Novel Light Weight Compressed Data Aggregation Using Sparse Measurements for IoT Networks

Amarlingam M, Pradeep Kumar Mishra, P Rajalakshmi,
Sumohana S. Channappayya, and C. S. Sastry[†]

Department of Electrical Engineering

[†] *Department of Mathematics*

Indian Institute of Technology Hyderabad,
Kandi, Sanagareddy-502285

Abstract

Optimal data aggregation aimed at maximizing IoT network lifetime by minimizing constrained on-board resource utilization continues to be a challenging task. The existing data aggregation methods have proven that compressed sensing is promising for data aggregation. However, they compromise either on energy efficiency or recovery fidelity and require complex on-node computations. In this paper, we propose a novel Light Weight Compressed Data Aggregation (LWCDA) algorithm that randomly divides the entire network into non-overlapping clusters for data aggregation. The random non-overlapping clustering offers two important advantages: 1) energy efficiency, as each node has to send its measurement only to its cluster head, 2) highly sparse measurement matrix, which leads to a practically implementable framework with low complexity. We analyze the properties of our measurement matrix using restricted isometry property, the associated coherence and phase transition. Through extensive simulations on practical data, we show that the measurement matrix can reconstruct data with high fidelity. Further, we demonstrate that the LWCDA algorithm reduces transmission cost significantly against baseline approaches,

[*] Amarlingam M
Email address: `ee13p1003@iith.ac.in` (Indian Institute of Technology Hyderabad, Kandi, Sanagareddy-502285)

implying thereby the enhancement of the network lifetime.

Keywords: Compressed sensing, data aggregation, Internet of Things, network lifetime.

1. Introduction

Internet of Things (IoT) is a communication paradigm that benefits society in many ways by providing vast and diverse applications [1]. In IoT, Wireless Sensor Network (WSN) takes an active part in the integration of physical world information. WSN encompasses huge number of tiny devices called sensor nodes which are deployed in the sensing area of interest to measure the data. IoT enables access to a wide variety of WSNs for providing a plethora of services to the citizens [1]. In most of the IoT applications such as remote sensing and environmental monitoring, seamless data aggregation from the sensor nodes is the fundamental requirement for data processing (data analytics) to facilitate the user with a useful interface and ubiquitous access to the network data.

The sensor nodes used in IoT application deployments are typically, inexpensive, untethered and are powered through batteries [2]. However, relying on battery power limits the lifetime of the nodes. Further, regular recharging or replacement of batteries leads to additional cost and is a laborious task [3]. Thus, the network lifetime is a critical concern for data aggregation in IoT networks. Wireless transmission consumes significant amount of energy during the data aggregation [4]. Indeed, reducing the number of packet transmissions and minimizing routing path for data aggregation in the network can improve the network lifetime. Several approaches have been proposed to address this problem [5].

Compressed Sensing (CS) [6] is a signal processing technique that has been proven to be very promising for data aggregation [7]. CS provides a new perspective for data aggregation in IoT networks enabling the compression and route minimization jointly for energy efficiency over the network [8]-[11]. Most of the CS aided data aggregation techniques use either dense [8]-[11] or sparse random measurements [12]-[15]. These methods have proposed the encoding by utilizing the structural properties of the measurement matrix. In dense random measurements based data aggregation techniques, it is assumed that the individual columns of the measurement matrix are generated at the respective nodes and compute the corresponding measurement [8]-[11]. The sparse random measurements based data aggregation techniques computes the measurements by collecting the data from the interested nodes for each measurement, while assuming that the sparse measurement matrix is stored at each node [12]-[15]. These approaches [8]-[15] aggregate the measurements from all the nodes by minimizing the routing path to reduce the energy consumption in data aggregation.

Most of the existing CS aided data aggregation approaches do not consider the feasibility of hardware implementation [8]-[15]. The bottleneck for hardware implementation of the CS aided data aggregation techniques is in the encoding process at IoT nodes that are severely resource constrained. The size of the measurement matrix depends on sparsity of the sensing data and the number of nodes deployed in the network [7]. As IoT nodes are resource constrained devices, for sparse random measurements based data aggregation techniques, storage issues can crop up in large-scale network applications. In case of dense random measurements, the dependency of column size on sensing data sparsity poses multiple constraints in real-time implementation for the applications where data to be sensed has low sparsity [16]. In contrast, the measurement matrix content can be combined enroute to the sink instead of generating individual columns or storing the matrix while aggregating the data from the nodes using CS. This class of methods is called as routing measurements based data aggregation approaches. Some existing methods in the literature [17]-[19] have investigated data aggregation using routing measurements. However, these methods compromise either on recovery fidelity (due to poor coherence) [18] or energy efficiency (due to higher number of transmissions) [17] [19].

On the other-hand some existing approaches in the literature [2, 20] proposed data aggregation methods by using dense random projections [8] for optimizing the energy consumption [2] and recovery fidelity [20]. However, these methods do not consider the feasibility of real-time implementation. Further, the proposed method in [20] does not consider the energy optimization during the data aggregation while only concentrating on recovery. Recently, in [21], the authors proposed a data aggregation method by utilizing the short and long range communication resources for achieving energy efficiency. However, in this work, most of the nodes will be required to send their data to multiple aggregators (hubs) leading to energy inefficiency in data aggregation for large-scale network applications.

Designing a low complexity CS based data aggregation technique that minimizes total energy consumption as well as guarantees the reconstruction is still a challenging problem. To address this problem, in this article, we propose a data aggregation method called "Light Weight Compressed Data Aggregation (LWCDA)", which is light-weight (low complexity), energy efficient and provides good recovery fidelity. In contrast to some existing approaches [17]-[19], we utilize clustering for data aggregation which is proven to be promising for energy efficient routing [15] [22]. In addition, the aggregated data from cluster heads is collected using a minimum spanning tree to minimize energy consumption. In the proposed algorithm, each node measures a data sample followed by generating a random value from a Bernoulli distribution for computing the measurement. The cluster heads receive the measurements from their descendants, process them to compute the final measurement before transporting it to the sink. We find that the measurement matrix constructed from our algorithm is highly sparse and possesses properties to guarantee the recovery of data such as high incoherence, good recovery region and satisfy the Restricted Isometry Property (RIP) when combined with some popular bases.

The contributions of this article are summarized as follows:

1. Low complexity CS aided data aggregation technique that constructs a sparse measurement matrix from the network.
2. Performance evaluation of the measurement matrix with respect to RIP,

coherence and phase transition.

3. Comparative analysis of the algorithm in terms of reconstruction error and transmission cost using real data sets.
4. A practical implementation using IITH Motes [23] to demonstrate hardware feasibility of the proposed LWCDA algorithm.

The paper is organized as follows: Section II explains the basics of compressed sensing in IoT networks. Section III describes the proposed LWCDA data aggregation method. Section IV evaluates the RIP and coherence of the proposed measurement matrix and presents the phase transition analysis. Section V describes the hardware implementation performed and simulation results of LWCDA method are described in Section VI. Section VII concludes the paper.

2. Related work

2.1. Compressed Sensing for IoT Networks

For a given N dimensional signal (hereafter data and signal are used interchangeably) that can be sparsely represented using a basis, CS promises to deliver a full recovery of the signal with high probability from far fewer samples [24]. Let $X = [x_1, x_2, x_3, \ldots, x_N]^T \in \mathbb{R}^N$ be sparsely represented in a basis (e.g., Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelets Transform (DWT), etc.) $\Psi = [\psi_1, \psi_2, \ldots, \psi_N] \in \mathbb{R}^{N \times N}$ with k large coefficients (k-sparse), where $k \ll N$, i.e., $X = \Psi\theta$, $\theta = [\psi_1^T X, \psi_2^T X, \ldots, \psi_N^T X] \in \mathbb{R}^N$ and $\|\theta\|_0 \leq k$. The CS theory computes the compressed M-dimensional vector,

$$Y = \Phi X, \tag{1}$$

where $Y \in \mathbb{R}^M$ is the measurement vector and M is the number of measurements and $M < N$ which influences reconstruction of the signal. It has been shown that the number of random measurements required for successful reconstruction of a k-sparse signal is $M = \mathcal{O}(k \; log \; N)$ [25]. The matrix

$\Phi = [\varphi_1^T, \varphi_2^T, \ldots, \varphi_M^T]^T \in \mathbb{R}^{M \times N}$ is called the measurement matrix. The problem here is to reconstruct X from Y, which is under-determined and can have infinitely many solutions. CS theory shows that the problem of recovering X from its linear measurements can be posed as a l_0-minimization problem as shown in (2) and it is computationally intractable. A family of greedy algorithms has been proposed in [26] and [27] to solve the l_0-minimization problem.

$$\min_{\theta} \|\theta\|_0 \text{ subject to } \quad \Phi\Psi\theta = Y. \tag{2}$$

The most prevalent decoding technique to solve the problem in (2) is l_1-minimization, which is a convex optimization problem [28] and hence, computationally tractable [29],

$$\min_{\theta} \|\theta\|_1 \text{ subject to } \quad \Phi\Psi\theta = Y. \tag{3}$$

From the solution θ obtained using l_0 or l_1-minimization, X can be reconstructed as,

$$\widehat{X} = \Psi\theta. \tag{4}$$

The CS matrix $A = \Phi\Psi$ plays a crucial role in the recovery of the N dimensional original signal X. In [30], it is shown that the CS matrix A should satisfy the property known as RIP for successful recovery of X using l_1 minimization. A matrix $A \in \mathbb{R}^{M \times N}$ is said to satisfy the RIP of order k with constant $\delta_k \in (0, 1)$ if

$$(1 - \delta_k) \|u\|_2^2 \leq \|Au\|_2^2 \leq (1 + \delta_k) \|u\|_2^2, \forall u \in \Sigma_k, \tag{5}$$

where u is a k-sparse vector and Σ_k is set of all k-sparse vectors.

On the other hand, if X can be sparsely represented in Ψ domain, then to achieve successful recovery, the theory of CS requires low mutual coherence between the columns of the CS matrix $A = \Phi\Psi$. The mutual coherence of the CS matrix can be defined as

$$\mu(A) = \max_{1 \leq p \neq q \leq N} |\langle a_p, a_q \rangle|, \tag{6}$$

where a_p and a_q are normalized columns of A.

2.2. Literature Review

In this section we discuss the contributions of the relevant literature. Most of the CS aided data aggregation techniques can be classified into three classes, dense random measurements [8]-[11], sparse random measurements [12]-[15] and routing measurements [17]-[19] based data aggregation methods.

