Research Article

Data Gathering in Wireless Sensor Networks Based on Reshuffling Cluster Compressed Sensing

Lu Zhu, 1 Baishan Ci, 1 Yuanyuan Liu, 1 and Zhizhang (David) Chen 2

1 School of Information Engineering, East China Jiaotong University, Nanchang 330013, China
2 Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS, Canada B3J 2X4

Correspondence should be addressed to Lu Zhu; luyuanwanwan@163.com

Received 26 May 2015; Revised 2 November 2015; Accepted 5 November 2015

Copyright © 2015 Lu Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The existing compressed sensing (CS) based data gathering (CSDG) methods in wireless sensor networks (WSNs) usually assume that the sensed data are sparse or compressible. However, the sparsity of raw sensed data in some case is not straightforward. In this paper, we present reshuffling cluster compressed sensing based data gathering (RCCSDG) method to achieve both energy efficiency and reconstruction accuracy in WSNs. By incorporating CS into the cluster protocol, RCCSDG is able to reduce the energy consumption and support larger networks. Moreover, the sparsity of raw sensed data can be greatly improved by reshuffling pretreatment. A theoretical analysis to energy consumption of cluster head is performed, and the cost of the pretreatment is small enough to be neglected. Based on these natures, the raw sensed data can be recovered from fewer samples. Also, considering the sensed data to be of excellent temporal stability in a short time, we reshuffle them just one time in this stable period to further reduce the energy consumption of WSNs. In addition, the delay of RCCSDG is analyzed based on TDMA 2 scheduling scheme. We carry out simulations on real sensor datasets. The results show that the RCCSDG can effectively compress the data transmission and decrease energy consumption of WSNs while ensuring the reconstruction accuracy.

1. Introduction

With the development of wireless sensor networks (WSNs), a wide range of applications of WSNs are being used in many areas, such as climate monitoring, forest fire detection, and habitat and infrastructure monitoring [1]. Data gathering [2] is one of the most essential functions provided by WSNs, where the sensor nodes periodically collect the information of the monitoring area and transmit them to the sink. However, each sensor node of WSNs, being a microelectronic device, can only be equipped with a battery-powered source and cannot be recharged in most cases. The network lifetime is limited by the capacity of battery. A primary challenge of designing data gathering schemes lies in prolonging the network lifetime and not sacrificing the data accuracy.

Because the main energy consumption of sensor nodes is contributed to the data transmission, data compression can extend the network lifetime effectively. Data aggregation techniques [3] deal with large volumes of raw sensed data by some algorithms, and only a small amount of meaningful results is transmitted to the sink. Consequently, it can reduce the data transmission and prolong the lifetime of WSNs. Up to now, many data aggregation techniques have been heavily investigated to reduce the quantity of data to be transmitted. Madden et al. [4] adopt simple data aggregation methods, such as averaging, maximizing, or minimizing, to extract some statistics characteristics of sensed data and get rid of other unnecessary information. This method above is just suitable for the condition of low accuracy. In [5], Ciancio et al. proposed a data compression scheme based on distributed Wavelet transform. After transformation, only a few significant coefficients are needed to transmit to the sink and other insignificant coefficients are discarded. But it is not quite suitable to distributed processing because of its computational complexity. The data gathering on the basis of the distributed source coding scheme was put forward in [6, 7]. In this process, each node just needs to code separately and send the compressed information to the sink along the
shortest path. However, it requires nodes to know the global correlation structure which is difficult to be obtained in large-scale WSNs.

The emergence of compressed sensing (CS) theory [8–10] has opened up a new research approach for the in-network data aggregation [11–15]. According to CS theory, a sparse signal can be precisely recovered from far fewer samples than Nyquist criterion. This technique provides an effective way for reducing the data transmission by a simple compression at nodes in WSNs. At present, the applications of CS based data gathering (CSDG) in WSNs are mainly concentrated in planar route (linear structure) [16, 17]. And the basic idea of CSDG is illustrated in Figure 1. Using random measurement matrix, every node expands one sampling point into m-dimensional information and sends the product to its parent node. The parent node also expands a sampling point into m-dimensional vector, getting a new m-dimensional vector by adding the m-dimensional information from all its children nodes, and forwards the new vector to the next parent node. If the size of network nodes is n, the load of the entire network is n * m. CSDG requires that the sensor readings should be sparse or compressible enough while the real-world networks cannot always meet this requirement, resulting in significant errors when the sampling rate of sensed data is low [18, 19]. STCDG [18] makes use of both the low-rank matrix completion [19] and short term stability features to reduce the amount of traffic and improve the level of reconstruction accuracy, which is much more adaptable since it is independent of specific sensor networks. Although CSDG can solve the problem of network energy imbalance, its data volume of whole network is still high which will restrict the scale of networks.

Hierarchical route (cluster) is able to solve the problem of limited scale of planar routing network, so it is available to support larger networks. CS based cluster method [20] integrates adjacent nodes to form a cluster, and then the cluster head (CH) will compress all the sensed data within cluster by linear compressed projection. Because the nodes in cluster only send the reading itself without the requirement of linearly expanding data, the CS based cluster can further compress the data. Existing CS based cluster methods are built on the hypothesis that the raw sensed data are sparse or compressible in some domains such as DCT, infinite difference, or Wavelet. However, such assumption is not entirely tenable for real sensed data. In many practical cases, the sparsity of raw sensed data is not straightforward, which will make the CS based cluster less practical.

