CCS: Energy-Efficient Data Collection in Clustered Wireless Sensor Networks Utilizing Block-wise Compressive Sensing

Minh Tuan Nguyen∗, Keith A. Teague∗, and Nazanin Rahnnavard†

∗Oklahoma State University, Stillwater, OK 74078
†University of Central Florida, Orlando, FL 32816

Emails: {tuanminh.nguyen, keith.teague}@okstate.edu and nazanin@eecs.ucf.edu

Abstract

In this paper, we propose an integration of compressive sensing (CS) and clustering in WSNs utilizing block diagonal matrices (BDMs) as the measurement matrices. Such an integration results in a significant reduction in the power consumption related to the data collection. The main idea is to partition a WSN into clusters, where each cluster head (CH) collects the sensor readings within its cluster only once and then generates CS measurements to be forwarded to the base station (BS). We considered two methods to forward CS measurements from CHs to the BS: (i) direct and (ii) multi-hop routing through intermediate CHs. For the latter case, a distributed tree-based algorithm is utilized to relay CS measurements to the BS. The BS then implements a CS recovery process in the collected $M$ CS measurements to reconstruct all $N$ sensory data, where $M \ll N$. Under this novel framework, we formulated the total power consumption and discussed the effect of different sparsifying bases on the CS performance as well as the optimal number of clusters for reaching the minimum power consumption.

Acknowledgements: This work is supported by School of Electrical and Computer Engineering (ECE), Oklahoma State University and the National Science Foundation under Grant No. ECCS-1418710 and CCF-1439182.
I. INTRODUCTION

A. Motivation

Wireless sensor networks (WSNs) have found numerous uses in both military and civilian applications [1]. Sensors in WSNs are usually randomly dropped/deployed in a sensing area that needs to be monitored. They are often deployed in harsh conditions without maintenance or renewable power supply. Therefore, the connection and operation of these networks rely on these small and inexpensive devices under a severe energy constraint. Saving energy in data collection in such networks is always a critical problem that directly impacts network lifetime.

The spatial correlation of the sensor readings in WSNs results in an inherent sparsity of data in a proper basis. This sparsity facilitates the application of the compressive sensing (CS) [2], [3], [4] technique in data collection in WSNs [5], [6], [7]. CS offers a novel framework to reconstruct all sensor readings based on a small number of CS measurements, which creates an opportunity to significantly reduce power consumption.

In recent years, there have been several studies on the integration of CS and data collection in WSNs, e.g., [8], [9], [10], [11], [12]. In these methods, sensor readings are multiplied by a selected set of coefficients and are sent to the base station (BS) following some routing methods such as gossip-based, random walk, tree-based, or cluster-based. The CS measurements are collected at the BS as $\mathbf{Y} = \Phi \mathbf{X}$, where $\Phi$ is called the measurement matrix and vector $\mathbf{X}$ represents all unknown readings from all the sensors. The resulting measurement matrices can be sparse or dense with Gaussian coefficients depending on the underlying routing method.

In this paper we combine the clustering technique, which has been shown to save and balance energy consumption for WSNs, and block diagonal matrices (BDMs) as the CS measurement matrices. We propose an algorithm called Cluster-Based Compressive Sensing Data Collection (CCS) in which the CS measurements are generated at each cluster-head (CH) in the clustered networks. We consider two methods to send these measurements to the BS: directly (one-hop), denoted as $D$-CCS and multi-hop relaying through the intermediate CHs, denoted as $I$-CCS. In the BDM, the size of each sub-matrix (block) depends on the size of each cluster. We formulate the total power consumption and discuss the effect of different sparsifying bases on the CS performance as well as the optimal number of clusters for reaching the minimum power consumption. For our formulations, two common positions for the BS are considered: the
BS located at the center and outside the sensing area. Based on that, we can obtain the optimal number of clusters that provides the minimum power consumption for our networks. In our simulation, we consider both random sparse signals in canonical basis and real sensor readings. The real sensor readings, although not sparse in canonical domain, are sparse in frequency domains (DCT or Wavelet). We compare different combinations between the measurement matrices and the sparsifying matrices in our theoretical and simulation results. Our work shows promise not only in WSNs but also in mobile sensor networks or vehicle networks for data monitoring or similar purposes. The power saving in communications prolongs the lifetime in such networks.

B. Related work

Clustering is an effective way to enhance the performance and lifetime of a WSN and many different clustering algorithms have been studied so far [13], [14], [15], [16], [17]. Each cluster has a cluster head (CH) and CHs can be pre-determined [13] or be selected while doing clustering as in the following algorithms. K-means [18], [19], [20] is a well-known and simple clustering algorithm that chooses CHs for K clusters at the central point of each cluster. Since power consumption is dependent on transmission distance, this helps to minimize the intra-cluster power consumption. In general, CHs consume power much more than other sensors as they transmit the entire cluster’s data to the BS. In LEACH [17], sensor nodes randomly elect themselves to be CHs. This way, the high power consumption related to communication with the BS will be distributed among the nodes in the network. The HEED algorithm [21] chooses CHs based on the highest residual energy of sensors to balance network energy. EEUC [22] makes unequal size clusters and muti-hop links between CHs to reduce and balance the power consumption. Fault-tolerant clustering is considered in [23] in order to recover sensors in a failed cluster. Load-balancing clustering [24] makes the whole network consume power equally and [25] finds the optimal number of clusters to get the lowest power consumption for WSNs.

Utilizing CS is also an effective way to reduce the number of required samples from a sparse signal. Due to the correlation between the sensor readings in a WSN, the monitored signal can have a sparse representation in a proper domain such as DCT or wavelet. Accordingly, CS has found applications in data collection in WSNs [5], [6], [7]. In [9] a tree-based algorithm called CDG is proposed to balance the payload falling on nodes close to the BS. The measurement
matrix is a full Gaussian one that consumes more power than the sparse binary measurement matrices [26]. To reduce power consumption, other methods based on sparse CS matrices have been proposed [27], [28]. In [27] the authors proposed MTT, which is an heuristic algorithm to compute a minimum transmission spanning tree for data collection in WSNs using CS. Another tree-based study is TCS [28] that utilizes sensor storage to reduce the number of transmissions in a routing tree.