Dense random measurements based methods [8]-[11] achieve CS aided data aggregation by considering individual column generation of the measurement matrix at node level using pseudo-random sequences. These methods aggregate the measurements from all the nodes by minimizing routing path to achieve energy efficiency. The size of the measurement matrix depends on the number of nodes and sparsity of the data. IoT nodes are constrained devices possessing minimal on-board resources (in terms of physical memory, processing capability, internal memory, energy). Therefore, generating individual columns of the measurement matrix at a node in case of a large-scale network application where sensing data sparsity is low is computationally intensive and poses multiple constraints in real-time implementation.

Wang *et al*. [12] showed that sparse random measurements (projections) reduce communication cost per sensor node for data aggregation. In [13]-[15], data aggregation techniques have been proposed to achieve energy efficiency for IoT networks by using the sparse random measurements [12]. These algorithms find the optimal route to collect data from the interested nodes for each measurement, while assuming that the sparse measurement matrix is stored at each node. Since the measurement matrix depends on the network size, storage issues can crop-up for large-scale networks. In other words, commercially available nodes that have minimal on-board resources may not be able to support the storage large measurement matrices.

In contrast to dense and sparse random measurements, the routing measurements based data aggregation methods aggregate the measured data from the nodes by computing measurements on the fly enroute to the sink [17]-[19]. In [17], the routing paths are iteratively built through a greedy choice to minimize the coherence of the CS matrix and energy required for data aggregation.

However, building of routing paths in an iterative manner is computationally intensive and requires more transmissions rendering the process highly energy inefficient. In [18], the algorithm picks up a portion of the nodes randomly from the network to generate measurements by utilizing shortest path routing. However, such an approach does not achieve good performance with respect to coherence. In [19], the authors showed that data aggregation from fixed length random walks starting at randomly located nodes can reconstruct the data using CS. However, recovery performance of the method depends on the length of the random walks. An increase in the length of the walk increases the number of transmissions which in turns data aggregation to be energy inefficient.

2.3. Problem Statement

In most of the IoT applications, seamless data aggregation from sensor nodes is the fundamental requirement for data processing (data analytics) and facilitating the user with useful interface and ubiquitous access to the network data[1]. Typically, the sensor nodes used in IoT application deployments such as remote sensing and monitoring are resource constrained (e.g., battery powered and low-end computational resources). CS has been proved to be very promising for data aggregation in IoT networks [6, 7, 21]. However, most of the CS based data aggregation algorithms proposed in the literature do not address the aspects of low complexity, energy efficiency and recovery fidelity jointly which is an important requirement as sensor nodes are resource constrained. The approaches proposed based on dense random measurements [8]-[11] as well as sparse random measurements [12]-[15] are energy efficient but not real-time implementable. On the other hand, the approaches proposed in [17]-[18] are light weight, however, they are either energy inefficient or do not achieve good performance in terms of coherence and recovery. These limitations provide the motivation for this work. Specifically, the problem is to design a low-complexity (real-time) CS aided data aggregation method that is energy efficient and can guarantee a successful recovery of the measured data from IoT networks.

3. Proposed Data Aggregation Protocol

In this section we first present the network model that will be used in our analysis and next describe the proposed data aggregation protocol which forms the light weight measurement matrix.

3.1. Network Model

Consider an IoT network with N nodes deployed in a rectangular area (an example network with grid-wise deployment of $N = 100$ nodes is shown in Fig. 1). The network can be represented by a graph $G(V, E)$, where V is the set of vertices or nodes and E represents the set of edges or links between the nodes. The sink node S is the node that collects data from all the other nodes in the network. We assume that all the nodes are loosely time synchronized and have homogeneous transmission coverage. Unit disc coverage model is considered for all the nodes. We consider the communication range of the nodes to be $D = \sqrt{\frac{5}{N}} * a$ [18]. Here, a is the length of the maximum side of the considered area and N is the number of nodes. Data aggregation proceeds in cycles (rounds) and each node generates one sample per cycle. For example, the i^{th} node acquires data sample x_i in each cycle and N samples $X = [x_1, x_2, x_3, \ldots, x_N]^T \in \mathbb{R}^N$ will be acquired from all the nodes per cycle. We also assume that there is no packet loss in data aggregation. We consider both grid [31] [32] and random deployment [33] scenarios for analysis in further sections as these network deployments have their own significance in different application scenarios.

3.2. Proposed Data Aggregation Protocol

As described above, $X \in \mathbb{R}^N$ is a signal of length N that contains measurements from N nodes in the network. To aggregate data from all the nodes, M nodes are randomly picked such that each node is a Cluster Head (CH) with a probability $P_{CH} = \frac{M}{N}$. The remaining $(N - M)$ leaf nodes connect to their respective nearest CH through the shortest path (route with minimum distance). Accordingly, the whole network gets divided into M non-overlapping clusters to aggregate sensors data. The M clusters $\{c_1, c_2, \ldots, c_M\}$ can contain distinct $\{n_1, n_2, \ldots, n_M\}$ number of nodes. Every node in the cluster measures its data sample x_i (e.g., temperature, humidity, light intensity, etc.) and multiplies it with a random value α_i generated from a Bernoulli distribution with a success probability of 0.5. In other words, the i^{th} node performs $\alpha_i x_i$, where α_i is randomly drawn from the set $\{-1, 1\}$ with a Bernoulli distribution and $i \in [1, N]$. Each leaf node sends the measurement $\alpha_i x_i$ to its CH. The CH adds the received measurements from the leaf nodes including its own measurement. The final measurement at j^{th} CH, $y_j = \sum_{i \in c_j} \alpha_i x_i$ is the linear combination of α_i and x_i, where the nodes belonging to the cluster take non-zero values i.e., $\{\alpha_i \neq 0, x_i \neq 0\} \in c_j$ and the nodes that do not belong to the cluster can be assumed to be zeros i.e., $\{\alpha_i = 0, x_i = 0\} \notin c_j$. The CHs deliver the computed measurements to the sink node through the Minimum Spanning Tree (MST). Dijkstra's and Kruskal algorithms can be used to create MST of CHs along with the sink node. The CHs follow the pack and forward method [9] that provides the feasibility to encapsulate the current measurement of a CH with the relaying packet from descendant CHs along the MST towards the sink.

From the CS formalism in Section II, each cluster can be considered as a row of the measurement matrix Φ and each node in the network corresponds to a column of Φ. In other words, M randomly formed clusters and the nodes in each cluster correspond to rows and respective columns of Φ. The j^{th} cluster c_j forms the j^{th} row of Φ, i.e., φ_j. The support vector of φ_j is $\Delta_j = \{i : i \in [1, N], i \in c_j\}$, $\varphi_{j_{\Delta_j}} = \{\alpha_i : i \in \Delta_j\}$ and $\varphi_{j_{\Delta_j^c}} = 0$. In other words, the j^{th} row of Φ at respective columns of nodes that are connected as a cluster $i \in c_j$ will be assigned values from the set $\{-1, +1\}$ with a Bernoulli distribution. The remaining entries in the row will be zeros.

More concretely, $\Phi \in \mathbb{R}^{M \times N}$, $\Phi = [\varphi_1^T, \varphi_2^T, \ldots, \varphi_M^T]^T$ contains elements in each row

$$\varphi_{ji} = \begin{cases} -1 \text{ or } +1 & \text{if } i \in c_j \\ 0 & \text{otherwise .} \end{cases}$$

Packets received at the sink node from the MST contain elements of the measurement vector $Y = [y_1, y_2, \ldots, y_M]^T \in \mathbb{R}^M$ which are linear combinations of the measured data and the random values of nodes, i.e.,

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_M \end{pmatrix} \begin{pmatrix} X \end{pmatrix} = \Phi X, \tag{7}$$

where $X = [x_1, x_2, \ldots, x_N]^T$, $X \in \mathbb{R}^N$, $\varphi_m \in \mathbb{R}^N$, $y_m \in \mathbb{R}$ where $m \in [1, M]$.

To gain insight into the described LWCDA, we consider a network of grid-wise deployed 100 nodes with a sink node (S = 101), which is placed at the center of the network as shown in Fig. 1. Fig. 1 shows the measurement matrix formation from the network and the sink node. Consider the example node of 98 from Fig. 1, which is a CH and has two descendant nodes $88, 97$. The measurement matrix Φ contains a row which represents the cluster with the nodes $98, 88, 97$ and contains non-zero values from the set $\{-1, +1\}$ that are drawn from a Bernoulli distribution with a success probability of 0.5 at respective columns, as shown in Fig. 1.

To recover the original signal $X \in \mathbb{R}^{\mathbb{N}}$ from the measurement vector $Y \in \mathbb{R}^M$, the sink node needs to have the knowledge of Φ. The information of Φ can be shared with the sink by maintaining synchronized seeds and pseudo-random number generators between the nodes and the sink [8]. Practically, to share pseudo-random number seeds, each node has to send its seed to the sink [19] and this requires a large number of transmissions for large-scale networks. Another approach is to transmit information of Φ to the sink along with the measured data if the message overhead is negligible [18]. In our data aggregation algorithm, each node needs to send or share the information of $\alpha = -1$ or $+1$

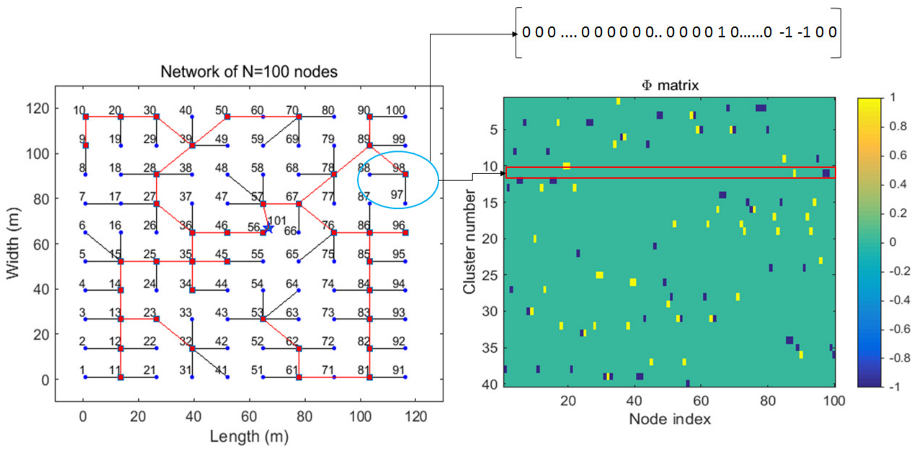


Figure 1: The procedure of measurement matrix designing from a network with $N = 100$ nodes and $M = 40$. The sink $S = 101$ is represented by a star, square boxes represent the CHs and remaining nodes are leaf nodes. The red line represents MST and black line represents connection from leaf node to CH.

with the sink, which can take a maximum of one octet of packet payload. We consider the case that every j^{th} CH sends individual α values of the nodes that belong to that cluster c_j and their indices Δ_j along with the measurement $\sum_{i \in c_j} \alpha_i x_i$ to the sink in the first cycle of data collection. The system of linear equations in (7) $(M < N)$ is under-determined and will give infinitely many solutions while recovering X from Y. The sink node reconstructs full dimension $\widehat{X} \in \mathbb{R}^N$ from the received measurement vector $Y \in \mathbb{R}^M$ by solving either of the optimization problems discussed in Section 2.