In this paper, we propose an efficient reshuffling cluster compressed sensing based data gathering (RCCSDG) scheme to solve the above challenges. Firstly, the LEACH (Low Energy Adaptive Clustering Hierarchy) [21] protocol is adopted to randomly select the cluster heads (CHs) among the whole network and balanced energy consumption for each sensor node. Secondly, we find that the sparsity of data can be greatly improved by the use of a simple pretreatment (reshuffling) on the raw sensed data. Here, the cost of pretreatment is small enough to be neglected. After receiving all the raw sensed data of the cluster, the CHs reshuffle them into ascending order and compress the preprocessing signals by linear compressed projection, and the compressed information will be transmitted to the sink, because the CHs have a certain computational capacity, and the cost of linear compressed projection and pretreatment is not high. So the RCCSDG method can improve the compression ratio dramatically just sacrificing little computational resource. Considering most sensor signals to be of excellent temporal stability in a short time [19], we reshuffle the sensor data only one time and keep the order in this stable period. By this operation, the proposed RCCSDG method can further reduce the energy consumption but ensure the reconstruction accuracy. The main contributions of this paper are listed as follows:

1. We present an efficient data gathering scheme by introducing the CS theory on the basis of the clustering structure, which can substantially reduce communication overhead and balance the nodes energy consumption. The algorithm based on reshuffling, especially, is capable of improving the sparsity of the data and further reducing the amount of data transmission.

2. Due to the fact that many sensor signals such as temperature and humidity will not change dramatically in a short time period, in this period, the data will be reshuffled just one time and keep the order. This method can reduce the computation burden. We also carried out a theoretical analysis in regard to energy consumption and delay of RCCSDG.

3. RCCSDG is verified by utilizing the real sensed data. The results of simulation show that RCCSDG can effectively reduce the energy consumption and achieve better reconstruction accuracy.

The rest of this paper is organized as follows. In Section 2, we present system model and motivation. Section 3 describes the details of the RCCSDG method we proposed and Section 4 presents the theoretical analysis on the energy consumption of CH and delay of the RCCSDG. The simulation results on both energy consumption and the accuracy of reconstruction are presented in Section 5. Finally, we conclude this paper and discuss future work in Section 6.

2. System Model and Motivations

In the RCCSDG model, we assume that N nodes have been randomly distributed in the sensing area and the proposed scheme implements a two-hop WSN (see Figure 2) that monitors a given physical scalar magnitude (e.g., temperature or humidity) [22]. In practical application, the topology of the WSNs can be abstracted as a weighted undirected graph \( G = (V, E) \), where \( V = \{V_i \mid i = 1, 2, \ldots, W\} \) is the set of nodes, \( W \) is the total number of nodes, and \( E = \{(V_i, V_j) \mid (V_i, V_j) \in V \ast V, i \neq j \} \) as the set of edges between nodes. Assume that the network has the following characteristics:

1. This is a static network with high density. When the deployment of wireless sensor network is finished, all the sensor nodes and the sink are assumed to be stationary, unless the nodes fail or die.
Sensor network is a homogeneous network. In addition to the base station, the other nodes are considered to have equal status and initial energy. Here, it is worthy to notice that the energy of nodes cannot be added in the process of data gathering.

All nodes have some storage space and certain capability of data fusion and can take turns to be cluster heads.

The system structure of the proposed RCCSDG method is depicted in Figure 2. In first phase, suppose that the W nodes of whole networks would form I clusters according to the clustering mechanisms, each cluster including one CH and \((N_i - 1)\) noncluster head (non-CH) nodes. It should be noted here that \(N_i\) is the number of nodes in cluster i. Let \(C_{CH_i}\) denote the cluster head of cluster i and \(d_i\) represent the sensor readings obtained by the i-th node in cluster i. In transmit phase, all non-CH nodes transmit their readings to that corresponding cluster head \(C_{CH_i}\) directly. Once these readings are received, the cluster head \(C_{CH_i}\) gets \(N_i\) readings which can be denoted as \(d_i = [d_{1i}, d_{2i}, ..., d_{Ni}]\), including \(N_i - 1\) readings from non-CH nodes and one reading from its own. Then, the raw readings \([d_{1i}, d_{2i}, ..., d_{Ni}]\) are converted into ascending order and form a new sequence \(d'_i = [d'_{1i}, d'_{2i}, ..., d'_{Ni}]\) at \(C_{CH_i}\), where \(d'_{1i} < d'_{2i} < \cdots < d'_{Ni}\). After that, \(C_{CH_i}\) multiplies the new sequence \(d'_i\) by a random matrix \(\Phi_i\) and then sends the product \(y_i\) to the sink. Notice that \(y_i = \Phi_i d'_i\) has \(m_i\) measurements. Similarly, each CH transmits their information to the sink similar to the process above. Finally, the sink receives \(y = \bigcup_{i=1}^{I}y_i\), the compression information of all clusters. And the sum of the measurements sent to the sink is \(\sum_{i=1}^{I}m_i = M\). At the sink, the original data can be reconstructed from \(y_i\) by using reconstruction algorithm. The reconstructed data of cluster i is denoted by \(\hat{d}'_i\).

Under such design mode, all intracluster non-CH nodes only transmit their readings to their CH, and each CH transmits \(m_i\) information to the sink. If each cluster contains the same number of nodes \(N_i = N\) and sends \(m\) information to the sink, the communication load of whole network is \(I * (N - 1) + I * m = W + I * (m - 1)\). Because of \(m \ll N\), the communication load of the RCCSDG method is far less than \(W * m\). Therefore, the RCCSDG method can further compress the data by using simple linear operation. In order to reduce the delay, an improved (TDMA) scheme is adopted in our data gathering scheme. To make the delay of data gathering as short as possible, we adopt an improved TDMA scheme (time-division multiplexing access) scheduling scheme which is composed of three phases. Within cluster, each sensor at

\[
y_m = \sum_{i=1}^{I} \Phi_{m,1x1} x_i
\]

\[
y_m = \sum_{i=1}^{I} \Phi_{m,2x2} x_i
\]
regular time intervals generates a field value and transports it to its CH according to the TDMA scheme. And the CH of cluster can gather the sensed data from the sensors of the cluster simultaneously.

