IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 2, 2019, accepted June 19, 2019, date of publication June 26, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924919

A Type of Energy-Balanced Tree Based Data Collection Strategy for Sensor Network With Mobile Sink

CHAO SHA, DANDAN SONG, RUI YANG, HANCHENG GAO, AND HAIPING HUANG, (Member, IEEE)
School of Computer Science, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Corresponding author: Chao Sha (shac@njupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672297, Grant 61872194, and Grant 61872196, in part by the Jiangsu Natural Science Foundation for Excellent Young Scholar under Grant BK20160089, in part by the Six Talent Peaks Project of Jiangsu Province under Grant JNHB-095), in part by the "333" Project of Jiangsu Province, Qing Lan Project of Jiangsu Province, Innovation Project for Postgraduate of Jiangsu Province under Grant SJCX18_0295, and in part by the 1311 Talents Project of Nanjing University of Posts and Telecommunications.

ABSTRACT In order to improve the sensing efficiency of the sensor network, a type of energy-balanced tree-based data collection strategy with mobile Sink (ETDC) is proposed. Based on the minimum cost data gathering tree, several sub-trees with similar scales and small differences in energy consumption are built for data gathering. Moreover, the roots of these sub-trees are regarded as the traversal points, and the moving trajectory of Sink is obtained with the help of these points. In addition, to further reduce energy consumption on communication and the end-to-end delay, roles of some traversal nodes and relay nodes are exchanged. The experimental results show that, compared with the typical mobile Sink based data collection methods such as VGDD and VGDR, ETDC performs well in reducing path length, enhancing the success rate of data transmission, and prolonging the network lifetime.

INDEX TERMS Energy-balanced tree, energy hole, mobile data collection, traversal paths.

I. INTRODUCTION

As a bridge between nodes and terminals, it is self-evident that the Sink with powerful sensing and communicating ability is indispensable in Wireless Sensor Networks (WSNs). In traditional way, one static Sink is located at the center of the network to collect data by one-hop or multi-hop data uploading mode [1]–[4]. However, in the former mode, energy consumption on nodes is higher. For example, in a cluster based sensor network, all nodes send data to their Cluster Headers (CHs) that increases the burden of these CHs as well as the unbalance of energy consumption. In multi-hop data uploading way, it is no guarantee of real-time and reliability. Moreover, nodes near the center of the network may be overloaded that causes the "energy hole" problem [5], [6]. All of these seriously hinder the development of this technology.

With the advent of the era of big data and mobile interaction, it is more and more necessary to improves the efficiency of multi-types of data collection in WSNs. For example, the interactive application of Unmanned Aerial Vehicle (UAV) [7], [8] is really a spatial data collection system based on single or multi-mobile Sinks. Another application example was illustrated in [9], sensor nodes are deployed along a highway to sense some statistical data related to traffic jam and then the line is patrolled by a mobile vehicle to collect this data efficiently. Similar to traditional WSNs, energy efficiency and time limitation are still the focus of research in the field of mobile data collection in sensor networks [10]. Xie *et al.* [11] have systematically analyzed the constraints on the expected energy consumption of nodes, packet loss rate, end to end delay, and network lifetime. Based on these constraints, an adaptive duty cycle optimization scheme by residual energy perception has then been proposed. He *et al.* [12] have focused on the correlation of measurement data and have proposed a series of schemes about routing, power control and medium access control. In these schemes,

The associate editor coordinating the review of this manuscript and approving it for publication was Tie Qiu.

values of the collected data have been greatly improved with the help of heuristic distributed algorithm. Moreover, Gai *et al.* [13] have proposed a dynamic energy-aware cloudlet-based mobile cloud computing model (DECM) focusing on solving the additional energy consumptions during the wireless communications by leveraging dynamic cloudlets (DCL)-based model. They have also proposed a model that is called Energy-Aware Heterogeneous Cloud Management (EA-HCM) model to realize heterogeneous task assignment [14]. This effectively reduces the total energy cost of the mobile heterogeneous embedded systems.

However, with the deep application of WSNs, the contradiction between the increasing network scale and the real-time performance of data interaction is becoming more and more prominent. Therefore, how to optimize the trajectory of Sink to further reduce energy consumption on nodes is the key to improve the efficiency of data collection.

II. RELATED WORK

Francesco *et al.* have already pointed out that, the energy of node is rather limited and the communication overhead is proportional to the transmission distance [15]. Thus, using one or more mobile data collectors in the whole network can not only effectively prolong network lifetime, but also reduce the rate of packet loss in the process of data uploading. Based on this, Yang *et al.* have focused on the real time problem of data collection and proposed a TSP based mobile Sink path planning method while the balance of energy consumption and data collection delay are the constraints [16]. Moreover, in [17], the authors have introduced a type of path length constrained single-hop data collection strategy in WSNs. Although it reduces the energy consumption of nodes, it can only be applied to small scale network due to its single-hop transmission mode. Similarly, Guo *et al.* propose a random compression based data collection scheme for the defect that nodes need to periodically broadcast their own information such as residual energy, coordinates and so on [18]. By probability calculation, some nodes are transformed into coordinators. Other nodes upload their data to these coordinators and then the compressed and aggregated data are sent to the random walking Sink. In general, the mobile modes of Sink in WSNs are divided into three types: random moving, moving along the fixed path and controllable moving.

A. RANDOM MOVING STRATEGY

In this strategy, the speed and direction of the mobile Sink are all random, and the sensing nodes often send data by the way of ‘‘opportunistic routing’’. The advantage of this is to reduce the overhead of path planning, but Sink cannot use the actual location of nodes as well as the content of the data packets to carry out targeted data collection.

Shi *et al.* have proved that, in random moving strategy, the network lifetime of WSNs can get the best polynomial solution [19]. Therefore, a large number of these strategies aim at prolonging network lifetime and optimizing the energy consumption of nodes. A random moving algorithm is designed by Jain *et al.* to make the Sink communicate with all nodes in the network with equal probability [20]. Although there is no complex mobile path planning process, it cannot ensure the fairness and real-time performance of data collection. On the other hand, in the ''Virtual Circle Combined Straight Routing Strategy'' [21], cluster headers are selected out with the help of the location of nodes near the virtual area center. And then, the periodic data collecting process is carried out according to the mobile characteristic of Sink as well as the spanning tree. The goal of this algorithm is to reduce the cost of rebuilding links and increase the success rate of data transmission. However, a large number of nodes' participation in routing selection will undoubtedly increase network load. In addition, to save the energy of high load nodes as far as possible, Lee *et al.* have established a mixed linear programming model with the help of Sink's initial location and routing cost [22]. On this basis, a type of greedy maximum residual energy algorithm as well as a distributed random trajectory selection strategy have then been proposed. However, the overhead on computing is slightly larger.

B. MOVING ALONG THE FIXED PATH

In this mode, Sink moves with a constant speed along a predetermined or generated path to carry out data collection. In general, this kind of strategy is mainly applied to the uniformly distributed network.

In [23], a delay constrained data gathering method based on fixed moving path has been introduced. Each node uploads data in a multi-hop way during its pre computed time sequence. Although it ensures no conflict and no packet loss, it is easy to cause uneven energy consumption. Based on this, Han *et al.* have proposed a type of mobile data collection strategy with the help of the minimum wiener index spanning tree [24]. Multiple mobile Sinks autonomously move along the shortest path to collect data. However, the cooperative processing mechanism between Sinks is rather complex.

Different from the above methods, TPDG (Tour Planning for mobile Data-Gathering) is another kind of moving strategy based on traveling salesman algorithm [25]. The mobile Sink gets the location of nodes by sending ‘‘Hello’’ messages and uses the spanning tree covering algorithm to find out the traversal points. However, time and space overhead on location calculation in TPDG is larger. Furthermore, a low latency data collection algorithm based on adaptive stopping times have then been proposed by Kinalis *et al.* Sink moves back and forth along the boundary of each sub-region to collect data in single hop mode. However, the length of this moving path is a little long [26]. In addition, Wang*et al.* have also put forward a type of periodic data collection method based on deterministic election [27]. The mobile Sink moves along a predetermined path back and forth which effectively alleviates the energy hole problem [28]. It runs well in nonuniform distributed small networks, but nodes near the moving trajectory die quickly in large-scale networks. That is because the selected traversal points are always fixed.

C. CONTROLLABLE MOVING STRATEGY

In this kind of strategy, the next moving direction of Sink is calculated out with the help of the real-time information of the network and the real-time state of nodes. Meanwhile, the velocity of Sink is also adjusted to ensure the optimality of one of its indicators.

To balance energy consumption and end-to-end delay in data collection, Gao and Zhang have designed a type of trajectory generation algorithm based on priority of virtual traversal point [29]. However, in order to generate the accurate trajectory, each sensor node needs to continuously update the current location of Sink, which consumes more energy.

On the other hand, ‘‘cost balance’’ is one of the main constraints of ‘‘controllable’’. Therefore, a type of Load Balanced Data Gathering algorithm (LBDG) in the cluster based network have been proposed by Zhao *et al.* [30]. Different from traditional clustering methods, in LBDG, multiple cluster headers are selected out in each cluster. Then, by cooperating with each other, these cluster headers communicate with the mobile Sink in a MIMO way. In addition, using some Rendezvous Points (RPs) to generate the moving trajectories is one of the commonly used methods at present. For example, in [31], Salarian *et al.* have described a type of controllable moving path selection algorithm based on weighted rendezvous planning. To reduce the load of relay nodes in multi-hop transmission, rendezvous points are selected out according to the weight that does not exceed the maximum cost. The shortest path is then constructed by using the traveling salesman algorithm. By moving along this shortest path, Sink can traverse all these RPs, which effectively reduces time delay on data transmission.

