

Research Article

Delay Tolerance and Energy Saving in Wireless Sensor Networks with a Mobile Base Station

Oday Jerew ¹ and Nizar Al Bassam ²

¹Asia Pacific International College, Sydney, Australia

²Middle East College, Muscat, Oman

Correspondence should be addressed to Oday Jerew; oday.jerew@gmail.com

Received 3 July 2018; Revised 30 October 2018; Accepted 17 December 2018; Published 12 February 2019

Academic Editor: Donatella Darsena

Copyright © 2019 Oday Jerew and Nizar Al Bassam. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent research shows that significant energy saving can be achieved in wireless sensor networks by using mobile devices. A mobile device roams sensing fields and collects data from sensors through a short transmission range. Multihop communication is used to improve data gathering by reducing the tour length of the mobile device. In this paper we study the trade-off between energy saving and data gathering latency in wireless sensor networks. In particular, we examine the balance between the relay hop count and the tour length of a mobile Base Station (BS). We propose two heuristic algorithms, Adjacent Tree-Bounded Hop Algorithm (AT-BHA) and Farthest Node First-Bounded Hop Algorithm (FNF-BHA), to reduce energy consumption of sensor nodes. The proposed algorithms select groups of Collection Trees (CTs) and a subset of Collection Location (CL) sensor nodes to buffer and forward data to the mobile BS when it arrives. Each CL node receives sensing data from its CT nodes within bounded hop count. Extensive experiments by simulation are conducted to evaluate the performance of the proposed algorithms against another heuristic. We demonstrate that the proposed algorithms outperform the existing work with the mean of the length of mobile BS tour.

1. Introduction

Data gathering in Wireless Sensor Networks (WSNs) is one of the most frequent and fundamental operations, which requires the sensor nodes to monitor the sensing field for as long as possible. As sensor nodes have limited energy resources and are powered by small batteries, energy consumption is a critical issue in the design of WSNs that affect the energy consumption of sensor nodes and hence the network lifetime.

Recent research [1–5] shows that significant energy saving can be achieved in wireless sensor networks by using mobile devices capable of carrying data mechanically. Mobile Base Stations (BSs) are proposed to reduce energy consumption by allowing a mobile BS to roam a sensing field and gather data from sensor nodes through a short transmission range. The energy consumption of each sensor node is then reduced, since fewer relays are needed for the sensor node to relay its data packet to the BS. As the speed of mobile BS is very slow compared with the speed of data packets, which travel in

multihop forwarding, the increased latency of data gathering when employing mobile BS presents a major performance bottleneck. Consequently, the time a mobile BS takes to tour a large sensing field may not meet the stringent delay requirements inherent in some mission-critical, real-time applications. Therefore, planning the tour of mobile BS needs to be considered in order to achieve the delay requirements.

Generally, the mobility of BS can be classified into three types, according to the mobility pattern of the entity upon which the BS is mounted, as follows:

- (1) Random mobility: this can be achieved, for example, when the BS is mounted on humans and animals. In this case, the probability of the BS collecting all sensing data is low since the BS opportunistically visits sensor nodes [6].
- (2) Predictable mobility: in this case, the BS is mounted on an entity that moves on a fixed track or path that cannot control its direction or speed, but which moves at a regular time, for example, a BS mounted on a bus

or train. Thus, the sensor nodes can predict when the BS may move around to send their data [7, 8].

- (3) **Controlled mobility:** when the BS is mounted on a robot or UAV plane, then the direction and speed of the BS can be controlled. Many algorithms are proposed to find the tour of the BS in order to achieve requirements such as maximised network lifetime and data-gathering delay using single and multihop relays [3, 5, 9].

Most literature studies the mobility of WSNs to be predictable or controlled to achieve network performance requirements (some literature focused on random mobility to improve network lifetime). This differs from mobile ad hoc networks in which nodes are assumed to move arbitrarily, which degrades the network performance by link failures.

In this paper, we deal with the data gathering problem with bounded delay time in sensor network using a controlled mobile BS, by exploring a balance between the relay hop count required for data relay and tour length of the mobile BS such that the energy consumption is minimized. Specifically, we assume the mobile BS selected a set of Collection Location (CL) nodes that temporarily cache and forward sensing data of other nodes. The sensor nodes are grouped into Collection Trees (CTs) rooted at CLs. Each node has to forward its data to the CL within a certain number of relay hops. The mobile BS traverses along a close tour and stops at each CL in the tour for data gathering.

Our major contributions, in this paper, are as follows. We propose two efficient heuristic algorithms: Adjacent Tree-Bounded Hop Algorithm (AT-BHA) and Farthest Node First-Bounded Hop Algorithm (FNF-BHA). The algorithms find the tour of the mobile BS consisting of CLs which meet the following criteria: (i) the energy consumption among the sensor nodes is balanced in order to prolong nodes lifetime; (ii) the total traversal time of the mobile BS on the tour is minimized; and (iii) the hop count between any sensor node and the CL is bounded by a given value. The AT-BHA constructs adjacent CTs by finding closest nodes to the last constructed CL. The FNF-BHA uses the farthest node from the BS in the construction of CL. Load balancing algorithm is proposed to balance the number of nodes in the CTs. Finally, the performance of the proposed algorithms is evaluated through experimental simulations. The experimental results demonstrate that the proposed algorithms are efficient.

The remainder of this paper is organized as follows. Section 2 reviews the related work on mobile data gathering. Section 3 presents two algorithms to solve data gathering with bounded hop count problem. Section 4 evaluates the efficiency of the proposed algorithms through extensive simulations. Finally, Section 5 concludes the paper.

2. Related Work

There is an extensive range of literature studying the use of mobile BS for data gathering [4, 5, 9–15]. Kansal et al. [12] combine multihop forwarding with a mobile BS. They performed an experimental evaluation for a small sensor network, assuming that a mobile BS moves back and forth

on a straight line (a fixed path). They employed a directed diffusion approach to gathering sensed data from the sensor nodes beyond the transmission range of the mobile BS.

Some existing studies assume that sensor nodes can cache sensing data of other nodes in order to forward it to the mobile BS when the BS moves around. Gao and Zhang [11] and Xing et al. [16] explored the delay requirement for data gathering assuming that the sensor nodes have an ability to cache sensing data of other nodes. In [11] sensor nodes send their data through multihop relays to the nodes (subsink) located within the direct-transmission range of the mobile BS. The subsink nodes cache data and send it to the mobile BS when it comes within transmission range. In [16], a rendezvous-based data-gathering approach is proposed, in which a subset of nodes is chosen as rendezvous points. The role of these points is to buffer and aggregate data originating from sensor nodes. When the mobile BS arrives within the transmission range of rendezvous points, the data will be forwarded to the mobile BS. Kaswanet et al. [17] proposed path plan algorithm for data collection by selecting set of rendezvous points. The algorithm minimize the tour length by reducing the number of rendezvous points by selecting the position closest to the location of the BS and allows the BS to collect data from maximum number of sensors via single hop communication. This algorithm does not consider a bounded hop count for data collection.