It is well known that CSDG is able to recover the raw sensed data with high probability from few measurements when the data are sparse or compressible in some certain domains such as DCT, finite difference, or Wavelet. However, the sparsity of most sensed signals in real world is not perfect. In some actual situations, the sensed data of adjacent nodes are not very uniform or vary greatly although they are close to each other on the physical position. When the sensed signals are not smooth enough, the sparsity of the data is not straightforward and even not sparse enough in transformed domains. As an example, Figure 3(a) shows a 72-dimensional out-of-order signal which has a great deal of volatility and worse smoothness. Obviously, it is not sparse itself. Figures 3(b), 3(c), and 3(d) give the corresponding sparse representation in three transformed domains, respectively. We set a threshold $h = 0.5$; if the value of coefficients is lower than $h$, they will be set as zero. Then the sparsity of signal is 57, 60, and 51 in TV, DCT, and DWT, respectively. Hence, the sparsity of the signal is also very poor in transformed domains. In this case, using current CS methods for data gathering usually cannot achieve good performance. As we know, the number of required measurements for reconstruction is in direct proportion to the sparsity of the signal. To guarantee the accuracy of data recovery, it needs to transmit more measurements in such situations. Therefore, if we can find a proper representation basis $\Psi$ that obtains the sparsest representation or improves the sparsity by some simple preprocessing, it can effectively reduce the number of measurements using CS.

Excitingly, we find that the signals are very smooth and have sparsest representation in TV domain when sorted into ascending order by their amplitudes; the results are shown in Figure 4. Figure 4(a) is the results by sorting the same original data of Figure 3(a) into ascending order; Figure 4(b) plots the sparse coefficients of this new signal in TV domain. One can see that most sparse coefficients are close to zero and only 5 values are relatively large; thus its sparsity is approximately 5. To check whether the other sensor data after reshuffling also have good sparsity, we compute the sparsity of light data, humidity data, and voltage data in different domains. In our tests, every type of sensor has 60 groups and every group contains 72 data items. We compute the sparsity of every group and the average sparsity of all groups as a result. The statistical results are presented in Table 1. We found that the sparsity of sensor data after reshuffling is always lower than the TV, DCT, and DWT in all the scenarios under investigation. These results indicate that the sensor data after sorting have a good sparsity in TV domain. Motivated by this investigation, we can improve the sparsity of the data signal through the use of a simple pretreatment on the raw sensed data. When the CH receives the raw sensed data $d$, which are sorted into ascending order through a simple preprocessing (namely, reshuffling operation) by CH firstly, then the result after preprocessing will be compressed by linear compressed projection and the compression information will be transmitted to the sink. Through the reshuffling operation and linear compressed projection, each cluster can effectively reduce the communication cost. And such process is reasonable because the CH has a certain ability of data processing and the additional computing power is so small that it can be ignored.

For the monitoring applications, the change of sensed data usually varies slowly within short time intervals. In other words, the sensed data have excellent temporal stability in a short time. So, in this short period, it can be assumed that the sparsity of this signal will not change and the sensed data can be arranged in the same order. Utilizing this feature, we just reorder data periodically according to empirical knowledge, which can further reduce energy consumption of the network.

### 3. Reshuffling Clustering Compressed Sensing Based Data Gathering Method

In this section, we will describe the details of our proposed RCCSDG scheme and its implementation. It consists of three parts: (1) sensing part, (2) data compressed on CH, and (3) data recovery.

#### 3.1. Sensing Part

As we know, the cluster route is capable of supporting the large-scale WSNs. In this subsection, we choose the LEACH to solve the problem of limited network scale of planar route. Here, the LEACH is a self-adaptive clustering algorithm whose execution process is cyclical, and each cycle is divided into two stages, namely, the establishment of cluster and data communication.

1. According to LEACH, the whole network is divided into $i$ clusters and each cluster has one cluster head. The non-CH nodes will independently join the corresponding cluster according to distance and then send the joined message to the CH.

2. When the cluster is set up, all non-CH nodes send the readings to their own CH. And the non-CH nodes only communicate with their own CH directly.

The topology of cluster is useful to the application of distributed algorithms, because it is suitable for the large-scale network. In addition, the clustering algorithm that uses periodic selection of cluster head can effectively balance the network energy consumption and prolong the lifetime of the network.

| Table 1: Statistical average sparsity of sensor data in different domains. |
|-------------------------------------|
| Light data | Humidity data | Voltage data |
| TV         | 23.5833       | 68.03        | 68.383      |
| DCT        | 69.2167       | 67.85        | 68.83       |
| DWT        | 46            | 66.93        | 71.95       |
| Reshuffling + TV | 8.13       | 22.9167  | 24.05       |
3.2. Data Compression on CH. Due to all the sensed data in the cluster being collected by the CH, these received signals can be compressed by the CH using CS theory. The premise of CS theory is that the signal is \( K \)-sparse in a certain domain, and the amount of measurements \( M \) is proportional to sparsity \( K \). CS can turn an \( N \)-dimensional signal into \( M \)-dimensional \((M < N)\) while still keeping the information capacity. If we need to decrease the amount of data transmission, the sparsity \( K \) of sensor readings should be reduced. However, how to exploit the sparsity of sensor readings is not straightforward in actual situation. It is well known that the smoother the sensed data is, the sparser those signals will be, and it is the most sparsest when the sensed data are sorted into ascending order. In light of this investigation, we proposed a new compressed sensing data aggregation algorithm based on reshuffling, which can decrease the data transmission of the cluster. And this data aggregation algorithm consists of two parts. The first is the reshuffling algorithm that aims to improve the sparsity. And the second is the linear compressed projection which can reduce the amount of data sampling by using compressed sensing technology.