On the basis of the above researches, we combine the advantages of these moving strategies and put forward a type of Energy-balance Tree based Data Collection strategy (ETDC). The contributions of this paper can be concluded as follows.

Firstly, to minimize the energy consumption on communication of the whole network, a data collection tree is constructed layer by layer. On this basis, this tree is divided into some sub-trees with almost the same size, which ensures that there is a considerable number of traversal points exist in the network, which effectively balances the workload on different nodes during data uploading.

Secondly, by limiting the length of the Sink’s moving trajectory, the balance between the sizes of the sub-trees and their quantity is achieved. Thus, the burden on the root nodes of the sub-trees is reduced to the greatest extent, and the real-time performance of data collection is effectively enhanced.

At last, to prolong the network lifetime, we further adjust the relationship between several nodes in some sub-trees. Moreover, the role of a few traversal nodes are also changed according to the state of the network. In this way, the efficiency of data uploading is improved to some extend.

III. NETWORK MODEL

A. MINIMUM COST DATA GATHERING TREE

As mentioned before, in a static sensor network, the Sink with fixed position is often taken as the root of a data gathering tree. In this data uploading mode, total energy consumption of the whole network is minimal if and only if the Sink is located at or near the center of the network [1], [6], [7]. Thus, in ETDC, the node which is closest to the network center is selected as the root to construct a data gathering tree layer by layer. However, according to the conclusion proposed by Heinzelman [30], energy consumption on one hop data transmission is proportional to the square or four square of the distance. So, in a tree-based sensor network, nodes which are close to the root may easily exhaust their energy and then the network is disconnected. Therefore, the balance between the load on nodes and the single hop transmission distance is important in constructing the data gathering tree.

The experimental results of [20] and [23] have proved that in a tree-based WSNs, when the direct children of root are close to it, total energy consumption on data transmission can decline significantly. In this case, these direct children upload data sensed by all nodes during a round of data collection. For this reason, the direct child which is closest to the root has the highest priority to select its child nodes. Nodes which are located in its communication range and which have not yet joined to this tree become its direct children. Then, the direct child that is next-closest to the root selects its direct children in the same way. When all nodes in the same layer have finished the ‘‘direct children selection tasks’’, nodes in the next layer of the tree start to perform the same operation similarly according to their priority.

Nevertheless, this method only takes into account the distance from the current node to its direct parent (the next hop node) and it does not pay attention to the cost of the entire data uploading path. It easily falls into local optimal and the energy holes appear inevitably. On the other hand, for a node already in the data gathering tree, it is not a wise way to select all nodes which are located in its communication range as its direct children. In this case, the hop distance between one node and its parent node may be too long. Therefore, the following improvements are described as follows.

Step 1: The node which is closest to the geometric center of a network is regarded as the root of the data gathering tree. If there are more than one node satisfies this condition, we randomly select one of them as the root that is also defined as the first layer node.

Step 2: The root then broadcasts packets to all nodes within its one hop range. The node that receives the packet is regarded as the direct child of the root (the second layer node) and sends an acknowledge packet to it. Thus, these nodes join in the data gathering tree.

Step 3: We use k to represent the height of the data gathering tree at this moment. Then, nodes in the k_{th} layer broadcast their packets to all nodes within their one hop range in order to make more nodes join in the tree.

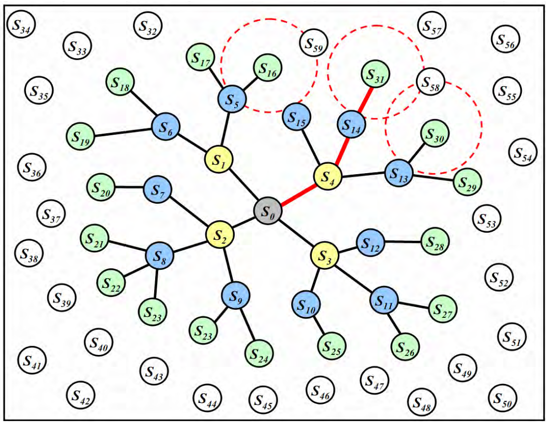


FIGURE 1. **Nodes join in the data gathering tree layer by layer.**

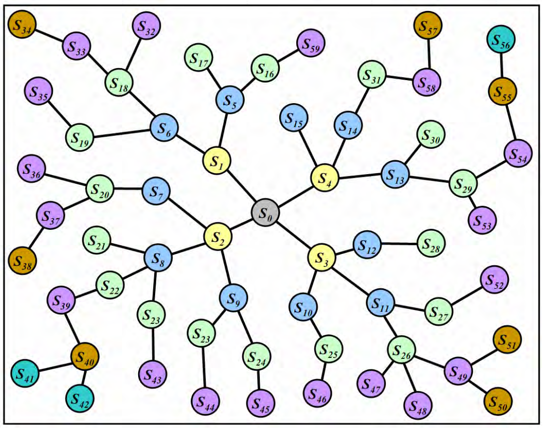


FIGURE 2. **The minimum cost based data gathering tree.**

Step 4: For a node S_j which does not yet join in the tree (e.g., S_{59} in Figure 1), if it can only receive the message from S_i (e.g., S_{16} in Figure 1), S_j is then regarded as the direct child of S_i. On the contrary, if S_j (e.g., S_{58} in Figure 1) can receive message from more than one nodes, it calculates the weight of data transmission between these nodes and itself (marked as $W(S_i,S_j)$) separately according to formula (1). Then, the node with the smallest value of $W(S_i,S_j)$ is regarded as the direct parent node of S_j. So, S_j sends an acknowledge packet to S_i and then join in the data gathering tree.

$$W(S_i, S_j) = W'(S_i) \times dis(S_i, S_j) \tag{1}$$

In formula (1), S_i is named as the "candidate direct parent node" of S_j and $dis(S_i,S_j)$ is defined as the Euclidean distance between S_i and S_j. We do not need to know the specific geographic coordinates of nodes, but only calculate the distance between them. In fact, in the process of building the data collection tree, nodes need to frequently communicate with each other. At this time, each node can get the signal strength of its neighbor, and so the distance between them can be calculated out.

$W'(S_i)$ is the "candidate weight" and the value of it is calculated by equation (2).

$$W'(S_i) = \alpha \times sum\_dis\,(S_i, S_0) + (1-\alpha) \times dis\,(S_i, Parent\,(S_i)) \tag{2}$$

In the data gathering tree, the total length of the path from S_i to the root S_0 (e.g., the black rough line in Figure 1) is defined as $sum\_dis(S_i,S_0)$. "$Parent(S_i)$" represents the direct parent node of S_i, and α is an adjustable parameter whose value is between 0 and 1.

Step 5: After all the qualified nodes join in the tree, we make $k=k+1$. Then, step 3,4,5 are implemented again until all nodes join in the tree. Thus, the minimum cost based data gathering tree is constructed, as shown in Figure 2.

B. TRAVERSAL POINT SELECTION

In WSNs, the location of traversal points and the trajectory of the Sink are important for the efficiency of mobile data collection. Therefore, in this section, we take the duration of the data collection cycle as well as the length of the moving path as two constraints and divide the minimum cost based data gathering tree into several sub-trees. Thus, the optimal traversal points can be selected out and the cost on data uploading is also reduced. Definitions and constraints are described as follows.

1) TRAVERSAL NODES

Nodes that can communicate with Sink in one hop.

2) A ROUND OF DATA COLLECTION TIME

The total time spending on communicating with each traversal node as well as traversing them once. Without loss of generality, it is defined as T. Moreover, the velocity of Sink is v, and each sensor node collects k bit of data in T.

3) THE UPPER LIMIT VALUE OF T

It is necessary to ensure the freshness of data in a high real-time requirement application scenario. In ETDC, the maximum value of T is defined as T_{th}. Thus, in a round of data collection time, the maximum length of the path that the Sink moves along is $T_{th} \times v$.

The process of dividing the data gathering tree into sub-trees starts from the root, S_0. At first, S_0 broadcasts a message whose content is "tree division". Each of the direct subnodes receives the message and calculates the value of $P(S_i)$ according to formula (3). Then, the node with the maximum value of $P(S_i)$ is no longer the child of its parent (e.g., S_2 in Figure 2). Hence, the data gathering tree is divided into two sub-trees, and the roots of them are S_0 and S_i, respectively (Figure 3).

$$P\,(S_i) = \frac{Num\_t\,(Parent\,(S_i)) \times Num\_t\,(S_i)}{D\,(S_0, \ldots, S_k, S_i) - D\,(S_0, \ldots, S_k)} \tag{3}$$

In equation (3), "$Num\_t(Parent(S_i))$" represents the total number of children of $S_i's$ direct parent. It is not difficult to know that, to balance energy consumption, the differences between the sizes of each sub-tree should be as small as possible. In addition, to further reduce the burdens on

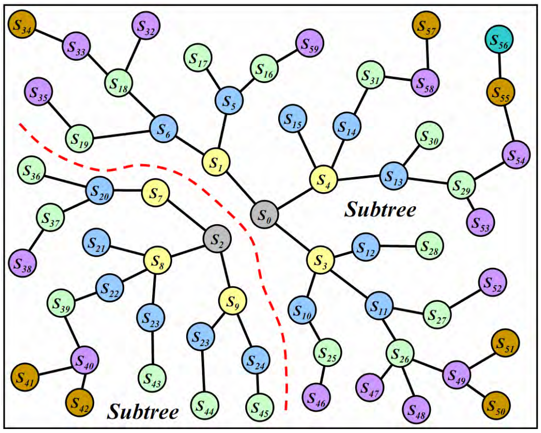


FIGURE 3. The data gathering tree is divided into sub-trees.