Hybrid data collection schemes where both raw data and combined samples are in traffic are mentioned in [29], [30], [31]. In [29] CS operation requires each node in the WSN to send exactly $M$ packets. $M$ is denoted as the number of CS measurements required to reach a given reconstruction quality. A spanning tree is used to partition the network into sub-nets. [30] and [31] propose a scheme called hybrid CS aggregation that combines the non-aggregation and plain CS mentioned in [29] to reduce the traffic loads sending from each node. The non-aggregation method is used if a node receives less than $(M - 1)$ raw readings from its downstream nodes. Otherwise, plain CS is used. In [29], [30], [31], each sensor needs to send up to $M$ samples to the BS to contribute $M$ CS measurements for signal recovery. The BDM is mentioned in [31] to reduce partially the total samples being sent from each sub-net. In [32], [33], a WSN is partitioned into clusters. Sensor readings are sent to CHs and the CHs send the received data to the BS. Since the measurement matrix is full Gaussian, each cluster needs to generate $M$ samples to contribute to $M$ CS measurements at the BS.

This paper extends our previous studies [12], [34]. In our proposed CCS algorithm, all non-CH sensors send their own readings to their corresponding CHs only once during $M$ rounds of measurement collection. The CHs generate sub-matrices with Gaussian coefficients ($\phi_i$) and generate $m_i$ CS measurements using $y_i = \phi_i \bar{x}_i$, where $y_i$ is the measurement vector including $m_i$ CS measurements collected from the $i^{th}$ cluster, and $\bar{x}_i$ represents all readings in the $i^{th}$ cluster ($n_i$ sensors in cluster $i^{th}$ provide $n_i$ readings). We also prove $(\frac{m_i}{M} = \frac{n_i}{N})$ in the next sections. The CS measurements are either sent directly from CHs to the BS or in a multi-hop fashion. We provide Table I to compare our data collection method with other related studies. In the table, we focus on the network structures, the number of times each sensor sends its reading to its CH or the subnet-head to contribute to $M$ CS measurements, and the number of combined data packets being sent from each CH or subnet-head to contribute to $M$ CS measurements. As shown in Table I, our proposed CCS significantly reduces certain number of packets, non-CH
sensors only transmit once their readings and the CHs transmit $m_i$ CS measurements.

**TABLE I**

**Comparison between the existing data collection methods and CCS**

<table>
<thead>
<tr>
<th>Related papers</th>
<th>Network structure</th>
<th>Measurement matrix</th>
<th>Number of times sending data from each sensor</th>
<th>Number of packets sending data from each cluster-head or subnet-head</th>
<th>CS measurements are generated at</th>
</tr>
</thead>
<tbody>
<tr>
<td>[29], [30], [31]</td>
<td>Spanning tree</td>
<td>Full dense Gaussian</td>
<td>Up to $M$ times</td>
<td>$M$</td>
<td>Base-station</td>
</tr>
<tr>
<td>Partly [31]</td>
<td>Spanning tree</td>
<td>BDM</td>
<td>$m_i$ times</td>
<td>$m_i$</td>
<td>Subnets</td>
</tr>
<tr>
<td>[32], [33]</td>
<td>Cluster</td>
<td>Full dense Gaussian</td>
<td>Only once</td>
<td>$M$</td>
<td>Base-station</td>
</tr>
<tr>
<td>CCS</td>
<td>Cluster</td>
<td>BDM</td>
<td>Only once $m_i$</td>
<td>$m_i$, ($\frac{m_i}{M} = \frac{n_i}{N}$)</td>
<td>Cluster-heads</td>
</tr>
</tbody>
</table>

*Note: $m_i$ and $n_i$ represent the number of CS measurements required and the number of sensor nodes from cluster $i$th, respectively.

The main contributions in this work are summarized as follows:

1- Two versions of the CCS algorithm (D-CCS forwards CS measurements directly to the BS and I-CCS forwards CS measurements through intermediate CHs to the BS), which combine clustering and BDMs as CS matrices, are proposed. CCS significantly reduces transmission power consumption in WSNs.

2- Expressions for power consumption in both D-CCS and I-CCS are formulated.

3- The optimal numbers of clusters are suggested for the networks in different scenarios to consume the least power.

The remainder of this paper is organized as follows. The Background and Problem Formulation are addressed in Section II. The CCS algorithm is stated in Section III. In Section IV and Section V, the two models to forward the CS measurements to the BS are presented with corresponding transmission power consumption analysis and simulation results. In Section VI use of the DCT and sending only $k$ large transformed coefficients is considered for comparison purpose since $k \ll M \ll N$. Finally, conclusions and suggestions for future work are presented.
II. BACKGROUND AND PROBLEM FORMULATION

A. System Model

In the network model considered here, we assume that $N$ sensors have been distributed uniformly at random in a sensing area. $N_c$ out of $N$ nodes in the network are selected uniformly at random as CHs with probability $\frac{N_c}{N}$ and the other nodes connect to the closest CH, as mentioned in LEACH [17]. This creates $N_c$ non-overlapped clusters with each cluster having one CH and $(\frac{N}{N_c} - 1)$ non-CH sensors on average. Each CH is assumed to have enough capacity to store the data vector collected from its own non-CH sensors and to generate a number of CS measurements required based on the number of sensors in the cluster.

We assume that each node can adjust its power level based on its distance from its CH and this can be done based on the received signal strength [35]. The consumed power for reaching a destination node $j$ with distance $d_{ij}$ from the node $i$ is $P_{ij} = d_{ij}^{\alpha}$. Parameter $\alpha$ is called the path loss exponent, which is usually between 2 and 4, depending on the characteristics of the channel [36]. In this paper, we assume $\alpha = 2$. For the reconstruction error related to CS signal recovery we consider the normalized reconstruction error $\frac{\|\hat{x} - \Phi x\|_2}{\|x\|_2}$.

B. Compressive Sensing (CS) Overview

1) Sparse presentation of signals: Compressive sensing (CS) is a novel technique for recovering random or compressible signals from undersampled random projections, also called measurements. A signal $X = [x_1 \ x_2 \ldots x_N]^T \in \mathbb{R}^N$ is defined to be $k$-sparse if it has a sparse representation in a proper basis $\Psi = [\psi_{i,j}] \in \mathbb{R}^{N \times N}$, where $X = \Psi \theta$ and $\theta$ has only $k$ non-zero elements. Based on the CS paradigm, a $k$-sparse signal can be under-sampled and be recovered from only $M \ll N$ random measurements $Y = [y_1 \ y_2 \ldots y_M]^T \in \mathbb{R}^M$.