Grouping sensor nodes in the sensing field into clusters and collecting data from cluster heads are proposed in [13–15]. Ma and Yang [13] proposed a heuristic for finding routing paths for the mobile BS, which assumes that the moving path of the mobile BS consists of a series of line segments. Sensor nodes closest to each line segment are selected as cluster heads. A specified configuration is applied, where the mobile BS starts data gathering from the left side of the path, moves towards the right side, and then comes back to the left side again. This scheme maximizes network lifetime. However, it may cause packet losses at each cluster head if the path length to the cluster head is too long, since the time required for the cluster head to send the data of cluster nodes to the BS is not considered in cluster forming. Saad et al. [14] proposed path planning algorithm for mobile BS that visit a set of cluster heads. Nodes are randomly grouped to form clusters and the node with the largest residual energy is selected as the cluster head. The clusters are merged, ensuring that the number of hops from the cluster head to any cluster node is no more than two hops. The cluster head gathers and buffers cluster sensing data. The BS visits the cluster heads to gather data through single-hop communication. However, this work does not consider data gathering delay for the mobile BS. Zhang et al. [15] extended the work in [16] and regarded cluster heads as the rendezvous point. They considered event based data gathering and proposed reduction in data gathering delay by only allowing the mobile BS to visit cluster heads that generate new data.

Bounded hop count for data gathering is studied in [4, 5, 18, 19]. Zhao and Yang [5] proposed SPT-DGA and BRH-MDG data gathering algorithms that select set of CL for the mobile BS to visit for data gathering. Each CL aggregates the local data from its affiliated sensors within a certain

number of relay hops. However, the number of nodes in the constructed CT is small. Consequently, many CTs are constructed for data gathering. Thus, the length of the BS tour increased. Chen et al. [19] analysed and improved the BRH-MDG algorithm by minimizing the degree of the CTs in order to prolong the network lifetime. Xu et al. [4] proposed heuristic algorithm to find the tour for a mobile BS consists of CL nodes. However, the algorithm does not consider the effect of some critical sensor nodes on the separation of other nodes. These critical nodes affect the location of CLs and hence the length of the mobile BS. The hop count and hop distance are used to plan the route of the mobile BS in [18]. In this work, set of rendezvous points are selected such that the mobile BS can collect data from the sensor nodes within a permissible delay with minimum possible hop distance and hop counts. However, this work does not consider minimizing the BS tour length.

Dividing sensing field into subareas is proposed in [20, 21]. Chen et al. [20] introduced algorithm to improve the network lifetime considering data transmission delay and number of hops. The algorithm divided the sensing field into grids; the mobile sink has to stay at the grid center for data collection. However, this work assumes the sensor nodes send data to the BS only when the hop count between the sensor and the BS is at or below a specific hop count. Otherwise the sensor enters a sleeping state. In addition, the hop count is estimated using the distance between the sensor nodes and the BS which is not accurate at low node density (node degree less than six). A delay bound path algorithm is proposed in [21]. The algorithm divided the sensing field into hexagonal cells whose centers are considered as the positions of rendezvous point. The algorithm minimizes the number of rendezvous point by selecting the location that covers as many number of sensor nodes and at minimal distance from the center. Data gathering within bounded hop count is also studied in [22]. The authors proposed reducing the number of collection points by selection of the CL within the convergence area that overlaps maximum number of sensor nodes. However, the work in [21, 22] does not consider the effect of isolated sensor nodes on the tour length of the BS.

Other literature involved sensor node residual energy and bounded hop count in the construction of the data gathering trees [2, 23]. Zhu et al. [23] suggested assigning weight for each node based on nodes residual energy, distance to the BS and number of hops. The weight is used to select rendezvous points for data collection. Jerew and Al Bassam [2] examined the sensor nodes residual energy to select a set of cluster heads with high residual energy and bounded hop count. Sensor node sends its data within bounded hop count to the cluster head and the BS visits all the cluster heads for data gathering. Chang and Shen [24] introduced a cluster routing structure, to decide the routing path based on the location of BS, the distances between the sensor nodes, and the residual energy of each sensor node. The closest node to the BS with residual energy more than the average network residual energy is selected as cluster head. The proposed algorithm reduced the energy consumption and extended the sensor lifetime by balancing the network load. However, this work did not consider data gathering within a bounded hop count.

2.1. Preliminaries and Problem. We consider a wireless sensor network $G = (V, E)$ consisting of $n = |V|$ stationary sensors and E is the set of links. There is a link between two sensors or a sensor and the BS if they are within the transmission range of each other. For the sake of simplicity, we assume the transmission range of each sensor node and the BS are fixed and identical, and all sensor nodes have identical initial energy. The storage of a sensor node is limited, so that it cannot buffer a large volume of data. Sensor nodes are densely deployed in the sensing region. Accordingly, the number of hops in a path is approximately proportional to the distance between the nodes. The BS moves with constant velocity. Thus, there is sufficient time to establish communication and send one or more data packets during the time the BS takes to travel across the transmission range of a sensor node. We further assume the mobile BS replenishes its energy periodically so there is no energy concern with the mobile BS. Finally, sensor nodes and the mobile BS are assumed to know their own physical locations via GPS or a location service in the network.

Problem. Given a network with a mobile BS and assuming that the number of hops for data gathering between the BS and sensor nodes is bounded by h , the problem is to find the shortest tour for the mobile BS such that the energy consumption among the sensor nodes is balanced.

3. Proposed Heuristic Algorithms

Due to the NP-hardness of this problem [5], a heuristic algorithm is proposed in this paper. In order to find the optimal CLs among sensors, relay routing paths and the tour of the mobile BS should jointly be considered. Based on these observations, we deal with the tour length minimization problem by devising scalable heuristic algorithms. In this section, we proposed two algorithms: the Adjacent Tree-Bounded Hop Algorithm (AT-BHA) and the Farthest Node First-Bounded Hop Algorithm (FNF-BHA). Table 1 shows the main symbols used in the proposed algorithms.

3.1. Adjacent Tree-Bounded Hop Algorithm (AT-BHA). The basic idea of AT-BHA algorithm is to find a set of CLs in the network that the BS can visit to collect sensing data of all the sensors. Visiting CLs in specific order determines the length of the tour of the mobile BS. The value of h determines the delay on data delivery and affects the number of sensor nodes contained in each tree and, hence, the number of CLs. For example, if $h = 1$ the number of CLs is equal to the number of sensor nodes and the BS tour is the longest tour as the BS has to visit each node for one hop data collection.

In any network, the node degree (number of neighbor nodes) significantly affects the network connectivity. As the node degree $d < 6$, there is a high probability of partitioned networks [25]. The first task of AT-BHA as shown in Algorithm 1 is to construct Minimum Spanning Trees (MSTs) that cover all sensors in the network. Each sensor node associated in a single MST referred to as T_p . Thus, the AT-BHA can be applied to connected and disconnected networks.