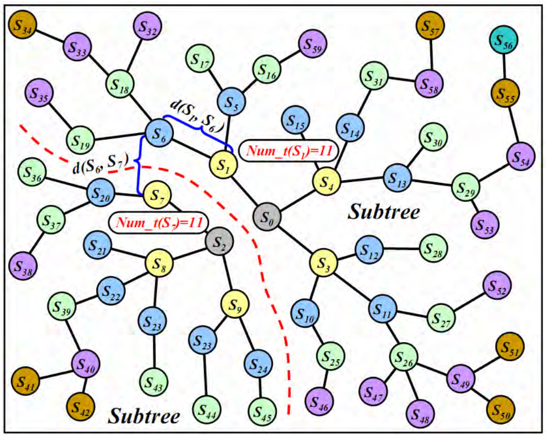


FIGURE 4. An example about calculating the value of *Num_t(S_i)*.

traversal nodes, there should be more sub-trees in the network. So, the priority of S_i is in proportion to the value of *Num_t(Parent(S_i))*.

Num_t(S_i) is defined as the possible value of the total number of children of S_i. If there exists a neighbor node of S_i (e.g., S_j) in the current data gathering tree and if the following two constraints are satisfied simultaneously, S_j is temporarily considered as a direct child of S_i. Thus, the sub-tree whose root is S_j temporarily join in the sub-tree whose root is S_i. That is, *Num_t(S_i)* =*Num_t(S_i)*+*Num_t(S_j)*.

- S_j is not the child of S_i, and it does not calculate the priority at this moment.
- The distance between S_j and S_i is shorter than the distance between S_j and its direct parent.

If there is no such node, the value of *Num_t(S_i)* is unchanged.

Up to now, there are two data gathering sub-trees in the network, as shown in Figure 4. Then, each of the direct children of S_0 and S_2 (the yellow nodes in Figure 4) calculates its $P(S_i)$ to see whether or not it is possible to become the root of a new sub-tree. Assuming that $d(S_6, S_7)$ is smaller than $d(S_1, S_6)$, nodes in the sub-tree whose root is S_6 are all regarded as the children of S_7. So, the value of *Num_t(S_7)* is 11.

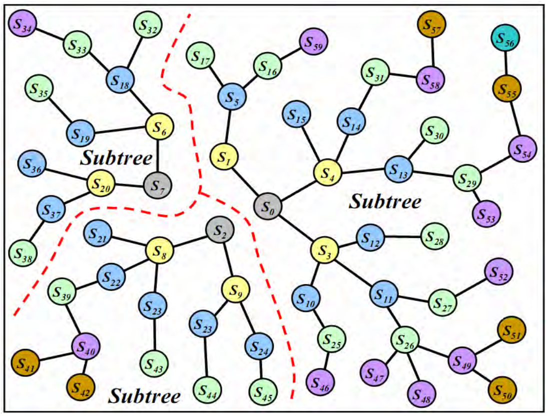


FIGURE 5. Parent node reselection.

After comparing the value of $P(S_i)$, if node S_i (e.g., S_7 in Figure 5) become the root of a new sub-tree, S_j (e.g., S_6 in Figure 5) then disconnects with its original parent node (e.g., S_1 in Figure 5) and takes S_i as its direct parent.

With the help of this traversal point selection algorithm, network topology is optimized and the communication distance between nodes is also shortened which effectively reduces energy consumption on data collection.

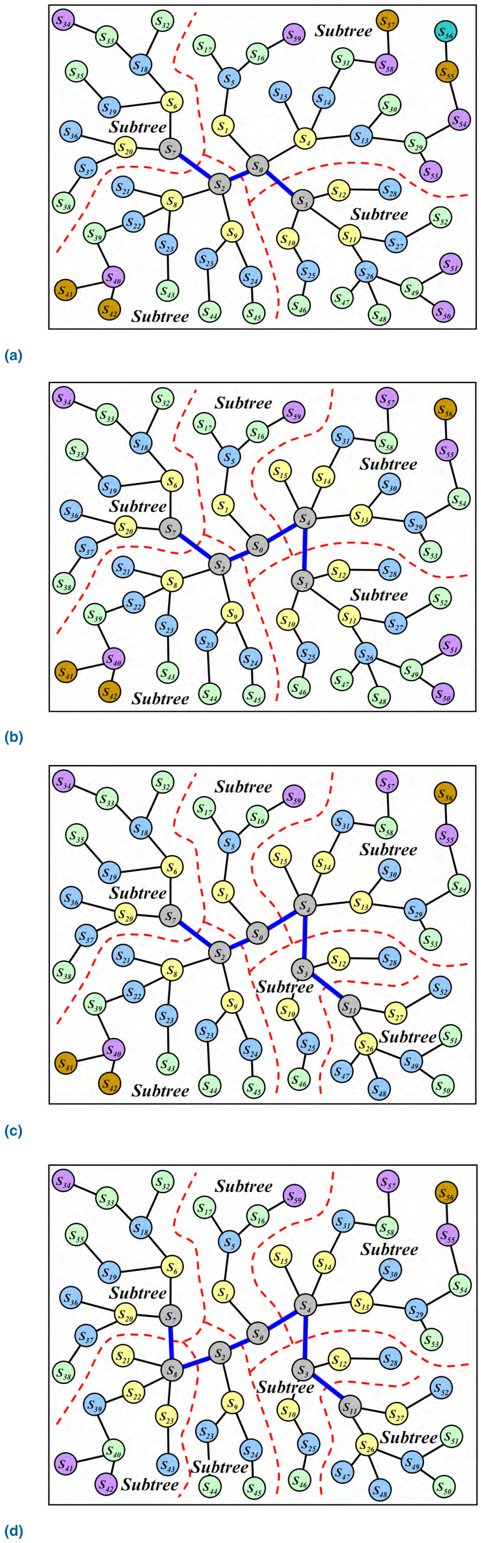
It is worth mentioning that, in formula (3), $D(S_0, \ldots, S_k)$ is defined as the minimum value of length of the moving path that is made up of all of the sub-tree roots (assuming that there are now *k+1* sub-trees in the network, and the roots of them are $S_0, S_1, \ldots, S_k$). Similarly, $D(S_0, \ldots, S_k, S_i)$ represents the minimum value of length of the new path when node S_i is added into the path. Thus, to reduce and balance energy consumption, it hopes that there are as many as possible sub-trees in the network. Moreover, in order to meet the requirement of the upper limit value of T_{th}, S_i should not be regarded as a sub-tree root if $D(S_0, \ldots, S_k, S_i) > l$.

An example about sub-tree construction and moving path (the blue solid line) generation is shown in Figure 6.

C. MOVING PATH OPTIMIZATION BASED ON CURVE FITTING

As mentioned above, there are now several sub-trees in the network and the length of the moving path is close to l. However, a lot of researches show that when collecting data, the Sink does not need to move to the positions where the traversal nodes located at. It just need to make sure that the Sink can directly communicate with each node. Therefore, we adopt the polynomial curve fitting by using least square method to further optimize the moving trajectory of the Sink. Assuming that there are *k+1* traversal nodes in the network. Thus, the curvilinear equation of the path is expressed as equation (4).

$$y = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1} + a_k x^k \tag{4}$$



(a)

(b)

(c)

(d)

FIGURE 6. **Sub-tree construction and trajectory generation.**

On the other hand, the sum of squared distances between these traversal nodes to the trajectory is

$$d^2=\sum_{i=0}^{k}\left((x_i-x)^2+\left(y_i-\left(a_0+a_1x+\cdots+a_kx^k\right)\right)^2\right) \quad (5)$$

Then, the partial derivative of $a_0, a_1, \ldots, a_k$ are calculated as follows.

$$2\sum_{i=0}^{k}\left(y_i-\left(a_0+a_1x+\cdots+a_kx^k\right)\right)=0 \quad (6)$$

$$2\sum_{i=0}^{k}\left(y_i-\left(a_0+a_1x+\cdots+a_kx^k\right)\right)x=0 \quad (7)$$

$$\cdots$$

$$2\sum_{i=0}^{k}\left(y_i-\left(a_0+a_1x+\cdots+a_kx^k\right)\right)x^k=0 \quad (8)$$

Formula (6), (7) and (8) are transformed as follows.

$$a_0k+a_1\sum_{i=1}^{k}x_i+\ldots+a_k\sum_{i=1}^{k}x_i^k=0 \quad (9)$$

$$a_0\sum_{i=1}^{k}x_i+a_1\sum_{i=1}^{k}x_i^2\ldots+a_k\sum_{i=1}^{k}x_i^{k+1}=0 \quad (10)$$

$$\cdots$$

$$a_0\sum_{i=1}^{k}x_i^k+a_1\sum_{i=1}^{k}x_i^{k+1}\ldots+a_k\sum_{i=1}^{k}x_i^{2k}=0 \quad (11)$$

From formula (9) and (11), the following matrix can be obtained.

$$\begin{bmatrix} k & \sum_{i=1}^{k}x_i & \cdots & \sum_{i=1}^{k}x_i^k \\ \sum_{i=1}^{k}x_i & \sum_{i=1}^{k}x_i^2 & \cdots & \sum_{i=1}^{k}x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{k}x_i^k & \sum_{i=1}^{k}x_i^{k+1} & \cdots & \sum_{i=1}^{k}x_i^{2k} \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}=\begin{bmatrix} \sum_{i=1}^{k}y_i \\ \sum_{i=1}^{k}x_iy_i \\ \vdots \\ \sum_{i=1}^{k}x_i^ky_i \end{bmatrix} \quad (12)$$

Hence, the value of the coefficient matrix $[a_0, a_1, \ldots, a_k]^T$ could be calculated out and the equation of the trajectory after fitting can also be solved out.