3.2.1. Node-level Complexity for Encoding

The node-level computation complexity required for computing the measurements during the data aggregation is computed in terms of generating or storing the number of random values. The proposed data aggregation algorithm constructs Φ on the fly while data is being aggregated from the nodes. Note that each node is required to generate only a random value -1 or $+1$ from a Bernoulli distribution as discussed above. The node level complexity of our method in terms of generating or storing number of random values is $\Theta(1)$ which is independent of sensing data sparsity and network size. The $\Theta(\ )$ refers the formal notation for stating the exact bound on growth of resource needs (computation and storage) of an algorithm. Baseline data aggregation approaches [8]-[11] which utilize dense random measurements require the generation of the respective columns at each node which is the size of $\Theta(M)$ units. In case of sparse random measurements based data aggregation methods [13]-[15], every node in the network stores the complete Φ. The size of the required storage is $\Theta(MN)$. Some of the methods which use sparse random measurements such as [12] generate the respective row of Φ at every node and the row size is $\Theta(N)$ units. The values of M and N are proportionally related and depend on sensing data sparsity and the network size. This dependency poses multiple constraints on the real-time implementation of the large-scale network applications where the data to be sensed has low sparsity. The proposed approach is lightweight as it completely eliminates the burden of generating a specific column or storing the entire Φ at the node to perform data aggregation in the network. Consequently, the proposed method can be implemented in commercially available low end IoT nodes.

The measurement matrix Φ should satisfy certain properties as discussed in Section 2 for it to allow data recovery. In the following section we evaluate the properties of the Φ and demonstrate how it can guarantee the reconstruction.

4. Measurement Matrix Analysis

To analyze the proposed measurement matrix Φ, we rely on RIP, coherence and Phase Transition (PT) [34] analyses. We considered both grid and random deployments scenarios as both deployments have their own significance for different application scenario [31]-[33]. We considered DCT, DFT, DWT, Laplacian and Diffusion Wavelet (DiWT) bases (Ψ) for the analyses. The DCT, DFT and DWT bases (Ψ) can sparsify data from regular (grid-wise) IoT deployments [8], [19]. In case of randomly deployed networks, the Laplacian [19] and Diffusion wavelet (DiWT) [35] can accommodate irregularity and provide a sparse representation of the data.

4.1. Numerical Experiments: RIP Analysis

As discussed in Section 2, RIP is a standard tool to analyze near-orthonormal performance of a CS matrix while operating with sparse input vectors. This property measures the performance of a compressed sensing matrix in terms of the Restricted Isometry Constant (RIC) δ_k. As a result, δ_k can be used to evaluate the ability to recover a sparse signal from the measurement vector. From the definition of RIP of a matrix $A \in \mathbb{R}^{M \times N}$, for k-sparse vectors with a constant δ_k, (5) can be rewritten as,

$$\delta_k = \max_{T \subset [N], |T| \leq k} \|A_T^* A_T - Id\|_{2 \to 2}, \tag{8}$$

where $Id \in \mathbb{R}^{|T| \times |T|}$ is an identity matrix and T is the support set of k-sparse vector [24].

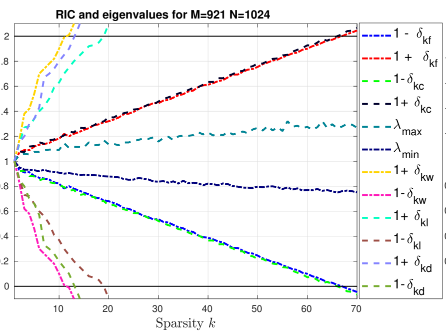
For any matrix A that satisfies RIP with a RIC of δ_k, the following condition holds:

$$(1 - \delta_k) \leq \lambda_{min}(A_T^* A_T) \leq \lambda_{max}(A_T^* A_T) \leq (1 + \delta_k), \tag{9}$$

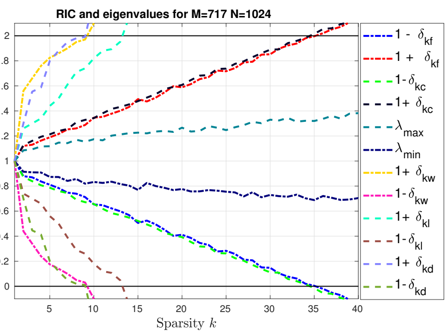
where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of the symmetric matrix $A_T^* A_T$ respectively.

4.1.1. Numerical verification of δ_k, λ_{min} and λ_{max}

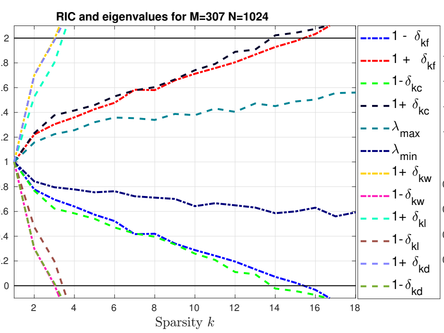
The DFT, DCT, DWT, Laplacian and DiWT matrices are separately considered as bases $\Psi_{N \times N}$ for the empirical evaluation of δ_k. To verify with DWT, we evaluated the performance of the CS matrix A across several popular wavelets such as Daubechies, Symlets, Coiflets and chose the Daubechies-4 wavelet for all our analysis as it gives a better range for k compared to other wavelets. The compressed sensing matrix $A_{M \times N} = \Phi_{M \times N} \Psi_{N \times N}$ with $N = 1024$, at different compression rates with $M = 103, 308, 717, 922$ are considered for evaluation. For a compressed sensing matrix $A \in \mathbb{R}^{M \times N}$, the compression rate $\Gamma\%$ can be written as, $\Gamma\% = \left(1 - \dfrac{M}{N}\right) \times 100$.



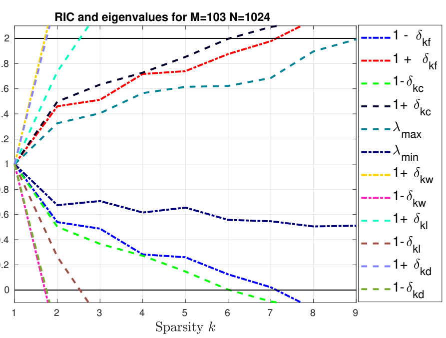
(a) $\Gamma = 10\%$.



(b) $\Gamma = 30\%$.



(c) $\Gamma = 70\%$.



(d) $\Gamma = 90\%$.

Figure 2: RIC δ_k and eigenvalue bounds (λ_{min}, λ_{max}) for the proposed CS matrix $A = \Phi\Psi$. Here, Φ is the measurement matrix constructed using LWCDA algorithm and analyzed for different Ψ (DCT, DFT, DWT, Laplacian and DiWT) matrices. (a) - (d) show values of δ_k, λ_{max} and λ_{min} for matrix A at different compression rates Γ. CS matrix A gives a better range of k with DFT and DCT compared to DWT basis in grid deployment. In the random deployment case, CS matrix A gives slightly better range of k with Laplacian compared DiWT.

Table 1: Sparsity value k where RIC $\delta_k \in (0,1)\ \forall\ u \in \Sigma_k$ for different Γ.

Sparsity value k					
	Regular deployment			Random deployment	
Compression rate Γ	DFT	DCT	DWT	Laplacian	DiWT
90%	7	6	1	2	1
70%	15	14	3	4	3
30%	36	35	9	13	9
10%	67	66	11	19	13

The procedure followed for empirical evaluation of δ_k, λ_{min} and λ_{max} is described below:

1. Generate the measurement matrix Φ and the basis Ψ for fixed $N = 1024$ and for each M.
2. For a combination (N, M), k is varied across $[1 : M]$.
3. Consider a k-sparse vector $u \in \mathbb{R}^N$. The vector u contains non-zero values at k randomly chosen locations and the values themselves are chosen from a normal distribution.
4. Find the support set for u, i.e., T.
5. Repeat steps 2 and 3 for 10000 iterations for each combination (N, M, k) and calculate δ_k from (8).
6. Compute $\lambda_{min}(A_T^* A_T)$, $\lambda_{max}(A_T^* A_T)$, where T is the support set corresponding to δ_k from step 5.

The calculated RIC δ_k values, λ_{max}, λ_{min} with respect to sparsity value k at different compression rates Γ, are plotted in Fig. 2. In Fig. 2, δ_{kf}, δ_{kc}, δ_{kw}, δ_{kl}, δ_{kd} refer to RICs of CS matrix A where Ψ is DFT, DCT, DWT, Laplacian and DiWT respectively. λ_{min}, λ_{max} refer to the minimum and the maximum eigenvalues of CS matrix A respectively when Ψ is DFT. Similar behavior of eigenvalues is also observed with DCT, DWT, Laplacian and DiWT bases.

Sparsity values k obtained while $\delta_k \in (0, 1)$ for the proposed CS matrix A with different bases are tabulated in Table I (the same can be observed from Fig. 2 as well). The interesting observation made from Table I is that the CS matrix A gives better range for k with DFT compared to that of DCT and DWT bases. In the random deployment case, CS matrix A gives slightly better range for k with Laplacian then DiWT basis.

