A Compressibility-Based Clustering Algorithm for Hierarchical Compressive Data Gathering

Kun-Chan Lan and Ming-Zhi Wei

***Abstract*— Data gathering in wireless sensor networks (WSNs) is one of the major sources for power consumption. Compression is often employed to reduce the number of packet transmissions required for data gathering. However, conventional data compression techniques can introduce heavy in-node computation, and thus, the use of compressive sensing (CS) for WSN data gathering has recently attracted growing attention. Among existing CS-based data gathering approaches, hierarchical compressive data gathering (HCDG) methods currently offer the most transmission-efficient architectures. When employing HCDG, clustering algorithms can affect the number of data transmissions. Most existing HCDG works use the random clustering (RC) method as a clustering algorithm, which can produce significant number of transmissions in some cases. In this paper, we present a compressibility-based clustering algorithm (CBCA) for HCDG. In CBCA, the network topology is first converted into a logical chain, similar to the idea proposed in PEGASIS [1], and then the spatial correlation of the cluster nodes' readings are employed for CS. We show that CBCA requires significantly less data transmission than the RC method with a little recovery accuracy loss. We also identify optimal parameters of CBCA via mathematical analysis and validate them by simulation. Finally, we used water level data collected from a real-world flood inundation monitoring system to drive our simulation experiments and showed the effectiveness of CBCA.**

***Index Terms*— Wireless sensor network, data gathering, compressive sensing, clustering algorithm.**

I. Introduction

MOST wireless sensor networks (WSNs) are battery powered. Hence, energy consumption constitutes a crucial issue in relation to WSNs. Wireless transmission is a major contributor to power consumption in every sensor node [2]. Data gathering serves as one of the main functions of sensor networks, and it introduces considerable wireless transmission overhead. For this reason, minimizing the amount of wireless transmissions in data gathering is a direct way to reduce energy consumption in a sensor network.

Raw Data Gathering (RDG) is the conventional methodology that is used for data gathering [3]–[7] in sensor networks. Each node transmits raw data to the sink over multi-hop without compression. As shown in Figure 1, x_i, $i = 1, 2, \ldots, n$, is the data sensed at each node. Node S_1 transmits x_1 to S_2 and S_2 transmits x_2 and relayed x_1 to S_3, etc. By the end of the route, S_n transmits all n data readings to the sink.

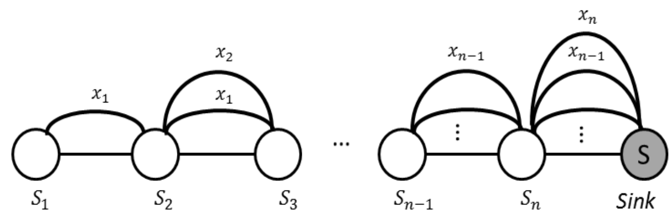


Fig. 1. Raw data gathering.

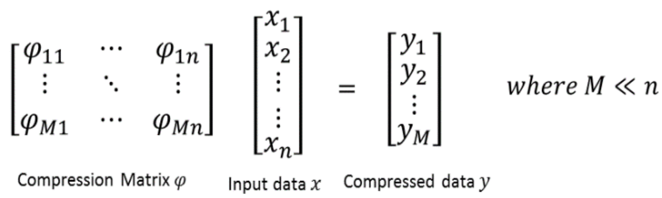


Fig. 2. The CS compression formula.

RDG typically involves $O(n^2)$ data transmission. As wireless transmission is the major contributor to power consumption in every sensor node, data compression is the obvious way to reduce data transmission. However, conventional compression techniques [8]–[10] generally require explicit data communication between sensors and often introduce significant in-node computation and control overhead [11].

The Compressive Sensing (CS) [12]–[14] has been recently proposed for data gathering in sensor networks [15]. The data compression of CS is "light-weight" and can be applied right after Analog-to-Digital Converter (ADC) at the sensor, while the decompression is usually computation-intensive. In the context of WSNs, the decompression of CS normally performed at the sink node which is often a full-fledged machine with good computing power. The basic ideas of CS are as follow. Assuming that $x = [x_1, x_2, \ldots, x_n]$ denote a set of sensor reading from n nodes, as shown in Figure 2, if raw data x is sparse [2], we can multiply a compression matrix φ by x to obtain y which is theoretically has less number of entries (i.e. M) than x's (i.e. n) but include most information of x (so that x can be recovered back from y at the sink node).

Compressive Data Gathering (CDG) [11] was the first method proposed for applying CS theory for data aggregation in a tree topology. The number of data transmissions using CDG methods are generally close to $O(Mn)$, where M is the size of compressed data and $M \ll n$. However, for nodes close to the leaf, the number of data transmissions in CDG are actually more than that in the traditional RDG (as discussed in Figure 5 later). To alleviate this problem,

Manuscript received December 1, 2016; revised January 20, 2017; accepted January 30, 2017. Date of publication February 14, 2017; date of current version March 22, 2017. The associate editor coordinating the review of this paper and approving it for publication was Dr. Roozbeh Jafari.

The authors are with the Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan (e-mail: klan@csie.ncku.edu.tw; xtcnrs456@gmail.com).

Digital Object Identifier 10.1109/JSEN.2017.2669081

the Hybrid CS Aggregation (HCS) [16]–[18] methods were proposed, in which nodes close to the leaf implement RDG and the remaining nodes implement CDG. As a result, HCS can reduce the number of transmissions to (hn). Here h is a function of the percentage of leaf nodes in the network and $h \le M$. Given that h can still be large for a large-scale network, Hierarchical CDG (HCDG) [19], [20] was then proposed by decomposing the network into several clusters in order to have a bounded h. In every cluster, HCDG uses a smaller compression matrix φ_i to replace the original φ. HCDG can further reduce the number of transmissions to $O(M_i n)$, where M_i is a function of cluster size and generally $M_i \le E[h]$.

The number of transmissions in HCDG is a function of the average compression ratio of the clusters; therefore, the clustering algorithm can significantly affect the performance of HCDG. Existing HCDG studies randomly choose the size of the cluster and each cluster has the same size; here, we refer to this approach as the *Random Clustering* (RC) method. RC does not consider the compressibility of each cluster, and thus the average compression ratio of each cluster could be unbounded. In this paper, we propose a Compressibility-Based Clustering Algorithm (CBCA). In CBCA, the network is first converted into a logical chain, similar to the idea proposed in PEGASIS [1], and then sensor nodes are grouped based on the compressibility of their readings on top of this chain. For this reason, the CBCA requires less number of transmissions than the RC method. In our experiments, the recovery quality levels at the sink node can maintain a Percentage Root-mean squared Distortion (PRD) of less than 5% with an average compression ratio around 40% when CBCA is used.

The remainder of this paper is structured as follows. In section II, we discuss related works. We describe our methodology in section III. The simulation experiment results are shown in section IV. Finally, we conclude this paper in section V. In Appendix, we show how we determine the optimal CBCA parameters via mathematical analysis.

II. Related Work

In this section, we first introduce the concept of sparsity and compressive sensing (CS). We then describe the basic ideas of CDG, HCS and HCDG.

A. Sparsity

Sparsity is a vital prerequisite for data compression, and it expresses "compressibility" of a signal. Numerous natural signals are compressible in the sense that they have concise representations when expressed properly [12]. We generally consider that "a vector $x \in \mathbb{R}^n$ is $K-sparse$" if $\exists \psi is\ an\ invetible\ matrix\ s.t\ \|\psi x\|_0 \le K, or$ we can say that "the sparsity of x is K.". That is, most of the information about$x$can be preserved by K components in ψx. Note that few real-world signals are *truly sparse*. Therefore, here we use an alternative definition of sparsity [21] which considers "a vector $x \in \mathbb{R}^n$ is $K-sparse$ if $|\psi x|$ has K elements greater than a sparse threshold ε, where $\varepsilon \cong 0$". *Compressive Sensing* (CS) is a non-adaptive compression method that

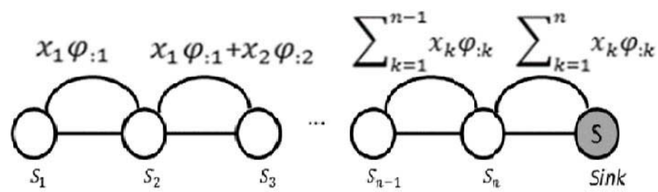


Fig. 3. Compressive data gathering.

assumes the sparsity of the raw data. More specifically, in CS, when a one-dimensional signal $x \in R^n$ is $K-sparse$ in ψ and K $\ll$ n, the following compression strategy [22]–[24] can be realized (intuitively, the sparse threshold ε will affect the compression ratio and recovery accuracy in CS).

$$\text{Encoder}: y = \varphi x, \varphi \in \mathbb{R}^{M*n}$$
$$\text{Decoder}: min\ \left\|\hat{z}\right\|_0\ \ s.t\ \varphi\psi^H\hat{z} = y$$
$$\hat{x} = \psi^H\hat{z}$$

B. Compressive Data Gathering

The Compressed Data Gathering (CDG) [11] was the first method proposed for applying CS theories to data aggregation in a sensor network, as shown in Figure 3. We let $x = [x_1, x_2, \ldots, x_n]$ denote sensor readings collected by nodes and assume that $\upsilon_i = x_i\varphi_{:i}$ ($\varphi_{:i}$ is the i^{th} column of the compression matrix φ in Figure 2). Here let's define the F_i as "fused-data of node S_i" which is the summation of υ_i and all S_i's children's F. If S_i is a leaf node, then $F_i = \upsilon_i$. For example, $F_3 = x_1\varphi_{:1} + x_2\varphi_{:2} + x_3\varphi_{:3}$ in Figure 3.

