Transmission-Efficient Clustering Method for Wireless Sensor Networks Using Compressive Sensing

Ruitao Xie and Xiaohua Jia, *Fellow, IEEE, Computer Society*

Abstract—Compressive sensing (CS) can reduce the number of data transmissions and balance the traffic load throughout networks. However, the total number of transmissions for data collection by using pure CS is still large. The hybrid method of using CS was proposed to reduce the number of transmissions in sensor networks. However, the previous works use the CS method on routing trees. In this paper, we propose a clustering method that uses hybrid CS for sensor networks. The sensor nodes are organized into clusters. Within a cluster, nodes transmit data to cluster head (CH) without using CS. CHs use CS to transmit data to sink. We first propose an analytical model that studies the relationship between the size of clusters and number of transmissions in the hybrid CS method, aiming at finding the optimal size of clusters that can lead to minimum number of transmissions. Then, we propose a centralized clustering algorithm based on the results obtained from the analytical model. Finally, we present a distributed implementation of the clustering method. Extensive simulations confirm that our method can reduce the number of transmissions significantly.

Index Terms—Wireless sensor networks, compressive sensing, data collection, clustering

1 INTRODUCTION

In many sensor network applications, such as environment monitoring systems, sensor nodes need to collect data periodically and transmit them to the data sink through multihops. According to field experiments, data communication contributes majority of energy consumption of sensor nodes [1]. It has become an important issue to reduce the amount of data transmissions in sensor networks. The emerging technology of compressive sensing (CS) [2], [3], [4] opens new frontiers for data collection in sensor networks [5], [6], [7], [8], [9], [10], [11], [12] and target localization in sensor networks [13]. The CS method can substantially reduce the amount of data transmissions and balance the traffic load throughout the entire network.

The basic idea of CS works is as follows, as shown in Fig. 1. Suppose the system consists of one sink node and N sensor nodes for collecting data from the field. Let x denote a vector of original data collected from sensors. Vector x has N elements, one for each sensor. x can be represented by Ψs, i.e., $x = \Psi s$, where Ψ is an $N \times N$ transform basis, and s is a vector of coefficients. If there are at most k ($k \ll N$) nonzero elements in s, x is called k-sparse in the Ψ domain. When k is small, instead of transmitting N data to the sink, we can send a small number of projections of x to the sink, that is, $y = \Phi x$, where Φ is an $M \times N$ ($M \ll N$) random matrix (called the measurement matrix) and y is a vector of M projections. At the sink node, after collecting y, the original data x can be recovered by using ℓ_1-norm minimization [14], [15] or other heuristic algorithms, such as orthogonal matching pursuit [16]. More background of CS and related works can be found in Section 1 of the online supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.90.

In data gathering without using CS, the nodes close to tree leaves relay fewer packets for other nodes, but the nodes close to the sink have to relay much more packets. By using CS in data gathering, every node needs to transmit M packets for a set of N data items. That is, the number of transmissions for collecting data from N nodes is MN, which is still a large number. Hybrid approaches were proposed in [8], [10]. In the hybrid method, the nodes close to the leaf nodes transmit the original data without using the CS method, but the nodes close to the sink transmit data to sink by the CS technique. Xiang et al. [10] applied hybrid CS in the data collection and proposed an aggregation tree with minimum energy consumption. The previous works use the CS method on routing trees. Since the clustering method has many advantages over the tree method [17], [18], [19], [20], [21], [22], such as fault tolerance and traffic load balancing, we use the CS method on the clustering in sensor networks. The clustering method generally has better traffic load balancing than the tree data gathering method. This is because the number of nodes in clusters can be balanced when we divide clusters. In addition, the previous works ignored the geographic locations and node distribution of the sensor nodes. While in sensor networks, the information of node distribution can help the design of data gathering method that uses less data transmissions [17], [18], [19], [20], [21], [22].

In this paper, we propose a clustering method that uses the hybrid CS for sensor networks. The sensor nodes are

- *The authors are with the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong. E-mail: ruitao.xie@my.cityu.edu.hk, csjia@cityu.edu.hk.*

Manuscript received 20 Sept. 2012; revised 11 Jan. 2013; accepted 11 Mar. 2013; published online 26 Mar. 2013.
Recommended for acceptance by V. Misic.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2012-09-0970.
Digital Object Identifier no. 10.1109/TPDS.2013.90.

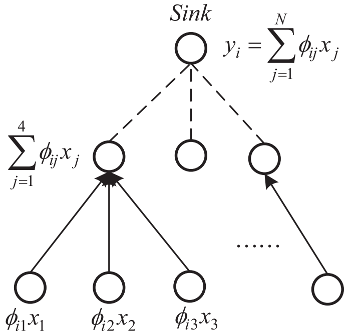

Sink
$y_i = \sum_{j=1}^{N} \phi_{ij} x_j$
$\sum_{j=1}^{4} \phi_{ij} x_j$
......
$\phi_{i1}x_1$
$\phi_{i2}x_2$
$\phi_{i3}x_3$

Fig. 1. Data collection with the pure CS method in the tree structure.

organized into clusters. Within a cluster, nodes transmit data to the cluster head (CH) without using CS. A data gathering tree spanning all CHs is constructed to transmit data to the sink by using the CS method. One important issue for the hybrid method is to determine how big a cluster should be. If the cluster size is too big, the number of transmissions required to collect data from sensor nodes within a cluster to the CH will be very high. But if the cluster size is too small, the number of clusters will be large and the data gathering tree for all CHs to transmit their collected data to the sink will be large, which would lead to a large number of transmissions by using the CS method. In this regard, we first propose an analytical model that studies the relationship between the size of clusters and number of transmissions in the hybrid CS method, aiming at finding the optimal size of clusters that can lead to minimum number of transmissions. Then, we propose a centralized clustering algorithm based on the results obtained from the analytical model. Finally, we present a distributed implementation of the clustering method.

Extensive simulations have been conducted. When the number of measurements is 10th of the number of nodes in the network, the simulation results show that our method can reduce the number of transmissions by about 60 percent compared with the clustering method without using CS. Meanwhile, our method can reduce the number of transmissions by 50 percent compared with the data collection method using the shortest path tree (SPT). In addition, our method can reduce the number of transmissions up to 30 percent compared with the data collection method using SPT with the hybrid CS. Even for the nonhomogenous networks in the irregular sensor field, our method can significantly reduce data transmissions compared with these data collection methods. Our simulation results demonstrate that the proposed distributed method is efficient in terms of the low communication cost and effective in reducing the number of transmissions.