Input: A sensor network $G(V, E)$, hop count h, r, β
Output: Set of CL nodes v_{CL}, BTC

- (1) $N_G \leftarrow V$; /*set of nodes in the network */;
- (2) $k = 0$;
- (3) $j = 0$;
- (4) **while** N_G is not empty **do**
- (5) Find the closest vertex v_c to the BS position from N_G ;
- (6) Find an MST T_P that covers vertices from N_G ;
- (7) $N_P \leftarrow T_P$; /* set of nodes in partition tree T_P */ ;
- (8) **while** N_P is not empty **do**
- (9) Find the farthest leaf vertex v_f in T_P ;
- (10) Find the routing path $R(v_f, v_c)$;
- (11) **if** $h \geq |R(v_f, v_c)|$ **then**
- (12) $v_{CL} \leftarrow v_c$;
- (13) **else**
- (14) $v_{CL} \leftarrow$ vertex from $R(v_f, v_c)$ at h hops from v_f ;
- (15) $j = j + 1$;
- (16) Find an MST $TC_{j,h}(v_{CL}, N_P)$;
- (17) $N_P \leftarrow N_P - TC_{j,h}$;
- (18) **if** N_P is empty **then**
- (19) **break**;
- (20) Find n_f node that with $Max.\{d(N_P, v_c)/d(N_P, v_{CL})\}$;
- (21) Find N_{cr} set of nodes with $d(N_P, n_f) \leq hr\beta$; /* nodes within a circle of $hr\beta$ radius */ ;
- (22) Find the closest vertex v_n to v_c from N_{cr} ;
- (23) $j = j + 1$;
- (24) $v_{CL} \leftarrow v_n$; /* set v_n as a CL */ ;
- (25) Find an MST $TC_{j,h}(v_{CL}, N_P)$;
- (26) $N_P \leftarrow N_P - TC_{j,h}$;
- (27) $(TC, N_P) = \text{Update_CT}(v_{CL}, j, N_P, TC, T_P)$;
- (28) $N_G \leftarrow N_G - N_P$
- (29) $[BTC, v_{CL}] = \text{Call LBA}(v_{CL})$;

ALGORITHM 1: The Adjacent Tree-Bounded Hop Algorithm (AT-BHA).

TABLE 1: Definitions of the main symbols used throughout the algorithms.

Symbol	Description
V	Set of network nodes
β	Between -1 and 1, to calculate hop progress
τ	Threshold value to decide tree update
h	Number of hops
T_P	Set of nodes in a partition tree
v_c	Root node of the MST that is closest to the location of BS
v_f	Farthest node from the BS
$R(v_f, v_c)$	Set of nodes in the routing path between v_f and v_c
v_{CL}	Set of the collection location nodes
$TC_{j,h}$	Set of nodes contained in the partial j th tree within h hops away from the root
N_P	Set of the nodes at network partition tree
$d(v_1, v_2)$	Euclidean distance between vertices v_1 and v_2
n_f	Farthest node to the BS and closest to the $TC_{j,h}$ tree
r	Transmission range of a sensor node
N_r	Set of nodes to be removed from a tree

The root of the MST is selected at sensor node v_c that is the closest to the location of the BS. We set the root close to the BS in order to select CLs as close as possible to each other and close to the BS initial position.

The next task of AT-BHA is to find the farthest node, v_f , from the BS position in the partition tree. The routing path between the farthest node and root of T_P is referred to as $R(v_f, v_c)$ and it is used to find the candidate CL node. We assume the set of nodes in R is sorted in decreasing order according to the number of hops to v_c . In the case of the number of nodes in the routing path being less than h , then the root of the T_P tree is selected as v_{CL} . While the node at h hop away from the v_f is selected as v_{CL} if the route length is higher than h .

The Breath-First-Search (BFS) algorithm with bounded hop count is used to find the Minimum Spanning Tree (MST) rooted at the CL. A partial BFS tree is constructed hop by hop, and the expansion continues until all nodes within h hops are explored. The set of sensors contained in the partial j th tree is referred to as $TC_{j,h}$. The set of the nodes covered in the j th CT is removed from the set of partition tree nodes, N_P .

In this algorithm, we propose to find the next CT adjacent to the last constructed tree in order to avoid partitioned node

```

Input:  $v_{CL}, j, N_p, TC, T_p, \tau$ 
Output:  $TC, N_p$ 
(1) if  $v_{CL}$  is the first time for check then
(2)    $N_{CL} \leftarrow TC_{j,h}$  ;
(3)   if  $|N_{CL}|/d(v_{CL}, v_c) \leq \tau$  then
(4)     Find the routing path  $R(v_{CL}, v_c)$  using partition tree  $T_p$  ;
(5)     if  $h \geq |R(v_{CL}, v_c)|$  then
(6)        $N_p \leftarrow N_p \cup R(v_{CL}, v_c)$  ;
(7)        $v_{CL} \leftarrow v_c$  ;
(8)     else
(9)       Find vertex  $v_h$  from  $R(v_{CL}, v_c)$  at  $h$  hops from  $v_{CL}$  ;
(10)       $N_p \leftarrow N_p \cup R(v_{CL}, v_h)$  ;
(11)       $v_{CL} \leftarrow v_h$  ;
(12)
(13)     Find an MST  $TC_{j,h}(v_{CL}, N_p)$  ; /* Update the  $j$ -th tree*/ ;
(14)      $N_p \leftarrow N_p - TC_{j,h}$  ;
(15) Return( $TC, N_p$ )

```

ALGORITHM 2: Update_CT.

between the CTs. The partitioned node increases the tour length since the BS has to visit the node for data collection. To find an adjacent tree to the j th tree, the algorithm calculates the ratio $\alpha = d(N_p, v_c)/d(N_p, v_{CL})$. The $d(v_1, v_2)$ represents Euclidean distance between vertices v_1 and v_2 . The node from N_p with maximum α is selected and referred to as n_f . This node represents the farthest node to BS and closest to the $TC_{j,h}$. The closest node to the BS that is within the distance $hr\beta$ from n_f is selected as the second v_{CL} , where r is the sensor transmission range and β between -1 and 1 . The value $r\beta$ represents the hop progress and its ranges from $-r$ to r depend on the network node density [25]. New MST is constructed rooted at v_{CL} to find set of nodes within h hops.

The number of nodes within each tree and the position of the CL nodes have a signification effect on the tour length. When the CT consists of many nodes, a less number of CLs are required and the tour length decreases. In addition, the tour length is also decreased as the selected CLs are close to the BS. To improve the tree size and location of v_{CL} , the Update_CT algorithm is proposed in Algorithm 2.

In the Update_CT algorithm, the ratio of the number of tree nodes to the distance between the v_{CL} and v_c is used as a measure of tree usefulness. A threshold value, τ , is used to decide the updating of the CT. The range of τ is $(1/d(L_{BS}, N_F), n/d(L_{BS}, v_{BS}))$, where n is the network nodes, and L_{BS} , v_{BS} , and N_F are, respectively, the position of BS, the closest node to the BS, and the farthest node to the BS.

In order to avoid infinite looping, the algorithm first checks whether the CT is previously updated since in some cases it is impossible to update the tree because of limited number of nodes. In the case the CT needs to update, the routing path between v_{CL} and v_c is used to select a new CL node. N_p is updated by including the set of CT nodes and the set of routing nodes to the selected v_{CL} . Finally, the updated CT is constructed and returns the updated CT and N_p to the AT-BHA algorithm. The AT-BHA algorithm continues until all the network nodes are associated with CTs.

3.2. Farthest Node First-Bounded Hop Algorithm (FNF-BHA).

This algorithm constructs CTs to include the farthest nodes to the BS. The FNF-BHA starts similar to AT-BHA by finding the nodes closest to the position of the BS and construction partitions trees. It then finds the route of data packets to v_f in the current partition tree to determine the CL node. The nodes involved in the data relay between v_f and the collection node are critical for data gathering from v_f . In the case of any of the relay node participating in another CT, the BS has to visit v_f for data collection increasing the tour length. Therefore, in this case, the algorithm first identifies and then removes all the relay nodes from other CTs by calling the Node Removal Algorithm (NRA) as shown in Algorithm 3. The set of removed nodes is used in the construction of new CTs at the next iteration.

The algorithm tests whether the CL node is already selected as CL of another tree. In this rare case, the set of CT nodes are merged with that tree and new CT is constructed. The new tree includes all the nodes of previous CT in addition to v_f , since all the relay nodes are available in the set of nodes, whereas, if the CL node is not selected, a new CT is constructed with a new tree index to include v_f . The algorithm is terminated when all the partitioned tree nodes are covered.