2) Signal sampling and the measurement matrix: The CS measurements are generated by $Y = \Phi X$, where $\Phi = [\phi_{i,j}] \in \mathbb{R}^{M \times N}$ is called the measurement matrix and is often a dense Gaussian matrix or a sparse binary matrix [26]. The $i^{th}$ element in the measurement vector $Y$ is formed by $y_i = \sum_{j=1}^{N} \phi_{i,j} x_j$.

1In fact, we have $P_{ij} \propto d_{ij}^{\alpha}$ [35]. However, since we are interested in a comparison of different schemes, and not the exact values of the power, without loss of generality, we can consider the constant factor as one.
3) Signal recovery: It has been shown that a $k$-sparse signal can be reconstructed with high probability from only $M = \mathcal{O}(k \log N / k)$ CS measurements employing the following $l_1$ optimization problem [37], [3]

$$\hat{\theta} = \arg\min_{\theta} \| \theta \|_1, \text{ subject to } Y = \Phi \Psi \hat{\theta},$$  

(1)

where $\| \theta \|_1 = \sum_{i=1}^{n} |\theta_i|$ and $\hat{X} = \Psi \hat{\theta}$. The $l_1$ optimization problem can be solved with linear programming techniques such as Basis Pursuit (BP) [37]. In reality, we also have to consider the noise while sampling and sending the measurements (in our case we collect measurements and send them to the base-station): $Y = \Phi X + e$, with $\| e \|_2 < \epsilon$ and recover

$$\hat{\theta} = \arg\min_{\theta} \| \theta \|_1, \text{ subject to } \| Y - \Phi \Psi \hat{\theta} \|_2 < \epsilon.$$  

(2)

C. Block Diagonal Matrices

As mentioned before, our goal is to utilize block-wise CS for data collection in clustered WSNs. Since CS measurements are formed at each CH, the overall CS measurement matrix formed at the BS will no longer have the form of the conventional CS matrices, such as a dense matrix with all the entries being i.i.d. Gaussian or Rademacher. Instead, our CCS algorithm results in block diagonal matrices (BDMs), in which $\phi_i$, the $i$th block in $\phi$, corresponds to the $i$th cluster and has i.i.d. Gaussian entries. Let $x_i$ denote a vector of size $N_i$ consisting of the sensor readings of the nodes in the $i$th cluster, and $y_i = \phi_i x_i$ denote a vector including $M_i$ CS measurements collected from the $i$th cluster. We have

$$\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_{N_c}
\end{bmatrix} =
\begin{bmatrix}
  \phi_1 \\
  \phi_2 \\
  \vdots \\
  \phi_{N_c}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_{N_c}
\end{bmatrix}$$  

(3)

The BDM is created due to the new way to collect CS measurements. Fortunately the matrix satisfies the restricted isometry property (RIP) to be able to work with CS. The RIP of BDMs has been studied in [38], [39] and it has been shown that BDMs can satisfy RIP and therefore can be used as efficient measurement matrices. The required number of the measurements though

---

2A Rademacher random variable takes a value of +1 or -1 with equal probability.
depends on the basis in which the signal is sparse. According to [38], [40], the number of measurements required for a BDM, consisting of \( N_c \) blocks with Gaussian entries, to satisfy RIP with high probability is given as [38]

\[
M = O(k \tilde{\mu}^2 \log^2(k) \log^4(N)),
\]

(4)

where \( \tilde{\mu} = \min\{ \sqrt{N_c}, \mu \} \) and \( 1 \leq \mu \leq \sqrt{N} \) is the coherence between \( \psi \) and canonical basis and defined as \( \mu = \sqrt{N} \max |\psi_{ij}| \). From (4) several very interesting points can be concluded. If the sparsifying basis has a small coherence with the canonical basis (such as the Fourier basis or DCT basis), then increasing \( N_c \) (which results in a more sparse matrix) does not increase \( M \). On the other hand, if the sparsifying basis has a large coherence with the canonical basis (such as Wavelet or Canonical bases), then for \( N_c < \mu^2 \), \( M \) is a linear function of \( N_c \). These results will be seen in our simulations later (Figures 4, 5, 8).

D. Problem Formulation

Consider a vector \( \underline{X} = [x_1, x_2, \ldots, x_{N_c}] \) in which \( x_i \) represents unknown sensor readings from cluster \( i \). We assume the BS needs \( M \) CS measurements collected from the network to recover precisely all raw readings. The \( i^{th} \) CH (\( i = 1, 2, \ldots, N_c \)) generates an \( m_i \times n_i \) block of Gaussian coefficients (\( \phi_i \)), where \( n_i \) is the number of sensors in the \( i^{th} \) cluster. The CH then generates \( m_i \) CS measurements using \( y_i = \phi_i x_i \) and sends them to the BS. The exact value of \( m_i \) out of \( M \) is calculated based on \( \frac{n_i}{N} \) which is shown in Lemma 1 in the next section.

The BS receives random seeds to generate the BDM and the measurement vector \( \underline{Y} = [y_1, y_2, \ldots, y_{N_c}] \) separately from \( N_c \) CHs. The greater number of measurements, the better the accuracy of the reconstruction. In the following sections we will analyze the use of BDMs in the CS recovery processes with different types of signals; sparse in canonical basis or in frequency domain. Transmission power consumptions for data collection in the networks are formulated, analyzed and finally simulated in arbitrary networks. The optimum number of clusters such that power consumption is minimized will be determined for each network.

III. CCS: CLUSTER-BASED COMPRESSIVE SENSING FOR DATA COLLECTION IN WSNs

The proposed CCS algorithm is summarized in Algorithm 1 below. CCS is divided into two parts. The first is the underlying clustering that can be based on different methods such as
Algorithm 1: CCS - Cluster-Based Compressive Sensing Algorithm

(1)-Clustering phase:
- $N_c$ out of $N$ sensors are randomly chosen as $CH_i$ ($i = 1,...,N_c$) with probability $p = \frac{N_c}{N}$.
- Assign each sensor ($S_j$) in the entire network to the nearest cluster $C_i$ as
  $$S_j \in C_i \text{ if } \| S_j - CH_i \| < \| S_j - CH_t \|$$
  for $j = 1,...,N$, $i \neq t$ and $i = 1,...,N_c$

(2)-Measurement generating phase
- Non-CH sensors send their data once to their CHs. Data vector $x_i$ is stored at $CH_i$
- CS measurements are generated at each CH as
  $$\text{for } i = 1 \text{ to } N_c \text{ do}
  \begin{align*}
  \phi_i &= \text{randn}(m_i, n_i) \quad \% \text{ sub-matrix is created at each cluster} \\
  y_i &= \phi_i x_i \quad \% m_i \text{ measurements are created at } CH_i
  \end{align*}
  \text{end}
$$

(3)-Measurement collection and Data recovery phase
- $M = \sum_{i=1}^{N_c} (m_i)$ CS measurements are forwarded separately from CHs to the BS.
- Given the BDM $\Phi$, all unknown values ($X$) are reconstructed based on ($Y$).