3.2.1. Reshuffling Algorithm. Assume that \( \mathbf{d}_i = [d_1, d_2, \ldots, d_N]^T \) is the original sensors data sequence received by the CH, where \( d_i \) denotes the reading of node \( i \). We set \( d_j \) representing the \( j \)th element of sequence \( \mathbf{d} \) and compare all adjacent data from 1 to \( N \), in turn, such as \( d_j \) and \( d_{j+1} \). If \( d_j > d_{j+1} \), exchange the elements in the \( j \)th and \( (j+1) \)th position; then compare with the next data in the same way. Otherwise, keep the data unchanged in the \( j \)th and the \((j+1) \)th position, and directly compare with the data in next position. It will generate a new sequence \( \mathbf{d}_i' \) when the comparison is finished, and then repeat the previous operations for every new \( \mathbf{d}_i' \) until the elements of data are in an ascending sort order \( d_1' < d_2' < \cdots < d_{N-1}' < d_N' \). The elements of initial data can be sorted into ascending order through such operations as that shown in Algorithm 1. And the new data vector can be represented by

\[
\mathbf{d}_i' = Z_A(A^{-1}(\mathbf{d}_i)), \tag{1}
\]

where \( Z_A(A^{-1}(\mathbf{d}_i)) \) is the reshuffling operation for sorting the elements of vector \( \mathbf{a} \) in ascending order. It is easy to show that the algorithm needs \( N_j * (N_j - 1)/2 \) comparisons and \( N_j * (N_j - 1)/2 \) shift operations to transfer \( \mathbf{d}_i \) into \( \mathbf{d}_i' \) for the worst-case scenario when the original data sequence is in a reverse order.

3.2.2. Linear Compressed Projection. Compressing the signals of sensors is the principal aim of data aggregation for WSNs. Each CH of the network utilizes the linear compressed projection to realize data compression. After getting the reshuffling preprocessing data \( \mathbf{d}_i' \), each CH synchronously generates a Gaussian random matrix \( \Phi_i \), and then the CH multiplies the Gaussian random matrix \( \Phi_i \) by the data vector \( \mathbf{d}_i' \) to produce the projection \( \mathbf{y}_i \), where \( \mathbf{d}_i' \) is an ascending sequence. By the linear compressed projection, the dimensions of the data are reduced to \( M (M < N) \) dimensions from \( N \) dimensions. Thereby it can decrease the communication overhead. Linear compressed projection model for whole network can be represented as

\[
\begin{pmatrix}
\mathbf{y}_1 \\
\mathbf{y}_2 \\
\vdots \\
\mathbf{y}_M
\end{pmatrix} = \begin{pmatrix}
\Phi_1 \\
\Phi_2 \\
\vdots \\
\Phi_M
\end{pmatrix} \begin{pmatrix}
\mathbf{d}_1' \\
\mathbf{d}_2' \\
\vdots \\
\mathbf{d}_W'
\end{pmatrix}. \tag{2}
\]

Through the data aggregation, each CH just needs to transmit the measurement vector \( \mathbf{y}_i \), but not all sensor data, to the sink. It can be concluded from (2) that the compression
collect the intra-cluster readings $\mathbf{d}_i = [d_1, d_2, \ldots, d_{N_i}]^T$.

for $K = 1$ to end do (K is the comparison round)
    for $j = 1, j \in (1, N_i)$
        do compare the data on the adjacent location $d_j, d_{j+1}$
        if $d_j \leq d_{j+1}$
            then exchange the elements of $j$th and $(j + 1)$th.
        else keep the data in $j$th and $(j + 1)$th position.
    end if
end for
end for
 output ascending sequence $\mathbf{d}'_i = [d'_1, d'_2, \ldots, d'_{N_i}]^T$

**Algorithm 1:** Reshuffling data into ascending order.

for $i = 1$ to $I$ do
    collect the intra-cluster readings $\mathbf{d}_i(t_0) = [d_1, d_2, \ldots, d_{N_i}]^T$ at $t_0$.
    for $t = t_0$ to end do
        if $(t - t_0) \% T = 0$
            using Reshuffling Algorithm get sequence $\mathbf{d}'_i(t) = [d'_1, d'_2, \ldots, d'_{N_i}]^T$
        else
            collect readings according to the data structure of $\mathbf{d}'_i(t - \Delta t)$ as $\mathbf{d}_i(t)$.
        end if
        linear projection $y_i(t) = \Phi_i \mathbf{d}_i(t)$ or $\Phi_i \mathbf{d}'_i(t)$
        then transmit $y_i(t)$ to the sink.
    end for
end for

**Algorithm 2:** Linear compressed projection.

information y has $M$ measurements, far less than the number of original data $W$. Beyond that, data sorted into ascending order through reshuffling algorithm can effectively reduce the data sparseness and thus decrease the total number of measurements to $M$. And using this way in data compression, we can reconstruct the raw data of all nodes in the sink through numerical methods.

Since many sensed signals such as temperature and humidity have excellent temporal stability in a short time and the data sorted into ascending order is more sparse than the original order, as a result, the sensed data collected at next interval can also be considered to be sparse when organized in the same order. But as we know, with the monitoring time growing, the signals will change and the sparsity of them may degrade. When sparsity is poor, more measurements will be needed for accurate reconstruction; otherwise the reconstruction will fail. Meanwhile, reordering the elements of data signal every time to obtain optimal measurements for exact reconstruction will increase extra complexity and energy consumption. To cope with this situation, we update the ordering periodically. We set an updated cycle $T$ according to the prior knowledge firstly. During the updated cycle $T$, the sensed data will be reshuffled just one time and keep the order. When the time interval between the current acquisition and first order is integer times of $T$, the CH will update the ordering of $\mathbf{d}_i(t)$ and use the new arrangement for data compression in the following $T$. The main process is shown in Algorithm 2.

3.3. Data Recovery. CS theory points out that when the number of measurements $M$ satisfies (3), a $K$-sparse signal may be exactly reconstructed:

$$M \geq c \cdot K \cdot \log \left( \frac{N}{K} \right),$$

(3)

where $c$ is a positive constant and $N$ is the length of signal here. Equation (3) also indicates that the smaller the $K$ is, the fewer the measurements are needed for accurate reconstruction. In practice, $M = 3K \sim 4K$ can usually satisfy the condition of (3).