D. TRAVERSAL NODE ADJUSTMENT

It is not difficult to know that, after optimizing the moving path, there may be two cases about the distance between the traversal node and the trajectory.

Case 1: The distance between the sub-tree root and the traversal node is larger than r_t, which is the maximum communication range of a sensor node (although it is unlikely to happen). In this case, the sub-tree root is unable to transmit data to the mobile Sink.

Case 2: Due to the randomness of distribution, some non-root nodes may be near to this trajectory. In this case, they are able to communicate with the mobile Sink directly.

Therefore, we should further make some non-root nodes be the traversal nodes. Correspondingly, the unsuitable roots should not be regarded as the traversal nodes.

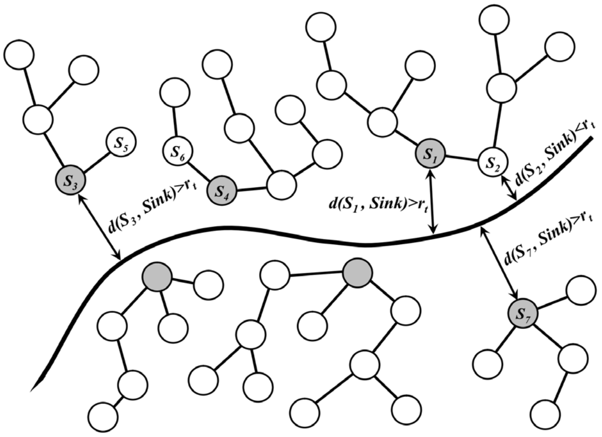


FIGURE 7. The data collection sub-tree before adjustment.

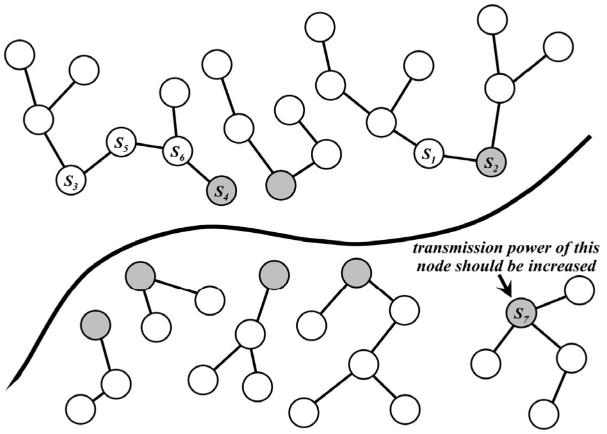


FIGURE 8. The data collection sub-tree after adjustment.

The traversal node adjustment strategy is described as follows. For convenience, $d(S_i, path)$ is defined as the shortest distance between S_i and Sink' s moving trajectory

Step 1: As mentioned above, for the sub-tree root S_i, if $d(S_i, path)>r_t$, it shows that S_i is unable to upload data to Sink. For this sub-tree, the node (e.g., S_j) with the smallest value of $d(S_j, path)$is selected out.

Step 2: If $d(S_j, path)\leqq r_t$, S_j is regarded as the new root of this sub-tree, otherwise we go to step 3. Other nodes (include S_i) in this sub-tree then establish a single-hop or multi-hop connection with S_j in turn according to the algorithm described in section 3.1. After doing that, it jumps to step 4.

For example, in Figure 7, the shortest distance between the sub-tree root S_1 and the moving path is larger than r_t. So, the structure of this sub-tree must be adjusted. It is easy to see that, only node S_2 satisfies the constraint $d(S_2, path)\leqq r_t$. Thus, S_2 becomes the new root of the sub-tree, and the restructured sub-tree is shown in Figure 8.

Step 3: For each node in this sub-tree (e.g., S_k), it assumes that S_l is the neighbor of S_k and it is not in the current sub-tree. If $d(S_l, path)\leqq r_t$, we calculate the value of $W''(S_l)$ according to formula (13). The node with the maximum value of $W''(S_l)$ is then regarded as the new root of this sub-tree.

$$W''(S_l) = E_r(S_l)/(dis(S_k, S_l) + dis(S_l, path)) \quad (13)$$

In Figure 7, it is shown that $d(S_3, path)>r_t$. It means the shortest distance between the sub-tree root and the trajectory is still longer than the communication radius of the sensor node. Meanwhile, in this sub-tree, there are no other nodes that satisfy the condition $d(S_l, path)\leqq r_t$. Thus, after each node in this sub-tree calculating out their weight $W''(S_l)$, S_6(it is in another sub-tree) is selected as the new sub-tree root, as shown in Figure 8.

If there is still no node in the neighbor sub-tree satisfies the condition $d(S_l, path)\leqq r_t$, this sub-tree is regarded as the "isolated sub-tree". For example, the sub-tree whose root is S_7 in Figure 7 is an "isolated sub-tree".

In this case, we should enhance the transceiver power of the sub-tree root or relax the constraints about the length of one data collection cycle as well as the length of the moving path. Then, the ETDC algorithm should be carried out again.

Step 4: After doing step 1, 2 and 3, all nodes that satisfies the condition $d(S_i, path)\leqq r_t$ are defined as the traversal nodes (the white nodes in Figure 8). For any traversal node S_i, if it is already the root of the current sub-tree, it does not do any more. Otherwise, it disconnects with its parent node and forms a new data collection sub-tree. The purpose of this is to make full use of the nodes that can be communicate with the mobile Sink and to build more data collection sub-trees as much as possible. Thus, the load on nodes can be reduced and their energy consumption is also balanced.

E. SUMMARY OF CORE ALGORITHMS IN ETDC

As mentioned above, there are three phases in the ETDC algorithm. The first one is the "***Minimum cost Data Gathering Tree Construction***" (Algorithm 1). Steps of this algorithm is summarized as follows.

Step 1: The node which is closest to the geometric center of a network is regarded as the root of the data gathering tree.

Step 2: The root node broadcasts packets to all nodes within its one hop range. The node that receives the packet joins in the data gathering tree.

Step 3: Nodes in the $k_{\rm th}$ layer broadcast their packets to all nodes within their one hop range.

Step 4: For a node S_j which does not yet join in the tree, if it receive the message from other nodes, it selects its direct parent node.

Step 5: We check whether all nodes have joined the data gathering tree. If so, the algorithm goes to step 5, otherwise, it goes to step 3 again.

The second phase of the proposed method is the "***Traversal Point Selection***" (Algorithm 2). Its implementation process is as follows.

Step 1: The root node of the data gathering tree broadcasts a message whose content is "tree division".

Step 2: Each of the direct sub-nodes receives the message and calculates the value of $P(S_i)$. Then, the node with the maximum value of $P(S_i)$ is no longer the child of its parent. Hence, the data gathering tree is divided into two sub-trees.

Step 3: Each of the direct children of the roots of the sub-trees calculates its $P(S_i)$. If the value of $P(S_{k+1})$ is the maximum one, we compare the value of $D(S_0, \ldots, S_k, S_{k+1})$ and l. Here, S_0, $S_1, \ldots,$ S_k are all the sub-tree root nodes. If $D(S_0, \ldots, S_k, S_{k+1}) > l$, S_{k+1} becomes the root of a new sub-tree. In addition, it disconnects with its original parent node, and then goes to step 3 again. Otherwise, the traversal point selection algorithm end.

The last phase of the proposed method is the "***Traversal Node Adjustment***" (Algorithm 3). Steps of this algorithm is summarized as follows.

Step 1: For the sub-tree root S_i, if $d(S_i, path) > r_t$, the node (e.g., S_j) with the smallest value of $d(S_j, path)$is selected out from this sub-tree.

Step 2: If $d(S_j, path) \leqq r_t$, S_j is regarded as the new root of this sub-tree. Then, other nodes in this sub-tree establish a single-hop or multi-hop connection with S_j in turn according to algorithm 1. After doing this, it jumps to step 4. If $d(S_j, path) > r_t$, we go to step 3.

Step 3: For each node in this sub-tree (e.g., S_k), it assumes that S_l is the neighbor of S_k and it is not in the current sub-tree. If $d(S_l, path) \leqq r_t$, the node with the maximum value of $W''(S_l)$ is then regarded as the new root of this sub-tree. If there is still no node in the neighbor sub-tree satisfies the condition $d(S_l, path) \leqq r_t$, this sub-tree is regarded as the "isolated sub-tree".

Step 4: For any traversal node S_i, if it is not the root of the current sub-tree, it disconnects with its parent node and forms a new data collection sub-tree.

It is worth mentioning that with the continuous running of the system, failure nodes will inevitably appear in the network, which may lead to the disconnection of the data collection sub-trees. The computational complexity of network reconstruction is analyzed as follows.