The remainder of this paper is organized as follows: Section 2 presents an overview of the clustering method by using hybrid CS for data collection. Section 3 presents an analytical model for analyzing the relationship between the size of clusters and the number of transmissions, and determining the optimal cluster size. Section 4 presents a centralized algorithm for sensor nodes clustering with minimum number of transmissions. Section 5 presents a distributed clustering algorithm and its implementations. The simulations and performance evaluations are presented in Section 6. Section 7 concludes the paper.

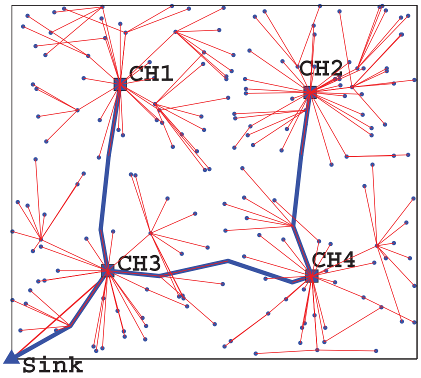


Fig. 2. The hybrid CS data collection method in cluster structure.

2 Overview of Sensor Nodes Clustering for Hybrid Compressive Sensing

We first make the following assumptions:

- The sensor nodes are uniformly and independently distributed in a sensor field. Such a deployment can be modeled as a Poisson point process [21], [22], [23], [24].
- All sensor nodes have the same fixed transmission power and transmission rate.
- Each sensor node is aware of its own geographic location, which can be obtained via attached GPS or some other sensor localization techniques [25], [26]. The location information is used in the distributed implementation.

In our method, sensor nodes are organized into clusters, and each cluster has a cluster head, represented by the solid square as shown in Fig. 2. Sensor nodes in each cluster transmit their original data to the CH without using CS. We assume each CH knows the projection vectors (in measurement matrix Φ) of all nodes within its cluster. In real systems, the measurement coefficient ϕ_{ij} can be generated using a pseudorandom number generator seeded with the identifier of the node v_j [5]. Thus, given the identifiers of the nodes in the network, the measurement matrix can be easily constructed at CHs or the sink locally. The measurement matrix Φ can be decomposed into submatrices, one for each cluster. Let Φ^{H_i} denote the submatrix for ith cluster. For ith cluster, let CH$_i$ denote the cluster head and x^{H_i} denote the data vector of the cluster. The CH$_i$ is able to compute the projections of all data x^{H_i} collected from the nodes in its cluster on the submatrix, that is $\Phi^{H_i}x^{H_i}$. The CH$_i$ generates M projections from the data within its cluster by using the CS technique. The value of M is determined by the number of nodes N and the sparsity level of the original data [5]. It then forwards them to the sink in M rounds along a backbone tree that connects all CHs to the sink. Taking the sensor nodes in Fig. 2 as an example, all sensors nodes are divided into four clusters. The four cluster heads, CH$_1$, CH$_2$, CH$_3$, and CH$_4$, are connected by a backbone tree to the sink. Data vector x can be decomposed as $[x^{H_1}\ x^{H_2}\ x^{H_3}\ x^{H_4}]^T$, and matrix Φ can be written as $[\Phi^{H_1}\ \Phi^{H_2}\ \Phi^{H_3}\ \Phi^{H_4}]$.

$$\begin{aligned} y &= \Phi x \\ &= \begin{bmatrix} \Phi^{H_1} & \Phi^{H_2} & \Phi^{H_3} & \Phi^{H_4} \end{bmatrix} \begin{pmatrix} x^{H_1} \\ x^{H_2} \\ x^{H_3} \\ x^{H_4} \end{pmatrix} \\ &= \sum_{i=1}^{4} \Phi^{H_i} x^{H_i}. \end{aligned} \tag{1}$$

As shown in (1), the projections of all data in the network on the measurement matrix Φ is the sum of the projections generated from the clusters. Thus in each round, the CH aggregates its own projection and the projections received from its children CHs in the same round and forwards it to the sink following the backbone tree. When the sink receives all M rounds of projections from CHs, the original data for all sensor nodes can be recovered.

There are two levels of transmissions in our clustering method using the hybrid CS: intracluster transmissions that do not use the CS technique and intercluster transmissions that use the CS technique. The data size in intercluster transmissions is the same as the data in intracluster transmissions. Thus, reducing the number of transmissions can effectively reduce the energy consumption of sensor nodes. For intracluster transmissions, we simply let sensor nodes transmit their data to the CH following the shortest path routing (in terms of number of hops). For intercluster transmissions, we construct a minimal cost (in terms of number of hops) backbone tree that connects all CHs to the sink and transmit the data projections along this backbone tree.

An important task of our method is to determine the cluster size. As cluster size increases, the number of intracluster transmissions would increase sharply. But when decreasing the cluster size, the number of clusters would increase and the number of intercluster transmissions would increase. Thus, there exists an optimal cluster size that minimizes the total number of data transmissions in the hybrid CS method. Our task is to determine the optimal cluster size and design a distributed clustering method, such that the total number of transmissions is minimized.

3 Analysis on the Optimal Cluster Size

There are N sensor nodes uniformly and independently distributed in a rectangle sensor field. Such a deployment can be modeled as a Poisson point process. Let λ denote the density of the underlying Poisson point process. The number of sensors located in a region with the area of A, $N(A)$, follows the Poisson distribution with mean of λA, i.e., $N(A) \sim \mathrm{Poi}(\lambda A)$. The assumption of uniform sensor distribution has been widely used in the performance analysis of large-scale wireless sensor networks [21], [22], [23], [24].

There is a sink node s located at the corner of the sensor field. We assume the coordinates of s are (0, 0), as shown in Fig. 3. This is because the sink is usually placed outside of the sensor field for easy installation. Our analysis can be easily modified to suit the cases that the sink is not located at the corner of the field. We assume that the transmission range of sensor nodes is r. That is, any two sensors whose euclidian distance is within r can communicate with each other.