K-means [19] and LEACH [17] which we use for comparison purposes in our simulations. The second is the generation of CS measurements based on BDMs and forwarding them to the BS either directly or in multiple hops to be addressed in Sections IV and V, respectively. In a real WSN each cluster may have a different number of sensors and accordingly different numbers of measurements are required from each CH. The following lemma relates $n_i$ and $m_i$.

Lemma 1: Let $M$ be the number of required measurements to be taken from all clusters to satisfy the RIP for a block diagonal matrix with blocks $\phi_i$ of size $m_i \times n_i$. To get the best CS performance in term of the reconstruction error, the number of measurements from the $i^{th}$ cluster ($m_i$) should be linearly proportional to the number of sensors in the cluster ($n_i$). In other words, $\frac{m_i}{M} = O(\frac{n_i}{N})$.

Proof: According to [41], the number of CS measurements required to reconstruct a $k$-sparse signal of length $N$ using a dense Gaussian measurement matrix of size $M \times N$ is given
as
\[ M = O(k \log \frac{N}{k}). \]  
(5)

Now assume the \( k \) non-zero elements are uniformly distributed in the vector \( X \), and \( X \) has been partitioned into sub-vectors of size \( n_i \). Therefore, we have
\[ \frac{n_i}{N} = \frac{k_i}{k}, \]  
(6)

where \( k_i \) is the average number non-zero elements in the sub-vector \( i^{th} \). Using (5), and considering that \( \phi_i \) is a dense Gaussian measurement matrix we have:
\[ m_i = O(k_i \log \frac{n_i}{k_i}). \]  
(7)

From (5), (6) and (7), we obtain
\[ \frac{m_i}{M} = \frac{O(k_i \log \frac{n_i}{k_i})}{O(k \log N/k)} = O(k_i) = O(\frac{n_i}{N}). \]

Hence,
\[ \frac{m_i}{M} = O(\frac{n_i}{N}). \]  
(8)

Although this lemma has been proved for the case that the signal \( X \) is sparse in the canonical domain, our simulation results, below, show that the lemma holds even when \( X \) is sparse in another domain. Here is an example. Assume a network of size \( N \) is divided into two clusters with sizes of \( n_1 = 0.7N \) and \( n_2 = 0.3N \). A fraction \( T \) and \( 1 - T \) of the total measurements are collected from cluster 1 and 2, respectively. Signal \( x \) is assumed to be sparse in DCT. Figure 1 depicts the reconstruction error versus \( T \), when \( N = 1000 \) and \( M = 250 \). As we see the minimum error occurs when \( T = m_1/M \) is exactly equal to \( n_1/N = 0.7 \). This is the same result stated in Lemma 1.

IV. DIRECTLY SEND CS MEASUREMENTS TO THE BS (DCCS)

A. Network Model

In this model, we assume a WSN with \( N \) sensors deployed in a square sensing area sized \( L \times L \) distance unit\(^2\). Non-CH sensors send their readings directly to the CHs they belong to based on real distances \((r)\). The CHs generate CS measurements and send them directly to the BS. The position of BS is changeable which can be inside or outside the sensing area.
B. Power Consumption Analysis for DCCS

We refer to the communication cost associated with the communication between the non-CH nodes and their CHs as the intra-cluster power consumption which is denoted as $P_{\text{intra-cluster}}$. The CHs create the CS measurements as combinations of all received data within each cluster ($y_i = \phi_i x_i$) and send the measurements directly to the BS. The corresponding power consumption is referred to as $P_{\text{to BS}}$. The total power consumption is formed as

$$P_{\text{total}} = (P_{\text{intra-cluster}} + P_{\text{to BS}}).$$

1) Analysis of $P_{\text{intra-cluster}}$: We assume a uniformly distributed WSN divided into $N_c$ clusters with the same number of sensors as $N/N_c$, consisting of one CH and $(N/N_c - 1)$ non-CH nodes. We have

$$P_{\text{intra-cluster}} = N_c (N/N_c - 1) E[r^\alpha],$$

where $r$ is a random variable representing the distance of a non-CH sensor to its corresponding CH and $\alpha$ is path loss exponent that we assume to be 2 throughout the paper. We can calculate $E[r^2]$ as following:

$$E[r^2] = \int \int (x^2 + y^2) \rho(x, y) \, dx \, dy$$

$$= \int \int r^2 \rho(r', \theta) \, r' \, dr' \, d\theta,$$
in which $\rho(x, y)$ is the node distribution. To make the analysis tractable, similar to [25], we assume each cluster area is a circle with radius $R = L/\sqrt{\pi N_c}$ and the density of the nodes is uniform throughout the cluster area, i.e. $\rho(r', \theta) = 1/(L^2/N_c)$. We have [25]:

$$E[r^2] = \frac{1}{(L^2/N_c)} \int_{\theta=0}^{2\pi} \int_{r'=0}^{R} r'^3 dr' d\theta = \frac{L^2}{2\pi N_c}. \tag{13}$$

and accordingly

$$P_{\text{intra-cluster}} = (\frac{N}{N_c} - 1) \frac{L^2}{2\pi}. \tag{14}$$

As we see, the total intra-cluster power consumption is a decreasing function of the number of clusters.