The CDG method involves four steps.

1. Every node S_i calculates its υ_i and F_i.
2. S_i transmits F_i to its upstream parent. Not that, as compared to traditional data gathering methods in which a node only sends raw data (i.e. x_i) , the transmission size at the leaf node increases from 1-tuple to M-tuple (here we assume that the size of each sensor reading take up 1-tuple), as shown in Figure 4.
3. The parent node receives all its children's $F$and calculate its own F and then forward to its upstream node.
4. Step 3 is repeated until the root node is reached and then $y = \sum_{k=1}^{n} x_k\varphi_{:k}\upsilon_k$ can be calculated, as shown in Figure 4.

The root node will then use y and φ to reconstruct $\hat{x}$. The transmission overhead of CDG is generally close to $O(Mn)$. One obvious drawback of this approach is that, for nodes close to the leaf, they are required to send more than what they do in the traditional RDG [3]–[7].

C. Hybrid CS Aggregation

Hybrid CS Aggregation (HCS) [11], [16], [17] was proposed as an improvement of CDG, as shown in Figure 5. The idea behind HCS is the "delayed-fusion" principle, in which the node only uses CDG if the size of aggregated data is larger than M. Otherwise, RDG is applied. The transmission overhead of HCS is close to $O\ (hn)$, where $h \le M$ and h is a function of M.

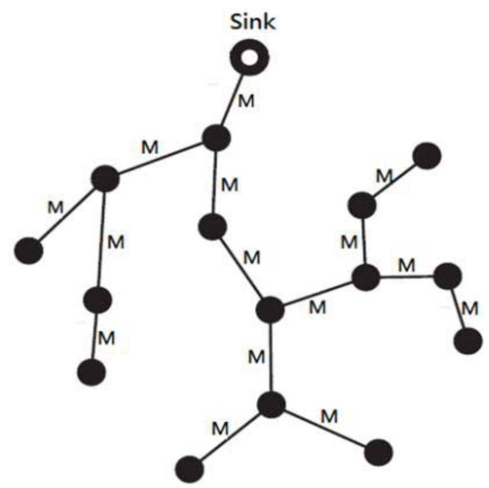


Fig. 4. Applying CDG for a tree topology.

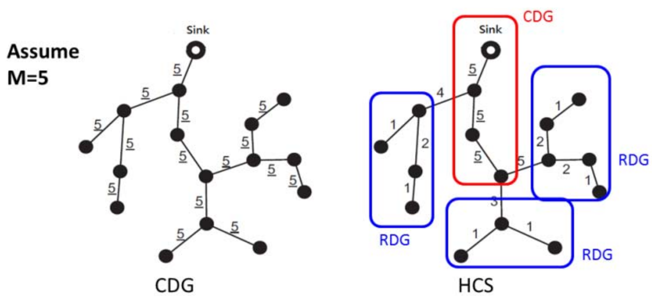


Fig. 5. Differences between CDG and HCS (assuming M = 5).

D. Hierarchical Compressive Data Gathering

Given that overhead of HCS can still be quite significant for a large-scale network [19], Hierarchical CDG (HCDG) [18], [19] was proposed. HCDG employs a hierarchical architecture (depicted in Figure 6) by dividing a network into several clusters. A smaller compression matrix $\varphi_i \in R^{M_i * n_i}$ is used in each cluster to replace the original φ (n_i is the i^{th} cluster size; M_i is a function of K_i which is the sparsity of the i^{th} cluster) used in HCS. Every cluster head collects data via intra-cluster CDG and relays it toward the sink. Based on HCDG, Ruitao Xie et al. proposed a clustering method [18] in which sensor nodes are randomly formed into clusters of a fixed size. Xi Xu et al. proposed a multiple level gathering strategy [19]. At each level, nodes of a fixed size are randomly grouped. These approaches, however, did not consider properties of collected data in a cluster. More specifically, if the collected sensor reading set is not compressible, the intra-cluster CDG might not be effective.

In this work, we propose a Compressibility-Based Clustering Algorithm (CBCA) for HCDG. In CBCA, the network topology is first converted into a logical chain, similar to the idea proposed in PEGASIS [1], and then a greedy clustering algorithm is implemented on top of this chain to minimize the average compression ratio (which is defined below) of all clusters.

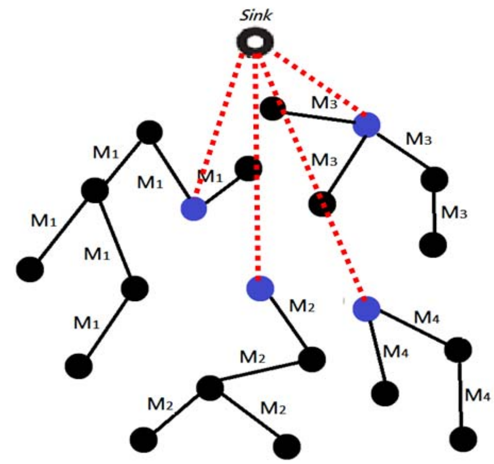


Fig. 6. Hierarchical Compressive Data Gathering (the dashed lines denote multi- hop transmissions from cluster heads to the sink).

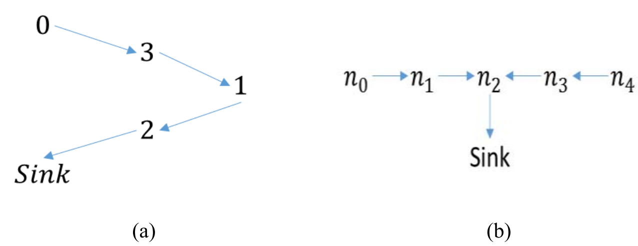


Fig. 7. (a) A chain topology (b) Leader node selection in PEGASIS.

PEGASIS is a RDG-based method which aims to increase the life time of a sensor network. The main idea in PEGASIS is for each node to receive from and transmit to close neighbors and take turns being the leader for transmission to the sink. This approach will distribute the energy load evenly among the sensor nodes in the network. In PEGASIS, the nodes will be first organized to form a chain, which can be accomplished using a greedy algorithm starting from some node. To construct the chain, PEGASIS starts with the furthest node from the sink in order to make sure that nodes farther from the sink have close neighbors, as in the greedy algorithm the neighbor distances will increase gradually since nodes already on the chain cannot be revisited, as the example shown in Figure 7(a). For gathering data in each round, each node receives data from one neighbor, fuses with its own data, and transmits to the other neighbor on the chain. Nodes take turns to become the leader for transmitting the aggregated data to the sink. In other words, the leader in each round of communication will be at a random position on the chain as shown in Figure 7(b). Note that CBCA is HCDG-based method which intuitively has less data transmission overhead compared to a RDG-based approach like PEGASIS, as discussed in the previous section.

III. Methodology

In this section, we propose the Compressibility-Based Clustering Algorithm (CBCA) for HCDG. The CBCA is a greedy clustering algorithm that aims to minimize the average compression ratio of all clusters. First, we determine whether a set of nodes is compressible based on its compression ratio and greedily select the sets that are ‘incompressible’ (defined below). We then try to maximize the number of compressible clusters based on the incompressible sets. After the clustering stage, we determine transmission modes for each cluster based on their compressibility levels. More specifically, if the compression ratio of a cluster is lower than a threshold, CDG is applied. Otherwise, RDG is applied. Since we first select incompressible set of sensor nodes greedily, the remaining part of the network is likely to contain mostly sparse data.