The sink gathers all measurements $y$ from every CH and takes a responsibility for recovering the sensor data $\mathbf{d}'$ from these measurements. Because the measurements $y_i$ are obtained by linear compressed projection on the sensors data, it contains enough information for exact reconstruction, since the compression information $y$ is an $M$-dimensional vector and the sensor data sequence $\mathbf{d}'$ is an $N$-dimensional vector, and $M \ll N$. Thus, (2) is an underdetermined equation and cannot recover the original signal directly. We have investigated in this paper that the sensors data after reshuffling have a better sparsity in TV domain. Putting this
prior knowledge into the signal model, the sink can reconstruct the raw sensed data via solving the $I_0$-minimization problem [23, 24]:

$$
\min \|x\|_0
\text{s.t.} \ y = \Phi d' = \Phi \Psi^{-1} x = \Phi' x.
$$

where $\Psi$ is the sparse matrix and $x$ is the sparse coefficients. Solving $I_0$-minimization problem can achieve precise reconstruction, but it is a NP problem. The Optimized Orthogonal Matching Pursuit (OOMP) [25] approach improved on the basis of the Matching Pursuit (MP) [26] and Orthogonal Matching Pursuit (OMP) [27] has a good convergence and high reconstruction accuracy. Given this, we choose OOMP algorithm to reconstruct the compressed data and the algorithm is summarized as Algorithm 3, where $a_n$ is the dictionary atom, $R_k$ is the $k$th order residue, and $l_k$ is the index for $a_{l_k}$ when $|\langle a_{l_k}, R_k \rangle|$ takes the maximal value, where $\langle \cdot, \cdot \rangle$ represents the inner product.

**Algorithm 3: OOMP algorithm.**

Therefore, the energy consumption of the CH can be formed as

$$
E_{CH} = (E_{DP} + E_{TR}).
$$

For simplicity, we only consider the situation of one cluster in the following analysis.

4.1.1. Analysis of $E_{DP}$. The energy consumption of CPU is determined by the number of operations for signal processing. In other words, energy consumption of the data processing is scaled with the operation during the process of signal processing. In RCCSDG scheme that we proposed, all sensed data within cluster are sorted into ascending order through reshuffling algorithm first and then acquire $m$ measurements through linear compressed projections on CH. So the energy consumption of CH for data processing also includes reshuffling cost ($E_{DP,RS}$) and data compression cost ($E_{DP-CS}$) except that of data reading and writing.

For reshuffling algorithm, it requires no more than $N \times (N - 1)/2$ comparisons and $N \times (N - 1)/2$ shift operations for the reverse order, as mentioned in Section 3. And the complexity of reshuffling algorithm is $O(N^2)$. The $m$ random measurements are acquired through a linear compressed projection on $N$ sensor data. It is noted that the linear compressed projection is a matrix multiplication operation in essence. And matrix multiplication is the process of multiplying the $m \times N$ measurement matrix by an $N$-dimensional data vector to get an $m$-dimensional vector. It needs to execute $m \times N(N - 1)$ additions and $m \times N$ multiplications. So the total energy consumption of cluster head for data processing can be expressed as a sum:

$$
E_{DP} = N\xi_{\text{mul}} + \frac{(N - 1)}{2}(\xi_{\text{cmp}} + \xi_{\text{sft}}) + \frac{m}{E_{DP-\text{RS}}} + \frac{mN\xi_{\text{add}} + m(N - 1)\xi_{\text{mul}} + m\xi_{\text{mwr}}}{E_{DP-\text{CS}}}.
$$

4. Energy Consumption and Delay Analysis of RCCSDG Method

4.1. Energy Consumption Analysis of RCCSDG. The previous section has described how to collect and recover the sensed data in RCCSDG scheme, and this section will investigate the energy consumption of RCCSDG. In the process of data gathering, non-CH nodes transmit their sensed data to the CH that they belong to, and the CH is responsible for aggregating the data and sending the results to the sink. Since the CH nodes are the main contribution to energy consumption in WSNs, thus here we only analyze the energy consumption of the CH. The total energy consumption of the CH is comprised of two main components: data processing energy consumption-$E_{DP}$ and data transmission energy consumption-$E_{TR}$.
where $\xi_{\text{mrd}} = 9.90 \text{nJ}$, $\xi_{\text{cmp}} = 3.30 \text{nJ}$, $\xi_{\text{fs}} = 3.30 \text{nJ}$, $\xi_{\text{mul}} = 9.90 \text{nJ}$, and $\xi_{\text{mwr}} = 9.90 \text{nJ}$ are the energy consumption values for memory reading, comparison, shift operation, addition operation, multiplication, and memory writing in CPU of sensor node [28]. If the cluster head refreshes the ordering of data every $T$, the computation energy can be further reduced (reduce $(T - 1) \times E_{\text{DP-RS}}$) and represented by

$$E_{\text{DP}}(T) = \frac{N (N - 1) (\xi_{\text{cmp}} + \xi_{\text{fs}})}{2} + T(N\xi_{\text{mrd}} + mN\xi_{\text{mul}} + m(N - 1)\xi_{\text{mul}} + m\xi_{\text{mwr}}).$$

### 4.1.2. Analysis of $E_{\text{TR}}$

In the process of data communication, the CH receives the sensed data from all non-CH nodes and transmits the compression information to the sink. Thus the energy consumption of $E_{\text{TR}}$ includes sending message ($E_{\text{TR-SD}}$) and receiving message ($E_{\text{TR-RE}}$). Here, we adopt wireless transmission energy consumption model proposed in [21] for analysis of $E_{\text{TR}}$. Depending on distance between the transmitter and the receiver, free space model and the multipath fading model are utilized, respectively. When the transmission distance $d$ is less than the threshold $d_0$, we choose the free space model. Otherwise, the multipath fading model is adopted. To send $l$-bit data for a distance of $d$, the energy consumption model of wireless transmission can be presented as follows:

$$E_{\text{TR-SD}} = \begin{cases} l \times E_{\text{elec}} + l \times \epsilon_{\text{fs}} \times d^2, & d \leq d_0 \\ l \times E_{\text{elec}} + l \times \epsilon_{\text{mp}} \times d^4, & d > d_0 \end{cases}.$$  