4.2. Coherence Analysis

As discussed in Section 2, if X can be sparsely represented in an arbitrary basis Ψ, then for successful recovery, CS theory requires low mutual coherence between columns of the matrix $A = \Phi\Psi$. The mutual coherence μ of the matrix A with different bases at various compression rates Γ is calculated using (6), i.e., the CS matrix $A_{M\times N} = \Phi_{M\times N}\Psi_{N\times N}$ where $N = 1000$ and M is chosen to vary from 100 to 900 in steps of 100 ($M = 100 : 100 : 900$) for calculating μ.

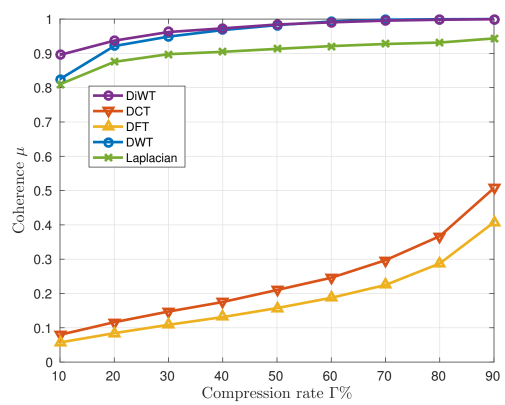
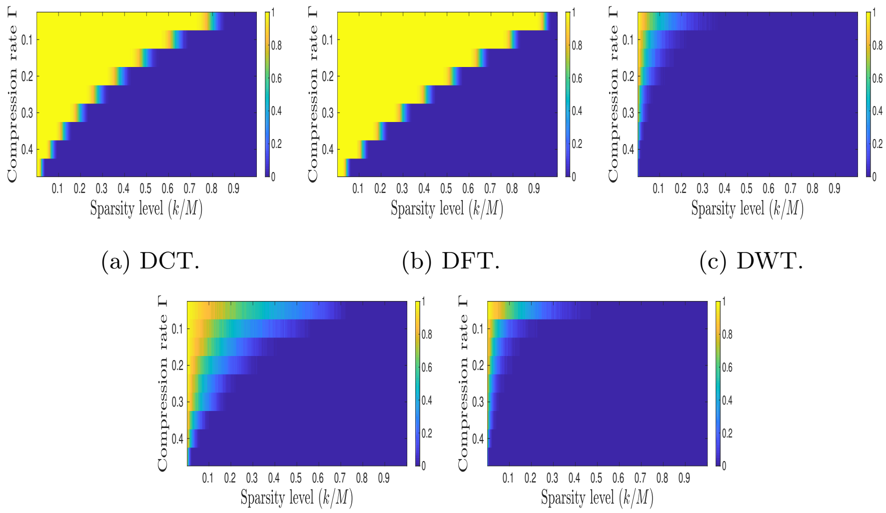


Figure 3: Comparison of mutual coherence μ of the CS matrix $A = \Phi\Psi$ with different bases where matrix Φ is constructed from LWCDA. Columns of the matrix A are highly incoherent with DFT basis among all.

The resultant mutual coherence with different bases are shown in Fig. 3. The CS matrix A provides better incoherence for the DCT and DFT bases compared to the DWT basis where Φ is constructed from grid deployment. In case of random deployment, the coherence of the matrix A with Laplacian is fairly better compared to DiWT basis across all compression rates. It is observed from Fig. 3 that among all the bases, DFT provides high incoherence for all compression rates.

4.3. Phase Transition Analysis



(a) DCT. (b) DFT. (c) DWT.

(d) Laplacian. (e) DiWT.

Figure 4: Phase transition analysis of CS matrix $A = \Phi\Psi$ for different bases, where Φ is the proposed measurement matrix and Ψ is the basis matrix. The color bar indicates successful recovery probability P_s. CS matrix A with DCT and DFT basis yields promising recovery region.

For a given CS matrix, the phase diagram can be generated as a numerical representation of successful recovery probability P_s over the space $(k/M, 1-(M/N)) \in [0,1]^2$, as in [34]. This space is discretized and we performed multiple compression and decompression experiments at each grid point. The phase diagram is finally approximated by using successful recovery probability $P_s = Pr\{e \leq e_{TH}\}$, where the reconstruction error $e = \left\|X - \hat{X}\right\|_2 / \left\|X\right\|_2$, with an appropriately selected threshold e_{TH}. We considered error threshold $e_{TH} = 10^{-8}$ in our analysis. For PT analysis, $A_{M\times N} = \Phi_{M\times N}\Psi_{N\times N}$ is considered with $N = 1000$ and evaluated for different compression rates (Γ) with $M =$

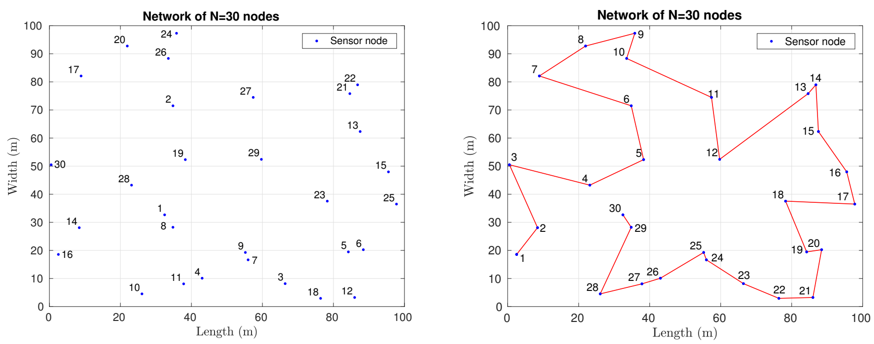
100 : 100 : 900. Fig. 4 shows the phase diagram of CS matrix $A = \Phi\Psi$, where Φ is the measurement matrix and Ψ is the basis. Fig. 4 also illustrates that the proposed measurement matrix Φ with DCT and DFT bases provides promising recovery region compared to DWT where Φ is constructed from grid deployment. In case of random deployment, Laplacian basis provides slightly better recovery region compared to DiWT basis.

This evaluation has shown that the proposed measurement matrix Φ gives better performance with DCT and DFT bases compared to the DWT basis in terms of RIC, coherence and PT analysis where Φ is designed from grid-wise deployed network. Further, in random deployment scenario, Laplacian and DiWT bases give comparable performance. The proposed matrix Φ with DCT and DFT bases (Ψ) has the ability to recover the signals successfully even though they have fairly low sparsity. Whereas in case of DWT, Laplacian and DiWT, the matrix Φ can recover the signals on the condition that they are highly sparse.

To extend the proposed LWCDA method to fairly low sparse data cases especially in random deployment scenario, we propose a technique called spatial logical node mapping, which is described in the following subsection.

4.4. Spatial Logical Node Mapping

Before invoking the LWCDA algorithm, we first model the network as a logical chain based on the Euclidean distance between the nodes. The algorithm starts from any random node and gives sequential node IDs along the chain. The method used to form the logical chain is similar to that in [36]. We consider that in the initial phase, each node sends the distance information of the nodes that are in its coverage area to the sink. The sink maps the new node IDs from old node IDs and sends it back to the nodes to change. Fig. 5 provides more insight into the Spatial Logical Node Mapping (SLNM) with an example network of $N = 30$ nodes. This preprocessing will introduce spatial correlation in the data since adjacent nodes in the chain tend to be the nodes which are geographically close to each other [36] [37]. The spatial correlation among the samples generated from the nodes which are geographically close to each other can make the signal sparse in the regular DFT and DCT bases. SLNM adds the advantage to LWCDA to guarantee the recovery of the measured data from the random deployment as it introduces sparsity for the data in DFT and DCT bases.



(a) Randomly deployed nodes.

(b) Logical mapped nodes.

Figure 5: Logical node mapping.

The data aggregation algorithms proposed in the literature such as [8]-[19] do not discuss hardware implementation details. The implementation procedure and assumptions considered for software simulations differ when it comes to real-time hardware implementation. For the completeness of the proposed algorithm and analysis, we describe an optimal way of implementation which shows the efficacy of the proposed method in a real-time scenario.

5. Real-time Implementation

The in-house IITH Motes [23] are used for implementing the proposed data aggregation algorithm (LWCDA). The IITH Mote is a ZigBee system-on-chip combining a 2.4 GHz IEEE 802.15.4 radio transceiver with a 8 MHz, 8-bit processor having 128 kB of flash memory and 8 kB of RAM. TinyOS [38] is used to program the proposed LWCDA algorithm on the nodes. Based on the required compression rate Γ, the threshold value T_{hr} will be decided. From the selected M CHs, the probability of the i^{th} node becoming a CH is $P_{CH} = \frac{M}{N}$ as discussed in Section III. Let T_{u_i} denote the generated uniform random value at the i^{th} node, i.e., $T_{u_i} \in U \sim [0, 1]$. If $T_{u_i} \leq T_{hr}$ then the i^{th} node becomes a CH. CH probability can be rewritten as $P_{CH} = Pr\{T_{u_i} \leq T_{hr}\} = T_{hr}$ where $i \in [1, N]$. For example, if the threshold is considered to be $T_{hr} = 0.3$ then on an average 30% of the nodes become CHs ($P_{CH} = 0.3$) and $\Gamma = 70\%$ compression can be achieved. The sink node broadcasts a starting packet with the specified threshold T_{hr} value. Each node in the network broadcasts this packet once so that the threshold value reaches every other node in the network. The nodes calculate Received Signal Strength Indication (RSSI) values from the received packets and stores them in a table. It is important to note that each node will have RSSI values of all the other nodes that are in its radio range (communication range). Using the created RSSI table, the nodes, which are selected as leaf nodes, connect to nearest CHs and CHs form MST.

As the sink node requires the knowledge of Φ, i.e., $\{\alpha_i\}$ values and respective indices Δ_j, where $i \in c_j$ and $j \in [1, M]$, in the initial phase (i.e., first cycle of data aggregation), CH sends $\{\alpha_i\}$, Δ_j to the sink along with the final measurement $\sum_{i \in c_j} \alpha_i x_i$. This is a small overhead as α and the respective node index (node address) together can take a maximum of three octets when short address mode is considered. By the end of the initial phase, all the nodes register their respective destination node addresses. In data sensing phase (i.e., from second data aggregation cycle on-wards), in each cycle, all the leaf nodes compute their measurements and send them to their respective destined CHs. Further, each CH computes the final measurement and forwards it to the sink. Pseudo code of the node level implemented algorithm is described in Algorithm 1.