Definition 1 (Compression Ratio): The compression ratio (CR) is also known as the measurement rate [11], [16], [17]. The CR measures the efficiency of compression and is defined as (i.e. a smaller CR has a better compression ratio)

$$CR = \frac{compressed\ data\ size}{raw\ data\ size}$$

Definition 2 (Compressible Cluster): We define a *compressible cluster* as:

$$\begin{aligned}&suppose\ x(i)\ is\ the\ reading\ set\ of\ Cluster_i\\&if Cluster_i\ is\ compressible,\ then\ the\ CR\ of\\&x(i)\ is\ lower\ than\ a\ threshold\ called\ ICT\end{aligned}$$

To minimize the data transmission in each cluster, we select a threshold (referred to ICT) to determine the transmission mode of a cluster. The ICT is generally a function of the number of nodes and the number of hops from a cluster head to a sink. In the later section, we will discuss how to find a good ICT.

A. Architecture

The system architecture is depicted in Figure 8. The clustering algorithm is performed at the sink node. In the initialization phase, the sink first collect some raw data from all sensor nodes through RDG to run CBCA, and then broadcast the results back to every node. Based on received CBCA results, every cluster runs either CDG or RDG for sensor data gathering. Given that the condition of the environment might be changing over time, the sink will run CBCA periodically (say, every t minutes, depending on the dynamics of the environment to monitor, as shown in Figure 8). Table 1 summarizes the notations used in remaining sections of this paper. Table 1I summarizes the notations used in the Appendix.

In this work, we make the following assumptions. First, the network contains n sensor nodes and the information of φ *and* ψ are stored at the sink. Second, as shown in previous studies, the sensor data can be reconstructed with high probability when $M = 3K \sim 4K$ [11]. In this work, we assume the sensor data is K-sparse and let $M = 4K$ for a better recovery quality. Third, we assume that the network can be converted into *a logical chain* as the shown in the prior work [1], [25], and every node S_i in the network satisfies the following conditions: $\forall i = 1 \sim n, S_i\ and\ S_{i+1}\ are\ neighbors$, here S_{n+1} is the sink. In reality, many sensor network can satisfy this condition, such as chain and mesh networks. Forth, data transmission is one of the major sources of power consumption in a wireless sensor. Finally, due to the space limitation, we do not consider the issues of packet loss in our experiments and leave it as our future work.

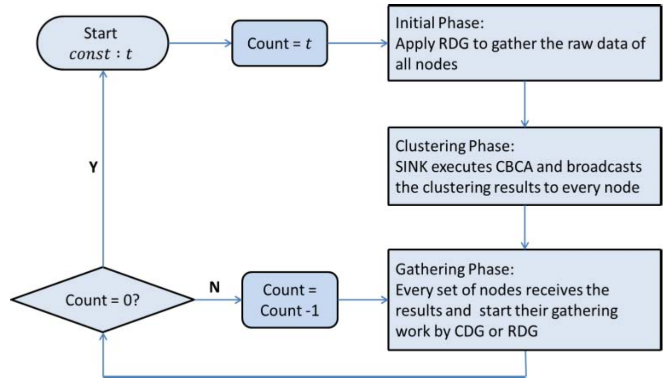


Fig. 8. System architecture.

TABLE I
NOTATIONS USED IN THE REMAINING SECTION

Notations	Meaning
φ	*Compression matrix*
ψ	*Sparse representation*
n	*total number of sensor nodes*
n_i	*number of nodes in the* i^{th} *cluster*
M	Size of *compressed data for the whole network*
M_i	*size of compressed data for* i^{th} *cluster*
K	*Sparsity of the whole network*
K_i	*Sparsity of the* i^{th} *cluster*
x	*The sensor reading at the node*
Z	selected window size
ICT	*Incompressible threshold*
ε	*Sparse threshold*
r	*Compression ratio of the whole network*
r_i	*Compression ratio of the* i^{th} *cluster*
$Cluster_i$	i^{th} *cluster*

TABLE II
NOTATIONS USED IN THE APPENDIX

Notations	Meaning
n	*the total number of sensor nodes*
n_i	*the number of nodes in* i^{th} *cluster*
D	*the number of clusters*
M_i	*Size of compressed data for the* i^{th} *cluster*
M_{avg}	*Average size of compressed data* $M_{avg} = \frac{M_1 + M_2 + \cdots + M_D}{D}$
r_i	*Compression ratio of* i^{th} *cluster*
r_{avg}	*Average compression ratio of all clusters* $r_{avg} = \frac{r_1 + r_2 + \cdots + r_D}{D}$
f	*Average number of hops from cluster heads to sink*

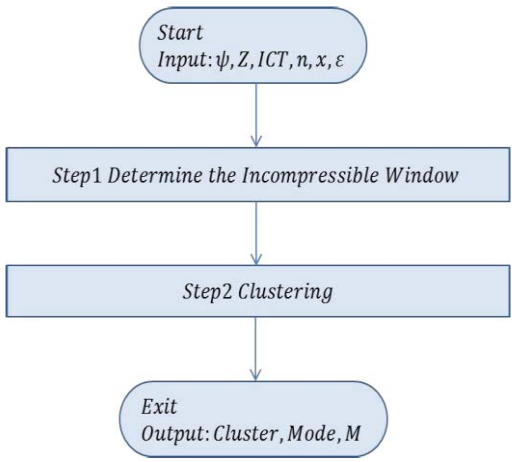


Fig. 9. CBCA flowchart.

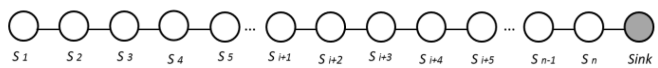


Fig. 10. Model the network as a chain.

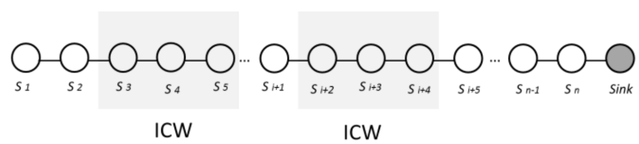


Fig. 11. Phase I: Identify the most incompressible set of nodes (ICWs).

B. Compressibility-Based Clustering Algorithm (CBCA)

Prior to running CBCA, we first model the network as a logical chain based on nodes' Euclidean distance [1], [25] and every node is given an unique ID accordingly, as shown in Figure 10. This pre-processing is used to introduce spatial data correlations since adjacent nodes in the chain tend to be nodes which are geographically close to each other [1], [25]. As shown in Figure 9, there are two phases in CBCA:

- Identify the most incompressible set of nodes (referred to "Incompressible Windows" as described shortly) based on a sliding window approach.
- Cluster formation based on the results from the phase I.

The details of phase I are as follows:

- Use a window (with a size of Z) to slide through the chain (starting from the first node of the chain, and advancing the window by one node at a time). This will create $(n - Z + 1)$ possible windows.
- Calculate the CR of each of these $(n - Z + 1)$ windows
- Mark those windows which have a CR $>$ ICT
- Among those marked windows, select a set of non-overlapping windows which have the highest average CR, and labeled these selected windows as Incompressible Window (ICW), as shown in Figure 11.

The process of phase II is similar to phase I. We again use a window (also with a size of Z) to slide through the ICW-marked chain from phase I, as shown in Figure 12(a). We refer this window as "Default Window (DW)" for the remaining of this paper. When sliding a DW over the ICW-marked chain, three cases can happen:

Case 1: DW is completely disjoint with any ICW.
Case 2: DW is completely overlapped with a ICW.
Case 3: DW is partially overlapped with a ICW.

For case 1 and case 2, nodes in the DW will form a cluster, as shown in Figure 12(a) and Figure 12(b). For case 3, let's first define DW1 = DW – (DW∩ICW) (e.g. nodes S_5, S_6, S_7 in Figure 12(c)) and MC as the most recently formed cluster (e.g. C_1 in Figure 12(c)). Two conditions can be found in case 3 based on the compression ratio of MC (i.e. CR_{MC}) and the compression ratio of DW1 (i.e. CR_{DW1}).

- When $CR_{MC} >$ ICT: if $CR_{DW1} <$ ICT, then nodes in DW1 will form a cluster by themselves, and nodes in the ICW that do not overlap with the DW also form a cluster, as the C_2 and C_3 shown in Figure 13(a). Otherwise, nodes in DW1 will merge with nodes in ICW to form a cluster, as the C_2 shown in Figure 13(c).
- When $CR_{MC} <$ ICT or when MC is an empty set (i.e. no cluster has been formed yet): if $CR_{DW1} <$ ICT, then nodes in DW1 will merge with nodes in MC and form a cluster, and nodes in the ICW that do not overlap with the DW form a cluster, as the C_1 and C_2 shown in Figure 13(b). Otherwise, nodes in DW1 will merge with nodes in the overlapped ICW to form a cluster, as the C_2 shown in Figure 13(c).