We now describe NRA algorithm in more detail. The pseudocode is given in Algorithm 4. In order to remove the set of nodes in N_r from the CTs, the tree index of each node needs to be found first. The set of nodes that need to be removed from each CT are identified and removed. A new CT is reconstructed after node removal. In some cases, some of the tree nodes, N_o , are partitioned, since the removed nodes provide a relay path for them. N_o may be covered by other CTs. Therefore, all the CTs are reconstructed considering N_o . If a node is associated with CT, the node is removed from N_o . The algorithm continues until all the nodes in N_r are removed from the CTs.

Input: A sensor network $G(V, E)$, hop count h
Output: Set of CL nodes v_{CL}, BTC

- (1) $N_G \leftarrow V$; /*set of nodes in the network*/ ;
- (2) **while** N_G is not empty **do**
- (3) Find the closest vertex v_c to the BS position from N_G ;
- (4) Find an MST T_p that covers vertices from N_G ;
- (5) $N_p \leftarrow T_p$; set of nodes in partition tree T_p ;
- (6) **while** N_p is not empty **do**
- (7) Flag \leftarrow false ; /*no node need to remove from any tree ;
- (8) Find the farthest leaf vertex v_f to v_c on T_p ;
- (9) Find the routing path $R(v_f, v_c)$;
- (10) **if** $h \geq |R(v_f, v_c)|$ **then**
- (11) $v_{CL} \leftarrow v_c$;
- (12) **else**
- (13) $v_{CL} \leftarrow$ vertex from $R(v_f, v_c)$ at h hops from v_f ;
- (14) $N_r \leftarrow$ set of nodes in $R(v_f, v_{CL})$ not in N_p ;
- (15) **if** N_r is not empty **then**
- (16) Flag \leftarrow true ;
- (17) $N_{rm} = \text{NRA}(N_r, N_p)$; /* call Node Removal Algorithm */
- $N_p \leftarrow N_p \cup N_{rm}$;
- (18) **if** Flag = false **then**
- (19) **if** v_{CL} selected as CL **then**
- (20) $j =$ index of tree rooted on v_{CL} ;
- (21) $N_p \leftarrow N_p \cup TC_{j,h}$;
- (22) **else**
- (23) select v_{CL} as Collection Location ;
- (24) $j = j + 1$;
- (25)
- (26) Find an MST $TC_{j,h}(v_{CL}, N_p)$;
- (27) $N_p \leftarrow N_p - TC_{j,h}$;
- (28) $N_G \leftarrow N_G - N_p$;
- (29) $[v_{CL}, BTC] = \text{Call LBA}(v_{CL})$;

ALGORITHM 3: The Farthest Node First-Bounded Hop Algorithm (FNF-BHA).

Input: A sensor network $G(V, E)$, hop count h, N_r, N_p
Output: N_r, TC

- (1) $IX \leftarrow$ set of tree index of all CTs ;
- (2) $IX_{rt} \leftarrow$ set of tree index of all N_r nodes ;
- (3) $N_{rm} \leftarrow \phi$;
- (4) $N_o \leftarrow \phi$;
- (5) **for each** j in IX_{rt} **do**
- (6) $N_{rt} \leftarrow N_r \cap TC_{j,h}$ /* Set of nodes to remove from the j-th tree */ ;
- (7) $N_{nt} \leftarrow TC_{j,h} - N_{rt}$ /* Set of node of the j-th tree after node removal */ ;
- (8) get v_{CL} from tree j ;
- (9) Find an MST $TC_{j,h}(v_{CL}, N_{nt})$;
- (10) $N_p \leftarrow N_{nt} - TC_{j,h}$ /* Set of partitioned nodes */ ;
- (11) $N_o \leftarrow N_o \cup N_p$;
- (12) **if** N_o is not empty **then**
- (13) **for each tree index** k in IX **do**
- (14) get v_{CL} for tree k ;
- (15) $n_t \leftarrow TC_{k,h}$;
- (16) $n_{set} \leftarrow TC_{k,h} \cup N_o$;
- (17) Find an MST $TC_{k,h}(v_{CL}, n_{set})$;
- (18) $N_{dif} \leftarrow TC_{k,h} - n_t$;
- (19) $N_o \leftarrow N_o - N_{dif}$;
- (20) $N_{rm} \leftarrow N_{rm} \cup N_{rt} \cup N_o$;
- (21) Return(N_{rm}) ;

ALGORITHM 4: Node Removal Algorithm (NRA).

Input: A sensor network $G(V, E)$, hop count h , tree index j , TC
Output: Set of CL nodes CL , BTC

- (1) $[BTC, PN] = \text{Call CTBA}(G, TC)$;
- (2) **while** PN is not empty **do**
- (3) $TC = BTC$; copy all the CTs to TC ;
- (4) $j \leftarrow$ number of trees in TC ;
- (5) Find the farthest leaf vertex v_f from PN using T_p ;
- (6) Find the routing path $R(v_f, v_c)$;
- (7) **if** $h \geq |R(v_f, v_c)|$ **then**
- (8) $v_{CL} \leftarrow v_c$;
- (9) $R_n \leftarrow |R(v_f, v_c)|$;
- (10) **else**
- (11) $v_{CL} \leftarrow$ vertex from $R(v_f, v_c)$ at h hops from v_f ;
- (12) $R_n \leftarrow$ nodes at h hops from V_f ;
- (13) $PN \leftarrow PN \cup R_n$;
- (14) $j = j + 1$;
- (15) Find an MST $TC_{j,h}(v_{CL}, PN)$;
- (16) $PN \leftarrow PN - TC_{j,h}$;
- (17) $[BTC, PN] = \text{Call CTBA}(G, TC)$;
- (18) Return(CL, BTC) ;

ALGORITHM 5: Load balanced algorithm (LBA).

Input: A sensor network $G(V; E)$, Set of CTs $TC_{j,h}$, hop count h
Output: set of balanced CTs $BTC_{j,h}$, Remain Nodes PN

- (1) $K \leftarrow$ number of CL nodes ;
- (2) $N_G \leftarrow V$;/* set of nodes in the network */ ;
- (3) $N_{GB} \leftarrow N_G$;
- (4) $BTC_{j,hc} \{j = 1 : K, hc = 1 : h\} \leftarrow \phi$;
- (5) Sort TC_j trees in increasing order according to the number of nodes in each tree ;
- (6) $N_{GB} = N_G - v_{CL}$ // remove CL nodes from testing nodes ;
- (7) **for** $hc = 1$ to h **do**
- (8) **for each** v_{CL} **do**
- (9) $j = j + 1$;
- (10) **if** $hc > 1$ **then**
- (11) $N_{GB} = N_{GB} \cup BTC_{j,hc-1}$;
- (12) Find an MST $BTC_{j,hc}(v_{CL}, N_{GB})$;
- (13) $N_{GB} = N_{GB} - BTC_{j,hc}$ /* remove tree nodes from testing nodes */ ;
- (14) $PN = N_G - N_{GB}$;
- (15) Return(BTC, PN) ;

ALGORITHM 6: CT balanced algorithm (CTBA).

3.3. Load Balancing and Further Improvement. The proposed algorithms AT-BHA and FNF-BHA can be further improved by balancing the number of nodes in the CTs. The proposed algorithms find the set of CL nodes to collect data from sensor nodes within h hops. However, there are large differences between the number of nodes in the CTs. In particular, the first constructed tree covers the maximum number of nodes, while the last tree covers only the remaining set of nodes.