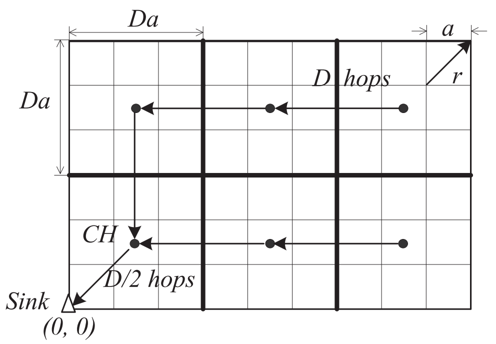


Fig. 3. The sensor field is partitioned into small grids with size $a \times a$. All nodes in a cluster-square of the size $Da \times Da$ form a cluster, where the cluster head (CH) is located at the center.

The sensor field is partitioned into small grids of size $a \times a$ as shown in Fig. 3. The edge length a of a grid is set to $\frac{r}{\sqrt{2}}$, so that any two nodes in a grid are within the transmission range of each other. Our purpose is to divide the sensor field into cluster-areas, such that nodes can be organized into clusters. Suppose each cluster-area is a square of size $Da \times Da$. All nodes in a cluster-square form a cluster as shown in Fig. 3. The largest feasible value D_{MAX} is

$$D_{\mathrm{MAX}} = \sqrt{\frac{N}{\lambda a^2}}. \tag{2}$$

The value of D lies in the interval $[1, D_{\mathrm{MAX}}]$, and it will be determined later through our analysis. Given the poisson distribution with density λ, there are $\lambda D^2 a^2$ sensor nodes in each cluster on average. Thus, the sensor field has $\frac{N}{\lambda D^2 a^2}$ clusters on average.

In our hybrid CS method with the cluster structure, the data transmission from the sensor nodes to the CH does not use CS. The sensor nodes within a cluster transmit their data to the CH via the shortest path routing. We assume the CH is located at the center of the cluster-square, which is the case that produces the minimal number of transmissions to collect data within the cluster when nodes are uniformly distributed. Considering the small grids inside a cluster-square as shown in Fig. 4, the nodes in the center grid that contains the CH take only one hop to transmit their data to the CH. The nodes in the next layer of grids around the center grid take two hops to reach the CH, and the nodes in the third layer of grids are three hops away from the CH. Following this pattern, the nodes in hth layers take h hops to transmit data to the CH. The number of grids in the hth layer is $8(h-1)$ for $h \geq 2$. Since the sensor nodes are distributed uniformly with the density λ, the number of nodes in each grid is λa^2. Thus, the number of data transmissions for all sensor nodes within a cluster to transmit their data to the CH is

$$\left(1 + \sum_{h=2}^{\frac{D+1}{2}} 8(h-1) \cdot h\right) \cdot \lambda a^2 = \left(\frac{D^3 - D}{3} + D^2\right) \cdot \lambda a^2. \tag{3}$$

Since the total number of clusters in the system is $\frac{N}{\lambda D^2 a^2}$, the total intracluster transmissions of all clusters, without using CS, are

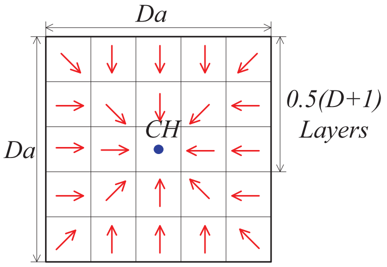


Fig. 4. The intracluster transmissions in the cluster-square, where the cluster head (CH) is located at the center.

$$\left(\frac{D^3 - D}{3} + D^2\right) \cdot \lambda a^2 \cdot \frac{N}{\lambda D^2 a^2} = \left(\frac{D}{3} - \frac{1}{3D} + 1\right) \cdot N. \tag{4}$$

The number of intracluster transmissions calculated above is an upper bound. Since in each grid, the nodes close to the inner layer take one hop less than the nodes close to the outer layer to transmit data to the CH.

To get the lower bound of the number of intracluster transmissions, considering the small grids inside a cluster-square as shown in Fig. 4, the nodes in the next layer of grids around the center grid that contains the CH take one hop to reach the CH, and the nodes in the third layer of grids are two hops away from the CH. Following this pattern, the nodes in hth layers take $h-1$ hops to transmit data to the CH. That is, each node takes 1 hop less than in the analysis of the upper bound of intracluster transmissions. Thus, the lower bound of the number of intracluster transmissions is N less than that in (4) in total. That is,

$$T_{intra} = \left(\frac{D}{3} - \frac{1}{3D}\right) \cdot N. \tag{5}$$

From (5), we can see that with fixed clusters' area, the number of intracluster transmissions is proportional to the number of nodes N, while with fixed N, it increases as the edge length of cluster-square D increases.

In our hybrid CS method, data are compressed by using CS at the CHs. The data projections generated at each CH are forwarded to the sink in M rounds along a backbone tree that connects all the CHs to the sink. We call CHs in adjacent clusters neighboring CHs. In each round of transmission, the projection of each CH is forwarded to its neighboring CH via some intermediate relay nodes along the routing tree. In our analytical model, we assume that backbone tree is structured as shown in Fig. 3: 1) all CHs transmit data to their left-neighbor CH until reaching the left most cluster; 2) for clusters at the left most column, CHs transmit data to down-neighbor CHs until reaching the left-bottom cluster; and 3) the CH of the left-bottom cluster transmits data to the sink.

It takes D hops to transmit a projection from its CH to a neighboring CH. For the cluster at the left-bottom corner (i.e., the cluster whose left-bottom corner is at (0, 0)), it takes approximately $\frac{D}{2}$ hops to transmit a projection from the CH to the sink. The projections of the same round generated from different clusters are aggregated at CHs as they are forwarded along the backbone tree. Thus, each intermediate node on the backbone tree does M transmissions. Since the number of clusters in the sensor field is $\frac{N}{\lambda D^2 a^2}$, the total number of intercluster transmissions with CS used for M rounds is

$$\begin{aligned} T_{inter} &= \left(\frac{N}{\lambda D^2 a^2} - 1\right) \cdot D \cdot M + \frac{D}{2} \cdot M \\ &= \frac{NM}{\lambda a^2} \cdot \frac{1}{D} - \frac{M}{2} \cdot D. \end{aligned} \tag{6}$$

From (6), it can be observed that with fixed D, the number of intercluster transmissions is proportional to the number of projections M; with fixed M, the number of intercluster transmissions decreases as D increases.