A new DW will start right after the most recently formed cluster on the chain and repeat the above process until the end of the chain.

Finally, we determine the transmission mode of each cluster by computing its CR. If the CR of a cluster is lower than the ICT, CDG is used in the cluster for the data gathering task. Otherwise, RDG is employed.

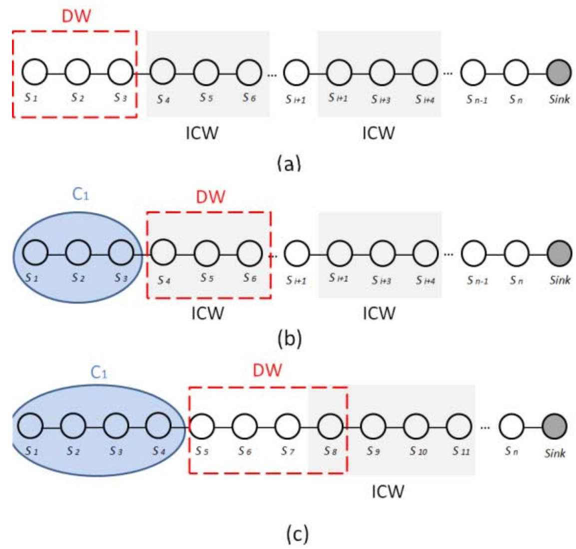


Fig. 12. The relation between DW and ICW.

IV. Experiments

In this section, we evaluate the performance of CBCA through trace-driven simulations based on data obtained from a real-world inundation monitoring sensor network.

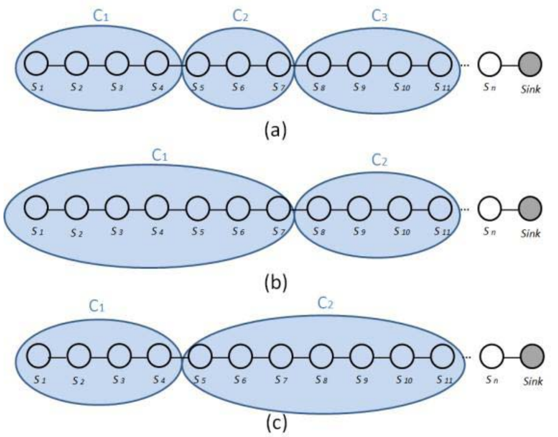


Fig. 13. Cluster formation for case 3.

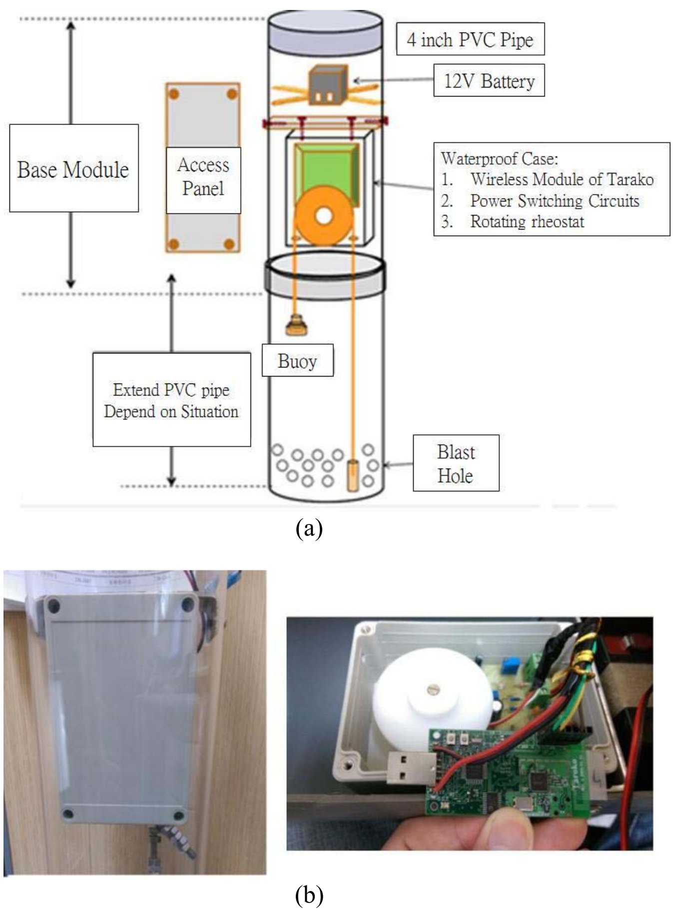


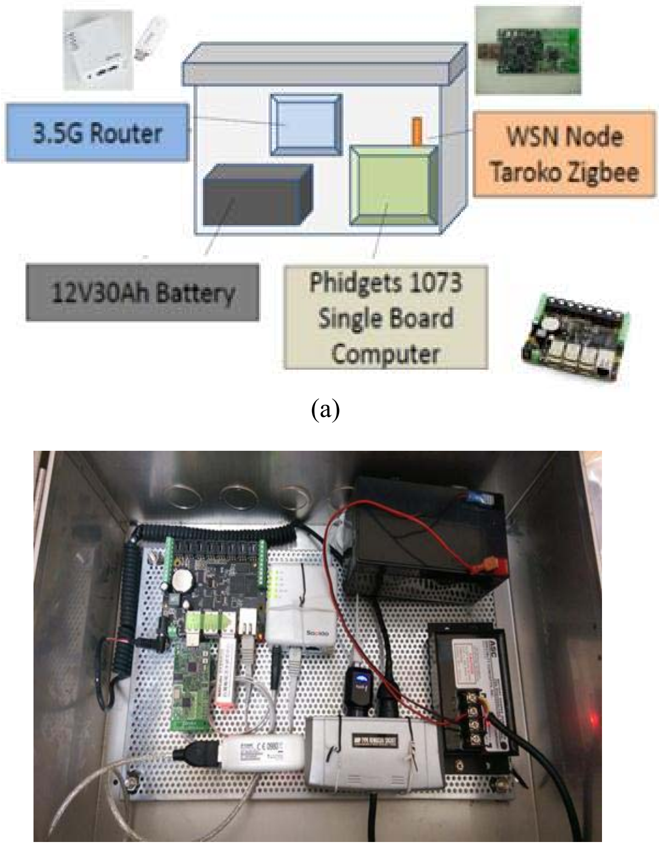
Fig. 14. (a) Structure and (b) physical appearance of the sensor node.

A. A Sensor Network Testbed for Inundation Monitoring

We implemented CBCA on a real-world sensor network containing seventeen nodes (deployed in a linear topology) used to collect water-level information for inundation monitoring. The water-level sensor is composed of a rotating rheostat and buoy, as shown in Figure 14. When flooding occurs, the buoy rotates the rheostat, and the rheostat transmits different levels of voltage to the sensor platform. We use the Taroko sensor platform, a modification of the TelosB mote [26] that was originally designed at UC Berkley. Taroko is a programmable, low-power wireless sensor platform. The Taroko platform uses a TI MSP430-F1611 microcontroller unit with 16-bit RISC [27]. The MSP430 has 48K bytes of flash memory and 10K bytes of RAM that support serial communications (e.g., UART, I2C, SPI, and Digital I/O). The Taroko platform is also equipped with a CC2420 RF transceiver [28], which is a low cost device for wireless communications in 2.4GHz based on IEEE 802.15.4 [29]. The maximum radio distance spans roughly 100 m. It also supports the USB interface and uses an FTDI chip [30]. The Taroko platform can also use the USB interface to connect to a computer for recharging, program uploading and data collection. The sensor network testbed is located in a suburb area of Kaohsiung city in Taiwan, as shown in Figure 15. The nodes are distributed along the road covering a length around 600 meters at a distance of 30-50m apart. Many of the nodes are in a clear line-of-sight of their neighbors. The sink/gateway node (circled in red) is located in a residential house. The roads where the testbed is deployed are about 5- 7m wide and surrounded mostly by farms.

Fig. 15. Testbed topology.

The sink node is shown in Figure 16. We use a single board computer as the gateway to upload the sensor data to the 3G network. We implement the gateway using the PhidgetSBC3 platform [31], which is a Single Board Computer with an integrated PhidgetInterfaceKit 8/8/8 [32]. In its most basic form, it can be viewed of as a Phidget that can be connected using a network cable rather than a USB. The PhidgetSBC3 also includes six high-speed ports that allow one to use a normal USB Phidgetsover as a network connection. This can extend the effective range of a Phidget from a USB's maximum range of 15 feet to any network range. The PhidgetSBC3 serves as a simple interface for setting up and running custom applications on-board. This allows the PhidgetSBC3 to operate autonomously without the ongoing use of a graphical interface or remote connection. For more advanced users, the PhidgetSBC serves as an embedded computer that runs Debian GNU/Linux. Phidget offers full shell access via a built-in SSH server with access to the full Debian package repository and with all standard command line tools expected of a modern Linux system. An integrated PhidgetInterfaceKit 8/8/8 [32] allows one to connect devices to any of the eight analog inputs, eight digital inputs and eight digital outputs. It serves as a generic, convenient tool for interfacing a PC and



(a)

(b)

Fig. 16. (a) Structure and (b) physical appearance of the sink node.