The proposed LWCDA algorithm is independent of the deployment scenario. As an example to verify the implementation efficacy of the LWCDA, we deployed 50 nodes grid-wise in an area of 321.44 ft^2 as shown in Fig. 6. The sink node is connected to a PC that collects measured data from all the nodes in the network. For illustration, we considered a threshold $T_{hr} = P_{CH} = 0.3$ and obtained 14 CHs among the deployed 50 nodes in a particular realization of the experiment,

Algorithm 1 Pseudo code for the data aggregation algorithm at node level

Require: T_{hr}

Data collection round r←0

Generate uniform random value $T_{u_i} \in U \sim [0, 1]$ (i refers node number)

if ($T_{u_i} \leq T_{hr}$) **then**

  Type ←CH

else

  Type ←Leaf node

end if

while $r \geq 0$ **do**

  if (*Type* is CH) **then**

    $r \leftarrow r + 1$

    if (r is 1) **then**

      Broadcast CH packet

      Generate uniform random value $R_i \in U \sim [0, 1]$

      if ($R_i \leq 0.5$) **then**

        $\alpha_i \leftarrow -1$

      else

        $\alpha_i \leftarrow 1$

      end if

      Discover the next hop destination node CH_{dest}: other CH or the leaf node in MST towards the sink

    end if

    Measure data sample x_i

    Compute: $\alpha_i x_i$

    Receive data packets from all the leaf nodes and descendant CHs

    Compute: $\sum_{i \in c_j} \alpha_i x_i$

    Send CH data packet to CH_{dest} using pack and forward method

  else

    $r \leftarrow r{+}1$

    if (r is 1) **then**

      Find $RSSI_h \leftarrow \underset{h}{\text{argmax}}$\{RSSI of CHs which are in the radio range\}

      $Leaf_{dest} \leftarrow CH_h$

      if ($Leaf_{dest}$ is NULL) **then**

        Discover the next hop destination node $Leaf_{dest} \leftarrow$ neighboring leaf node in the shortest path towards nearest CH

      end if

      Generate uniform random value $R_i \in U \sim [0, 1]$

      if $(R_i \leq 0.5)$ **then**

        $\alpha_i \leftarrow -1$

      else

        $\alpha_i \leftarrow 1$

      end if

    end if

    Measure data sample x_i

    Compute: $\alpha_i x_i$

    Send the data packet to $Leaf_{dest}$

  end if

end while

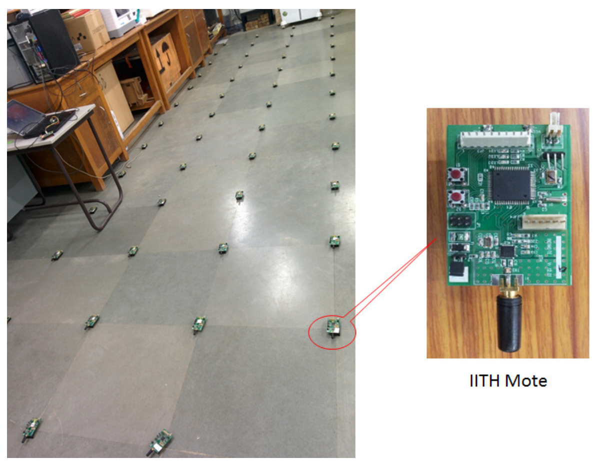


Figure 6: Experimental setup with $N = 50$ nodes deployed in an area of 321.44 ft^2.

while the remaining nodes are connected to their respective CHs. Accordingly, 14 clusters were formed, and thus the rows of the measurement matrix $\Phi_{14\times50}$ were generated. The resultant measurement matrix $\Phi_{14\times50}$ is shown in Fig. 7. To evaluate the efficacy of the proposed method, we considered coherence as the metric. We repeated the above experiment for a range of threshold values $T_{hr} = 0.1 : 0.1 : 0.9$, and in each case of T_{hr}, the measurement matrix Φ was constructed. To compute the coherence of Φ against all the compression rates, we obtained Φ for 10 realizations and for each T_{hr}. Each realization gives one mutual coherence value μ for a pair of Φ and Ψ. We then averaged μ over 10

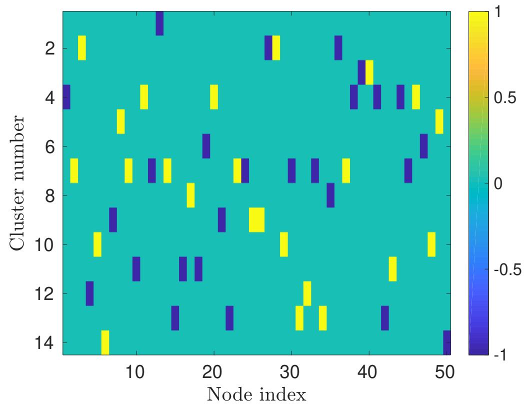


Figure 7: Measurement matrix $\Phi_{14\times 50}$ constructed from the real field deployment with $N = 50$ nodes and $\Gamma = 70\%$.

realizations for each T_{hr}. To compare with the real deployment, we simulate a similar scenario in software. Average coherence values of the matrix A designed from both the experiment (exp.) as well as the simulation (sim.) are plotted in Fig. 8. Fig. 8 illustrates that the coherence values of the matrix A where the proposed measurement matrix is constructed from the experiment as well as the simulation with DFT, DCT and DWT bases are in excellent agreement. These results show efficacy of the proposed method in a real-time implementation. It justifies our claim that the proposed method does not require any extra computational overhead (such as the generation of the individual columns of the matrix Φ, storage of Φ etc.). Hence, the proposed method can be implemented on low end commercial off-the-shelf IoT nodes.

In the following section, we evaluate the performance of the proposed data aggregation method in an application scenario.

6. Results And Evaluation

In this section, the performance of the proposed data aggregation method is analyzed using the following metrics:

1. Reconstruction error.

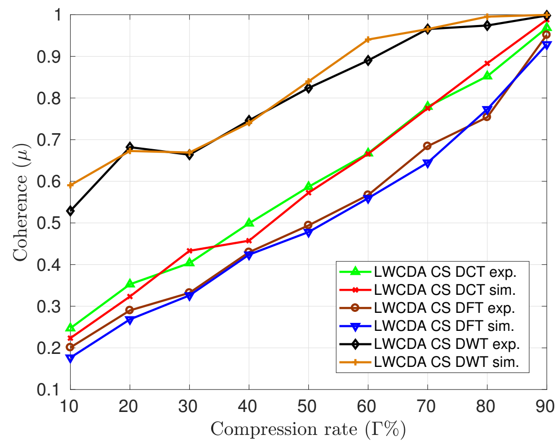


Figure 8: Comparison of the mutual coherence μ for the CS matrix $A = \Phi\Psi$ where Φ is constructed from the real-time deployment and simulations against different compression rates Γ, with $N = 50$ nodes and for different bases DFT, DCT and DWT. Mutual coherence curves from simulation and real-time deployment are very close and demonstrate the efficacy of the proposed method.

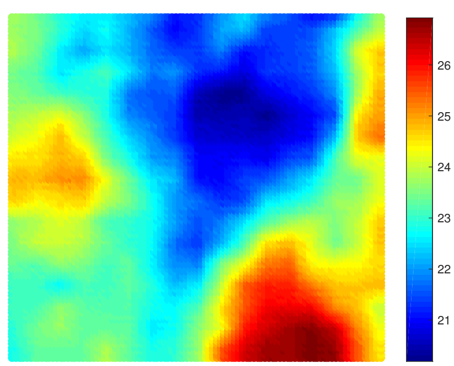


Figure 9: Temperature sensing field.

2. Transmission cost.

6.1. Reconstruction Error Analysis

We extend the application of the proposed algorithm for compressible signals by using a real data set for evaluation. Real temperature data which is obtained

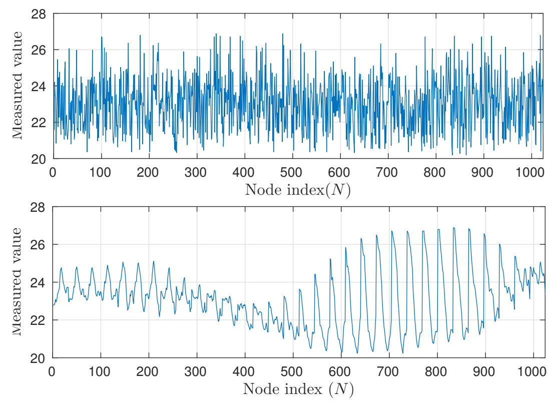


Figure 10: Measured data from random (top) and grid-wise (bottom) deployed 1024-node network.

by capturing thermal images from the top view of an area $100m \times 100m$ is considered for analysis. Thermal images were captured in a region using a mobile phone interfaced thermal camera [39] in the campus of IIT Hyderabad on August 14, 2016 centered at location of Latitude:17.593978 and Longitude:78.123359. Fig. 9 visualizes the considered temperature data for evaluating recovery performance of the proposed method. Fig. 10 shows the measured data from random and grid-wise deployed 1024 sensor nodes on the field. We used MATLAB R2015b software for performing all our simulations. Ideally the sparsity value k of X in a basis Ψ is measured using the l_0 norm, $k = \|\theta\|_0$, where $\theta = \Psi X$. For real-time data which is approximately sparse, only few large coefficients contribute a large proportion of the total energy. We use numerical sparsity [35] as the measure of sparsity which represents the number of effective large coefficients. If a vector X can be represented using a sparsifying basis Ψ as $X = \Psi\theta$, then the numerical sparsity of X can be calculated as

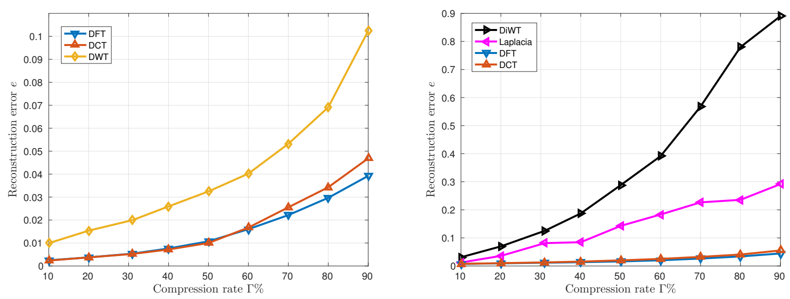
$$s = \frac{\|\theta\|_1{}^2}{\|\theta\|_2{}^2}. \tag{10}$$

Numerical sparsity of the considered temperature data with different bases (Ψ) (DFT, DCT, DiWT and Laplacian) are tabulated in Table II.