- If the dead node is just a leaf of the data collection sub-tree, it has no effect on the network topology. That is to say, the topology does not need to be reconstructed at this time.
- If the dead node is the intermediate node of a sub-tree, only those nodes that were originally its descendants need to reselect their parent. From the description of the proposed algorithm, it can be seen that the distance between nodes in the same sub-tree is short. Therefore, within the single-hop communication range, each node can probably find its new parent and reconstruct the topology by communicating with other nodes only once. Assuming that the total number of this kind of nodes is n, then the computational complexity of reconstructing the topology is $O(n)$.
- If the dead node is the root of a sub-tree, the whole data collection sub-tree will be reconstructed. Assuming that there are n alive nodes in this data collection sub-tree, and each node needs to calculate the value of $d(S_j, path)$ once. Subsequently, the new root node (S_j) reconstructs the data collection sub-tree layer by layer. In the worst case, nearly $(n\text{-}1)/(l\text{-}1)\times(n\text{-}1)/(l\text{-}1)$ times of comparisons are required for each layer (it is assumed that the sub-tree has l layers). It can be seen that the cumulative calculation of $(n\text{-}1)^2/(l\text{-}1)$ times is needed. In other words, the computational complexity of reconstructing the data collection sub-tree in this case is $O(n^2)$.
- If a large number of nodes die, all the data collection sub-trees need to be reconstructed according to the process described in Sections 3.1 and 3.2. Assuming that the total number of surviving nodes in the network is n in this case, so according to the conclusion mentioned above, the computational complexity of reconstructing the data collection tree is $O(n^2)$. On this basis, the computational complexity of reconstructing the sub-trees is at most $O(n)$ because each node only needs to calculate the value of $P(S_i)$ once according to formula (3).

IV. SIMULATION RESULTS

In order to verify the effect of ETDC on balancing energy consumption, prolonging the network lifetime and reducing time delay on data collection, simulation experiments were carried out with the help of Eclipse4.5 and Matlab8.5. All the experiments were carried on a server (the operating system is Win 10) with Intel Xeon (E3-1225V6) 3.3GHz CPU, 16GB memory(DDR4, 2400MHZ), 8MB cache and 2TB hard disk. All the algorithms were implemented via Java code. Moreover, the proposed method is also compared with two typical mobile Sink based data gathering schemes, VGDD [28] and VRDG [29]. All the data of each experiment were obtained after 100 times of simulations.

VGDD (Virtual Grid based Data Dissemination) is a grid based data collection method for Wireless Sensor Networks. A rectangular network is divided into several virtual grids. In each grid, the node closest to the center is selected as the cluster header. It is also assumed that two cluster headers in the adjacent grids can communicate with each other.

The mobile Sink moves to one of the edge grids along the inner tangential circle of the network. Then, the data collection tree is constructed and the root of it is the cluster header of this grid. Moreover, the cluster headers in other grids are regarded as the intermediate nodes or the leaf nodes of this data collection tree, as shown in Figure 9. Thus, in VGDD, the mobile Sink only needs to communicate with the cluster headers in the edge grids and the length of each grid can be adjusted dynamically according to the size of the network and the number of nodes. In this way, VGDD has a good performance on real time data collection. However, frequently constructing the data collection tree will cause more energy consumption.

Similar to VGDD, VRDG (Virtual Region based Data Gathering) is also a distributed data collection method based on mobile Sink. In this scheme, the sensing region of

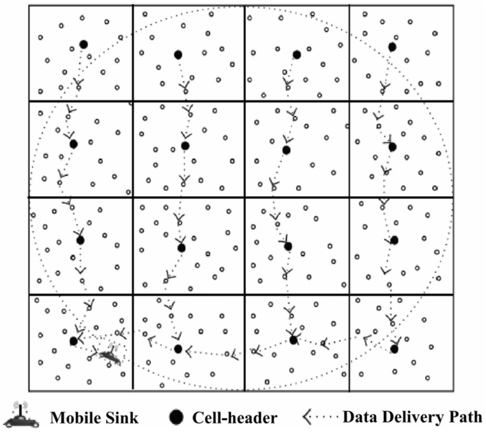


FIGURE 9. Network model of VGDD [28].

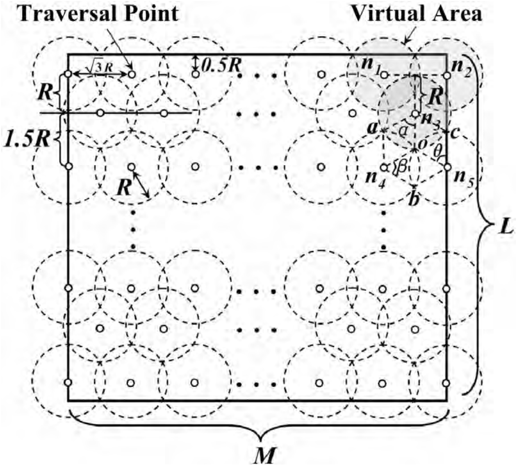


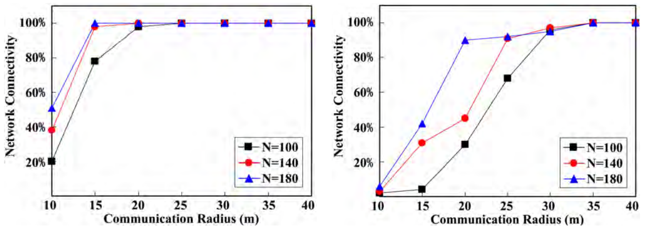
FIGURE 10. Network model of VRDG [29].

three adjacent nodes are defined as a ''virtual area'' and the traversal points are selected out from each ''virtual area'' according to the residual energy and distribution of nodes, as shown in Figure 10. Compared with the ''virtual grid'', the ''virtual area'' can basically ensure the full coverage of the network. Moreover, in VRDG, the time when the mobile Sink arrives at each ''virtual area'' can be accurately calculated out. Thus, nodes are able to go into sleeping mode until the mobile Sink is near. This greatly reduces the energy cost of the network.

It is worth mentioning that, we have also developed a type of wireless sensor node based on CC2530 module. Values of the data collection rate, the energy consumption rate on communication, the cache capacity and the unit energy consumption of the circuit are all measured in real scenes. Some of these values have been used in the simulation experiments. Without loss of generality, it is defined that *n* stationary nodes are randomly deployed in a rectangular network. The initial energy and the threshold of the residual energy of one node are set to 1.0J and 0.4J, respectively. From the network model and the implementation process of ETDC, it can be seen that the size of network, the number of nodes, the moving speed of Sink and the time interval of data packet transmission will have an important impact on the simulation results. Therefore, in the following experiments, we focus on the analysis of the experimental results under different values of these parameters.

TABLE 1. Values of parameters in experiment about network connectivity.

Parameter	Symbol	Value	Unit
Total Number of Nodes	*N*	*100, 140,*	
Moving Speed of Sink	*v*	*5*	*m/s*
Time Interval about Packet Sending	*Δt*	*5*	*s*
Communication Radius of Node	*R*	*10~40*	*m*



(a) 100m×100m Network

(b) 200m×200m Network

FIGURE 11. Network connectivity.

A. NETWORK CONNECTIVITY

In order to verify the robustness of ETDC, we first analysis the network connectivity. Values of key parameters in this experiment are shown in Table 1, and the results are shown in Figure 11.

It is not difficult to see that in the 100m×100m network, when the number of nodes is 140 or 180, there are hardly any isolated nodes in the network if and only if the communication radius of each node is larger than 15m. In this case, the density of nodes is enough to make the whole network fully connected. So, it avoids the situation that nodes cannot join the data collection tree due to channel conflict to the greatest extent (the data collection tree is constructed through layer-by-layer broadcasting and response). This can reduce the energy consumption on communication to a certain extent and enhance the efficiency of data collection.

When the network size increases to 200m×200m, even if the number of nodes is only 100, the network connectivity is still basically ensured when the communication radius is 30 meters. In ETDC, nodes are uniformly deployed and in the process of establishing a minimum cost based data collection tree, the nodes search for their neighbors as far as possible with the maximum communication radius. Therefore, nodes that exist in the connected graphs will not become the isolated nodes in ETDC, thus ensuring the adaptability of this algorithm to the actual network environment. In addition, the ETDC algorithm also makes adjustment to some traversal nodes. That is to say, the data collection sub-tree that cannot communicate with the mobile Sink can be merged with other sub-tree, or the power of the root node of this sub-tree can be increased to ensure the network be connected for a long time.

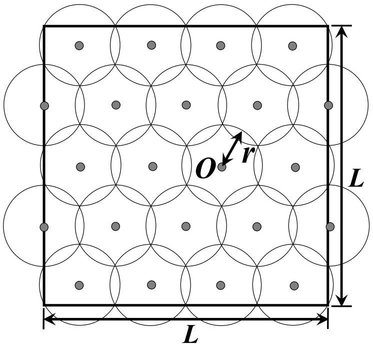


FIGURE 12. Minimum density distribution model without blind area.

When the node density is low, there will be a considerable proportion of isolated nodes in the network. This will have a negative impact on the establishment of the minimum cost based data collection tree as well as the optimization of the moving path. According to [17], under the condition of uniform deployment, when the density of nodes is less than $2/3\sqrt{3}\,r^2$, the blind area appears (Figure 12). Therefore, in the 200m×200m network, when the number of nodes is 100, whether the length of the communication radius is 10m or 15m, there are a large number of isolated nodes in the network.