PhidgetSBC with a wide variety of devices, and it operates in exactly the same way as an external PhidgetInterfaceKit.

Through *mathematical analysis* and *extensive trace-driven simulations* (the simulation experiments are described in the next section and the analytical proofs are shown in the Appendix), we found that CBCA performs best when the selected window size Z $= \left\lfloor \frac{4}{r_{avg}} \right\rfloor$ (here r_{avg} is the average compression ratio of all clusters) and ICT is set to $\frac{Z^2+Zn+Z}{2Z^2+Zn}$. Based on the insight from our analytics analysis, Z and ICT in CBCA were set to 10 and 0.76 respectively on our 17-node testbed. OMP [34] was chosen as the data recovery algorithm and a Gaussian matrix [12] was used for the compression matrix φ. We set the sparse threshold to 2cm after consulting with some hydraulic experts.

We use PRD to evaluate the performance of CBCA on the testbed, and it is defined as:

$$PRD = \frac{\|x - \hat{x}\|_2}{\|x\|_2}$$

where x and $\hat{x}$ are the original and reconstructed signals, respectively, and $\|.\|$ denotes the Euclidean norm. We observed that CBCA can generally achieve a good balance between the compression ratio and the recovery accuracy. Specifically, when parameters of CBCA are optimized as above, we can obtain a compression ratio of 40% with a recovery accuracy of more than 95% (i.e. a PRD of less than 5%) on our 17-node testbed.

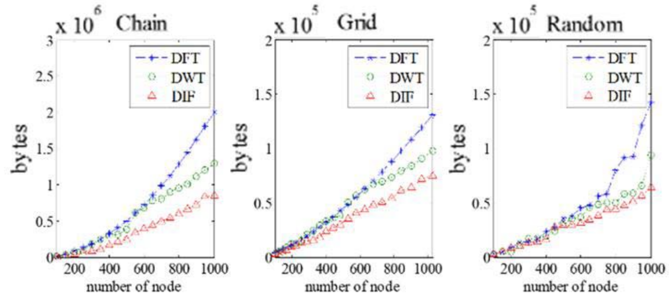


Fig. 17. Effect of different ψ.

B. Simulation Experiments

The performance of CBCA can be affected by many parameters, such as the topology, choices of different ψ, Z, *ICT*, φ, and recovery algorithms. In this section, we evaluate the effects of these parameters on the performance of CBCA. Given that our system only contains seventeen nodes while, in reality, a flooded area can cover hundreds of square kilometers. We thus evaluate the performance of CBCA through trace-driven simulations (trying to maintain the property of that closer nodes have more similar sensor readings) using sensor data obtained from the network testbed. The simulations are implemented in MATLAB and we use *the amount of data transmission* (in bytes) as the performance metric. We assumed that four bytes are used to store each sensor reading for the water-level. For simplicity and space limitation of this paper, we do not consider packet loss in our simulations and leave that as our future work.

We consider three different topologies in our simulations: chain, grid (a node has at most 4 neighbors) and random networks (nodes are uniformly distributed). The number of nodes in all our simulations is 1024.

1) Comparisons of Different ψ for Networks of Different Sizes: In HCDG, the data gathering overhead highly depends on the selection of sparse representation. Generally speaking, if the sparisty of a cluster i can satisfy $\sum K_i \approx K$, HCDG is more likely to exhibit good performance (in terms of the amount of data transmission). In this work, we compare three most commonly-used sparse representations in the literatures, including Difference Transform(*DIF)*, Discrete Fourier Transform(*DFT)* and Discrete Wavelet Transform (*DWT)* [14], [20], [33], [34]. In this simulation, the selected window size Z is set to $\left\lfloor \frac{4}{r_{avg}} \right\rfloor$ and ICT is set to $\frac{Z^2+Zn+Z}{2Z^2+Zn}$ to optimize CBCA, and a Gaussian matrix [12] is used for the compression matrix φ. As shown in Figure 17, for different network sizes, transmission overhead when using *DIF* [33] as the sparse representation is lower than that of the other two approaches.

The average compression ratio r_{avg} obtained in our simulation experiments is similar to that obtained from our 17-node testbed. This is not surprising given that the simulations are driven based on the sensor data collected from the testbed.

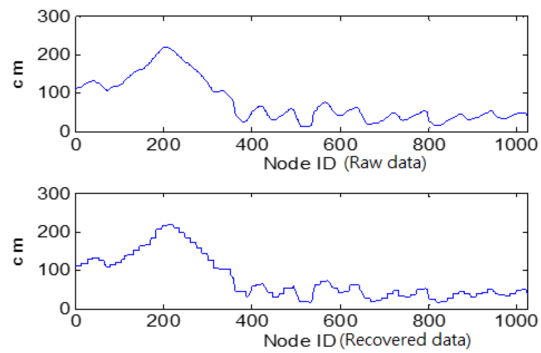


Fig. 18. Raw and recovered data. The upper figure denotes raw data, and the bottom figure denotes recovered data. Y-axis is the water level of flood inundation.

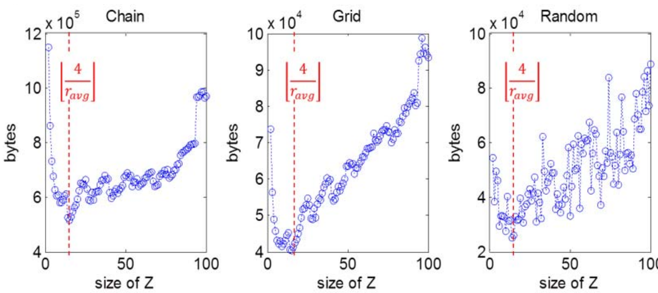


Fig. 19. Effect of different Z values for CBCA.

Figure 18 shows a snapshot of original data and reconstructed data.

2) Transmission Overhead for Different Z Values in CBCA: The selected window size Z is also an important parameter in the CBCA algorithm. Later in the Appendix, we will show through a mathematical analysis that a HCDG network will have the lowest data gathering overhead when Z is equal to $\left\lfloor \frac{4}{r_{avg}} \right\rfloor$. We verify this result in our simulation, as shown in Figure 18. We set ICT to 0.3. *DIF* is used for the sparse representation ψ and a Gaussian matrix [12] is used for the compression matrix φ.

As suggested in Figure 19, the choice of Z could have different effects for different topologies. For chain and grid topologies, the amount of *data transmission* generally increases as Z increases when Z is greater than $\left\lfloor \frac{4}{r_{avg}} \right\rfloor$. Such a correlation is less obvious for random topology. In addition, Figure 19 shows that a chain network generally requires more data transmission. This is expected since less aggregation can be performed with such a network topology.

3) Transmission Overhead for Different ICT Values in CBCA: The incompressible threshold (ICT) is another important parameter that affects the performance of CBCA. We also show it mathematically in the Appendix that a HCDG network will have the minimal data transmission for data gathering when the ICT value is equal to $\frac{Z^2+Zn+Z}{2Z^2+Zn}$, as shown in Figure 20. Here Z is the selected window size in the CBCA algorithm. In this experiment, Z is set to $\left\lfloor \frac{4}{r_{avg}} \right\rfloor$. Again, *DIF* is used for the sparse representation and a Gaussian matrix [12] is used for the compression matrix φ.

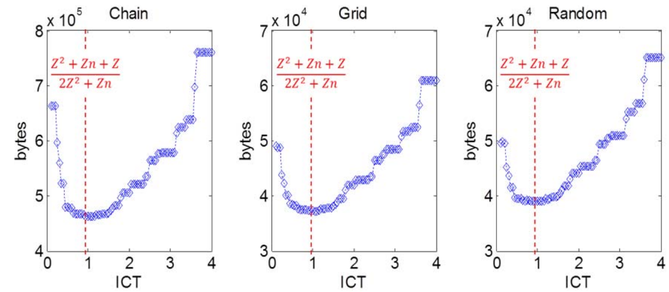


Fig. 20. Data gathering overhead with CBCA for different ICT.