Our objective is to minimize total number of transmissions of the hybrid CS method in cluster structure, which is the sum of the intracluster transmissions and the intercluster transmissions. That is,

$$\begin{aligned} T &= T_{intra} + T_{inter} \\ &= \left(\frac{N}{3} - \frac{M}{2}\right) \cdot D + \left(\frac{NM}{\lambda a^2} - \frac{N}{3}\right) \cdot \frac{1}{D} \\ &= \frac{N}{3}\left(1 - \frac{3M}{2N}\right) \cdot D + \frac{N}{3}\left(\frac{3M}{\lambda a^2} - 1\right) \cdot \frac{1}{D} \\ &= c_1 \cdot D + c_2 \cdot \frac{1}{D}. \end{aligned} \tag{7}$$

Considering the above (7), T is a function of D, where D lies in the interval $[1, D_{\mathrm{MAX}}]$.

With different M, the optimal value of D to minimize T is different. When M is less than $\frac{2}{9}N$, the optimal value D^* is calculated as

$$D^* = \begin{cases} \sqrt{\frac{\frac{3M}{\lambda a^2} - 1}{1 - \frac{3M}{2N}}}, & M < \frac{2}{9}N; \\ D_{\mathrm{MAX}}, & \frac{2}{9}N \le M \le N. \end{cases} \tag{8}$$

When $M \ge \frac{2}{9}N$, and in the extreme case when $M = N$, the optimal value D^* is D_{MAX}. That is, the sensors in the network are organized to a single cluster, which is degenerated into the optimal tree structure using hybrid CS [10]. The detailed derivation of the optimal value D^* is in Section 2 of the online supplemental material.

The optimal cluster size N_c^* of hybrid CS method in cluster structure, in terms of the number of nodes in each cluster, is

$$N_c^* = \lambda (D^* a)^2 = \begin{cases} \frac{3M - \lambda a^2}{1 - \frac{3M}{2N}}, & M < \frac{2}{9}N; \\ N, & \frac{2}{9}N \le M \le N. \end{cases} \tag{9}$$

That means, when there are N_c^* nodes in each cluster, the total number of transmissions in the clustering with hybrid CS is minimized.

4 Minimum Transmission Clustering Algorithm

4.1 Overview of Centralized Clustering Algorithm

The sensor network is modeled by a graph $G = \langle V, E \rangle$, where V consists of the sink node v_0 and N sensor nodes. If two nodes in V are within the communication range of each other, then there is a link between the two nodes.

As the centralized algorithm, we assume the sink node has the full knowledge of the network topology. That is, it knows the network graph $G = \langle V, E \rangle$. The sink will divide the sensor nodes into clusters, choose a CH for each cluster, and construct a backbone tree that connects all CHs to the sink. After computing the clustering, the sink can broadcast the clustering information to all sensor nodes and start data collection subsequently.

From the theoretical analysis in the last section, we can find the optimal cluster size N_c^* for a given number of N sensor nodes uniformly distributed in a field. Thus, the optimal number of clusters in the system is:

$$C = \left\lceil \frac{N}{N_c^*} \right\rceil. \tag{10}$$

In our method, within a cluster, each sensor node transmits its data to its designated CH via the shortest path. The routes that sensor nodes use to send their data to the CH form a shortest path tree in each cluster. The total number of intracluster transmissions is the sum of the distance of all sensor nodes to their CHs. Thus, the clustering problem for minimizing intracluster transmissions becomes a well-known k-median problem, that is to find the locations to place C CHs in the network $G = \langle V, E \rangle$ such that the total distance from all sensor nodes to their nearest CHs is minimized. The distance between two nodes is defined as the number of hops of the shortest path between them.

Data collected from sensor nodes is compressed by the CS method at the CHs. The data projections generated at each CH are forwarded to the sink in M rounds along the backbone tree. At each CH in the backbone tree, it aggregates its own data projection with the projections received from other CHs by using the CS method and forwards the aggregated projection upward toward the sink along the tree. There are usually multihops between two CHs. Thus, the problem of constructing a backbone tree that connects all CHs to the sink and has the minimum number of links in the tree is the well-known minimum Steiner tree problem, which is NP-hard. We will use an efficient heuristic method to construct the backbone tree.

4.2 Centralized Clustering Algorithm

In this section, we present the centralized clustering algorithm. Given the network $G = \langle V, E \rangle$, our algorithm has two major steps: 1) select C CHs from the set V of N sensor nodes and divide the sensor nodes into C clusters and 2) construct a backbone routing tree that connects all CHs to the sink.

The k-median problem is NP-hard. A lot of heuristic algorithms have been proposed to solve the k-median problem [27], [28], [29]. We adopt an efficient method that iteratively closes to the near-optimal solution. Our algorithm starts from an initial set of CHs, which is randomly selected. At each iteration, the algorithm proceeds following steps:

1. Connect sensor nodes to their closest CHs. Ties break arbitrarily.
2. For each cluster, choose a new CH, such that the sum of the distances from all nodes in this cluster to the new CH is minimized.
3. Repeat the above two steps until there is no more change of the CHs.

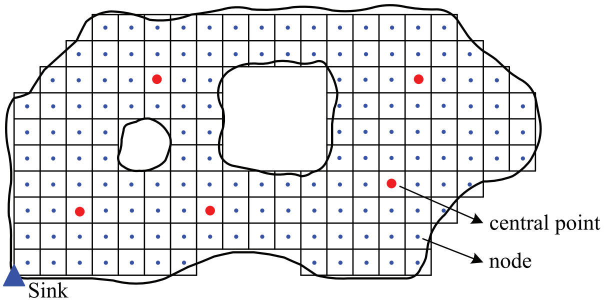


Fig. 5. An example of calculating the central points of cluster-areas: an irregular sensor field is roughly divided into small grids, and a virtual node is placed at the center of each grid. Five nodes are chosen as the approximate central points.

This algorithm converges quickly. The simulations show that it takes four or five iterations on average for the algorithm to compute the CHs of clusters (see Section 4.3 of the online supplemental material).

We use a minimum spanning tree (MST)-based method to compute the backbone tree that connects all CHs and the sink. Given a set U of CHs obtained from the above algorithm, we introduce a graph $G_{\mathrm{CH}} = \langle V_{\mathrm{CH}}, E_{\mathrm{CH}} \rangle$, where V_{CH} consists of the sink node v_0 and the set U of CHs. There is an edge between any pair of nodes in V_{CH}. That is, the graph G_{CH} is a complete graph. The distance of an edge $(\mathrm{CH}_i, \mathrm{CH}_j)$ in E_{CH} is the length of the shortest path between CH_i and CH_j in G. Then, we compute the MST of G_{CH}, which spans all nodes in V_{CH}. From this MST, we obtain a backbone routing tree, where each edge in the MST is its corresponding shortest path in G.

