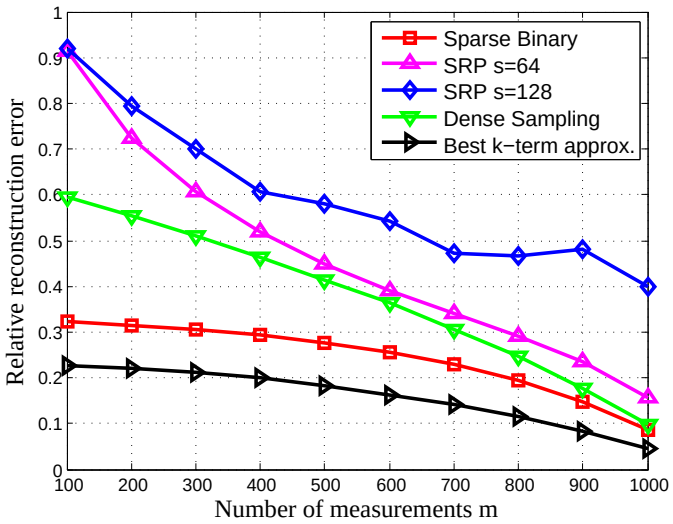


Fig. 7: Recovery result of a sampled compressible signal $x = 4n^{-\frac{7}{10}}$, with an average of 100 experiments using LP recovery method evaluated by different measurements.

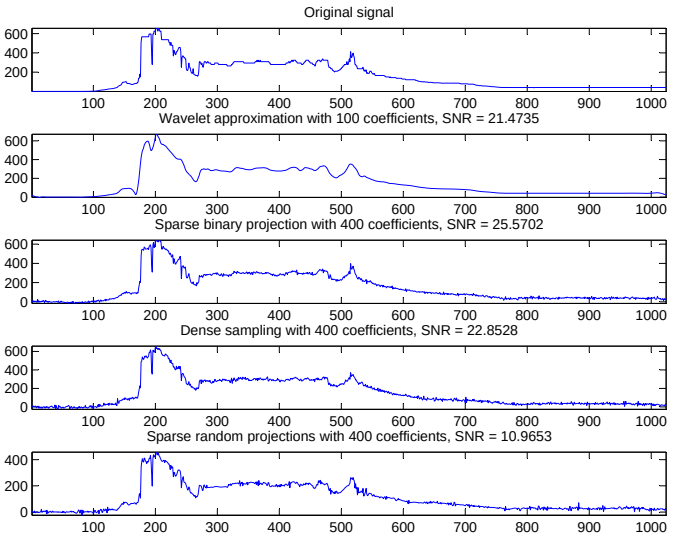


Fig. 8: Recovery result of real Intel lab signal using 100 wavelet coefficients and 400 CS measurements with different measurement matrices.

the number of measurements is less than 300 except for the wavelet approximation. When the number of measurements increases, the sparse binary matrix has the closest recovery results with random selection sampling, which in turn demonstrates the fairness of our sparse binary matrix on sensor selection in Section III. The random selection of sensors works good when the number of measurements is around 700 and performs poorly when the number of measurements is either too small or too large. For SRP, the recovery fails when the measurements equal to 100 and 200. We also observe an interesting relation between the relative recovery error and the number of measurements for sparse random projections, dense sampling, and random selection sampling, where increasing the number of measurements actually degrades the recovery per-

formance. The reason is that the real signal in the transformed wavelet domain is not sparse enough for reliable CS recovery.

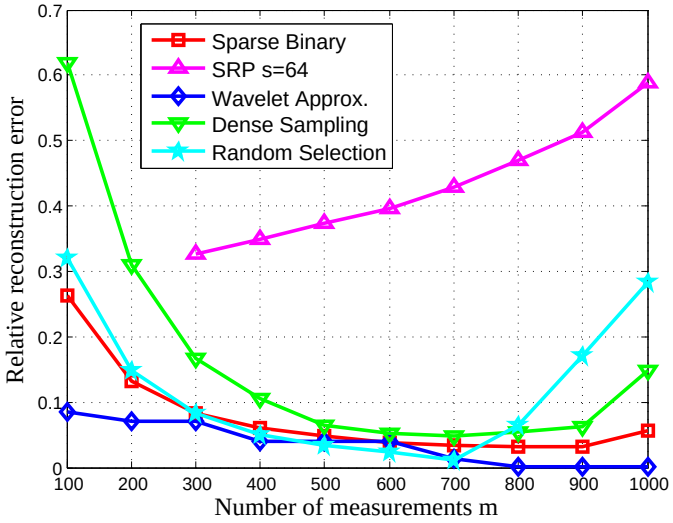


Fig. 9: Recovery result of real Intel lab signal, with an average of 100 experiments using LP recovery method evaluated by different measurements.

## VII. Conclusion

In this paper, we presented a novel distributed compressed sensing algorithm based on sparse binary matrix for data recovery in WSNs. We constructed a sparse binary matrix that resembles the unbalance expander graph with high probability for CS measurement matrix. We showed that the sparse binary matrix can be used for sensor selection for distributed data aggregation to the designated sensors in the network while keeping intercommunication cost very low. The experimental results showed that using the sparse binary matrix, the accuracy of recovery can be as good as the traditional dense CS measurement matrices and the proposed sparse binary measurement matrix worked the best on compressible data. Results from both synthetic and real data experiments demonstrated the usefulness of the proposed distributed CS algorithms using sparse binary matrix for both sensor selection and sparse recovery.

## References


[1] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] C. E.J. and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.

[3] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] M. Sartipi and R. Fletcher, "Energy-efficient data acquisition in wireless sensor networks using compressed sensing," in *Data Compression Conference (DCC), 2011*. IEEE, 2011, pp. 223–232.

[5] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank, "Efficient and robust compressed sensing using optimized expander graphs," *Information Theory, IEEE Transactions on*, vol. 55, no. 9, pp. 4299–4308, 2009.

[6] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Joint source–channel communication for distributed estimation in sensor networks," *Information Theory, IEEE Transactions on*, vol. 53, no. 10, pp. 3629–3653, 2007.

[7] C. Luo, F. Wu, J. Sun, and C. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 145–156.

[8] L. Xiang, J. Luo, and C. Rosenberg, "Compressed data aggregation: Energy-efficient and high-fidelity data collection," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, p. 1, 2012.

[9] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 331–339.

[10] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "On the interplay between routing and signal representation for compressive sensing in wireless sensor networks," in *Information Theory and Applications Workshop, 2009*. IEEE, 2009, pp. 206–215.

[11] X. Wang, Z. Zhao, Y. Xia, and H. Zhang, "Compressed sensing for efficient random routing in multi-hop wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 7, no. 3, pp. 275–292, 2011.

[12] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega, "Spatially-localized compressed sensing and routing in multi-hop sensor networks," *Geosensor Networks*, pp. 11–20, 2009.

[13] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[14] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[15] T. Cai, L. Wang, and G. Xu, "New bounds for restricted isometry constants," *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4388–4394, 2010.

[16] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[17] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, pp. 4203–4215, 51 2005.

[18] V. Chandar, "A negative result concerning explicit matrices with the restricted isometry property," *preprint*, 2008.

[19] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 798–805.

[20] E. Candes and J. Romberg, "$\ell_1$-magic: Recovery of sparse signals via convex programming," 2005.

[21] S. Madden, "Intel lab data. web page," http://db.lcs.mit.edu/labdata/labdata.html, 2004, [Online; Accessed on 1/12/2013].