Thus, we proposed the load balanced algorithm (LBA) with the pseudocode given in Algorithm 5. First, the algorithm balanced the number of nodes for each CT using the CT balanced algorithm (CTBA) shown in Algorithm 6. The CTBA first sorts the CTs in increasing order according to

the number of nodes in order to start constructing the CT with the minimum number of nodes. The algorithm gradually constructs all the CTs by finding the nodes with one hop from the CL nodes. Then the number of nodes in the CTs increases with the hop count. In some cases, some nodes does not associate to any tree since they are partitioned as the relaying nodes are associated to other CTs. Finally, the LBA returns the balanced CTs (BTC) and partitioned nodes (PN). When there is no PN the algorithm finalizes the BTC and no further process is required. However, a new set of v_{CL} needs to be selected from the PN for data collection. This is achieved by finding the farthest node v_f to the BS from PN. The routing path between v_f and closest node to the BS is used to find

the v_{CL} at h hops. A new CT is constructed and rooted at v_{CL} to cover PN nodes. Finally, the algorithm rebalanced the load by calling the LBA. The LBA is terminated when there is no partition node in the PN.

4. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms through simulations. We assume that sensor nodes in the network are randomly deployed with uniform distribution in a $400\text{ m} \times 400\text{ m}$ square sensing field. Each sensor node has a transmission range of $r = 25\text{ m}$. We first vary the hop count to evaluate the performance of the algorithms. We also vary the number of nodes in the network to emulate the change in the node degree. The node degree is used as a metric of node density. We compare our algorithms with the SPT-DGA algorithm proposed in [5]. For the AT-BHA, we set $\beta = 0.5$ and $\tau = 0.1$. For each instance of deployment, the network performance metrics are calculated, and the result is the average over 500 instances for each node degree and hop count.

We adopt the nearest neighbor algorithm [26] in our simulation for the travelling salesman problem to determine the tour of the mobile BS. In order to calculate the energy consumption of CL in transmitting CT sensors data to the BS on wireless communication per time unit, we adopt the energy model [2], $EC_{CL}(CL) = r_a \cdot L_R \cdot (NT_{CL} + 1) \cdot E_t$, where $EC_{CL}(CL)$ is the energy consumption of CL, r_a is data generation rate, L_R is the length of single data reading, and NT_{CL} is the number of sensors the CL node has to forward their data to the BS. E_t is the amount of power consumed by transmitting a unit-length of data. E_t can be represented by $E_t = \alpha_1 + \alpha_2 r^\gamma$, where α_1 is a constant that represents the energy consumption to run the transmitter circuitry which is negligibly small, α_2 is a constant that represents the transmitter amplifier, and r is the transmission range. The exponent γ is determined by the field measurements, which is typically a constant between 2 and 4. In this simulation we assume $r_a = 1\text{ Kbps}$, $L_R = 1\text{ Kb}$, $\alpha_1 = 0$, $\alpha_2 = 150\text{ nJ/bit/m}^2$, and $\gamma = 2$.

4.1. Varying Hop Count. We first study the effect of changing the hop count on the network performance. Figure 1 plots the performance of our algorithms and the SPT-DGA in terms of tour length and hop count, h . When h is small, the number of nodes within each CT is small and thus the number of CLs increased, which increases the length of the mobile BS. The result also shows that the tour length in the FNF-BHA algorithm is the shortest since the FNF-BHA algorithm considered the farthest nodes.

The maximum number of nodes in the CTs with the number of hops is shown in Figure 2. The result shows that as the hop count increases the number of nodes in the CTs increased. This makes sense as increasing the number of hops added extra nodes to the CTs. The result also shows that the number of nodes in the CTs constructed by SPT-DGA algorithm is always less than that in our algorithms.

We also study the number of CLs with the hop count as shown in Figure 3. The result shows that the numbers of CLs

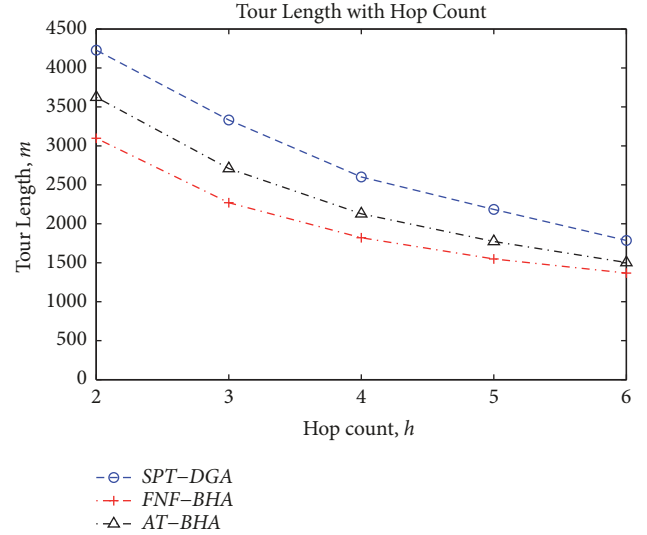


FIGURE 1: The tour length of the mobile BS with the hop count for the SPT-DGA, FNF-BHA and AT-BHA algorithms, for $d = 12$.

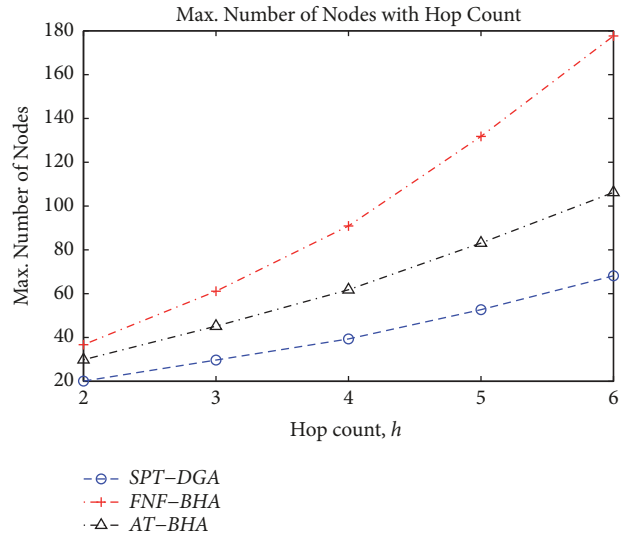


FIGURE 2: The maximum number of nodes in the CT with the hop count for the SPT-DGA, FNF-BHA and AT-BHA algorithms, for $d = 12$.

in our algorithms are significantly less than that in SPT-DGA; this is because the proposed algorithms try to construct CTs with maximum number of nodes by finding routes to the partition nodes and adjacent CTs.