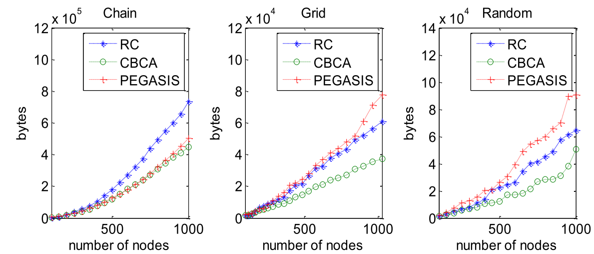


Fig. 21. Performance of CBCA and RC for networks of different sizes.

In addition, as observed from Figure 19, the amount of data transmission for all three topologies show similar increasing trends when the ICT value is greater than $\frac{Z^2+Zn+Z}{2Z^2+Zn}$.

4) Compare CBCA and Random Clustering (RC) Methods for Different Network Sizes: Next, we compare the performance of CBCA against the RC method used by all the prior work. In this simulation, *DIF* and a Gaussian matrix [12] are used for the sparse representation ψ and the compression matrix φ respectively. Z is set to $\left\lfloor \frac{4}{r_{avg}} \right\rfloor$ while ICT is set to $\frac{Z^2+Zn+Z}{2Z^2+Zn}$. As shown in Figure 21, the data gathering overhead with CBCA is significantly lower than that of RC for all three different topologies. In addition, we compare PEGSIS against both CBCA and RC methods. The performance of PEGSIS is close to CBCA in the chain topology when network sizes are small; but worse than both CBCA and RC methods in grid and random topologies, which is not surprising though since PEGSIS is a RDG-based method which inherently has a higher data transmission overhead (i.e. $O(n^2)$) than a CDG-based method like CBCA or RC. On the other hand, since the level of aggregation is low in a chain topology (i.e. each node only has one child (or downstream) node, RDG and CDG could have similar performance in such a case.

For the same reason stated above, the chain topology always has the worse performance for all three protocols while grid and random topologies exhibit similar results, as shown in Figure 22.

V. Conclusion and Discussion

In this work, we propose a novel clustering algorithm for hierarchical compressive data gathering (HCDG) called CBCA and we identify the best parameters for CBCA via mathematical analysis and simulation experiments. We show that CBCA enables less data transmission than the Random Clustering method previously used for HCDG. In our experiments, CBCA

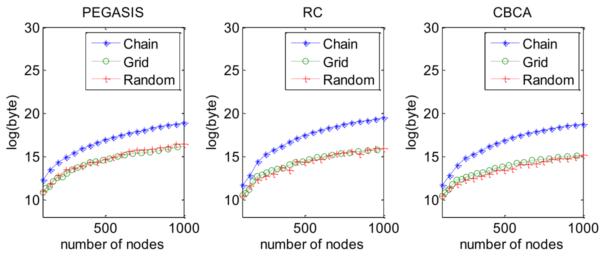


Fig. 22. Performance of different topology.

achieves a PRD of less than 5% (with a compression ratio of 40%) when the best parameter setting of CBCA is used.

In our current study, we assume that the physical network can always be modeled as a logical chain [1], [25] (i.e. $\forall i = 1 \sim n, S_i\ and\ S_{i+1}\ are\ neighbors$). While this is applicable to many sensor networks such as chain or mesh networks, it might fail for some particular types of networks. One possible direction to improve this limitation is to adopt a similar multi-level data gathering approach as in [19] by first dividing the network into multiple groups which are CBCA-compatible (meaning each of these subgroups can be modeled as a logical chain), and then the data can be aggregated to the sink through the roots of these subgroups. In addition, we do not consider packet loss and other networking issues (such as routing and packet re-ordering) which will be addressed in our future work.

Appendix

In this appendix, we mathematically derive the optimal values for two important parameters used in CBCA, including "Window Size" (Z) and the "Incompressible Threshold" (ICT), and analytically compare the performance of CBCA against that of Random Clustering (RC) method. The concerned performance metric here is the number of total data transmission from sensors to the sink.

A. Assumptions

In HCDG, data transmissions occur during intra-cluster data gathering and forwarding from cluster heads to the sink. In this analysis, we consider a *chain topology* which is generally the worst case scenario in HCDG because it requires more forwarding transmissions.

In order for HCDG to perform well, generally speaking, the summation of sparsity for each cluster ($\sum K_i$) should be greater than or equal to K. In addition, a small $\sum K_i$ value is desirable. In other words, a good representation basis (ψ) for HCDG should satisfy $\sum K_i - K \approx 0$. In the following analysis, we adopt $Different\ Matrix\ (DIF)$ [33] as our basis of representation the following analysis, as it can meet the above-mentioned condition (i.e. $\sum K_i - K \approx 0$).

$$\mathrm{DIF} = \begin{bmatrix} -1 & 1 & 0 & & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & & -1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -\gamma \end{bmatrix}$$

B. "Window Size (Z)" Selection

For simplicity, we assume that the size of each cluster is equal to Z and therefore $Z = \frac{n}{D}$. When n is known, we can determine once D is known. Thus, our goal is to find a D value that can minimize the number of data transmission Tn_{HCDG}. Here, we first model the number of data transmission using RDG and CDG respectively, as a function of number of nodes (n) and size of compressed data (M). Therefore, the number of transmission in RDG: $Tn_{RDG}(n) = \frac{n^2+n}{2}$ and the number of transmission in CDG: $Tn_{CDG}(M, n) = Mn$. We next model the number of data transmission in HCDG (Tn_{HCDG}) as a function of D. Specifically, the number of data transmission in HCDG

$$\begin{aligned}(Tn_{HCDG}) &= \sum_{for\ all\ cluster} [(intra-cluster\ transmission) \\ &\quad + (forwarding\ transmission\ from\ cluster\ head \\ &\qquad to\ sink)] \\ &= \sum_{i=1}^{D} \left[M_i n_i + M_i \left(n - \sum_{j=0}^{i} n_j \right) \right], n_0 = 0 \\ &= \sum_{i=1}^{D} \left(n - \sum_{j=0}^{i-1} n_j \right) * M_i, \quad n_0 = 0 \\ &\quad For\ simplicity,\ assuming\ n_1 = n_2 = \ldots = n_D = \frac{n}{D}\end{aligned}$$

the number of data transmission in HCDG

$$\begin{aligned}(Tn_{HCDG}) &= \sum_{i=1}^{D} \left[n - (i-1)\frac{n}{D} \right] * M_{avg} \\ &= \sum_{i=1}^{D} \left[\frac{Dn - (i-1)\,n}{D} \right] * M_{avg} \\ &= n \left(\frac{M_{avg}}{D} \right) \sum_{i=1}^{D} [D + 1 - i] \\ &= n \left(\frac{M_{avg}}{D} \right) \left(\frac{D\,(D+1)}{2} \right) \\ &= \left(\frac{D+1}{2} \right) * n * M_{avg} \\ \because r_{avg} &= \frac{M_{avg}}{n/D} \\ Tn_{HCDG} &= \left(\frac{D+1}{2} \right) * n * \left(\frac{n}{D} * r_{avg} \right) \\ &= \left(1 + \frac{1}{D} \right) \left(\frac{n^2}{2} * r_{avg} \right) \qquad (1)\end{aligned}$$

Furthermore, $for\ each\ Cluster_i$,

$$\begin{aligned}M_i &= 4K_i\ if\ K_i \geq 1 \\ &\Rightarrow M_i \geq 4 \\ &\Rightarrow M_{avg} \geq 4 \\ &\Rightarrow \left(\frac{n}{D} * r_{avg} \right) \geq 4 \\ &\Rightarrow 1 \leq D \leq \frac{n}{4} r_{avg} \qquad (2)\end{aligned}$$

Based on (1) and (2), we know that Tn_{HCDG} can be minimized when $D = \frac{n}{4} r_{avg}$. Therefore, we select $Z = \frac{n}{D} = \frac{n}{4r_{avg}}$ as the "Windows Size."

C. "Incompressible Threshold (ICT)" Selection

To minimize the transmission of the cluster, we determine the transmission mode of the cluster based on its compression ratio (CR) and *ICT*. Intuitively, the selection of *ICT* should enable that the number of transmissions are the same using either RDG $\left(denote\ Tn_{R_{Mode}}\right)$ or CDG $\left(denote\ Tn_{C_{Mode}}\right)$. In other words, our goal is to $find\ a\ compression\ ratio\ such\ that\ Tn_{C_{Mode}} = Tn_{R_{Mode}}$. Specifically,

$$\begin{aligned}
Tn_{C_{Mode}} &= (intra-cluster\ transmission) \\
&\quad + (forwarding\ transmission) \\
&= Tn_{CDG}(M_i, Z) + (M_i * f) \\
&= (M_i * Z) + (M_i * f) \\
&= M_i(Z + f) \\
&= r_i(Z^2 + Zf) \\
Tn_{R_{Mode}} &= (intra-cluster\ transmission) \\
&\quad + (forwarding\ transmission) \\
&= Tn_{RDG}(Z) + (Z * f) \\
&= (1 + 2 + \cdots + Z) + (Z * f) \\
&= \frac{Z^2 + Z}{2} + Zf
\end{aligned}$$

Here, the average number of hop is obtained by the equation as shown at the top of next page.