B. NETWORK LIFETIME

In ETDC, the lifetime of any node can be expressed as equation (14).

$$Round\ (S_i) = \left\lfloor \frac{E_{init}\ (S_i)}{k\ (Num\_c\ (S_i)\ e_r + (Num\_c\ (S_i) + 1)\ e_t\ (S_i))} \right\rfloor \tag{14}$$

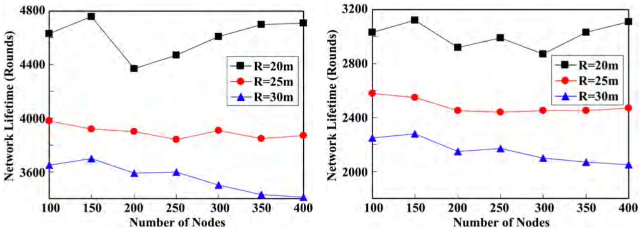
$E_{init}(S_i)$ is the initial energy of S_i, and $Num\_c(S_i)$ is defined as the total number of descendant nodes of it. e_r and $e_t(S_i)$ are the energy consumption on receiving and sending 1 bits of data, respectively. In addition, the value of e_r is equal to that of E_{elec} according to the wireless communication model proposed by Heinzelman *et al.* [30]. Moreover, the value of $e_t(S_i)$ is calculated by formula (15).

$$e\ (S_i) = \begin{cases} E_{elec} + \varepsilon_{fs} d^2 & d < d_0 \\ E_{elec} + \varepsilon_{amp} d^4 & d \geq d_0 \end{cases} \tag{15}$$

E_{elec} is the circuit loss during transmitting one bit of data and the value of it is $50nJ \times b^{-1}$. ε_{fs} and ε_{amp} are defined as the signal amplification factors of the free space and the multi-path fading environment. The values of them are $10pJ \times (b/m^2)^{-1}$ and $0.0013pJ \times (b/m^4)^{-1}$ respectively. Moreover, d_0 is the threshold of the reference distance about single-hop communication. Its value is $(\varepsilon_{fs}/\varepsilon_{amp})^{1/2}$. It is worth mentioning that, in (15), for the non-traversal node, the value of d is equal to the distance between it and its parent.However, for the traversal node, the value of d is just the distance between this node and the mobile Sink. Values of key parameters in this group of experiments are shown in Table 2.

TABLE 2. Values of parameters in experiments about network lifetime.

Parameter	Symbol	Value	Unit
Total Number of Nodes	*N*	*100~300*	
Moving Speed of Sink	*v*	*1, 3, 5*	*m/s*
Time Interval about Packet Sending	*Δt*	*1, 3, 5*	*s*
Communication Radius of Node	*R*	*20*	*m*



(a) 100m×100m Network

(b) 200m×200m Network

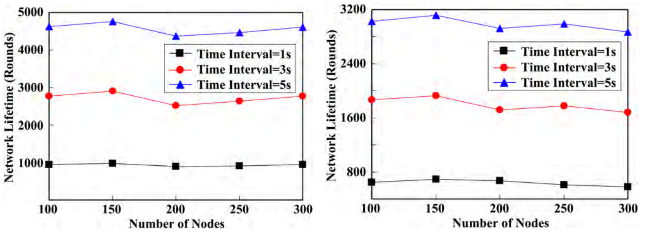
FIGURE 13. Network lifetime under different communication radius.

Figure 13 shows the network lifetime under different number of nodes and different length of communication radius. The packet generation rate of each node is fixed to 80bit/s, and the data is uploaded to its parent node every 5 seconds. Moreover, the velocity of the mobile Sink is 5m/s.

It can be seen that the network lifetime of ETDC is not changed significantly with the increase of the number of nodes when the node's communication radius is large. For example, in the 200m×200m network, when $R = 25$m, the standard deviation of network lifetime in ETDC is only 52.3 rounds. This is because there are hardly any isolated nodes in the network at this time. According to the strategies described in section 3.2 and 3.4, it is known that no matter what the number of nodes is, the size of each data collection sub-tree is basically the same. However, when the number of nodes is too large and the communication radius is longer, the network lifetime is slightly shorter because the difference between the data collection sub-trees appears.

Meanwhile, it can also be seen from Figure 13 that when the communication radius increases, the network lifetime decreases. For example, in the 100m×100m network, the average network lifetime is 4607 rounds when $R = 20$m, while in the same simulation scenario where $R = 30$m, it is only 3554 rounds. From formula (15), it is known that the energy consumption on communication is proportional to the square (or even four square) of the distance. Thus, the increase of communication radius will lead to a steep increase in energy consumption.

On the other hand, with the increase of the node's communication radius, the number of layers of the minimum cost based data collection tree decreases, and the number of nodes in the same layer will be much more. In this case, the number of traversal nodes as well as the length of the moving path will increase. This not only prolongs the data collection time of each round, but also aggravates the burden on traversal nodes and forwarding nodes. Thus, the network lifetime is shortened.



(a) 100m × 100m Network (b) 200m × 200m Network

FIGURE 14. Network lifetime under different time intervals of data collection.

However, it is necessary to point out that as ETDC specifies the upper limit of the path length, it can be seen from Figure 13 that the range of changes in the network lifetime is not significant.

Figure 14 shows the network lifetime under different time intervals of data collection (1s, 3s and 5s). Velocity of Sink in this experiment is set to 5m/s. Without loss of generality, we assume that each node samples 400 bits of data each time. The communication radius of each node is set to 20m, and the moving speed of Sink is still 5 m/s. It can be seen that the network lifetime is basically proportional to the time interval of data collection. For example, in the 200m×200m network, when the time interval for data collection is extended from 1s to 5s, the average network lifetime increases from 640 rounds to 2986 rounds accordingly. Moreover, it is obvious that when the packet generation rate is stable, the network lifetime of ETDC will not change significantly with the increase of the number of nodes. This is because the scale and energy consumption of the sub-tree is nearly the same as each other in this case. When the number of nodes increases, the total number of sub-trees increases accordingly as well. On the contrary, the load of each sub-tree does not increase significantly.

Figure 15 shows the network lifetime under different moving speeds of Sink. In this experiment, the length of the communication radius is set to 20m, and the data generation rate is 80bit/s. Moreover, time intervals of data collection in Figure 15 is 5s. Similarly, the network lifetime is basically proportional to the moving speed of Sink, regardless of the network size. Obviously, this is due to that when the speed of Sink increases, the time on completing a round of traversal is reduced. In addition, energy consumption on communication also becomes lower. Thus, the network lifetime is prolonged.

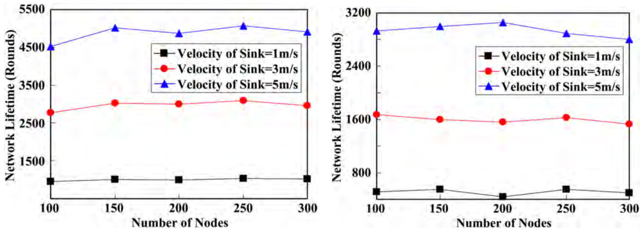
C. SUCCESS RATE OF DATA TRANSMISSION

In ETDC, the success rate of data transmission refers to the ratio of the amount of data collected by the mobile Sink to the amount of data generated by all nodes during a round of data collection. That is,

$$P_s = \sum_{i=1}^{m} Data\,(S_i)/n \times u \times (l/v) \tag{16}$$

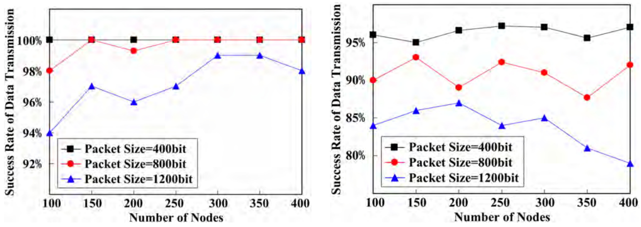
TABLE 3. Values of parameters in experiment about success rate of data transmission.

Parameter	Symbol	Value	Unit
Total Number of Nodes	*N*	*100~400*	
Moving Speed of Sink	*v*	5	*m/s*
Time Interval about Packet Sending	*Δt*	5	*s*
Communication Radius of Node	*R*	*20*	*m*



(a) 100m × 100m Network (b) 200m × 200m Network

FIGURE 15. Network lifetime under different moving speeds of Sink.



(a) 100m × 100m Network (b) 200m × 200m Network

FIGURE 16. Success rate of data transmission under different packet sizes.

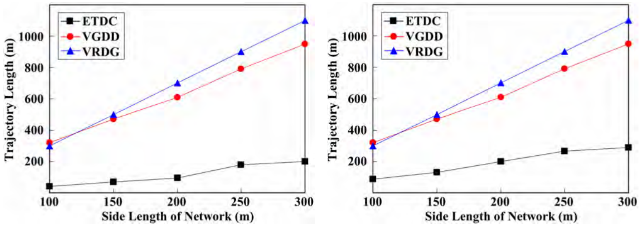
In (16), m is defined as the number of traversing nodes, and u is regarded as the rate of data generation. While, *Data(S_i)* represents the total amount of data uploaded to Sink within a round of data collection time by S_i. Thus,

$$Data\,(S_i) = \begin{cases} Cache & \\ \quad if\ \ (Num\_t\,(\mathrm{S_i})+1) \times u \times l/v > Cache & \\ (Num\_t\,(\mathrm{S_i})+1) \times u \times l/v & \\ \quad else & \end{cases} \tag{17}$$

As mentioned before, *Num_t(S_i)* is the number of all descendant nodes of S_i, and "Cache" represents the size of node's buffer. Without loss of generality, the value of it is set to 4KB.

Figure 16 shows the success rate of data transmission under different packet sizes. Values of key parameters in this experiment are shown in Table 3.

The data packet generation time interval of each node is set to 5s, and the generated data are 400bit, 800bit and 1200bit, respectively. That is, the data packet transmission rate is 80bit/s, 160bit/s and 240bit/s. Moreover, the velocity of Sink is still 5m/s. It can be seen from Figure 16 that the greater the amount of data, the lower the transmission success rate. For ETDC, the energy balanced data collection tree generation strategy is adopted, and the load as well as the energy consumption of traversal nodes is fully considered



(a) Total Number of Nodes is 100

(b) Total Number of Nodes is 400

FIGURE 17. Trajectory length comparison.