5 Distributed Implementation

This section presents a distributed implementation of the clustering method. We assume that 1) every sensor node knows its geographic location. This location information can be obtained via attached GPS or some other sensor localization techniques [25], [26]; 2) the sink knows the area of the whole sensor field, but does not need to know the location information of all sensor nodes. This is a reasonable assumption, since in most applications of the sensor networks, the sink usually knows the area that has sensors deployed for surveillance or environmental monitoring [30].

In our distributed algorithm, the sink divides the field into C cluster-areas, calculates the geographic central point of each cluster-area, and broadcasts the information to all sensor nodes to elect CHs. The sensor node that is the closest to the center of a cluster-area is selected to be the CH. The CHs then broadcast *advertisement* messages to sensor nodes to invite sensor nodes to join their respective clusters.

5.1 Calculating Central Points of Cluster-Areas

Given a sensor field and the number of cluster C to be divided to, the sink needs to find out the central points of C cluster-areas. We first divide the whole sensor field into small grids, as shown in Fig. 5. Then, we place a virtual node at the center of each grid to represent the grid. C nodes in the grids will be chosen as the approximate central points of the cluster-areas. We use an auxiliary graph $G_A = \langle V_A, E_A \rangle$ to help finding the central points, where V_A is the set of nodes in the grids, and each node v_i in V_A has an edge to each of the nodes in its neighboring grids. Each grid, except those on the border of the sensor field, has eight neighboring grids (as shown in Fig. 5). The distance of all edges in E_A is set to 1. We compute a subset of nodes V_C, $V_C \subset V_A$ and $|V_C| = C$, such that the total distance from all nodes in V_A to their nearest nodes in V_C is minimized. The nodes in V_C are the approximate central points of the C cluster-areas in the sensor field. We use the same iterative algorithm presented in Section 4.2 to compute the set of nodes V_C from V_A in graph G_A.

After computing V_C, the sink can calculate the geographic locations of the nodes in V_C, which are the approximate locations of C central points of the cluster-areas. The sink then broadcasts the locations information of central points to all sensor nodes for CHs election. The size of the grids that the sink divides the sensor field to depends on the accuracy of locating the central points. The smaller the size is, the more accurate the locations information will be, but it incurs more computation cost in this case. In our simulation, we simply set the grid size as $a \times a$, where a is defined in Section 3.

5.2 Cluster Head Election

Given the geographic location of the central point of a cluster-area, the sensor node that is the closest to the central point will become the CH. Since the sensor nodes do not know who is the closest to the central point of a cluster-area, and we do not know if there is a sensor node falling into the close range of the central point, we let all nodes within the range of Hr from the center be the CH candidates of the cluster, where r is the transmission range of sensors. The value of H is determined such that there is at least one node within H hops from the central point of a cluster (The detailed discussion on H is in Section 3 of the online supplemental material). To elect the CH, each candidate broadcasts a *CH election* message that contains its identifier, its location and the identifer of its cluster. The *CH election* message is propagated not more than $2H$ hops. After a timeout, the candidate that has the smallest distance to the center of the cluster among the other candidates becomes the CH of the cluster.

In the extreme case that no sensor node falls within H hops from the central point so that there is no CH for this cluster-area, the nodes in this cluster-area accept the invitation from neighboring CHs and become members of other clusters. Thus, no node will be left out of the network.

5.3 Sensor Node Clustering

After a CH is elected, the CH broadcasts an *advertisement* message to other sensor nodes in the sensor field, to invite the sensor nodes to join its cluster. An *advertisement* message carries the information: the identifier and location of the CH, and the number of hop that the message has traveled. The hop count is initialized to be 0.

When a sensor node receives an *advertisement* message, if the hop count of message is smaller than that recorded from the same CH, it updates the information in its record including the node of previous hop and the number of hop to the CH, and further broadcasts the message to its neighbor nodes; otherwise, the message is discarded. The maximal hop count for the *advertisement* message is set to $\lceil D^* \rceil$ hops (D^* is from (8)), so that all nodes can receive the *advertisement* messages from at least one CH.

After the advertisement of CH is complete, each non-CH node decides which cluster it joins. The decision is based on the number of hops to each CH. The routing from a sensor node to its CH follows the reverse path in forwarding the *advertisement* message. The data of sensor nodes within a cluster is collected by this routing tree.

5.4 Backbone Tree Construction and Network Maintenance

A backbone tree is constructed in a distributed fashion to connect all CHs and the sink. Through the broadcasting of the *advertisement* messages from CHs, each CH receives the *advertisement* messages from the other CHs that are close to it. Thus, it has the knowledge about the locations of its nearby CHs and the number of hops to them. Since the sink needs to broadcast the central points information to all sensor nodes, all sensor nodes know the location of the sink and the hop distance to it. For each CH, we define its upstream CHs as the set of CHs (including the sink) that are closer to the sink than itself in terms of euclidean distance. We take a distributed method of an approximate MST algorithm to construct the backbone tree. For each CH, it chooses the CH that has the minimum number of hops to it from the set of its upstream CHs as its parent CH in the backbone tree.

After constructing the backbone tree, each CH has the knowledge about its children CHs in the backbone tree. When M projections are generated at the CH, they are transmitted to the parent CH along the backbone tree in M rounds.

When a CH fails or runs out of energy, the neighboring nodes of the CH will detect the failure of the CH. These nodes will broadcast a message to all the nodes in this cluster to start the new CH election. The new CH election algorithm and the new backbone construction follow the same methods as presented in Sections 5.2 and 5.4. As there are many distributed routing algorithms that were proposed for sensor networks [31], [32], we simply use the existing method [31], [32] for route maintenance.

6 Performance Evaluation

In this section, we evaluate the performance of the clustering method using hybrid CS. Our method is compared with four other data collection methods. We first evaluate the performance of our method on a regular sensor field. In Section 6.2, we demonstrate that our method can significantly reduce the number of transmissions. In Section 6.3, we demonstrate the impact of the cluster size on the number of transmissions. We also evaluate the performance of our method on the nonhomogenous networks in an irregular sensor field. Refer to Section 4.2 of the online supplemental material. All results confirm that our method can save the number of transmissions significantly. In addition, the comparison between analytical results and simulation results is shown in

Section 4.1 of the online supplemental material. It demonstrates that our analytical model is strong in analyzing the number of transmissions. The evaluation of the iteration times to converge of our iterative algorithm in Section 4.2 is shown in Section 4.3 of the online supplemental material. The results demonstrate that our algorithm is efficient and scalable in large-scale networks.