2) Analysis of $P_{\text{to BS}}$: Next, we need to determine $P_{\text{to BS}}$, which is based on the distances between CHs and the BS and the total number of measurements $M$ required to be transmitted from each CH to the BS. We assume the BS is located at the location $(L_i, \frac{L}{2})$ with respect to our reference point (see Figure 2). The average consumed power by all CHs is given by

$$P_{\text{to BS}} = ME[d^2], \tag{15}$$

where $d$ is a random variable representing the distance between the CHs and BS. Assuming that all CHs are randomly distributed in the sensing area, the expected squared distance between CHs and the BS is given by

$$E[d^2] = \int_0^L \int_0^L [(x - L_i)^2 + (y - \frac{L}{2})^2] f(x, y) dxdy \tag{16}$$

$$= \frac{1}{L} \left[ \frac{(L - L_i)^3}{3} + \frac{L_i^3}{3} + \frac{L^2}{12} \right], \tag{17}$$

in which $f(x, y) = \frac{1}{L^2}$ (uniform distribution of CHs). From Equations (15) and (17) we conclude that $P_{\text{to BS}}$ is independent of the number of the clusters. Using (9), (14), (15), and (17), the total power consumption can be formulated as

$$P_{\text{total}} = \left( \frac{N}{N_c} - 1 \right) \frac{L^2}{2\pi} + \frac{ML^2}{12}. \tag{18}$$

We usually have two common positions for the BS, at the center of the sensing area ($L_i = L/2$) and outside the sensing area ($L_i \geq L$). For the former case, (18) is simplified as

$$P_{\text{total}} = \left( \frac{N}{N_c} - 1 \right) \frac{L^2}{2\pi} + \frac{ML^2}{6}. \tag{19}$$
According to (4) [38], we can see that for canonical and wavelet bases, the number of required measurements is a linear function of $N_c$. Based on this, we can state the following lemma to find the optimal number of clusters $N_c^*$ for minimizing the power consumption.

**Lemma 2:** Assume the number of required measurements is a linear function of the number of clusters, i.e., $M = aN_c + b$, where $a$ and $b$ are appropriate constants. In order to achieve the lowest power consumption with CCS, the optimal number of clusters is given by

$$N_c^* = \sqrt{C \cdot N} = O(\sqrt{N}), \quad (20)$$

where

$$C = \frac{6L^3}{4\pi a [(L - L_i)^3 + L_i^3] + \pi a L^3}. \quad (21)$$

**Proof:** Adding the linear function of $M$ mentioned in the lemma into the general equation of the total power consumption (18), we have

$$P_{total} = \left(\frac{N}{N_c} - 1\right) \frac{L^2}{2\pi} + \frac{aN_c + b}{L} \frac{(L - L_i)^3 + L_i^3}{3} + \frac{(aN_c + b)L^2}{12}. \quad (22)$$

We have

$$\frac{dP_{total}}{dN_c} = -\frac{NL^2}{N_c^22\pi} + \frac{a}{L} \frac{(L - L_i)^3 + L_i^3}{3} + \frac{aL^2}{12}. \quad (23)$$

By forcing $\frac{dP_{total}}{dN_c} = 0$, we can obtain the optimal number of cluster $N_c^*$ calculated as

$$N_c^* = \sqrt{\frac{6L^3N}{4\pi a [(L - L_i)^3 + L_i^3] + \pi a L^3}} = \sqrt{C \times N}. \quad (24)$$

So,

$$N_c^* = O(\sqrt{N}). \quad (25)$$

---

**C. Simulation Results for DCCS**

In this section, we work with both random $k$-sparse signals (sparse in a canonical basis, i.e., $\psi$ is the identity matrix) and real sensor readings (which are sparse in DCT or wavelet bases). We create a random network with number of sensors $N = 2000$ and size $L = 100$ according to the network model from Sections II-A and IV-A. We use K-means and LEACH clustering algorithms to arrange sensors into $N_c$ clusters. Then, we apply our CS-based data collection and calculate the total power consumption of the network for collecting $M$ CS measurements that is
required for reaching a target error rate of 0.1. The number of measurements from each cluster is linearly proportional to the size of the cluster based on Lemma 1. Simulation results based on K-means and LEACH clustering as well as the analytical results derived in Section IV-B are provided below.

Figure 3 shows the histogram for the number of sensors in each cluster for both K-means and LEACH when $N_c = 10$. K-means generates clusters that are more uniform in size, resulting in a lower expected intra-cluster power consumption since it aims to minimize the within-cluster sum of squares [19]. Next we find the number of measurements required based on CS to satisfy a target error for our network when it is clustered into different number of clusters ($N_c = [1, 2, \ldots, 50]$). Each clustering method provides a different BDM as the measurement matrix. For deriving our analytical results, we assumed clusters with equal size, while K-means and LEACH have different size blocks. For comparison, we also generated a BDM with all equal size blocks as the measurement matrix and found the number of required measurements to reach the target error. This is referred to as CS-based uniform clustering to compare with other methods. We choose a fixed target error in all our simulations as $target\ error= 0.1$. After finding the number of measurements required, we will find the power consumption for different choices for the location of the BS as mentioned before. We present our first simulation with random $k$-sparse signals, then real sensor readings as actual temperatures.

1) $\mathbf{X}$ as a random $k$-sparse vector: In this example, we consider $\mathbf{X}$ to be sparse in the canonical basis. We create a 100-sparse vector $\mathbf{X}$ with length $N = 2000$. The measurement matrix is an $M \times N$ BDM, where $M$ is the number of measurements required to satisfy the target error of 0.1. We obtain the number of required measurements for three algorithms as shown in Figure 4.
As shown in Figure 4, increasing $N_c$ leads to a degradation in the CS performance and a linear increase in the number of required measurements (as discussed in Section II-C). This increases $P_{to\ BS}$. On the other hand, $P_{intra\-\ cluster}$ is a decreasing function of $N_c$. Therefore, there is an optimal $N^*_c$, for which the total power consumption is minimized. Figure 6(a) depicts $P_{total}$ when the BS is at the center of the sensing area. In this case, we have $N^*_c = 14$.

Figures 6(b)(c)(d) depict $P_{total}$ when BS is outside the sensing area at different locations. In the first case, the minimum power consumption occurs for $N^*_c = 9$. Figures 6(c) and (d) show $P_{total}$ when the BS is far from the sensing area. The optimal number of clusters is $N^*_c = 4$ and $N^*_c = 2$ for $L_i = 2L$ and $L_i = 3L$, respectively. In such cases, $P_{to\ BS}$ will be the dominating factor in $P_{total}$ and accordingly $N^*_c$ becomes smaller. It is worth noting that the results with K-means clustering match the analytical results much better than the case with LEACH clustering. This is due to the nonuniform cluster sizes in LEACH compared to K-means (Figure 3).