The CL consumes more energy than any other network nodes since it has to forward the sensing data of the CT to the BS. Figure 4 illustrates the maximum energy consumption of CL for forwarding sensing data of a single sensor node by dividing the total energy consumption of the CL to the number of sensors in the CT. The result shows that the energy consumption decreases with the hop count as the number of nodes in the CT increased with hop count. The result also shows that the energy consumption of the CL at FNF-BHC algorithm is lower than other algorithms since the FNF-BHC

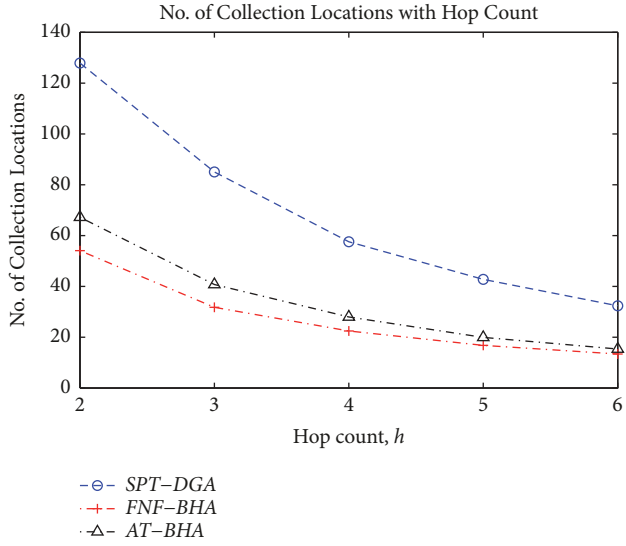


FIGURE 3: The number of CLs with the hop count for the SPT-DGA, FNF-BHA and AT-BHA algorithms, for $d = 12$.

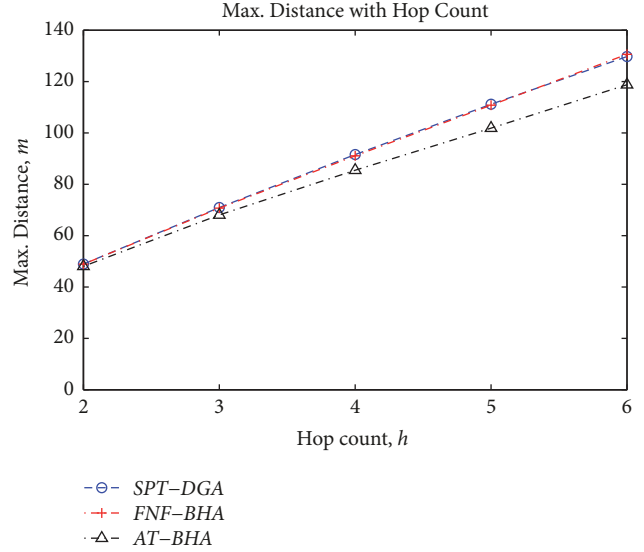


FIGURE 5: The maximum distance between the CL and the farthest sensor node with the hop count for the SPT-DGA, FNF-BHA, and AT-BHA algorithms, for $d = 12$.

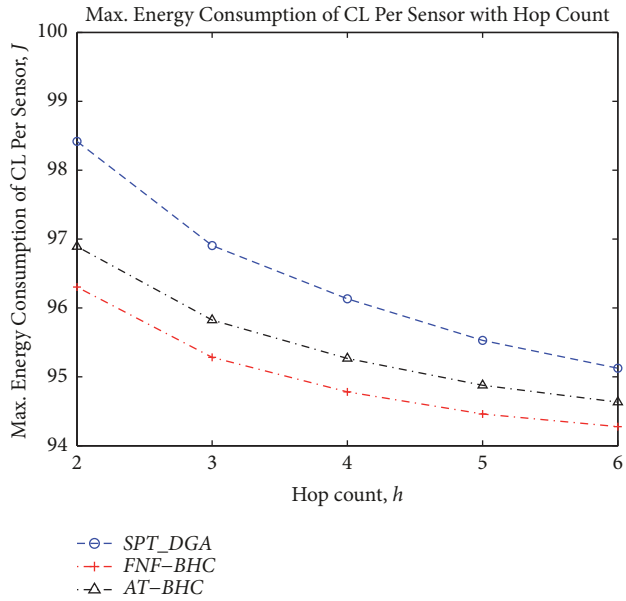


FIGURE 4: The maximum energy consumption of the CL sensor node for forwarding sensing data of single sensor node with the hop count for the SPT-DGA, FNF-BHA, and AT-BHA algorithms, for $d = 12$.

algorithm constructed CTs with the maximum number of nodes.

The maximum distance between the CL and the farthest sensor nodes is examined in Figure 5. The CLs selected by the AT-BHA are the closest to the sensor nodes since the AT-BHA algorithm constructed adjacent trees using hop progress βr rather than dealing with the farthest nodes.

To better understand the results, we give examples in Figures 6, 7, and 8 where 245 sensors are scattered over $200\text{ m} \times 200\text{ m}$ field ($d = 12$) with the initial position of data BS located at the center of the area. The h is set to 4,

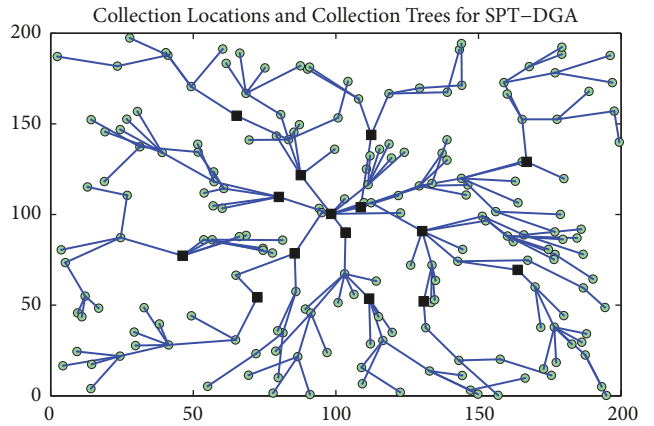


FIGURE 6: An example illustrates the CLs and CTs constructed using SPT-DGA for a network of 244 sensors. The tour length is equal to 599.2 m with 15 CL nodes, $h = 4$ and $d = 12$.

which means that it is required for each sensor to forward its data to the CL within four hops. The figures also show the construction of CTs rooted at CLs for each algorithm.

The SPT-DGA constructs 15 routing trees for data collection as shown in Figure 6. The CLs are selected very close to the center of the sensing area in order to minimize the tour length. However, for the same sensors positions, the AT-BHA and FNF-BHA constructed 6 and 5 routing trees, respectively. The position of the CLs selected in the FNF-BHA is closer to the center of the sensing field than that in AT-BHA.

4.2. Varying Node Degree. We then vary the node degree in the network by varying the number of network nodes according to $n = d * A / (\pi * r^2)$, where A is the network area and r is the node transmission range.

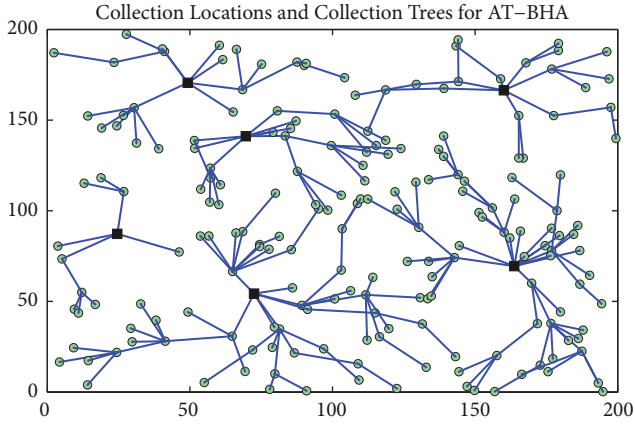


FIGURE 7: An example illustrates the CLs and CTs constructed using AT-BHA for a network of 244 sensors. The tour length is equal to 464.6 m with 6 CL nodes, $h = 4$ and $d = 12$.