As $f = \frac{n}{2}$, in order to have

$$\begin{aligned}
Tn_{C_{Mode}} &= Tn_{R_{Mode}} \\
&\Rightarrow r_i\left(Z^2 + \frac{Zn}{2}\right) = \frac{Z^2 + Zn + Z}{2} \\
&\Rightarrow r_i = \frac{Z^2 + Zn + Z}{2Z^2 + Zn} = ICT
\end{aligned}$$

D. Compare CBCA Against the RC Method

1) Assumption:

- For simplification, we assume that the size of each cluster is Z.
- As noted above, when a good representation basis (ψ) *is selected,* we can have $\sum K_i \approx K$, and in this analysis we assume $\sum K_i = K$.
- Finally, we assume the number of clusters $D \geq 2$

2) Analysis: In probability theory, the binomial distribution with parameters$n$ and p is a discrete probability distribution that models the number of successes in a sequence of n independent yes/no experiments, each of which yields success rates with a probability of p. In this analysis, we assume that the size of cluster represents the number of experiments and that the number of non-redundant readings in the cluster denotes the number of successes. A non-redundant reading denotes a zero element in a sparse representation, and the number of non-redundant reading denotes the sparsity level K.

$$\begin{aligned}
&\text{assumingthedistributionofnon} \\
&\quad -\text{redundantreadingsisuniform} \\
&\quad \because \sum K_i = K \\
&\quad \therefore \textit{the probability for a node to have} \\
&\quad \textit{non-redundant reading} = \frac{K}{n} \\
&\quad \Longrightarrow The\ sparsity\ of\ an\ arbitrary\ cluster \\
&\quad K_c \sim B\left(Z, \frac{K}{n}\right) \\
&\quad \forall i\ P_r\ is\ the\ probability\ of\ Mode_i = RDG \\
P_r &= 1 - F_{K_c}\left(\frac{ICT}{4} Z\right), \text{Here } F \text{ is cumulative} \\
&\quad \text{distribution function} \\
&\quad Let\ C : K_c \geq \left\lfloor \frac{ICT}{4} Z \right\rfloor + 1 \\
&\quad to\ be\ The\ expectation\ of\ sparsity \\
&\quad for\ a\ cluster\ in\ RDG\ Mode \\
\mu_{K_c|C} &= \sum_{k=\left\lfloor \frac{ICT}{4} Z \right\rfloor + 1}^{Z} k * \wp_{K_c|C}(k), \\
\wp_{K_c|C}(k) &= \begin{cases} \dfrac{P(K_c = k)}{P_r}, & k \in C \\ 0, & k \notin C \end{cases}
\end{aligned}$$

With the above information, now we can calculate the number of transmissions using RC

$$\begin{aligned}
Tn_{RC} &= amount\ of\ cluster \\
&\quad * [(intra-cluster\ transmission) \\
&\quad + (forwarding\ transmisstion\ from\ cluster \\
&\quad head\ to\ sink)] \\
&= D * \left[Tn_{CDG}\left(4\mu_{K_c}, Z\right) + \left(4\mu_{K_c} * f\right)\right] \\
&= D * \left[\left(4\mu_{K_c} * Z\right) + \left(4\mu_{K_c} * f\right)\right] \\
&= D\left[4\left(\frac{K}{n} Z\right)(Z + f)\right]
\end{aligned}$$

The number of transmissions using CBCA

$$\begin{aligned}
&Tn_{CBCA} \\
&= amount\ of\ CDG\ Mode\ cluster \\
&\quad * [(in-cluster\ transmission) \\
&\quad + (forwarding\ transmisstion\ from\ cluster\ head \\
&\quad to\ sink)] + amount\ of\ RDG\ Mode\ cluster \\
&\quad * [(in-cluster\ transmission) + (forwarding \\
&\quad transmisstion\ from\ cluster\ head\ to\ sink)] \\
&= (D - R) * \left[Tn_{CDG}\left(4\frac{K - R * \mu_{K_c|C}}{n}, Z\right)\right. \\
&\quad \left. + \left(4\frac{K - R * \mu_{K_c|C}}{n} * f\right)\right] + R * [Tn_{RDG}(Z) + (Z * f)] \\
&= (D - R)\left(4\frac{K - R * \mu_{K_c|C}}{n} Z\right)(Z + f) \\
&\quad + R\left[\left(\frac{Z^2 + Z}{2}\right) + (Zf)\right] \\
&\quad \because \mu_{K_c|C} \geq \frac{Z}{4}
\end{aligned}$$

$$
\begin{aligned}
f &= \frac{\sum_{for\ all\ cluster} number\ of\ hops\ from\ cluster\ head\ to\ sink}{number\ of\ clusters} \\
&= \frac{0 + \frac{n}{D} + 2\frac{n}{D} + \ldots + (D-1)\frac{n}{D}}{D} \\
&= \frac{\frac{n}{D}(0 + 1 + \ldots + (D-1))}{D} \\
&= \frac{\frac{n}{D}\left(\frac{D(D-1)}{2}\right)}{D} \\
&= \frac{n(D-1)/2}{D} \\
&\approx \frac{n}{2}
\end{aligned}
$$

$$
\begin{aligned}
&Tn_{CBCA} \\
&= (D-R)\left(4 * \frac{K - R * \mu_{K_c|C}}{n} * Z\right)(Z+f) \\
&\quad + R\left[\left(\frac{Z^2+Z}{2}\right) + (Zf)\right] \\
&\le (D-R)\left(4 * \frac{K - R * \frac{Z}{4}}{n} * Z\right)(Z+f) \\
&\quad + R\left[\left(\frac{Z^2+Z}{2}\right) + (Zf)\right] \\
&= D\left[\left(\frac{K-RZ}{n}Z\right)(Z+f)\right] \\
&\quad - R\left[\left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right) - \left(\frac{Z^2+Z}{2} + Zf\right)\right] \\
&= Tn'_{CBCA}
\end{aligned}
$$

Here we want to prove that $Tn_{CBCA} \le Tn'_{CBCA} \le Tn_{RC}$ which will be true if $T'_{CBCA} - T_{RC} \le 0$

$$
\begin{aligned}
&\Rightarrow D\left[\left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right) - \left(4\left(\frac{K}{n}Z\right)(Z+f)\right)\right] \\
&\quad - R\left[\left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right) - \left(\frac{Z^2+Z}{2} + Zf\right)\right] \le 0
\end{aligned}
$$

Let

$$
\begin{aligned}
A &= D\left[\left(4\left(\frac{K}{n}Z\right)(Z+f)\right) - \left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right)\right] \\
B &= R\left[\left(\frac{Z^2+Z}{2} + Zf\right) - \left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&Tn'_{CBCA} \le Tn_{RC} \\
&\quad \Longrightarrow -A + B \le 0
\end{aligned}
$$

Next, let's try to prove that $B \le A$

$$
\begin{aligned}
&\because \sum K_i = K \quad and\ n_1 = n_2 = \ldots = n_D = \frac{n}{D} \\
&r_{avg} = \frac{r_1 + r_2 + \ldots + r_D}{D} = \frac{\frac{M_1}{n_1} + \frac{M_2}{n_2} + \ldots + \frac{M_D}{n_D}}{D} \\
&= \frac{\frac{4(K_1 + K_2 + \ldots + K_D)}{n/D}}{D} = \frac{4K}{n} = \mathrm{r} \\
&\Longrightarrow Z = \frac{4}{r_{avg}} = \frac{4}{r} = \frac{n}{K} = \frac{n}{D} \\
&\Longrightarrow D = K
\end{aligned}
$$