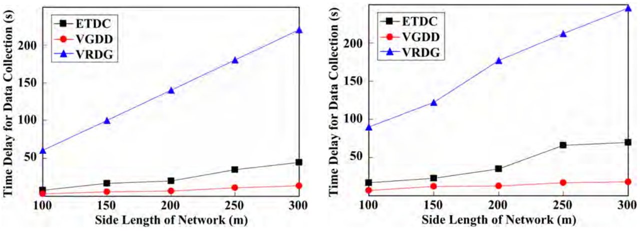
in constructing the sub-trees. Therefore, even if the number of data packets is large, the success rate of data transmission is still higher than 94% or 80% respectively in the network whose the size is 100m×100m or 200m×200m.

It is worth mentioning that, when the size of packet is 1200bit, the success rate of data transmission rises first and then declines slightly with the increase of the number of nodes. In these two experimental scenarios, the standard deviation of them are 1.64% and 2.60% respectively. This is because when the number of nodes is less, there are only a little data collection sub-trees, and the number of descendants of each traversal node is relatively large. In this case, the traversal node is prone to buffer overflow. When the number of nodes increases, the size of each data collection sub-tree is nearly the same, and the load on each traversal node is also reduced. So, the success rate of data transmission is rising. However, because ETDC specifies the upper limit of the path length, the number of the traversal nodes will not continuously increase. Therefore, when the number of nodes continues to rise, the data collection sub-tree will be enlarged accordingly. In this case, the buffer overflow occurs, and the success rate of data collection is reduced.

D. PERFORMANCE COMPARISON BETWEEN THREE TYPES OF MOBILE SINK BASED DATA COLLECTION METHODS

The following figures show the performance comparison of ETDC with VRDG and VGDD. Values of key parameters in this group of experiments are the same as those shown in Table 3.

Figure 17 shows the length of the trajectory in ETDC, VGDD and VRDG. As mentioned earlier, in VGDD, the trajectory of the mobile Sink is always the tangential circle of the rectangular network. The cluster head located in the boundary grid is responsible for establishing a data collection tree of the whole network. When the mobile Sink traverses this grid, the cluster head upload all the gathered data to it. Although this method effectively improves the real-time performance of data collection, the trajectory of Sink is too long and it grows linearly with the expansion of the network scale. In VRDG, the network is also divided into several data collection areas. The mobile Sink arrives at the traversal points in each area in a spiral-like manner, and it completes the information exchange with the ''leader nodes''. Due to the fact that the Sink can only moves sequentially among limited number of traversal points, the nodes in VRDG can predict the time for data collection and realize sleep scheduling, which effectively prolongs the network lifetime. However, it is undeniable that the length of the moving path in VRDG is still too long. In addition, with the expansion of network size, the number of traversal nodes in VRDG method will accordingly increase. Therefore, similar to VGDD, the length of Sink's moving trajectory in VRDG is positively correlated with the network size.



(a) Total Number of Nodes is 100

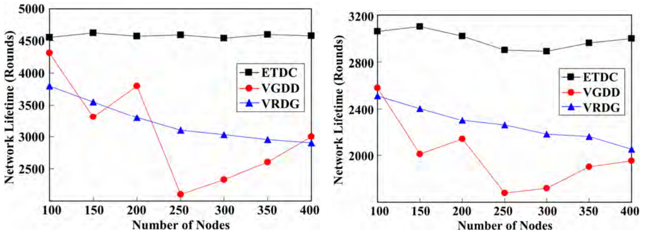
(b) Total Number of Nodes is 400

FIGURE 18. Time delay of data collection.

To further reduce the time delay of data gathering and to make full use of the advantages of tree based data collection structure, the upper limit value of the trajectory length is defined in ETDC. Constrained by this, the cost-balanced data collection sub-trees are constructed one after another, and the traversal nodes are also selected out. Thus, the curve fitting of Sink's trajectory can be realized. It is worth mentioning that setting this upper limit value can also alleviate the probability of node's buffer overflow caused by long-time waiting for the arrival of the mobile Sink, which ensures the success rate of data transmission. The experimental results in Figure 16 confirm this conclusion.

Figure 18 shows the comparison among the three algorithms in real-time of data collection. Without loss of generality, the delay of data collection is defined as the duration from the time when the node generates the data packet to the time when the data packet is finally received by the mobile Sink. As can be seen from this figure, the performance of ETDC on real-time of data collection is significantly better than that of VRDG. What's more, the time delay of ETDC will not significantly increase with the expansion of network scale. This is also because the ETDC method sets the upper limit value of the path length. Although this value can be appropriately increased due to the large number of data collection sub-trees and the traversal nodes when the network size is enlarged, it is still smaller than the side length of the network. In VRDG, the number of traversal nodes is positively correlated with the size of the network, because the network is divided into some data collection areas with the same size. In this case, the time delay on data collection increases linearly with the increase of the side length of the network.

It is worth mentioning that in VGDD, whenever Sink moves to a boundary grid, the cluster head in this grid will always establish a data collection tree for the whole network, as shown in Figure 9. Although the path length of the mobile



(a) 100m×100m Network (b) 200m×200m Network

FIGURE 19. Network lifetime comparison.

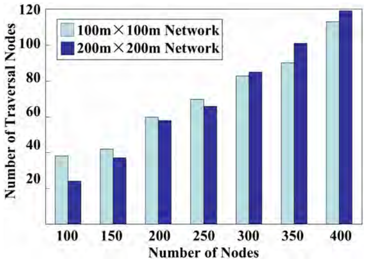


FIGURE 20. Number of traversal nodes in ETDC.

Sink in VGDD is the longest, the frequency of data collection is very high, that is, the real-time effect of it is the best. That is to say, the time delay on data collection of this method is almost equal to the time spent by Sink moving between two traversal nodes. However, it should be noted that such a centralized data collection method will inevitably increase the energy consumption of the whole network, as demonstrated by the experimental results in Figure 19.

Figure 19 is the network lifetime of the three data collection methods. As we know, in VGDD, each cluster heads builds a data collection tree of the whole network for data uploading. Although it guarantees real-time performance to some extent, it increases energy consumption on communication and aggravates the burden of cluster head nodes. In each round of data uploading, there is a cluster head that needs to receive data from all the nodes in the network and send them to the mobile Sink, which may lead to premature failure of this cluster head. In addition, cluster heads in each grid need to communicate with each other periodically to announce the current location of the mobile Sink in real time, which further increases the energy consumption of nodes. Therefore, in Figure 19, the network lifetime of VGDD is shorter than that of the other two methods.

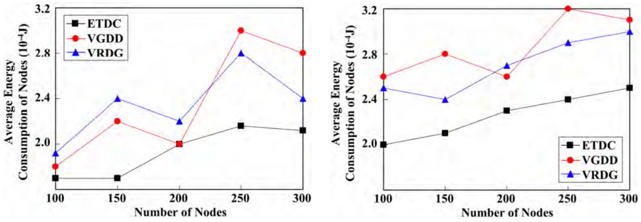
It is worth noting that, the broken line representing the VGDD method shows a trend different from that of the other two methods. Its volatility is relatively large, which is manifested in the trend of first decreasing, then rising, then rapidly declining, and then slowly rising. The standard deviation of network lifetime is as high as 741 and 280 rounds respectively in the 100m×100m and 200m×200m networks. The reasons for this fluctuation are described as follows.

When the number of nodes in the network is very small, the load of cluster headers in VGDD is lighter, so the network lifetime is longer. When the number of nodes increases slightly, according to the clustering strategy described in VGDD, the number of clusters does not significantly increase. However, the burden on each cluster head will inevitably increase, resulting in shorter network lifetime. As shown in Figure 19(a), when the number of nodes is 150, the network lifetime of VGDD is about 5% lower than that of VRDG. With the further increase of the number of nodes, most of the clusters will be reconstructed. At this time, the number of clusters slightly increases, and the size of them decreases. Therefore, the burden on these cluster headers slightly reduces and the network lifetime accordingly increases. Subsequently, when the number of nodes increases from 200 to 250, the number of clusters is unchanged, but the average number of nodes in each cluster increases by 25%. At this time, the load of cluster head increases sharply, resulting in a significant decline in the network lifetime of VGDD. With the number of nodes continuously increases (e.g., from 250 to 300), the clustering process will be re-executed. As a result, the number of clusters increases and the size of clusters decreases again. In this case, the network lifetime slightly increases again.

In VRDG, the mobile Sink only need to collect data in each virtual regions during a round of data gathering time. Moreover, the selected leader of each region is close to the mobile Sink when uploading data to it, which greatly reduces the energy consumption on communication to a certain extent. For these reasons, the network lifetime of VRDG is generally better than that of VGDD. However, when there are a large amount of nodes exist in the network, the burden of the leader will increase as the number of virtual regions remains unchanged. Although VRDG also adopts the ‘‘leader node reselection’’ strategy, there is few number of nodes satisfying the condition of becoming the new leader. Therefore, with the increase of the number of nodes, the network lifetime of VRDG decreases slightly.

Relatively speaking, the ETDC method proposed in this paper has the longest network lifetime and it is basically not affected by the number of nodes in the network. This is because as there are more and more nodes in the network, the number of traversal points and data collection sub-trees will accordingly increase. This ensures that the load on the root nodes of the sub-trees does not increase (even decrease in some cases). As shown in Figure 20, when the total number of nodes is 400, there are 113 traversing nodes in the 100m×100m network. That means the average number of nodes in each sub-tree is less than 4. It saves the energy of traversal nodes to a certain extent, and shows better adaptability.