2) $X$ as real sensor readings: We use real sensor readings from Sensorscope: Sensor Networks for Environmental Monitoring [42]. $X$ is dense in the canonical domain. In order to apply CS, as mentioned in the background section, we need a sparsifying basis. Next, we will consider the utilization of both DCT and wavelet bases for this purpose.

* Wavelet as the sparsifying basis: This case is similar to the case discussed in Section IV-C1 in the sense that the wavelet basis also has a large coherence $\mu$. As discussed in Section II-C, this causes a linear increase in the number of required measurements versus $N_c$. Our simulation results in Figure 5 depict this fact. Similarly, there will be an optimal $N^*_c$, for which the total
Fig. 6. Total power consumption for the network using 100-sparse signals when BS at different positions from the area

power consumption is minimized. Figure 7 depicts $P_{\text{total}}$ when the BS is at the center, $L_i = L$, $L_i = 3L$, and $L_i = 5L$, respectively. The optimal number of clusters are $N_c^* = 18$, $N_c^* = 12$, $N_c^* = 2$ or 3 (depending on the clustering scheme), and $N_c^* = 2$, respectively. As we make the BS farther from the sensing area, $P_{\text{to BS}}$ becomes a more dominating factor in $P_{\text{total}}$ and this leads to a decrease in $N_c^*$.

* **DCT as the sparsifying basis**: In this case we employ DCT as the sparsifying basis. As discussed in Section II-C, the DCT is incoherent with the canonical basis and the CS performance does not degrade with increasing $N_c$. This can be seen in our simulation results shown in Figure 8. The number of required measurements to reach a target reconstruction error is almost constant versus changing $N_c$. Given that $M$ does not change with $N_c$, we can see from Equation (18)
(a) Total power consumption when the BS is at the center of the sensing area. Here, $N^*_c = 18$.

(b) Total power consumption when $L_i = L$. Here, $N^*_c = 12$.

(c) Total power consumption when $L_i = 3L$. Here, $N^*_c = 2$ or $3$.

(d) Total power consumption when $L_i = 5L$. Here, $N^*_c = 2$.

Fig. 7. Total power consumption for the network using real data and Wavelet sparsifying matrix when BS at different positions.

Fig. 8. Number of measurements required when DCT is considered as the sparsifying basis.

Fig. 9. Total power consumption when the BS at the center of the sensing area.
that $P_{\text{total}}$ is a decreasing function of $N_c$. This is also shown in Figure 9 and 10 for the BS being at the center and $L_i = 3L$, respectively. On the other hand, since we are collecting $M$ measurements from the networks, we have $N_c \leq M$. Therefore, $N^*_c = M$ and the smallest size of each cluster on average is $N/M$ sensors.

![Figure 10. Total power consumption when the BS outside the sensing area at $3L$.](image)

![Figure 11. All transmissions in the clustered network with inter-cluster multi-hop routing when the BS at the center.](image)

*Remarks on the effect of the sparsifying basis on performance*: Based on our discussion, we can conclude that under the given cluster scenarios and assuming that the signal of interest is sparse in both wavelet and DCT bases, employing the DCT will be more energy efficient. This is because when $\psi$ is a DCT matrix, $\phi$ can become very sparse (by increasing $N_c$) without a considerable loss in the CS performance. Our analytical and simulation results showed that in this case the consumed power is a decreasing function of $N_c$ and more clusters results in more power savings and $N^*_c = M$.

V. INTER-CLUSTER MULTI-HOP ROUTING IN CCS (ICCS)

In this section, we propose a method for further energy saving during data collection where by CHs transmit the CS measurements through intermediate CHs to the BS. We refer to this method as inter-cluster multi-hop routing in CCS (ICCS). For networks with a small number of clusters, ICCS may not help because the multi-hop routing paths might require more power than transmitting directly. But with a large number of CHs, ICCS can significantly reduce the power needed to transmit the CS measurements.
Since we already have clusters formed by K-means or LEACH, we develop an iterative greedy distributed algorithm to form a tree that connects all CHs with the root at the BS. We assume all the CHs have the same transmission range ($R$) and that CHs within that range can communicate with one another. An appropriate $R$ should be chosen based on the number of CHs formed so that all CHs can be connected as an undirected geometry graph $G(V,E)$, where $V$ is the set of vertices referred to the number of CHs, and $E$ is the set of edges referred to the number of communications links between CHs. Based on the graph, we can deploy the GDA to form the routing paths for the CHs: All CHs broadcast their information about the number of hops away from the BS to their neighbors. At the first iteration, only the CHs which are close to the BS (their transmission ranges cover the BS) have the number of hops (NoH). They name their NoH as ”1” and broadcast their own updated information to their neighbors in the next iterations. The algorithm iterates running until there is no change in the communication links between all CHs. This algorithm is shown below as Algorithm 2.

**Algorithm 2: Distributed Tree-based Routing Algorithm**

**Initialization phase:**
- $N_c$ clusters marked as $CH_i$, $i = 1,...,N_c$; CH’s transmission range is $R$.
- NoH(BS) = 0; Nei = set of $CH_i$’s neighbors;

**Tree forming phase**

while the routing paths are changing do

for $i = 1$ to $N_c$ do

  for $j = 1$ to $N_c$ do

    if $d(i,j) < R$ then

      CH(i) chooses CH(j) if NoH(j) = min{NoH(Nei)};

      NoH(i) = NoH(j) + 1;

    end

  end

end
A. Network Model

To simplify the problem, in this model, we assume that the sensing area has a circular shape with radius $R_0$ in which the BS is at the center. With intra-cluster transmission, sensors still can adjust their power level to transmit data to the CHs, while all CHs only use one transmission range, denoted as $R$, to connect to other CHs and the BS. We still have $N$ sensors uniformly distributed in the area, and the path-loss exponent is assumed to be equal to $2$ ($\alpha = 2$).

As shown in Figure 11, all the transmissions show CHs receive readings from their cluster members as non-CH sensors and they transmit CS measurements through other CHs or directly to the BS at the center depending on their positions and $R$.