Table II shows considered data is more sparse in DCT and DFT bases com-

Table 2: Sparsity measure of the temperature data

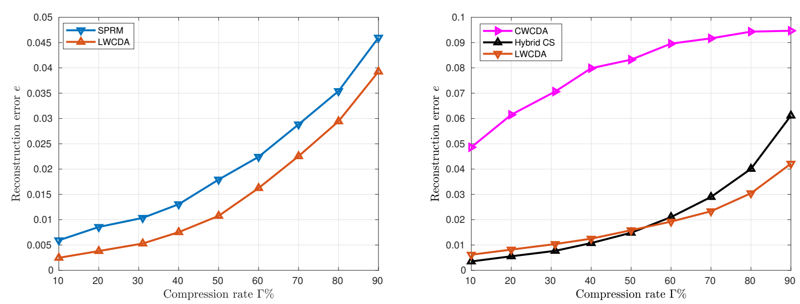
Numerical sparsity value s						
Regular deployment			Random deployment			
					SLNM	
DFT	DCT	DWT	Laplacian	DiWT	DFT	DCT
2.2205	2.5251	7.6707	15.925	53.0402	2.6219	2.7569



(a) (b)

Figure 11: Average reconstruction error against different compression rates. Here, 1024 temperature data points are considered. Figures (a) and (b) depicts recovery of grid-wise and Randomly deployed nodes measured data respectively. DFT basis provides low recovery error compared to all bases in both grid and random deployment scenario.

pared to the others in both grid-wise and random deployment scenarios. To evaluate recovery performance of the measured data from grid-wise deployed nodes the basic LWCDA is used to construct the measurement matrix Φ. In case of random deployment, Φ is constructed from LWCDA and SLNM. The OMP algorithm [27] is used for the recovery of the compressed data. We evaluated the performance of the proposed data aggregation method in terms of the reconstruction error e against the compression rate Γ. Fig. 11 compares average reconstruction error of our method with different bases. In our analysis, e is averaged over 100 iterations for each Γ. From Fig. 11a, it can be observed that DFT and DCT can recover the data which is measured from grid-wise deployed nodes with a low error compared to DWT for all compression rates. Fig. 11b illustrates the data recovery performance of the proposed method where the data is measured from randomly deployed sensor nodes. From Fig. 11b, it can be observed that DCT and DFT can recover the data with a low error across all compression rates compared to other bases. However, DiWT and Laplacian result in high recovery error as they require the signal to be highly sparse. This evaluation has shown that the proposed LWCDA method provides high recovery fidelity using the DFT basis for the data measured from both the random and grid-wise deployed nodes.



(a) (b)

Figure 12: Comparison of average reconstruction error against different compression rates for grid and random deployment scenarios. Here, 1024 temperature data points are considered. Figures 12a and 12b depict recovery performance of the data measured from grid-wise and randomly deployed nodes respectively. In the grid-wise deployment scenario, the proposed LWCDA performs superior to SPRM. However, in case of random deployment, the proposed LWCDA provides superior performance compared to CWCDA and a comparable performance with Hybrid CS.

On the other hand, we perform recovery error comparative analysis of the proposed algorithm with existing CS based data gathering methods. To demonstrate the efficiency of our algorithm, we compare with Shortest Path Routing Measurements (SPRM) [18] for the grid-wise deployment scenario, and Cluster-based Weighted Compressive Data Aggregation (CWCDA) [15], Hybrid Compressed Sensing (Hybrid CS) [8] methods for the random deployment scenario. SPRM [18] is closely related to our work, and CWCDA [15] is a data aggrega-

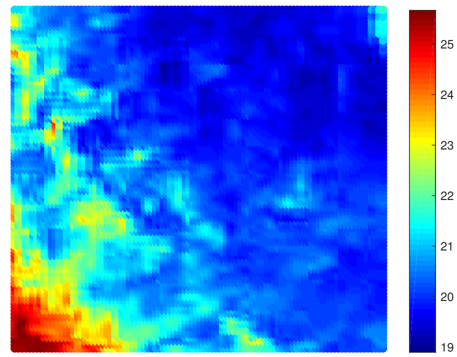


Figure 13: Sea surface temperature sensing field.

tion method based on sparse random measurements. In contrast, the Hybrid CS [8] method utilizes dense random measurements. Fig. 12 compares average reconstruction error of the proposed LWCDA method with baseline approaches against compression rate Γ. Here, e is averaged over 100 iterations for each Γ. From Fig. 12a, it can be observed that the proposed LWCDA method has lower recovery error compared to SPRM for all considered compression rates. Fig. 12b illustrates the data recovery performance of the proposed method where the data is measured from randomly deployed sensor nodes. From Fig. 12b, it can be observed that the proposed method yields superior recovery performance compared to CWCDA across all considered compression rates. Relative to Hybrid CS, the performance of the proposed method is comparable at lower compression rates ($\Gamma \leq 50$) and superior at higher compression rates ($\Gamma > 50$).

To evaluate the proposed data aggregation method on independently generated data set, we have considered sea surface temperature data of the region of Latitude:23-25 and Longitude:78-80 measured from INSAT 3D remote sensing system at 12:30 IST, December 21, 2017 [40]. Fig. 13, visualizes the considered sea surface temperature sensing field and Fig. 14 shows the measured data from random and grid-wise deployed 1024-node network upon the sensing field. Comparison of average reconstruction error of the proposed LWCDA method with

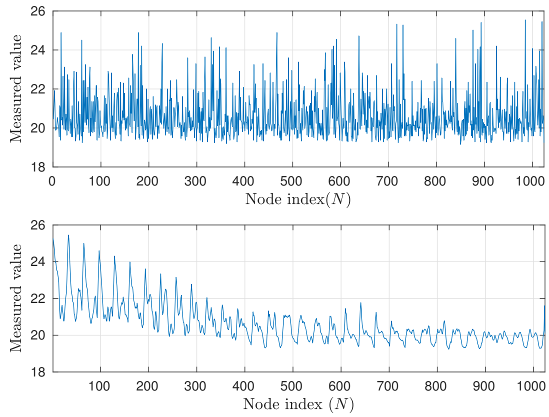
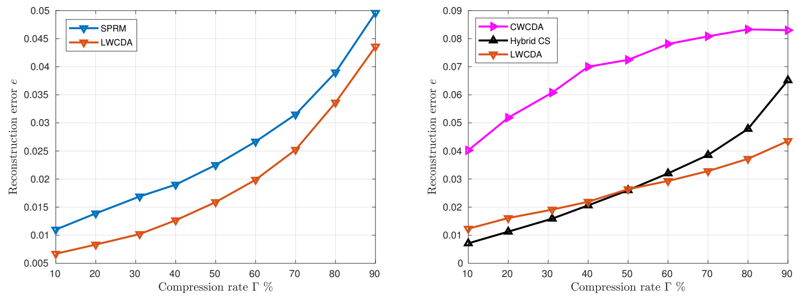


Figure 14: Measured data from random (top) and grid-wise (bottom) deployed 1024-node network upon considered sea surface temperature sensing field.

baseline approaches against compression rate Γ is depicted in Fig. 15. From Fig. 15a (where data is measured from grid-wise deployment), one can observe that the proposed LWCDA method has lower recovery error compared to SPRM for all considered compression rates. From Fig. 15b (where data is measured from random deployment), one can observe that the proposed method yields superior recovery performance compared to CWCDA across all considered compression rates. Relative to Hybrid CS, the performance of the proposed method is comparable at lower compression rates ($\Gamma \leq 50$) and superior at higher compression rates ($\Gamma > 50$). From Fig. 12 and Fig. 15, one can infer that the proposed LWCDA can recover the data with high fidelity in both grid and random deployment scenarios while being at par or better than competing algorithms.

In the following section, we perform a comparative analysis of the transmission cost of our algorithm with traditional CS based data gathering methods. To demonstrate the efficiency of our algorithm, we compare with SPRM for the grid-wise deployment scenario, CWCDA, Hybrid CS and a Non-compressed Sensing (Non-CS) method for the random deployment scenario.



(a) (b)

Figure 15: Comparison of average reconstruction error against different compression rates for grid and random deployment scenarios. Figures 15a and 15b depict recovery performance of the data measured from grid-wise and randomly deployed 1024 data points (nodes) on sea surface temperature sensing field respectively. In the grid-wise deployment scenario, the proposed LWCDA performs superior to SPRM. However, in case of random deployment, the proposed LWCDA provides superior performance compared to CWCDA and a comparable performance with Hybrid CS.

6.2. Transmission Cost Analysis

We considered a cost model for computing the required transmission cost (or energy) for data aggregation as in [8]. Let c_{ij} be the energy expense of sending one unit of data across link $(i, j) \in E$ where nodes $i, j \in V$. Assuming identical data rate and bandwidth for all links, we can show that c_{ij} is proportional to the path loss on the link (i, j), hence c_{ij} is a function of link length. Further, we considered c_{ij} as the Euclidean distance between the nodes i, j and one packet as one unit of traffic on the link $(i, j) \in E$. The total transmission cost (or energy consumption) of the network is computed as,

$$T_{cst} = \sum_{(i,j)\in E} t_{ij} c_{ij}, \tag{11}$$

where t_{ij} represents the traffic on the link $(i, j) \in E$. ZigBee protocol is considered for simulations as the ZigBee stack is one of the most commonly used protocols among commercially available off-the-shelf IoT solutions. The size of PHY layer data field of the packet of ZigBee is 128 bytes, of which 87 bytes can be used for application payload as the remaining octets are reserved for packet

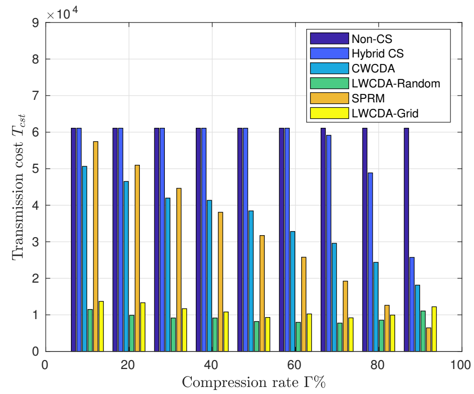


Figure 16: Comparison of the transmission cost required for data aggregation from 625 nodes deployed in an area $256m \times 256m$ using the proposed LWCDA and SPRM, Hybrid CS, CWCDA and Non-CS methods against the compression rate Γ. Transmission cost of the LWCDA is significantly low compared to all methods almost for all compression rates ($\Gamma \leq 80\%$).

header information of higher layers. The number of bits required to represent the data sample and the address field (short address mode) is considered to be 2 octets.