Also, in order to receive $l$-bit data, the sensor expends

$$E_{\text{TR-RE}} = l \times E_{\text{elec}},$$

where $E_{\text{elec}}$ is the energy consumption of transmission circuit to send or receive $l$-bit data. The $\epsilon_{\text{fs}}$ and $\epsilon_{\text{mp}}$ are represented as the power consumption of the launch amplifier to transmit $l$-bit data in different model. In RCCSDG scheme, we assume that there are $N$ nodes in cluster ($m$ CH node and $N - m$ non-CH nodes) and the size of each data packet is $L$ bytes. The CH receives $(N - 1) \times L$-byte data from all non-CH nodes within cluster and sends $m$-byte measurements to the sink. According to (8) and (9), the transmission energy consumption of CH in each gathering cycle can be formulated as

$$E_{\text{TR}} = E_{\text{TR-RE}} + E_{\text{TR-SD}} = \begin{cases} 8l \times ((N - 1)E_{\text{elec}} + mE_{\text{elec}} + m\epsilon_{\text{mp}}d^2), & d \leq d_0 \\ 8l \times ((N - 1)E_{\text{elec}} + mE_{\text{elec}} + m\epsilon_{\text{mp}}d^4), & d > d_0 \end{cases}.$$  

From (10), we can conclude that the energy consumption for transmission, $E_{\text{TR}}$, only depends on the number of measurements $m$ when the distance $d$ and the size of cluster $N$ are fixed. We have proved that the RCCSDG method can improve the sparsity of data by a simple pretreatment on original data and greatly reduce the number of measurements. Therefore, although the RCCSDG scheme increases some extra computation, the energy consumption of data transmission can be greatly reduced. In the next section, we will verify the theory analysis of energy consumption of CH by simulation.

#### 4.2. Delay Analysis of RCCSDG

In this section, we analyze the delay of RCCSDG. The analysis can be done in a way similar to [29]. Recall that the proposed model implements two-hop WSNs based on TDMA$^2$ scheduling scheme. In this scheme, each node within the same cluster sends data to cluster by TDMA-1 and the cluster headers compress the signals and transmit the random projections to sink by TDMA-2. The processing schedule in one round is shown in Figure 5.

To make the delay of data gathering as short as possible, we adopt a pipeline TDMA$^2$ scheduling scheme composed of three phases.

**First Phase.** The sink finds cluster $C_m$ containing the maximum number of nodes and the cluster head can firstly forward the random compressed data to sink.

**Second Phase.** Each node of cluster sends the data to the cluster head by TDMA scheduling method. After receiving the data of all nodes in this cluster, the cluster head compresses these data by random projection and the cluster head of $C_m$ firstly forwards their compressed information to the sink.

**Third Phase.** After the cluster head of the $C_m$ forwards their compressed information to the sink. Other cluster heads forward their randomly compressed information to the sink by TDMA$^2$ scheduling method.

**Definition 1.** The delay of data gathering $D$ is the time when the last random measurement reaches the sink. The delay of RCCSDG based on pipeline TDMA$^2$ scheduling scheme is

$$D = \sum_{i=1}^{m} T_{C_i} + T_{\text{DP+}} + \sum_{i=1}^{I} T_{C_{\text{Ch}}},$$

where $T_{C_i}$ is the $i$ node time slot of cluster $C_m$, $T_{\text{DP+}}$ is processing time of cluster $C_m$ including reshuffling time and compressing time, and $T_{C_{\text{Ch}}}$ is forwarding time of cluster $C_{\text{Ch}}$. Let $t_{\text{sen}}$ be the time of sending one bit and $t_{\text{proc}}$ is the time of processing one bit. Assume that the compressing ratio is $c_{\text{ratio}}$ and equal to all clusters. If we choose the quick sort method as reshuffling algorithm, its worst-case performance is $O(N^2)$, while this is rare. In practice choosing a random pivot almost certainly yields $O(N \log N)$ performance, and the complexity of reshuffling algorithm is $O(N \log N)$. Based on these conditions, (11) can be rewritten as

$$D = N_m \times t_{\text{sen}} + O(N_m \log N_m) \times t_{\text{proc}} + c_{\text{ratio}} \times N \times t_{\text{sen}} = (N_m + M) \times t_{\text{sen}} + O(N_m \log N_m) \times t_{\text{proc}}.$$  

International Journal of Distributed Sensor Networks
From (12), we can conclude that the worst delay case is partitioned to one cluster only and the best delay case is divided into $I$ uniform clusters. So the delay $D$ of RCCSDG is

$$D \leq (N + M) \cdot t_{\text{sen}} + O(N \cdot \log N) \cdot t_{\text{proc}} \leq D$$

$$\leq \left( \frac{N}{I} + M \right) \cdot t_{\text{sen}} + \left( \frac{O(N \cdot \log N)}{I} \right) \cdot t_{\text{proc}}.$$  \hspace{1cm} (13)

### 5. Simulation Results

In the following subsection, we first verify the efficiency of the RCCSDG scheme by energy consumption simulation and then evaluate reconstruction accuracy over real sensed data. The numerical results show that the RCCSDG method is effective in reducing energy consumption in deed. Furthermore, the reconstruction results also demonstrate the better performance of the RCCSDG method.

#### 5.1. Energy Consumption Simulation

The RCCSDG method can effectively reduce the energy consumption by sacrificing a small amount of computing resource. And during the data gathering, the CH, which plays an important role in data processing and data transmission, is the main aspect of network energy consumption. In this subsection, we will give the simulation results of CH energy consumption and verify the efficiency of the RCCSDG method. Table 2 lists the main parameters used in the simulations.