So now we can replace $Z\ with\ \frac{n}{D}$ and $D\ with\ K$ and simplify A and B as follows:

$$
\begin{aligned}
A &= D\left[\left(4\left(\frac{K}{n}Z\right)(Z+f)\right) - \left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right)\right] \\
&= D(Z+f)\left[\left(4\left(\frac{K}{n}Z\right)\right) - \left(\left(\frac{K-RZ}{n}Z\right)\right)\right] \\
&= D(Z+f)\left(4 - \frac{\frac{n}{Z} - RZ}{n}Z\right) \\
&= D(Z+f)\left(4 - \frac{n - RZ^2}{n}\right) \\
&= D(Z+f)\left(3 - \frac{RZ^2}{n}\right) \\
&= D(Z+f)\left(3 - \frac{R\left(\frac{n}{D}\right)^2}{n}\right) \\
&= \left(\frac{3D^2 - Rn}{D}\right)(Z+f) \\
B &= R\left[\left(\frac{Z^2+Z}{2} + Zf\right) - \left(\left(\frac{K-RZ}{n}Z\right)(Z+f)\right)\right] \\
&= R\left[\left(\frac{Z^2+Z}{2} + Zf\right) - \left(\left(\frac{\frac{n}{Z} - RZ}{n}Z\right)(Z+f)\right)\right] \\
&= R\left[Z\left(\frac{Z+1}{2} + f\right) - \left(\left(1 - \frac{R}{n}\left(\frac{n}{D}\right)^2\right)(Z+f)\right)\right] \\
&\le R\left[Z(Z+f) - \left(\left(1 - \frac{R}{n}\left(\frac{n}{D}\right)^2\right)(Z+f)\right)\right] \\
&= R(Z+f)\left[Z - \left(1 - \frac{Rn}{D^2}\right)\right] \\
&= R(Z+f)\left(Z + 1 + \frac{RZ}{D}\right) \\
&= B_1
\end{aligned}
$$

Here we want to show $B \leq B_1 \leq A$

$$\Longrightarrow R(Z+f)\left(Z+1+\frac{RZ}{D}\right) \leq \left(\frac{3D^2-Rn}{D}\right)(Z+f)$$
$$\Longrightarrow R\left(Z+1+\frac{RZ}{D}\right) \leq \left(\frac{3D^2-Rn}{D}\right)$$
$$\Longrightarrow RDZ - RD + RZ \leq 3D^2 - Rn$$
$$\Longrightarrow Rn - RD + RZ \leq 3D^2 - Rn$$
$$\Longrightarrow RZ - RD \leq 3D^2$$
$$\Longrightarrow R\left(\frac{n-D^2}{D}\right) \leq 3D^2$$
$$\because R \leq \frac{K}{Z/4} = \frac{4K}{Z} = \frac{4D}{Z}$$
$$\therefore R\left(\frac{n-D^2}{D}\right) \leq \frac{4D}{Z}\left(\frac{n-D^2}{D}\right) \leq 3D^2$$

So now if we can show $\frac{4D}{Z}\left(\frac{n-D^2}{D}\right) \leq 3D^2$ then we can prove $B \leq A$

$$\frac{4D}{Z}\left(\frac{n-D^2}{D}\right) = \frac{4}{n/D}\left(n-D^2\right)$$
$$= 4D - \frac{4D^2}{n}\left(\frac{n}{Z}\right)$$
$$= 4D\left(1-\frac{D}{Z}\right)$$
$$\text{case } 1: D \geq Z \Longrightarrow 4D\left(1-\frac{D}{Z}\right) \leq 0 < 3D^2$$
$$case\ 2:\ D < Z \Longrightarrow 4D\left(1-\frac{D}{Z}\right) = \alpha * 4D\alpha \in [0,1)$$
$$\Longrightarrow 4\alpha D < 4D \leq 3D^2 if D \geq 2$$
$$\Longrightarrow 4D\left(1-\frac{D}{Z}\right) \leq 3D^2$$

References

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

[2] 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.

[3] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, USA, 2009, pp. 1–14.

[4] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Orlando, FL, USA, Mar. 2002, pp. 350–355.

[5] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Dallas, TX, USA, Oct. 2001, pp. 304–309.

[6] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proc. 8th ACM Mobicom*, Atlanta, GA, USA, Sep. 2002, pp. 148–159.

[7] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," in *Proc. IEEE Int. Conf. Netw. (NETWORKS)*, Atlanta, GA, USA, Aug. 2002, pp. 1–9.

[8] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 41–54, Feb. 2006.

[9] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Nashvile, TN, USA, Sep. 2006, pp. 309–316.

[10] J. Acimovic, B. Beferull-Lozano, and R. Cristescu, "Adaptive distributed algorithms for power-efficient data gathering in sensor networks," in *Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput. (WiCom)*, ShenZhen, China, Jun. 2005, pp. 946–951.

[11] C. Luo, F. Wu, J. Sun, and C.-W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proc. 15th ACM MobiCom*, Beijing, China, 2009, pp. 145–156.

[12] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

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

[14] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[15] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[16] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *Proc. 8th IEEE SECON*, Salt Lake City, Utah, USA, Apr. 2011, pp. 46–54.

[17] J. Luo, L. Xiang, and C. Rosenberg, "Does compressed sensing improve the throughput of wireless sensor networks?" in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, Apr. 2010, pp. 1–6.

[18] R. Xie and X. Jia, "Transmission-efficient clustering method for wireless sensor networks using compressive sensing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 806–815, Mar. 2014.

[19] X. Xu, R. Ansari, and A. Khokhar, "Power-efficient hierarchical data aggregation using compressive sensing in WSNs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Sep. 2013, pp. 1769–1773.

[20] R. Chartrand, "Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, Boston, MA, USA, Sep. 2009, pp. 262–265.

[21] M. Davenport, "Random observations on random observations: Sparse signal acquisition and processing," Ph.D. dissertation, Dept. Elect. Comput. Eng., Rice Univ., Houston, TX, USA, Aug. 2010.

[22] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006.

[23] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[24] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[25] D. Xie, Q. Zhou, J. Liu, B. Li, and X. Yuan, "A chain-based data gathering protocol under compressive sensing framework for wireless sensor networks," in *Proc. 5th IEEE Int. Conf. Comput. Inf. Sci. (ICCIS)*, Apr. 2013, pp. 1479–1482.

[26] *TelosB*, accessed on Feb. 6, 2015. [Online]. Available: http://www.willow.co.uk/html/telosb_mote_platform.html

[27] *MSP430 F1611*, accessed on Feb. 6, 2015. [Online]. Available: http://focus.ti.com/docs/prod/folders/print/msp430f1611.html

[28] *CC2420*, accessed on Feb. 6, 2015. [Online]. Available: http://focus.ti.com/docs/prod/folders/print/cc2420.html

[29] *802.15.4*, accessed on Feb. 6, 2015. [Online]. Available: http://www.ieee802.org/15/pub/TG4.html

[30] *FTDI*, accessed on Feb. 6, 2015. [Online]. Available: http://www.ftdichip.com/Documents/DataSheets/ds232b18.pdf

[31] *PhidgetSBC3*, accessed on Feb. 6, 2015. [Online]. Available: http://www.phidgets.com/docs/1073_User_Guide

[32] *PhidgetInterfaceKit 8/8/8*, accessed on Feb. 6, 2015. [Online]. Available: http://www.phidgets.com/docs/1018_User_Guide

[33] X. Wu and M. Liu, "*In-situ* soil moisture sensing: measurement scheduling and estimation using compressive sensing," in *Proc. 11th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, New York, NY, USA, 2012, pp. 1–12.

[34] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of non-invasive fetal ECG via block sparse Bayesian learning," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 2, pp. 300–308, Feb. 2013.

[35] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing," *Inverse Problem*, vol. 24, no. 35020, pp. 1–14, 2008.

[36] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[37] J. Romberg, "Compressive sensing by random convolution," *SIAM J. Imag. Sci.*, vol. 2, no. 4, pp. 1098–1128, 2009.

[38] G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "Complex-valued sparse representation based on smoothed $\ell 0$-norm," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Las Vegas, NV, USA, Sep. 2008, pp. 3881–3884.

[39] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Sep. 2009.

[40] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[41] 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, Sep. 2007.

[42] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Annu. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Sep. 1993, pp. 40–44.

Kun-Chan Lan received the master's degree in computer science from the State University of New York, Stony Brook, NY, USA, in 1997, and the Ph.D. degree in computer science from the University of Southern California, Los Angeles, CA, USA, in 2004. From 1998 to 2004, he joined the ISI's Computer Networks Division, as a Graduate Research Assistant. During this period, he maintained and contributed codes to the popular network simulation software ns-2. From 2004 to 2007, he joined the Network and Pervasive Computing Program at National ICT Australia, Sydney, Australia, as a Researcher. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. His research interests include intelligent transport system, sensor network, and pervasive healthcare.

Ming-Zhi Wei was born in Tainan, Taiwan. He received the M.S. degree in computer science and information engineering from National Cheng Kung University. He has been with Mstar semiconductor since 2015.