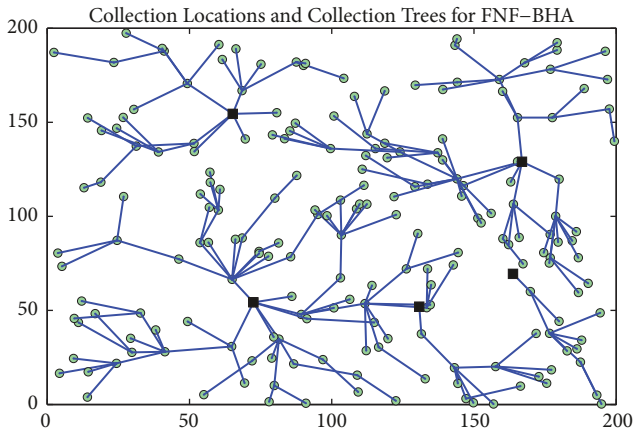


FIGURE 8: An example illustrates the CLs and CTs constructed using FNF-BHA for a network of 244 sensors. The tour length is equal to 360.5 m with 5 CL nodes, $h = 4$ and $d = 12$.

To study the effect of the node degree on the tour length, the tour of the mobile BS is calculated using the nearest neighbor algorithm (as shown in Figure 9 for our algorithms and the SPT-DGA algorithm). It can be seen that, under different d , the tour length for AT-BHC and FNF-BHA is shorter than that for the SPT-DGA algorithm. This is because the FNF-BHC selects the minimum number of CLs considering the farthest nodes to the position of the BS while the SPT-DGA selects more CLs, which increases the length of mobile BS tour. The result also shows that the tour length slightly increases with the node degree in SPT-DGA. However, increasing the node degree reduced the tour length for the AT-BHA and FNF-BHA since that improves network connectivity and finding routing paths to the farthest nodes. This reduces the number of CLs, as shown in Figure 10.

Figure 11 illustrates the maximum number of nodes in the CTs with the node degree as a function of h . The result shows that the number of nodes increases with the node degree since that increases the number of neighbors.

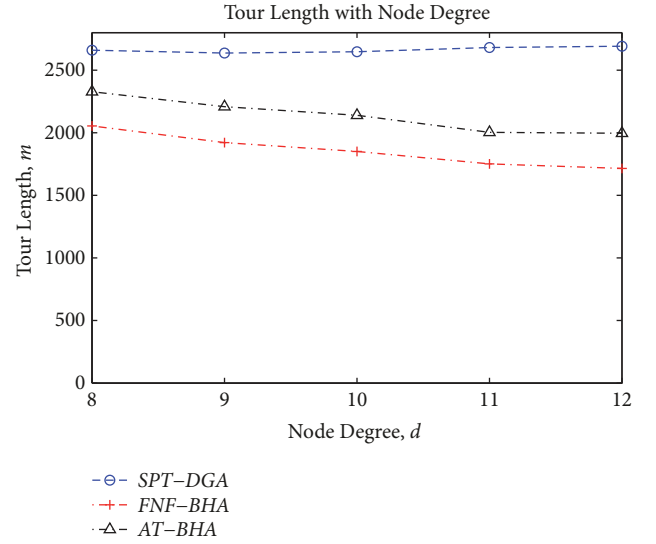


FIGURE 9: The tour length of the mobile BS with the node degree for the SPT-DGA, AT-BHA and FNF-BHA algorithms, for $h = 4$.

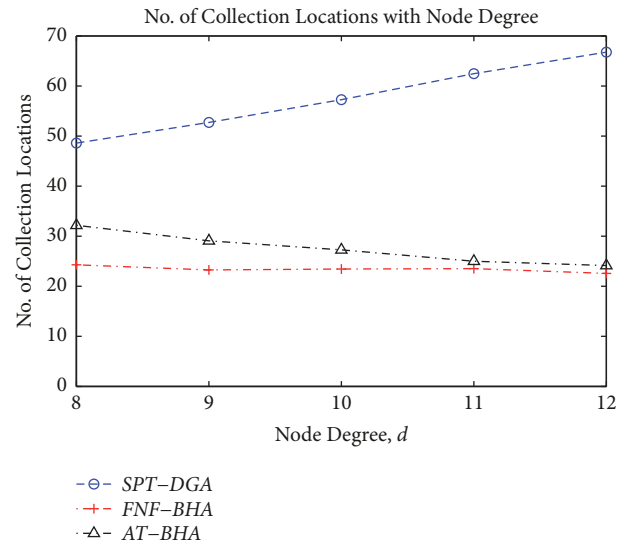


FIGURE 10: The number of CLs with the node degree for the SPT-DGA, AT-BHA and FNF-BHA algorithms, for $h = 4$.

Figure 12 shows the maximum energy consumption of the CL for forwarding sensing data of a single sensor node to the BS. The result shows that the energy consumption slightly decreases with the node degree. In addition, the CL at FNF-BHA consumes less energy than other algorithms since the number of nodes at the CT of the FNF-BHA is more than the number of nodes at the CTs of SPT-DGA and AT-BHA as shown in Figure 11.

We study the distance the packet travels to reach the BS by calculating the distance between the sensor nodes. The distance between the farthest sensor node and the CL is shown in Figure 13. The result shows that the maximum distance is higher in SPT-DGA and FNF-BHA than in AT-BHA. This is because the expansion of CTs in AT-BHA depends on the

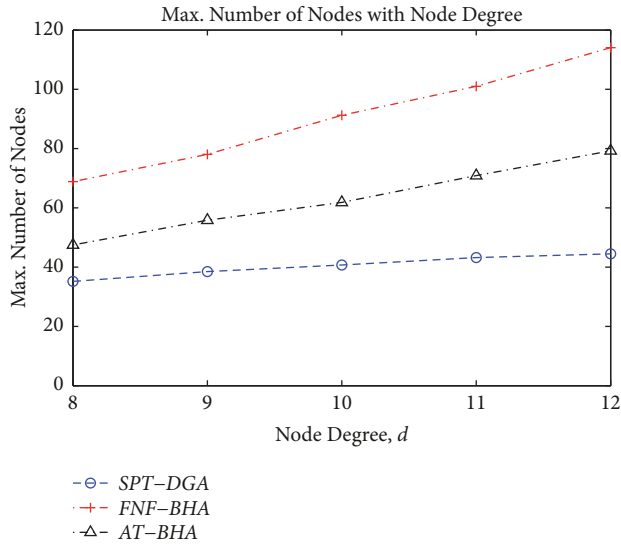


FIGURE 11: The maximum number of nodes with the node degree for the SPT-DGA, AT-BHA, and FNF-BHA algorithms, for $h = 4$.

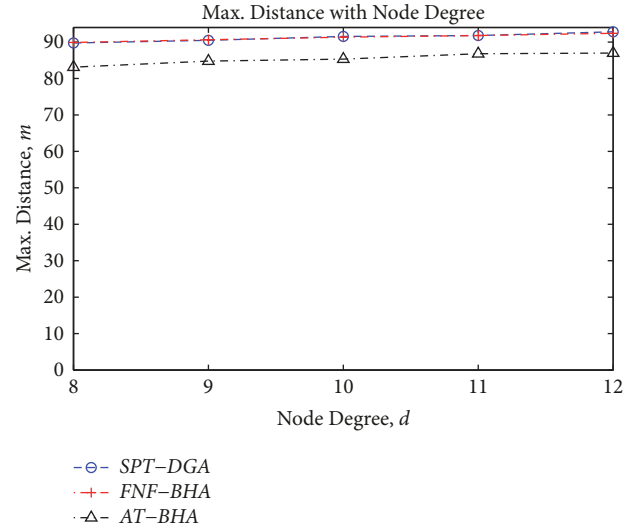


FIGURE 13: The maximum distance between the CL and sensor nodes with the node degree for the SPT-DGA, AT-BHA, and FNF-BHA algorithms, for $h = 4$.