B. ICCS Power Consumption Analysis

As before, we refer to the communication cost associated with the communication between the non-CH nodes to CHs as the intra-cluster power consumption and denote it as $P_{\text{intra-cluster}}$. The CHs create the CS measurements as the combinations of all reading data within each cluster ($y_i = \phi_i x_i$) and send the measurements to the BS in a multi-hop fashion. The corresponding power consumption is referred to as $P_{\text{to BS}}$. The total power consumption is formed as

$$P_{\text{total}} = (P_{\text{intra-cluster}} + P_{\text{to BS}}). \quad (26)$$

1) Analysis of $P_{\text{intra-cluster}}$: Similar to the assumption for Equation (10), we have

$$P_{\text{intra-cluster}} = N_C \left( \frac{N}{N_c} - 1 \right) E[r^2], \quad (27)$$

and

$$E[r^2] = \int \int r'^2 \rho(r', \theta) r' \, dr' \, d\theta. \quad (28)$$

We assume each cluster area is a circle with radius $R = R_0/\sqrt{N_c}$ and the density of the nodes $\rho(r', \theta) = 1/(\pi R_0^2/N_c)$. Hence,

$$E[r^2] = \frac{1}{(\pi R_0^2/N_c)} \int_{\theta=0}^{2\pi} \int_{r'=0}^R r'^3 dr' \, d\theta = \frac{R_0^2}{2N_c}, \quad (29)$$

and accordingly

$$P_{\text{intra-cluster}} = \left( \frac{N}{N_c} - 1 \right) \frac{R_0^2}{2}. \quad (30)$$

Equation (30) shows that $P_{\text{intra-cluster}}$ is a decreasing function of $N_c$. 
2) Analysis of $P_{\text{to BS}}$: $P_{\text{to BS}}$ is calculated based on the inter-cluster multi-hop routing as follows

$$P_{\text{to BS}} = \sum_{i=1}^{N_c} \text{NoH}_i \times R^2 \times M_i,$$

(31)

where $M_i$ is the number of measurements required taken from the $i$th cluster, and $R^2$ is the power consumption based on the CH’s radius spent on each hop with path-loss exponent $\alpha = 2$. For analysis we assume equal size clusters (equal number of sensor nodes). According to Lemma 1, the number of measurements required taken from each cluster should be linearly proportional to the number of sensors in each cluster or $M_i = \frac{M}{N_c}$. So, (31) can be written as

$$P_{\text{to BS}} = R^2 \times \frac{M}{N_c} \sum_{i=1}^{N_c} \text{NoH}_i,$$

(32)

where, $M$ is the total number of measurement required from the network to satisfy an error-target. In [43], Chandler calculated the average number of relay hops in randomly located radio networks. Based on this, (32) is given by

$$P_{\text{to BS}} = \text{NoH}_{\text{avg}} \times R^2 \times M,$$

(33)

where $\text{NoH}_{\text{avg}}$ is the average number of hops equal to $E[\text{NoH}]$ as mentioned in [43]. This expectation of the number of hops is calculated as

$$E[\text{NoH}] = \max(\text{NoH}) - \sum_{\text{NoH}=1}^{\max(\text{NoH})-1} \frac{P_{\text{NoH}}(x)}{P_{\max(\text{NoH})}(x)},$$

(34)

where $\max(\text{NoH})$ is the maximum number of hops allowed and $P_{\text{NoH}}(x)$ is the probability of being able to send data from a CH to BS at distance $x$ using $\text{NoH}$ or less hops. Interested readers are referred to [43] for the details on the calculation of $P_{\text{NoH}}(x)$.

3) Analysis of CH’s transmission range $R$: In each routing path, the number of hops is directly related to the broadcast radius $R$. If we increase $R$, a CH could reach more distant CHs and choose one of them to forward measurements. This means that the total number of hops can be reduced or increased with variable values of $R$ which may affect power consumption. For example, if we increase $R$ the number of hops in each routing path might decrease. But we have to deal with longer hop distance that consumes more power. In Figure 12, we have a 2000-node sensor network with 500 clusters formed by K-means and LEACH. We chose $R = \{10, 12, 14, 16, 18, 20\}$. Figure 12(a) shows the total number of hops reduced corresponding
to the increase in the radius. Figure 12(b) shows that the total consumed power increases as we increase $R$. Based on Figure 12 we should choose the smallest $R$ that results in the least consumed power for the network.

C. ICCS Simulation Results

In this simulation we form a network consisting of 2000 sensors randomly distributed in a circular area with radius $R_0 = 50$. The BS is set at the center of the sensing area. We use real sensory data collected from [42] and the sparsifying matrix $\psi$ as the DCT. In this case, as discussed in the previous sections, the total number of measurements required does not change as we increase the number of clusters. Hence, for any number of clusters, we chose $M = 500$ to satisfy the error-target of $0.1$. We only consider the maximum number of clusters up to $N_c = 500$ since $M = 500$. This means that each cluster should send at least one measurement to the BS for the data recovery process.

We apply K-means and LEACH clustering algorithms to form two different clustered networks. Figure 13(a) shows the total intra-cluster power consumption due to the total consumed power required to transmit data from all non-CH sensors to their CHs within all clusters. As shown in Figure 13(a), the intra-cluster consumed power becomes very small if the network is divided into many clusters. In this case, the total power consumption will be dominated by the power corresponding to the inter-cluster routing paths.

In Figure 13(b), the total inter-cluster power consumption is reduced as the network is divided
Fig. 13. Total intra-cluster, inter-cluster power consumption when BS at the center in a circular sensing area

into a larger number of clusters. For a large value of $N_c$, the density of CHs in the sensing area is large. Therefore, the CH’s transmission range needed to maintain inter-cluster connection becomes smaller. These numbers of clusters of $N_c = 10, 100, 200, 300, 400, 500$ correspond to values of $R = 50, 30, 25, 22, 18, 14, 11$, respectively, which explains the reduced inter-cluster consumed power.

As shown in Figure 13, the total power consumption is reduced by both intra-cluster and inter-cluster transmissions as we increase $N_c$. As compared with DCCS in Figure 14, ICCS significantly reduces the power consumption when the network is arranged into a large number of clusters ($N_c \geq 100$). Note that our calculation for power consumption for DCCS was originally based on the assumption of a square sensing area. These results are extended to the case of a circular sensing area with BS at the center in our previous work [34].