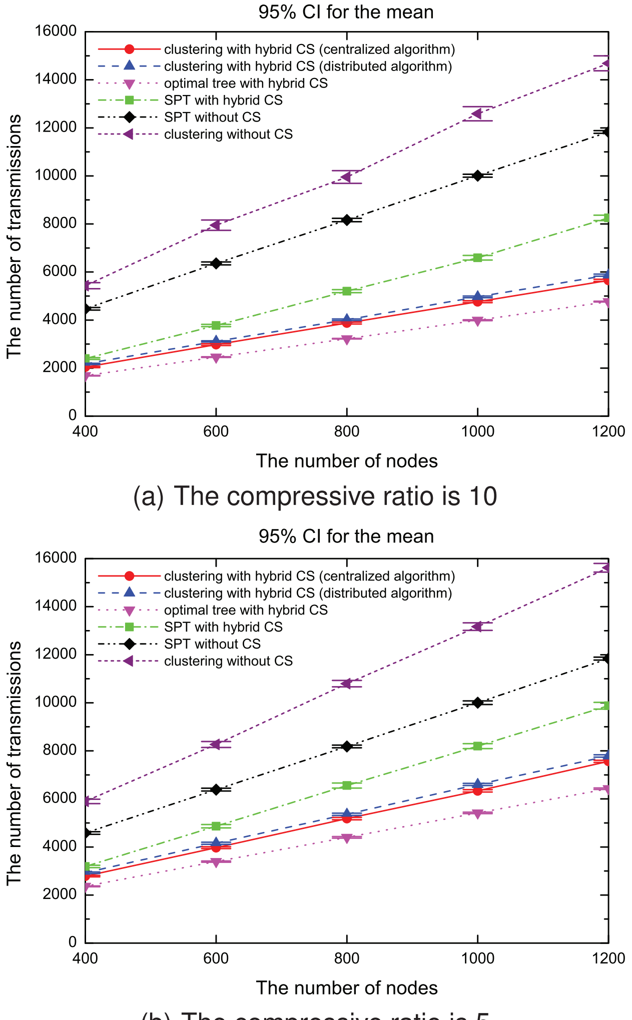
6.1 Simulation Metrics and Setup

We use two metrics to evaluate the performance of the *clustering with hybrid CS* proposed in this paper: the *number of transmissions* which is required to collect data from sensors to the sink, and the *reduction ratio of transmissions* (reduction ratio for short) of our method compared with other methods. Four other data collection methods are considered. In the *clustering without CS* method, the same cluster structure to our method is used, but CS is not used. In the *shortest path tree (SPT) without CS*, the shortest path tree is used to collect data from sensors to the sink. In the *SPT with hybrid CS*, the shortest path tree is used to collect data from sensors to the sink, and CS is used in the nodes who has more than M descendant nodes (including itself). In the *optimal tree with hybrid CS*, a tree having minimum transmissions is used. It is computed by the greedy algorithm in [10].

In all simulations on the regular sensor field, sensor nodes are uniformly and independently distributed in a rectangle sensor field of the size 20×10 square units. A sink node is located at the corner of the sensor field. It has coordinates (0, 0). The number of nodes N varies from 400 to 1,200, then the density of nodes λ varies from 2 to 6. The transmission range r is set to $\sqrt{2}$ unit. The edge length a of small grid in our analytical model is set to 1 unit. Let $\rho = N/M$, it is called compressive ratio. ρ is set to 5 and 10, so that the projections are sufficient to recover the original data with satisfied accuracy [6], [8], [10], [11], [12]. The measurement matrix Φ and the transform basis Ψ in CS could be selected as introduced in Section 1 of the online supplemental material. These parameters have no effect on the performance evaluation of our method. Each simulation result is averaged over 50 random network topologies.

6.2 Reduction of Transmission Number

We compare our method with other methods in terms of the number of transmissions. Fig. 6 shows the number of transmissions, where the bars around the symbols on the lines represent the 95 percent confidence interval (CI). As shown in Fig. 6, the CI of our algorithms is tight. It is obvious that the number of transmissions of our method is significantly smaller than that of the clustering method without using CS. The reason is that data are compressed using the CS method at the CHs in our method. Each node on the backbone tree does M transmissions for the intercluster data gathering. It is significantly less than the number of transmissions of the method without using CS. The number of transmissions of our method is also visibly smaller than that of SPT with the hybrid CS method. This is because in the cluster structure, sensor nodes transmit data to their cluster head, which is located nearly at the center of the cluster, while in the SPT, sensor nodes transmit data to the nodes near to the sink, which results in more transmissions than our method.



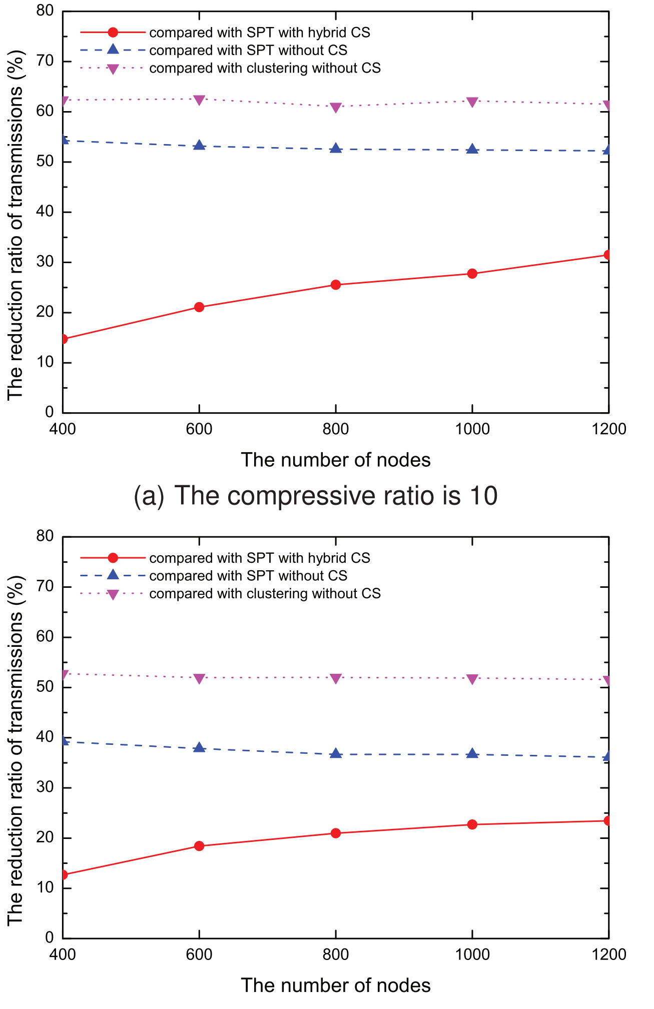
(a) The compressive ratio is 10

(b) The compressive ratio is 5

Fig. 6. The number of transmissions of data collection methods. The bars around the symbols on the lines represent the 95 percent confidence interval.