To reduce the energy consumption of CH, we need to know the factors which influence the energy consumption of CH. Here, we have conducted a numerical analysis to the factors that may influence energy consumption of the CH in terms of data transmission and data processing, and the results are shown in Figure 6. From (10), it indicates that the energy consumption of data transmission is determined by the transmission distance and the number of measurements when the size of cluster is fixed. And this is also shown in Figure 6(a). The energy consumption of data transmission is proportional to the amount of measurements when the distance is constant. And the closer the distance between node and CH is, the faster the energy is consumed. In particular, when the distance is greater than threshold $d_0$ (here $d_0 = 80$ m), the energy consumption will quicken significantly. Differently from transmission cost, the energy consumption of data processing is only determined by the number of measurements and the size of cluster. As shown in Figure 6(b), it is easy to note that the energy consumption of CH computation increases with the increase of the cluster size and measurements. Therefore, we can conclude that more energy can be saved with further decreasing the measurements when the distance and the size of cluster are fixed. Luckily, the RCCSDG scheme can recover the data signals with fewer measurements.

By using a pretreatment on original data at CH, the RCCSDG we proposed can decrease the measurements to be transmitted, but it introduces some computation cost. To validate the validity of this scheme, we need to demonstrate that the added computation is far smaller than the reduced transmission. For this, we plot in Figure 7 the energy consumption of transmission and computation of RCCSDG and conventional CS scheme when we set $d = 80$, $N = 200$. The black line is the transmission cost, the red line depicts the computation consumption of RCCSDG, and the pink line represents the computation energy consumption of conventional CS scheme. Obviously, as the compression ratio grows, both transmission cost and computation cost increase, and the transmission cost is always far larger than computation cost. Here, compression ratio is defined as the ratio of the number of measurements to the number of original data. The RCCSDG method can yield the same results as the conventional CS scheme with a lower compression ratio, which will be validated in next subsection. During our experiments, we suppose that RCCSDG has the same performance at ratio = 0.1 while the ratio of conventional.
5.2. Reconstruction Performance Simulation. In order to verify the reconstruction performance of the RCCSDG method we proposed, we use the following two evaluation criterions (RMSE and PSNR) in this paper. Formally, the RMSE (root-mean-square error) is defined as (14), and the PSNR (peak signal-to-noise ratio) is defined as (15):

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left( d_i - \tilde{d} \right)^2},
\]

\[
\text{PSNR} = 10 \times \log_{10} \left( \frac{d_{pp}^2}{\text{RMSE}} \right),
\]

where \( N \) is the length of the data signal, \( d_i \) is the original sensor data of node \( i \), \( \tilde{d} \) represents the reconstructed value of the \( i \)th node, and \( d_{pp} \) represents the peak-to-peak value of signal.

In our paper, we carry out simulation on real sensed datasets. The datasets contain 2666 humidity readings from 31 sensor nodes and they are collected by the Department of
information Engineering (DEI) of the University of Padova on March 24th, 2009. According to CS theory, the original data can be recovered at the sink by some algorithms.

Firstly, we compare the recovery performance of RCCSDG with other data collection schemes at moment \( t_0 \). Figure 8 shows the reconstruction RMSE of RCCSDG, CSDG-OOMP, and CSDG-TV under different dumping ratio and Figure 9 gives the results of PSNR under different sampling ratio. Paper [19] points out that sampling rate is inversely linked to the packet loss rate. This means that the low sampling rate of sensor data is equivalent to the case of high packet loss rate. In this paper, the dumping ratio is defined as \((1 - \text{sampling ratio})\). In order to avoid fluctuation, the reconstructions are repeated 100 times. From the results of Figure 8, we can see that the RMSE of RCCSDG is the lowest one. Figure 9 apparently indicates that the recovery performance by the three schemes increases as the sampling ratio \( r \) increases. Nevertheless, the growth rate of PSNR by the RCCSDG scheme is faster than the other two schemes and the value is always larger than them at the same ratio. For example, the RCCSDG method outperforms conventional CS scheme reconstruction by up to about 9 dB at ratio \( r = 0.4 \) in PSNR. And given the target PSNR, the ratio of RCCDG is lower than the other two schemes. For instance, to get the same PSNR = 30.49, the ratios of RCCDG, CSDG-OOMP, and CSDG-TV are 0.1, 0.7, and 0.8, respectively. The reason is that the sensor readings after reshuffling become piecewise smooth and more sparse. This means that, to achieve the same effect of reconstruction, the number of measurements required by the RCCSDG method is far less than CSDG-OOMP and CSDG-TV scheme. In other words, the RCCSDG method can achieve better recovery performance under a lower compression ratio.

Since the humidity data are not varied much in a short time and smooth when collected in the same order at next collecting moment, as shown in Figure 10, the sparsity of data can be regarded as not changed at this moment when data are arranged in the same order. But as monitoring time increases, the smoothness of the data will become worse, thus impacting the precision of data reconstruction. To handle this problem, the RCCSDG method rearranges the data ordering periodically based on the a priori knowledge. During this period, sensor data are reshuffled just one time and keep the order. By reordering data periodically, the RCCSDG method can make the sparsity of data always stay in a proper range, which can ensure achievement of accurate reconstruction with a small number of measurements. This is depicted by the simulation results in Figure 11. It shows the PSNR of data recovery with compression ratio \( r = 0.4 \) and the reordered period \( T = 20 \). From the results we can see that the average reconstruction PSNR is 34 and far better than the conventional CS scheme without ordering. Therefore, the RCCSDG method can effectively reduce the amount of data transmission and guarantee the reconstruction accuracy at the same time.