For transmission cost comparison, a network deployment of 625 nodes in an area of $256m \times 256m$ is considered. The comparison of the transmission cost for data aggregation using the proposed method (LWCDA), SPRM, Hybrid CS, CWCDA and a Non-CS method with respect to the change in compression rate (Γ) is shown in Fig. 16. In particular, we considered the conventional shortest path algorithm [41] for data gathering as the Non-CS approach, where each node in the network sends its data to the sink through the shortest path. From Fig. 16, it can be observed that our LWCDA (labeled as LWCDA-Random for random deployment) method requires very low transmission cost for data aggregation compared to Non-CS, Hybrid CS and CWCDA for all compression rates where nodes are randomly deployed. In case of grid deployment, compared to SPRM the proposed LWCDA (labeled as LWCDA-Grid for grid deployment) method outperforms until $\Gamma = 80\%$ for data aggregation. In the proposed LWCDA method, an increase in compression rate Γ decreases the number of required

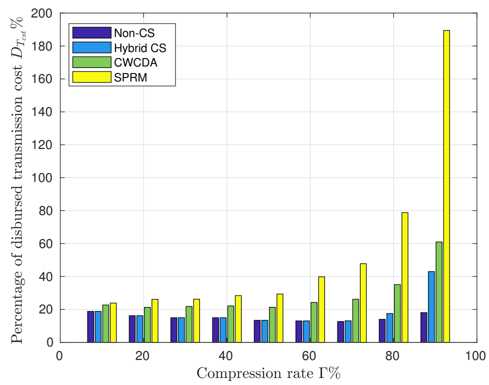


Figure 17: Comparison of the percentage of disbursed transmission cost of the proposed LWCDA with respect to SPRM, Hybrid CS, CWCDA and Non-CS methods against the compression rate Γ. Here, data aggregation is considered from 625 nodes deployed in an area $256m \times 256m$. Percentage of disbursed transmission cost of the LWCDA with respect to SPRM, Hybrid CS, CWCDA and Non-CS methods is low for almost all compression rates ($\Gamma \leq 80\%$).

clusters for data aggregation. A decrease in number of clusters increases the required transmission cost for data aggregation as the leaf nodes need to send their measurements to CHs from farther distance. Further, as Γ increases, the required transmission cost to collect measurements from CHs (using MST) also decreases. This results in an increase of total transmission cost T_{cst} at higher compression rates ($\Gamma > 80\%$) as shown in Fig. 16. Fig. 17 illustrates the percentage of disbursed transmission cost $D_{T_{cst}}$ of the proposed LWCDA with respect to that of Hybrid CS, LWCDA, SPRM and Non-CS methods. Percentage of disbursed transmission cost $D_{T_{cst}}$ of a given method P with respect to the method Q is defined as,

$$D_{T_{cst}}\% = \frac{T_{cst} \text{ of method } P}{T_{cst} \text{ of method } Q} \times 100. \quad (12)$$

The SPRM method at high compression rates ($\Gamma > 80\%$) results in lesser transmission cost as compared to that of the proposed LWCDA method. This in turn results in the percentage of disbursed transmission cost of LWCDA (LWCDA-Grid) to go beyond 100% as shown in Fig. 17 for higher compression rates.

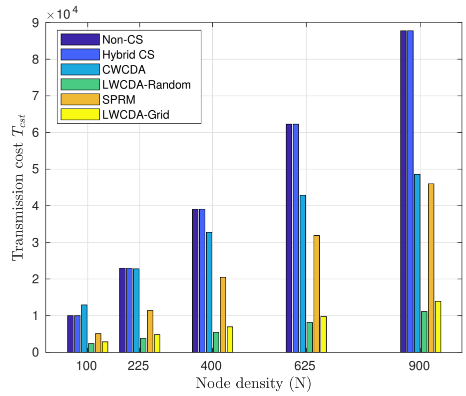


Figure 18: Transmission cost comparison of the proposed LWCDA method against node density with SPRM, Hybrid CS, CWCDA and Non-CS methods at $\Gamma = 50\%$. Here, an area of $256m \times 256m$ is considered for the network deployment and number of nodes deployed (N) are varied. Transmission cost of the LWCDA is significantly low compared to SPRM, Hybrid CS, CWCDA and Non-CS methods for all considered node densities.

This is because, in the SPRM method, very few randomly selected nodes are required to send data through the shortest path to the sink at high compression rates. Although, SPRM offers higher compression rates with lower transmission costs, it does not achieve good performance with respect to coherence leading to higher reconstruction errors at higher compression rates, thereby not guaranteeing a successful reconstruction (as discussed in [18]). For all compression rates in both grid-wise ($\Gamma \leq 80\%$) and random deployment scenario, the proposed LWCDA method can deliver the data to the sink with a lower transmission cost as illustrated in Fig. 16 and with a lower percentage of disbursed transmission cost as shown in Fig. 17, thereby enhancing the network lifetime as compared to the considered baseline approaches.

To evaluate the effect of changing the node density on the required transmission cost for data aggregation of the proposed LWCDA, we performed an experiment where the number of nodes deployed is varied in the considered area of $256m \times 256m$. The transmission cost of data aggregation with respect to the changing in node density with $\Gamma = 50\%$ compression rate is shown in Fig. 18.

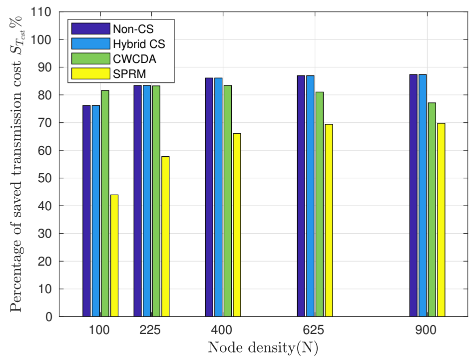


Figure 19: Comparison of the percentage of saved transmission cost of the proposed LWCDA with respect to SPRM, Hybrid CS, CWCDA and Non-CS methods against node density at $\Gamma = 50\%$. Here, an area of $256m \times 256m$ is considered for the network deployment and number of nodes (N) deployed are varied. The proposed LWCDA method offers savings in the transmission cost consistently with respect to the baseline approaches for all the considered node densities.

From Fig. 18, it is observed that the transmission cost increases with an increase in the node density. The interesting observation made from Fig. 18 is that the transmission cost for LWCDA is significantly low as compared to that of the traditional methods for all considered density levels in both deployment scenarios. Fig. 19 shows the percentage of savings in the transmission cost with respect to Non-CS, Hybrid CS, CWCDA and SPRM methods. Percentage of saved transmission cost $S_{T_{cst}}$ of a given method P with respect to the method Q is defined as,

$$S_{T_{cst}}\% = \left(1 - \frac{T_{cst} \text{ of method } P}{T_{cst} \text{ of method } Q}\right) \times 100. \tag{13}$$

From Fig. 19, one can observe that the proposed method consistently offers savings in transmission cost under the considered varying node densities. We can infer that for large-scale dense networks, LWCDA algorithm can achieve significant improvements in the network lifetime compared to traditional approaches.

The location of the sink node affects the required transmission cost for data aggregation [42]. To analyze the dependence of the transmission cost on the sink location for data aggregation, we considered a 625-node network deployed

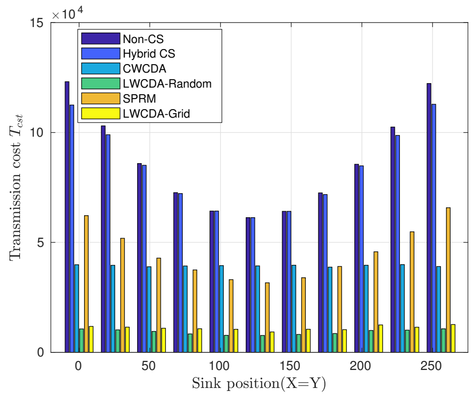


Figure 20: Comparison of the transmission cost required for data aggregation from 625 nodes deployed in an area of $256m \times 256m$ using the proposed LWCDA and SPRM, Hybrid CS, CWCDA and Non-CS methods against the sink location. The sink node location (X, Y) varies according to the line $X = Y$ where $X, Y \in [0, 256]$. Transmission cost of the proposed LWCDA method is robust and lower compared to all baseline approaches for all considered sink locations.

(grid and random deployment) in an area of $256m \times 256m$ with varying sink locations. Fig. 20 compares the transmission cost of the proposed LWCDA algorithm with that of SPRM (for grid-wise deployment) and Non-CS, Hybrid CS and CWCDA (for random deployment) with respect to various sink locations. Note that the variables $X, Y \in [0, 256]$ represent the geographic coordinates of the sink node on the considered area. The sink location (X, Y) varies on the line $X = Y$. The observation that can be made from Fig. 20 is that the transmission cost of baseline approaches except CWCDA strongly depends on the sink location. Transmission cost required for data aggregation with CWCDA is robust to sink location, but it requires more transmission cost compared to the proposed method across all sink locations. The considered traditional approaches (SPRM, Hybrid CS and NoN-CS) yield lower transmission cost when the sink is at the center of the considered area. In fact, if the sink is at the center of the considered area, every node can connect to the sink with the shortest distance. An interesting inference that can be made from Fig. 20 is that the transmission cost of the proposed LWCDA algorithm for data aggregation in both grid-wise and random deployment scenario is robust to the sink location and is much lower compared to that of the traditional methods for all the considered sink locations. This can be justified by noting that the proposed LWCDA algorithm aggregates data through clustering where required transmission cost is independent of the sink location. In addition, to aggregate measurements from randomly distributed CHs, which are connected through MST along with the sink node, incur almost same transmission cost irrespective of the sink location.