The number of transmissions of our method is slightly larger than that of the optimal tree with the hybrid CS method. However, the cluster structure can be organized in the distributed manner, while the optimal tree with hybrid CS is computed in the centralized manner. In addition, our distributed algorithm is fault tolerant. The greedy algorithm [10] iteratively computes an optimal tree with the input of network topology. The network topology may change due to the node failures or the power outage. Once the network topology changes, the resulting tree may not be energy efficient anymore. While in our distributed algorithm, the sink computes the approximate locations of central points of the cluster-areas based on the geographic area of the sensor field, instead of the network topology. Our algorithm can easily reorganize the cluster structure that has the similar quality in terms of the number of data transmissions when failures or power outage occur in the network, as discussed in Section 5.4.

Fig. 7 shows the reduction ratio of transmissions of our method compared with other methods. As shown in Fig. 7a,



(a) The compressive ratio is 10

(b) The compressive ratio is 5

Fig. 7. The reduction ratio of transmissions of clustering with the hybrid CS method compared with other methods.

when the compressive ratio is 10, our method reduces the number of transmissions by about 60 percent compared with clustering without the CS method. It reduces the number of transmissions by about 50 percent compared with SPT without the CS method. In addition, it reduces the number of transmissions by about 30 percent when the number of nodes is 1,200, compared with SPT with the hybrid CS method. The reduction ratio does not drop as the number of nodes increases. It demonstrates our method is scalable in large-scale networks. As shown in Fig. 7b, when the compressive ratio is 5, the reduction ratio of our method decreases only about 10 percent compared with the case that the compressive ratio is 10. It demonstrates that our method has significant improvements in the worst case.

6.3 Impact of the Cluster Size

In this section, we evaluate the impact of the cluster size on the performance of our method. The number of nodes is set to 1,000. The compressive ratio ρ is set to 10. In our simulations, the number of clusters C varies from 1 to 15. The cluster size is $N_c = N/C$. From the analytical model in Section 3,

$$N_c = \lambda (Da)^2, \tag{11}$$

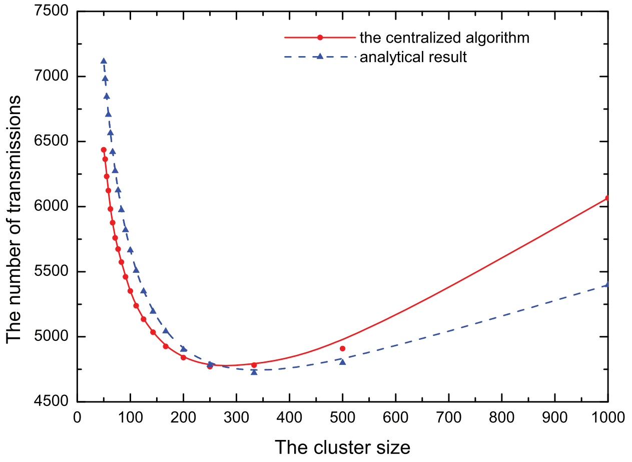


Fig. 8. The number of transmissions for different cluster size.

we get

$$D = \sqrt{\frac{N_c}{\lambda a^2}} = \sqrt{\frac{N}{\lambda a^2 C}}. \tag{12}$$

Thus, for different C or different N_c, D is calculated from (12). The number of transmissions $T(D)$ in analysis is calculated from (7).

Fig. 8 shows the change of the number of transmissions as the cluster size increases. For the curve obtained from the centralized algorithm, we make the following observation, which conforms to our analysis in Section 3. As the cluster size N_c increases, the number of transmissions decreases; when N_c reaches a certain value, the further increase of N_c would lead to the increase of transmissions.

Fig. 9 shows the change of the number of transmissions as the number of clusters increases. From the analysis in Section 3, it is known that the optimal number of clusters is 3. As shown in Fig. 9, by using the centralized algorithm, the number of transmissions of 3 clusters is near to the minimum number of transmissions.

7 Conclusion

In this paper. we used hybrid CS to design a clustering-based data collection method, to reduce the data transmissions in

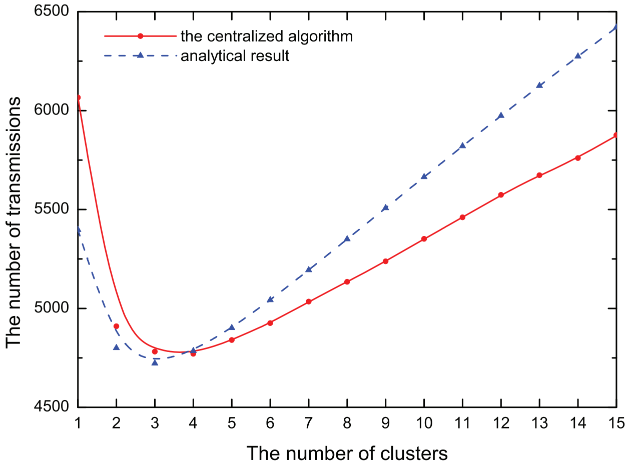


Fig. 9. The number of transmissions for different number of clusters.

wireless sensor networks. The information on locations and distribution of sensor nodes is used to design the data collection method in cluster structure. Sensor nodes are organized into clusters. Within a cluster, data are collected to the cluster heads by shortest path routing; at the cluster head, data are compressed to the projections using the CS technique. The projections are forwarded to the sink following a backbone tree. We first proposed an analytical model that studies the relationship between the size of clusters and number of transmissions in the hybrid CS method, to find the optimal size of clusters that can lead to minimum number of transmissions. Then, we proposed a centralized clustering algorithm based on the results obtained from the analytical model. Finally, we present a distributed implementation of the clustering method. Extensive simulations confirm that our method can reduce the number of transmissions significantly. When the number of measurements is 10th of the number of nodes in the network, the simulation results show that our method can reduce the number of transmissions by about 60 percent compared with clustering method without using CS. Meanwhile, our method can reduce the number of transmissions up to 30 percent compared with the data collection method using SPT with the hybrid CS. Even for the nonhomogenous networks in the irregular sensor field, our method can significantly reduce data transmissions compared with these data collection methods.