6. Conclusion

This paper describes an energy-efficient data gathering scheme for WSNs by reshuffling cluster compressed sensing as described. We have found that the sensed data arranged into ascending order have better sparsity. Based on this principle, the cluster heads just adopt a simple preprocessing on original data to resuffle the data into ascending order, which can greatly improve the sparsity and effectively minimize the amount of data transmission. We have investigated that the additional computation for preprocessing is small enough to be neglected. Besides, most sensor signals have excellent temporal stability in WSNs; we only update the ascending order of data periodically. By incorporating the temporal correlation, the energy consumption can be further reduced while guaranteeing the data reconstruction accuracy. Also we have demonstrated the theoretical analysis of the energy consumption and delay in detail when adopting the RCCSDG scheme. The simulation results based on real sensor data that...
validate the energy efficacy and reconstruction accuracy of the RCCSDG scheme have been proposed. Considering the fact that the sensor data also contains low-rank structure information, our future work will investigate matrix completion to further improve the reconstruction accuracy and reduce the computational complexity so as to conserve the energy and ulteriorly extend the lifetime of network.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work was supported by the National Nature Science Foundation of China (no. 61662015 and no. 31101081) and the Science and Technology Supported Project of Jiangxi Provincial (no. 2015BBIE50095).

References

[1] C. W. Chen and Y. Wang, “Chain-type wireless sensor network for monitoring long range infrastructures: architecture and protocols,” International Journal of Distributed Sensor Networks, vol. 4, no. 4, pp. 287–314, 2008.
[2] C. Wang, H. Ma, Y. He, and S. Xiong, “Adaptive approximate data collection for wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, pp. 1004–1016, 2012.
[3] P. Kasirajan, C. Larsen, and S. Jagannathan, “A new data aggregation scheme via adaptive compression for wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 9, no. 1, article 5, 26 pages, 2012.
[4] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TAG: a tiny aggregation service for ad-hoc sensor networks,” ACM SIGOPS Operating Systems Review, vol. 36, pp. 131–146, 2002.
[5] 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 Proceedings of the 5th International Conference on Information Processing in Sensor Networks, pp. 309–316, ACM, Nashville, Tenn, USA, 2006.
[6] K. Yuen, B. Liang, and B. Li, “A distributed framework for correlated data gathering in sensor networks,” IEEE Transactions on Vehicular Technology, vol. 57, no. 1, pp. 578–593, 2008.
[7] G. Hua and C. W. Chen, “Correlated data gathering in wireless sensor networks based on distributed source coding,” International Journal of Sensor Networks, vol. 4, no. 1-2, pp. 13–22, 2008.
[8] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.
[9] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489–509, 2006.
[10] E. J. Candés and M. B. Wakin, “An introduction to compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, 2008.
[11] S. Lee, S. Pattem, M. Sahami, B. Krishnamachari, and A. Ortega, “Spatially-localized compressed sensing and routing in multi-hop sensor networks,” in Geosensor Networks, pp. 11–20, Springer, Berlin, Germany, 2009.
[12] L. Xiang, J. Luo, and C. Rosenberg, “Compressed data aggregation: energy-efficient and high-fidelity data collection,” IEEE/ACM Transactions on Networking, vol. 21, no. 6, pp. 1722–1735, 2013.
[13] J. Wang, S. Tang, B. Yin, and X.-Y. Li, “Data gathering in wireless sensor networks through intelligent compressive sensing,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM ’12)*, pp. 603–611, Orlando, Fla, USA, March 2012.

[14] M. T. Nguyen and K. A. Teague, “Tree-based energy-efficient data gathering in wireless sensor networks deploying compressive sensing,” in *Proceedings of the 23rd Wireless and Optical Communication Conference (WOCC ’14)*, pp. 1–6, IEEE, Newark, NJ, USA, May 2014.

[15] W. Wang, M. Garofalakis, and K. Ramchandran, “Distributed sparse random projections for refinable approximation,” in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN ’07)*, pp. 331–339, ACM, Cambridge, Mass, USA, April 2007.

[16] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Compressive data gathering for large-scale wireless sensor networks,” in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom ’09)*, pp. 145–156, ACM, Beijing, China, September 2009.

[17] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Efficient measurement generation and pervasive sparsity for compressive data gathering,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 12, pp. 3728–3738, 2010.

[18] J. Cheng, H. Jiang, X. Ma et al., “Efficient data collection with sampling in WSNs: making use of matrix completion technique,” in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM ’10)*, pp. 1–5, IEEE, Miami, Fla, USA, December 2010.

[19] J. Cheng, Q. Ye, H. Jiang, D. Wang, and C. Wang, “STCDG: an efficient data gathering algorithm based on matrix completion for wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 850–861, 2013.

[20] Y. Y. Liu, L. Zhu, and W. L. Tang, “The data aggregation of wireless sensor networks based on compressed sensing and cluster,” *Journal of Computational Information Systems*, vol. 9, no. 9, pp. 3399–3406, 2013.

[21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[22] J. E. Barcelo-Llado, A. Morell, and G. Seco-Granados, “Amplify-and-forward compressed sensing as an energy-efficient solution in wireless sensor networks,” *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1710–1719, 2014.

[23] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[24] M. A. Iwen, “Simple deterministically constructible RIP matrices with sublinear fourier sampling requirements,” in *Proceedings of the 43rd Annual Conference on Information Sciences and Systems (CISS ’09)*, pp. 870–875, IEEE, Baltimore, Md, USA, March 2009.

[25] L. Rebollo-Neira and D. Lowe, “Optimized orthogonal matching pursuit approach,” *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 137–140, 2002.

[26] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[27] T. T. Cai and L. Wang, “Orthogonal matching pursuit for sparse signal recovery with noise,” *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

[28] C. Karakus, A. C. Gurbuz, and B. Tavli, “Analysis of energy efficiency of compressive sensing in wireless sensor networks,” *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1999–2008, 2013.

[29] H. Zheng, S. Xiao, X. Wang, X. Tian, and M. Guizani, “Capacity and delay analysis for data gathering with compressive sensing in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 917–927, 2013.