Figure 21 shows the average energy consumption of nodes after a round of data collection. It is not difficult to see that the advantages of ETDC is more obvious. The value of it is not only the smallest one among the three methods, but also the most stable one. For example, in the 100m×100m network, the standard deviation of the average energy consumption



(a) 100m×100m Network

(b) 200m×200m Network

FIGURE 21. Average energy consumption of nodes in a round of data collection (100m×100m).

of nodes in ETDC is only 2.00×10^{-4}J, while in the other two methods, the values are 2.88×10^{-4}J and 4.63×10^{-4}J respectively. This is due to the fact that in ETDC, based on the traversal node selection algorithm under multiple constraints, we have established the data collection sub-trees with approximate the same cost. As mentioned in Section III, the length of Sink's moving path is limited, so the size of these data collection sub-trees will not increase significantly regardless of the number of nodes. In this case, the load and energy consumption of each node in the network are relatively small. In addition, because the trajectory length of the Sink in ETDC is relatively short, the time spending on a round of mobile data collection is also short. That is to say, the amount of data uploaded by nodes during a single-round of data collection is less, so the average energy consumption of nodes in ETDC is the lowest.

As mentioned above, in VGDD, data is uploaded with the help of a data collection tree of the whole network, which aggravates the burdens on cluster heads and relay nodes close to them. So, the average energy consumption in VGDD is little higher. With the increase of the number of nodes in the network, the number and size of clusters in VGDD will also change, resulting in a slight increase and decrease in energy consumption of the cluster heads as time goes up. In the analysis of the experimental results in Figure 19, we have mentioned this. Therefore, although the data value of VGDD in Figure 21 shows an overall upward trend, the fluctuation of it is a little obvious.

For VRDG, the burden on the leaders will be aggravated with the increase of the number of nodes, so the average residual energy of nodes is higher than that of ETDC. It should be pointed out that in VRDG, Sink collects only the data uploaded by the nodes in the ''data collection area'' where the traversal point it arrives at each time. That is to say, there is no ''overload cluster head'' which carries the data of the whole network as that in the VGDD method. Therefore, the average energy consumption of VRDG method is less than that of VGDD. The more the number of nodes, the more obvious the difference of between the them.

V. CONCLUSION

A tree based data gathering method with a mobile Sink is designed and implemented in this paper. By constructing several low power data collection sub-trees, the traversal nodes are selected out which forms the optimal trajectory of Sink. Based on this trajectory, the number of traversal nodes is further adjusted to further balance the scale and load of each data collection sub-tree.

Nevertheless, with the arrival of the era of big data, the amount and types of sensing data will change accordingly. Therefore, how to effectively reduce the redundancy of the collected data and how to find the best locations for data collection are our future work.

REFERENCES

[1] Y. Yue, L. Cao, B. Hang, and Z. Luo, ''A swarm intelligence algorithm for routing recovery strategy in wireless sensor networks with mobile sink,'' *IEEE Access*, vol. 6, pp. 67434–67445, 2018.

[2] R. Deng, S. He, and J. Chen, ''An online algorithm for data collection by multiple sinks in wireless-sensor networks,'' *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 93–104, Jun. 2018.

[3] Y. Zhang, S. He, and J. Chen, ''Near optimal data gathering in rechargeable sensor networks with a mobile sink,'' *IEEE Trans. Mobile Comput.*, vol. 16, no. 6, pp. 1718–1729, Jun. 2017.

[4] K. Gai and M. Qiu, ''Reinforcement learning-based content-centric services in mobile sensing,'' *IEEE Netw.*, vol. 32, no. 4, pp. 34–39, Jul./Aug. 2018.

[5] I. Azam, N. Javaid, A. Ahmad, W. Abdul, A. Almogren, and A. Alamri, ''Balanced load distribution with energy hole avoidance in underwater WSNs,'' *IEEE Access*, vol. 5, pp. 15206–15221, 2017.

[6] H. Huang, H. Yin, G. Min, J. Zhang, Y. Wu, and X. Zhang, ''Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks,'' *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1339–1352, Jun. 2018.

[7] C. Zhan, Y. Zeng, and R. Zhang, ''Trajectory design for distributed estimation in UAV-enabled wireless sensor network,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 10155–10159, Oct. 2018.

[8] H. Chen, D. Li, Y. Wang, and F. Yin, ''UAV hovering strategy based on a wirelessly powered communication network,'' *IEEE Access*, vol. 7, pp. 3194–3205, 2018.

[9] A. Mehrabi and K. Kim, ''Maximizing data collection throughput on a path in energy harvesting sensor networks using a mobile sink,'' *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 690–704, Mar. 2016.

[10] C. Wang, S. Guo, and Y. Yang, ''An optimization framework for mobile data collection in energy-harvesting wireless sensor networks,'' *IEEE Trans. Mobile Comput.*, vol. 15, no. 12, pp. 2969–2986, Dec. 2016.

[11] R. Xie, A. Liu, and J. Gao, ''A residual energy aware schedule scheme for WSNs employing adjustable awake/sleep duty cycle,'' *Wireless Pers. Commun.*, vol. 90, no. 4, pp. 1859–1887, Oct. 2016.

[12] S. He, J. Chen, D. K. Y. Yau, and Y. Sun, ''Cross-layer optimization of correlated data gathering in wireless sensor networks,'' *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1678–1691, Nov. 2012.

[13] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, ''Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing,'' *J. Netw. Comput. Appl.*, vol. 59, pp. 46–54, Jan. 2016.

[14] K. Gai, M. Qiu, and H. Zhao, ''Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing,'' *J. Parallel Distrib. Comput.*, vol. 111, pp. 126–135, Jan. 2018.

[15] M. D. Francesco, S. K. Das, and G. Anastasi, ''Data collection in wireless sensor networks with mobile elements: A survey,'' *ACM Trans. Sensor Netw.*, vol. 8, no. 1, p. 7, Aug. 2011.

[16] Y. Yang, M. I. Fonoage, and M. Cardei, ''Improving network lifetime with mobile wireless sensor networks,'' *Comput. Commun.*, vol. 33, no. 4, pp. 409–419, 2010.

[17] A. Chakrabarti, A. Sabharwal, and B. Aazhang, ''Communication power optimization in a sensor network with a path-constrained mobile observer,'' *ACM Trans. Sensor Netw.*, vol. 2, no. 3, pp. 297–324, Aug. 2006.

[18] L. Guo, R. Beyah, and Y. Li, ''SMITE: A stochastic compressive data collection protocol for mobile wireless sensor networks,'' in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 1611–1619.

[19] Y. Shi and Y. T. Hou, ''Theoretical results on base station movement problem for sensor network,'' in *Proc. 27th Conf. Comput. Commun.*, Phoenix, AZ, USA, Apr. 2008, pp. 1–5.

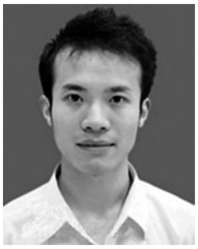
[20] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Netw. Appl.*, vol. 11, no. 3, pp. 327–339, Jun. 2006.
[21] T.-S. Chen, H.-W. Tsai, Y.-H. Chang, and T.-C. Chen, "Geographic convergecast using mobile sink in wireless sensor networks," *Comput. Commun.*, vol. 36, no. 4, pp. 445–458, Feb. 2013.
[22] K. Lee, Y.-H. Kim, H.-J. Kim, and S. Han, "A myopic mobile sink migration strategy for maximizing lifetime of wireless sensor networks," *Wireless Netw.*, vol. 20, no. 2, pp. 303–318, 2014.
[23] J. Luo and J.-P. Hubaux, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 871–884, Jun. 2010.
[24] S.-W. Han, S. H. Kang, and I.-S. Jeong, "Low latency and energy efficient routing tree for wireless sensor networks with multiple mobile sink," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 156–166, 2013.
[25] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1472–1483, May 2013.
[26] A. Kinalis, S. Nikoletseas, D. Patroumpa, and J. Rolim, "Biased sink mobility with adaptive stop times for low latency data collection in sensor networks," *Inf. Fusion*, vol. 15, pp. 56–63, Jan. 2014.
[27] J. Wang, Z. Zhang, F. Xia, W. Yuan, and S. Lee, "An energy efficient stable election-based routing algorithm for wireless sensor networks," *Sensors*, vol. 13, no. 11, pp. 14301–14320, Oct. 2013.
[28] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.
[29] S. Gao and H. K. Zhang, "Optimal path selection for mobile sink in delay-guaranteed sensor networks," *Acta Electron. Sinica*, vol. 39, no. 4, pp. 742–747, Apr. 2011.
[30] M. Zhao and Y. Yang, "A framework for mobile data gathering with load balanced clustering and MIMO uploading," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2759–2767.
[31] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.

CHAO SHA received the B.Eng., M.Eng., and Ph.D. degrees in computer science and technology from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2005, 2008, and 2010, respectively. He is currently an Associate Professor with the School of Computer Science, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications. His research interests include mobile data collection and energy hole avoidance in wireless rechargeable sensor networks.

DANDAN SONG received the B.S. degree from Liaocheng University, in 2017. She is currently pursuing the master's degree with the School of Computer Science, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications. Her research interests are data collection algorithm and the energy hole avoidance problem in sensor networks.

RUI YANG received the B.Eng. and M.Eng. degrees in computer science and technology from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2015 and 2019, respectively. His research interests are energy balancing problem and energy harvesting technology in sensor networks.

HANCHENG GAO is currently pursuing the bachelor's degree with the School of Computer Science, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications. His research interests are path planning technology for mobile data collection and vehicle scheduling problem in wireless rechargeable sensor networks.



HAIPING HUANG (M'12) received the B.Eng. and M.Eng. degrees in computer science and technology from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2002 and 2005, respectively, and the Ph.D. degree in computer application technology from Soochow University, Suzhou, China, in 2009. In 2013, he was a Visiting Scholar with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K. He is currently a Professor with the School of Computer Science, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications. His research interests include information security and privacy protection of wireless sensor networks.