VI. DCT COMPRESSION TRANSMITTING ONLY $k$ LARGE COEFFICIENTS

In this section, we consider transmitting only $k$ large DCT coefficients as proposed in RIDA [44]. All raw readings from non-CH sensors are sent to their respective CH and sorted at each CH in descending or ascending order. Either DCT or wavelet transform is used as the sparsifying matrix to achieve a $k$-sparse data vector. The mapping process given in [44] is used to match sensors to virtual indices. These sensors will multiply their readings with DCT coefficients and then only send $k$ significant large coefficients to the BS. The rest of the coefficients are considered as zeros and not sent to the BS. At the BS, all $k$ large coefficients are mapped to
zero-coefficients and recovered to return all the raw data.

In order to reduce transmission cost in RIDA, the idea of transmitting measurements in CCS [12], [34] is applied. First, all sensor readings from non-CH sensors are sent to their own CH. The data is sorted at the CHs. After being multiplied with a sparsifying DCT matrix, a large proportion of the signal energy is focused on the very \( k \) first large coefficients. Instead of sending \( M \) CS measurements from the CHs as with in CCS, only these \( k \) large coefficients are sent directly to the BS for the recovery process as mentioned in RIDA. All calculations in this section are based on the network model in the DCCS section, and \( k \) is much less than \( M \).

A. Network Model

We assume the WSN is deployed in a square sensing area sized \( L \times L \). This model is similar to the model defined for the analysis of the DCCS algorithm. The power consumption for each transmission is calculated as it was with DCCS, except for the number of coefficients \( k \) transmitted from CHs to the BS. We will not compare the total power consumption between DCCS and DCT compression because of \( k \ll M \). We formulate and also simulate the problem to show how this compression method works with clusters, noiseless or noisy signals.

B. Communication Power Consumption

We assume that all clusters have the same number of sensors. Hence, the number of large coefficients collected from all clusters should be equal. Hence, the total number of large coefficients is calculated as \( k = \sum_{i=1}^{N_c} k_i \), where \( k_i \) is number of coefficients collected from the \( i^{th} \)
cluster. Similar to Equation (18) in Section IV-B, the total power consumption for this method can be formulated in general as

\[ P_{\text{total}} = \left( \frac{N}{N_c} - 1 \right) \frac{L^2}{2\pi} + \frac{kL}{L} \left[ \frac{(L - L_i)^3 + L^3}{3} \right] + \frac{kL^2}{12}. \]  

(35)

When the BS at the center of the sensing area \((L_i = L/2)\), (35) is simplified as

\[ P_{\text{total}} = \left( \frac{N}{N_c} - 1 \right) \frac{L^2}{2\pi} + \frac{kL^2}{6}. \]  

(36)

C. Simulation Results

In this section, we consider both sorted and unsorted signals generated from 2000 sensors uniformly distributed in a square sensing area. These types of data provide different values of \(k\) that affects either the transmitting cost from the CHs to the BS or the reconstruction error at the BS.

Figure 15(a) shows unsorted sensor readings collected from a WSN [42] and their transformations in the DCT domain. All signal energy is preserved in the transformed vector but is now focused in a relatively small number of large coefficients. If we transmit only these \(k\) large valued coefficients to the BS, this results in much less consumed power than transmitting all the values as was done earlier with CCS.

Figure 15(b) shows sorted signals in decreasing order and the DCT coefficients. The large coefficients are concentrated in the lower numbered coefficients. The transmission cost can be reduced based on the smaller values of \(k\) compared to that in unsorted signals.

Both Figures 16(a)(b) show that increasing the number of clusters or reducing the total number of coefficients \(k\) transmitted to the BS will increase the reconstruction error. Transmitting more of the larger DCT coefficients to the BS can compensate for the error as we increase the number of clusters.

In a noiseless environment, using DCT compression consumes less energy than CCS since CCS transmits \(M\) measurements to the BS while DCT compression only sends \(k\) large transformed coefficients \((k \ll M)\). As shown in our simulation results, \(k\) is generally only about 20% as large as \(M\) to satisfy the error-target in signal recovery processes. In practical networks noise is problematic. CCS can work with noise contaminated measurements while DCT compression is quickly degraded. As shown in Figure 17(a), sensor readings can be recovered at the BS based
Fig. 15. Real unsorted and sorted data and their DCT transformed coefficients.

(a) Unsorted sensory readings from 2000 sensors and the DCT transformed coefficients.
(b) Descending sorted readings from 2000 sensors and the DCT transformed coefficients.

Fig. 16. Normalized reconstruction errors with different number of clusters and large coefficients (measurements).

(a) Reconstruction error versus number of measurements.
(b) Reconstruction error versus number of clusters.
on different numbers of noisy measurements. Increasing the CS measurements can recover the original signals with less error.

Figure 17(b) shows that with DCT compression in the presence of noise the reconstruction error increases as the total number of measurements is increased. It means that the method can not work with noise. So, in practical applications of WSNs DCCS and ICCS should be considered.

VII. CONCLUSION

In this paper we proposed an energy-efficient data collection method applied in WSNs that is based on an integration of clustering and block-wise CS, called CCS. It is well known that natural signals have spatial correlation and therefore the sensor readings in a WSN are sparse in a proper basis such as DCT or wavelet. This sparsity facilitates the utilization of CS for energy-efficient data collection in such networks. In contrast to previous work in this area, in this paper we introduced CCS in which all non-CH sensors send their readings only once to the CH they belong to. The CS measurements required are generated at the CH before being sent to the BS for the CS recovery processes in two possible ways: directly (one hop) or multi-hop, called DCCS and ICCS, respectively. The algorithm reduced significantly numerous data transmissions in the networks.

We formulated the total power consumption and discussed the effect of different sparsifying bases on CS performance as well as the optimal number of clusters for reaching the minimum
power consumption. We employed K-means, LEACH, and uniform clustering techniques in our simulations and found the optimal cluster size when the signal of interest is sparse in canonical, wavelet, and DCT bases. After choosing DCT as the best sparsifying basis for CCS, we showed choosing a larger number of clusters can achieve less power consumption utilizing DCT with real sensor readings as the intra-cluster power consumption is reduced. The optimum number of clusters was determined to be $N_c^* = M$. Furthermore, as we employ many clusters, ICCS outperforms DCCS based on multi-hop routing.

As a final case to compare with DCCS and ICCS, we considered transmitting only $k$ large coefficients in DCT transformed signals. This method cannot work in noisy environments as mentioned in simulation Section VI.

REFERENCES