Acknowledgments

This work was supported by NSF China (Grant No. 61173137).

References

[1] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04)*, pp. 214-226, Nov. 2004.

[2] E. Candes and M. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21-30, Mar. 2008.

[3] R. Baraniuk, "Compressive Sensing [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118-121, July 2007.

[4] D. Donoho, "Compressed Sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[5] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed Sensing for Networked Data," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92-101, Mar. 2008.

[6] C. Luo, F. Wu, J. Sun, and C.W. Chen, "Compressive Data Gathering for Large-Scale Wireless Sensor Networks," *Proc. ACM MobiCom*, pp. 145-156, Sept. 2009.

[7] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega, "Spatially-Localized Compressed Sensing and Routing in Multi-Hop Sensor Networks," *Proc. Third Int'l Conf. GeoSensor Networks (GSN '09)*, pp. 11-20, 2009.

[8] C. Luo, F. Wu, J. Sun, and C.W. Chen, "Efficient Measurement Generation and Pervasive Sparsity for Compressive Data Gathering," *IEEE Trans. Wireless Comm.*, vol. 9, no. 12, pp. 3728-3738, Dec. 2010.

[9] J. Luo, L. Xiang, and C. Rosenberg, "Does Compressed Sensing Improve the Throughput of Wireless Sensor Networks?" *Proc. IEEE Int'l Conf. Comm (ICC)*, pp. 1-6, May 2010.

[10] L. Xiang, J. Luo, and A. Vasilakos, "Compressed Data Aggregation for Energy Efficient Wireless Sensor Networks," *Proc. IEEE Sensor, Mesh, and Ad Hoc Comm. and Networks (SECON '11)*, pp. 46-54, June 2011.

[11] F. Fazel, M. Fazel, and M. Stojanovic, "Random Access Compressed Sensing for Energy-Efficient Underwater Sensor Networks," *IEEE J. Selected Areas Comm.*, vol. 29, no. 8, pp. 1660-1670, Sept. 2011.

[12] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data Gathering in Wireless Sensor Networks through Intelligent Compressive Sensing," *Proc. IEEE INFOCOM*, pp. 603-611, Mar. 2012.

[13] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang, "Sparse Target Counting and Localization in Sensor Networks Based on Compressive Sensing," *Proc. IEEE INFOCOM*, pp. 2255-2263, Apr. 2011.

[14] E. Candes, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," *IEEE Trans. Information Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.

[15] E. Candes and T. Tao, "Near-Optimal Signal Recovery from Random Projections: Universal Encoding Strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.

[16] J. Tropp and A. Gilbert, "Signal Recovery from Random Measurements via Orthogonal Matching Pursuit," *IEEE Trans. Information Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[17] M. Youssef, A. Youssef, and M. Younis, "Overlapping Multihop Clustering for Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1844-1856, Dec. 2009.

[18] S. Soro and W.B. Heinzelman, "Cluster Head Election Techniques for Coverage Preservation in Wireless Sensor Networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 955-972, 2009.

[19] O. Younis, M. Krunz, and S. Ramasubramanian, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," *IEEE Network*, vol. 20, no. 3, pp. 20-25, May/June 2006.

[20] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Comm.*, vol. 1, no. 4, pp. 660-670, Oct. 2002.

[21] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.

[22] S. Bandyopadhyay and E. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proc. IEEE INFOCOM*, vol. 3, pp. 1713-1723, Mar. 2003.

[23] D. Wang, L. Lin, and L. Xu, "A Study of Subdividing Hexagon-Clustered WSN for Power Saving: Analysis and Simulation," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1302-1311, Sept. 2011.

[24] S. Chen, Y. Wang, X.-Y. Li, and X. Shi, "Data Collection Capacity of Random-Deployed Wireless Sensor Networks," *Proc. IEEE GLOBECOM*, pp. 1-6, Dec. 2009.

[25] K. Yedavalli and B. Krishnamachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 1, pp. 81-94, Jan. 2008.

[26] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02)*, pp. 105-111, 2002.

[27] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering Data Streams: Theory and Practice," *IEEE Trans. Knowledge and Data Eng.*, vol. 15, no. 3, pp. 515-528, May/June 2003.

[28] K. Jain and V.V. Vazirani, "Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation," *J. ACM*, vol. 48, no. 2, pp. 274-296, Mar. 2001.

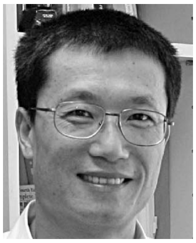
[29] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local Search Heuristic for k-Median and Facility Location Problems," *Proc. Thirty-Third Ann. ACM Symp. Theory of Computing (STOC '01)*, pp. 21-29, 2001.

[30] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[31] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, pp. 153-181, Kluwer Academic Publishers, 1996.

[32] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. IEEE Second Workshop Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90-100, Feb. 1999.

Ruitao Xie received the BEng degree from Beijing University of Posts and Telecommunications in 2008. She is currently working toward the PhD degree in the Department of Computer Science at the City University of Hong Kong. Her research interests include wireless sensor networks, cloud computing, and distributed systems.



Xiaohua Jia received the BSc and MEng degrees from the University of Science and Technology of China, Hefei, in 1984 and 1987, respectively, and the DSc degree in information science from the University of Tokyo, Bunkyo, Japan, in 1991. He is currently a chair professor in the Department of Computer Science at the City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. He is an editor of the *IEEE Transactions on Parallel and Distributed Systems* (2006-2009), *Wireless Networks, Journal of World Wide Web, Journal of Combinatorial Optimization,* and so on. He is the general chair of ACM MobiHoc 2008, TPC cochair of IEEE MASS 2009, area chair of IEEE INFOCOM 2010, TPC cochair of IEEE GlobeCom 2010—Ad Hoc and Sensor Networking Symposium, and panel cochair of IEEE INFOCOM 2011. He is fellow of the IEEE Computer Society.