The evaluation has shown that the proposed method can deliver data with high fidelity compared to SPRM (Figs. 12a, 15a) and CWCDA (Figs. 12b, 15b). In comparison with Hybrid CS, the proposed method is competitive for lower compression rates ($\Gamma \leq 50$) and yields superior performance for higher compression rates ($\Gamma > 50$) (Figs. 12b, 15b). However, Hybrid CS approach is computationally intensive and impose multiple constraints on hardware implementation. This is especially true for lower compression rates as the generation of the column size increases. In case of transmission cost, the evaluation has illustrated that the proposed LWCDA method requires less transmission cost for data aggregation compared to SPRM ($\Gamma \leq 80\%$), Hybrid CS and CWCDA for all compression rates. Transmission cost analysis with respect to the node density (shown in Fig. 18) has shown that the proposed method can achieve significant improvement in the network lifetime for large-scale dense networks. Further, transmission cost analysis with respect to sink location (Fig. 20) has illustrated that the proposed method's transmission cost is robust to the sink location. This makes our approach attractive for various IoT applications. Furthermore, the complexity analysis discussed in Section 3.2.1 illustrates that the proposed method eliminates the burden of storing or generating the measurement matrix information at the node level. This allows the proposed method to be implementable on low-end IoT nodes. Thus, the proposed method is attractive as it yields an optimal trade-off between transmission cost, recovery fidelity and complexity for numerous IoT applications.

7. Conclusion and Future Work

In this paper, we proposed a CS based data aggregation method for IoT networks which is both low-complex and energy efficient. In the proposed method, we exploited non-overlapped clustering for data aggregation where each node contributes to only one measurement. Hence, the columns of the measurement matrix constructed from the proposed algorithm are coherent and recovery is not possible for the data which is sparse in the canonical basis (Identity matrix). However, we showed that the measurement matrix, when combined with the popular bases (DFT, DCT, DWT, Laplacian, and DiWT) could guarantee the recovery of data with high fidelity. Unlike conventional methods, in the proposed data aggregation method the node-level complexity is independent of the network size and data sparsity. The comparison of the transmission cost with traditional approaches concludes that the proposed method is energy efficient and can aid in extending the network lifetime by achieving minimal transmission cost. Hardware implementation demonstrated the efficacy of the proposed algorithm in a real-time implementation. Further, through the analysis of the measurement matrix combined with the popular bases, we found that our data aggregation method using the DFT basis yields a better reconstruction quality compared to other bases. In future, we will pursue a thorough analysis of this discrepancy and present theoretical guarantees. Another future direction of the proposed work is to include the energy harvesting mechanism [43] to the sensor nodes which helps in conserving the energy from renewable resources (e.g., piezoelectric energy harvesters and thermoelectric devices) to extend the network lifetime. Further, improving the robustness of the proposed data aggregation algorithm to the effects of dynamic characteristics of energy harvesting (due to environmental impacts) [44] is another future direction of the proposed work. Optimizing the energy required for data aggregation in the presence of heterogeneity (with respect to energy resource and delay sensitivity) [45] of sensor nodes and channel interference is another future extension of the proposed work.

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things J.*, vol. 1, no. 1, pp. 22-32, Feb. 2014.

[2] M. Amarlingam, P. K. Mishra, K. V. V. D. Prasad and P. Rajalakshmi, "Compressed sensing for different sensors: A real scenario for WSN and IoT," in *Proc. IEEE 3rd World Forum on Internet of Things*, 2016.

[3] W. Xu, Y. Zhang, Q. Shi, and X. Wang, "Energy management and cross layer optimization for wireless sensor network powered by heterogeneous energy sources," *IEEE Trans. Wireless Commun.,* vol. 14, no. 5, pp. 2814-2826, May 2015.

[4] R. Jurdak, A. G. Ruzzelli, and G. M. P. O'Hare, "Radio sleep mode optimization in wireless sensor networks," *IEEE Trans. Mobile Comput.,* vol. 9, no. 7, pp. 955-968, Jul. 2010.

[5] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: A survey," *IEEE Commun. Surv. Tutorials,* vol. 8, no. 4, pp. 48-63, Fourth Quarter 2006.

[6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory,* vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[7] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.,* vol. 25, no. 2, pp. 92-101, Mar. 2008.

[8] L. Xiang, J. Luo, and C. Rosenberg,"Compressed data aggregation: Energy-efficient and high-fidelity data collection," *IEEE/ACM Trans. Netw.,* vol. 21, no. 6, pp. 1722-1735, Dec. 2013.

[9] C. Caione, D. Brunelli, and L. Benini, "Distributed compressive sampling for lifetime optimization in dense wireless sensor networks," *IEEE Trans. Ind. Informat.,* vol. 8, no. 1, pp. 30-40, Feb. 2012.

[10] J. Luo, L. Xiang, and C. Rosenberg, “Does compressed sensing improve the throughput of wireless sensor networks?," in *Proc. IEEE Int. Conf. Commun.,* 2010.

[11] L. Xiang, J. Luo, and A. Vasilakos, “Compressed data aggregation for energy efficient wireless sensor networks," in *Proc. 8th annu. IEEE commun. Soc. Conf. Sensor, Mesh, Ad hoc Commun., Netw.,* 2011.

[12] W. Wang, M. Garofalakis, and K. Ramchandran, “Distributed sparse random projections for refinable approximation," in *Proc. 6th Int. Symp. on Inform. Process. in Sensor Netw.,* Cambridge, MA, pp. 331-339, 2007.

[13] D. Ebrahimi and C. Assi, “Optimal and efficient algorithms for projection-based compressive data gathering," *IEEE Commun. Lett.,* vol. 17, no. 8, pp. 1572-1575, Aug. 2013.

[14] D. Ebrahimi and C. Assi, “A distributed method for compressive data gathering in wireless sensor networks," *IEEE Commun. Lett.,* vol. 18, no. 4, pp. 624-627, 2014.

[15] S. A. Daresari and J. Abouei, “Toward cluster-based weighted compressive data aggregation in wireless sensor networks," *J. Ad Hoc Netw.,* vol. 36, pp. 368-385, 2016.

[16] K. C. Lan and M. Z. Wei, “A Compressibility-Based Clustering Algorithm for Hierarchical Compressive Data Gathering", *IEEE Sensors J.,* vol. 17, no. 8, pp. 2550-2562.

[17] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega, “Compressed sensing and routing in multi-hop networks," *USC CENG, Tech. Rep.,* 2009.

[18] 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 *Proc. Inform. Theory and Applicat.,* Feb. 2009.

[19] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, and S. Xiao, “Data gathering with compressive sensing in wireless sensor networks: a random walk based approach," *IEEE Trans. Parallel and Distrib. Syst.,* 2015.

[20] E. Zimos, J. F. Mota, M. R. Rodrigues, and N. Deligiannis, “Internet-of-Things data aggregation using compressed sensing with side information," in *Proc. Int. Conf. Telecomm. (ICT),* May 2016.

[21] M. Mangia, F. Pareschi, R. Rovatti and G. Setti, “Rakeness-Based Compressed Sensing and Hub Spreading to Administer Short/Long-Range Communication Tradeoff in IoT Settings," *IEEE Internet of Things J.,* vol. 5, no. 3, pp. 2220-2233, June 2018.

[22] S. P. Tirani and A. Avokh, “On the performance of sink placement in WSNs considering energy-balanced compressive sensing-based data aggregation," *J. Network and Computer Applicat.,* vol. 107, pp. 38-55, 2018.

[23] P. Rajalakshmi. (2012). *IITH Mote-Wireless Sensor Communication Module.* [Online]. Available: http://www.iith.ac.in/~raji/downloads/IITH-mote-webpage.pdf.

[24] S. Foucart and H. Rauhut, “A Mathematical Introduction to Compressive Sensing," Basel, Switzerland: Birkhäuser, 2013.

[25] E. J. Candes and T. Tao, “The Dantzig selector: Statistical estimation when p is much larger than n," *Ann. Statist.,* vol. 35, no. 6, pp. 2313-2351, 2007.

[26] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. and Computational Harmonic Anal.,* vol. 26, no. 3, pp. 301-321, May 2009.

[27] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory,* vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[28] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory,* vol. 52, pp. 1289-1306, 2006.

[29] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.

[30] E. Candies and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory,* vol. 51, no. 12, pp. 4203-4215, Dec. 2005.

[31] K. Derr and M. Manic, "Wireless sensor networks-node localization for various industry problems," *IEEE Trans. Ind. Informat.,* vol. 11, no. 3, pp. 752-762, Jun. 2015.

[32] W. Poe and J. Schmitt, "Node deployment in large wireless sensor networks: Coverage, energy consumption, and worst-case delay," in Proc. *Asian Internet Eng. Conf.,* pp. 77-84, 2009.

[33] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proc. ACM 10th Annu. Int. Conf. on Mobile Computing and Networking,* 2004.

[34] D. Donoho and J. Tanner, "Precise undersampling theorems," *Proc. IEEE,* vol. 98, no. 6, pp. 913-924, Jun. 2010.

[35] C. Lva, Q. Wanga, W. Yanb, and Y. Shena, "Diffusion wavelet basis algorithm for sparse representation of sensory data in WSNs," *J. Signal Process.,* vol. 140, pp. 12-31, 2017.

[36] S. Lindsay and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information system," in *Proc. Int. Conf. Commun. (ICC),* Bejing, China, vol. 3, pp. 1125-1130, 2002.

[37] K. C. Lan and M. Z. Wei, "A compressibility-based clustering algorithm for hierarchical compressive data gathering," *IEEE Sensors J.,* vol. 17, no. 8, pp. 2550-2562, Apr. 2017.

[38] TinyOs. [online]. Available: http://www.tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Documentation_Wiki

[39] Seek thermal, Compact, [online]. Available:https://www.thermal.com/compact-series.html

[40] INSAT3D IMAGER TIR1 Count, Indian states. [online]. Available:http://www.rapid.imd.gov.in/

[41] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Commun.,* vol. 11, no. 6, pp. 6-28, Dec. 2004.

[42] H. Safa, W. El-Hajj, and H. Zoubian, "A robust topology control solution for the sink placement problem in WSNs," *J. Network and Computer Applicat.,* vol. 39, pp. 70-82, 2014.

[43] L. Li, "MEMS/NEMS and Their Applications in Internet of Things, Energy Harvesting, Sensors and Actuators," in *Proc. IEEE International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, pp. xxii-xxii, Wuhan, 2017.

[44] C. Li, Y. Fu, Z. Liu, X. Y. Liu, W. Wu, and L. Xiong, "Spectrum Trading for Energy-Harvesting-Enabled Internet of Things in Harsh Environments," *IEEE Access,* vol. 6, pp. 16712-16726, 2018.

[45] M. Amjad, M. K. Afzal, T. Umer and B. S. Kim, "QoS-Aware and Heterogeneously Clustered Routing Protocol for Wireless Sensor Networks," *IEEE Access,* vol. 5, pp. 10250-10262, 2017.