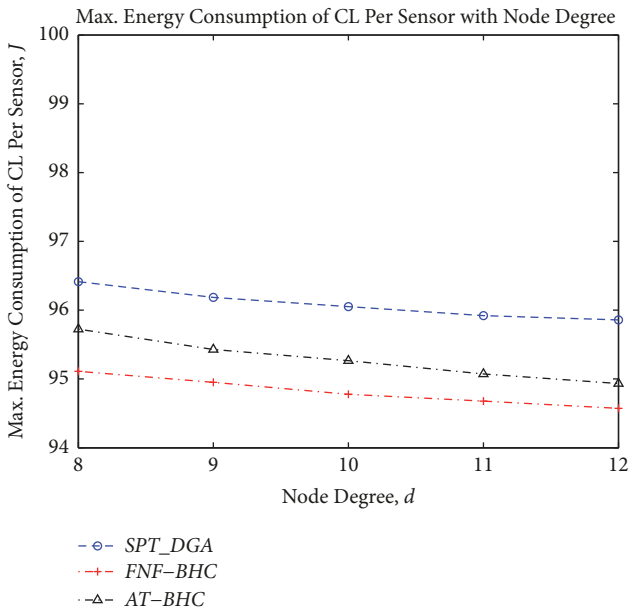


FIGURE 12: The maximum energy consumption of the CL sensor node for forwarding sensing data of single sensor node with the node degree for the SPT-DGA, AT-BHA, and FNF-BHA algorithms, for $h = 4$.

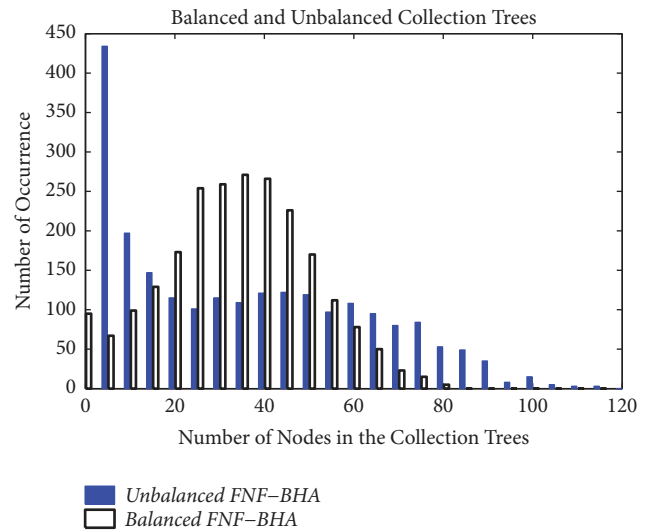


FIGURE 14: Comparison between the number of nodes at balanced and unbalanced CTs for FNF-BHA algorithm with $h = 4$ and $d = 10$.

settings of β and hop count, while it depends on hop count only for the SPT-DGA and FNF-BHA algorithms.

Finally, we study the effect of load balanced algorithm on the distribution of the number of sensor nodes in the CTs. Figure 14 shows the histogram of the number of nodes in the CTs before and after the execution of the load balanced algorithm. The result shows that there is high occurrence of trees, with the number of nodes less than 20 nodes for the unbalanced FNF-BHA. However, the number of nodes is very well distributed for the balanced FNF-BHA.

5. Conclusion

In this paper, we dealt with the problem of data gathering in the mobile BS environment. We studied the trade-off between the relay hop count of sensor nodes and the tour length of the mobile BS. Two heuristic algorithms, AT-BHA and FNF-BHA, are proposed towards finding the shortest tour length for the mobile BS subject to bounded hop count. The AT-BHA selects adjacent CTs during tree construction to avoid partitioned nodes that increased the tour length. The FNF-BHA uses the farthest nodes from the BS in the construction of the CTs. Extensive simulations have been carried out to validate the efficiency of the proposed algorithms against SPT-DGA. The experimental results demonstrated that the

proposed algorithms outperform the SPT-DGA significantly in terms of minimizing the BS tour length.

Data Availability

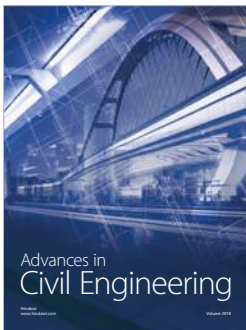
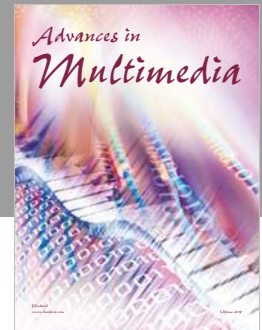
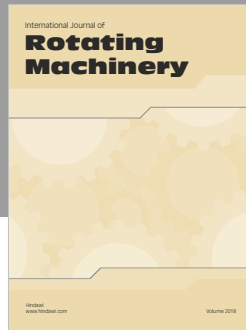
The data used in this study are generated via the simulation of the proposed algorithms. The proposed algorithms are provided and discussed within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] H. Huang and A. V. Savkin, "Optimal path planning for a vehicle collecting data in a wireless sensor network," in *Proceedings of the Chinese Control Conference (CCC '16)*, pp. 8460–8463, IEEE, July 2016.
- [2] O. D. Jerew and N. A. Bassam, "Energy aware and delay-tolerant data gathering in sensor networks with a mobile sink," in *Proceedings of the MEC International Conference on Big Data and Smart City (ICBDSC '16)*, pp. 1–5, IEEE, March 2016.
- [3] O. Jerew, K. Blackmore, and W. Liang, "Mobile Base Station and Clustering to Maximize Network Lifetime in Wireless Sensor Networks," *Journal of Electrical and Computer Engineering*, vol. 2012, Article ID 902862, 13 pages, 2012.
- [4] Z. Xu, W. Liang, and Y. Xu, "Network lifetime maximization in delay-tolerant sensor networks with a mobile sink," in *Proceedings of the 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 9–16, IEEE, 2012.
- [5] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *IEEE Transactions on Computers*, vol. 61, no. 2, pp. 265–277, 2012.
- [6] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the Sensor Network Protocols and Applications (SNPA '03)*, pp. 30–41, IEEE, 2003.
- [7] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Proceedings of the International Conference on Information Processing in Sensor Networks*, pp. 129–145, 2003.
- [8] P. Baruah and R. Urgaonkar, "Learning-enforced time domain routing to mobile sinks in wireless sensor fields," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 525–532, 2004.
- [9] F. N. Huang, T. Y. Chen, S. H. Chen, H. W. Wei, T. S. Hsu, and W. K. Shih, "ShareEnergy: Configuring Mobile Relays to Extend Sensor Networks Lifetime Based on Residual Power," in *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS '16)*, pp. 478–484, IEEE, Oct 2016.
- [10] Z. Chen, L. Kang, X. Li, J. Li, and Y. Zhang, "Constructing load-balanced Degree-constrained Data Gathering Trees in Wireless Sensor Networks," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC '15)*, pp. 6738–6742, IEEE, June 2015.
- [11] S. Gao and H. Zhang, "Energy efficient path-constrained sink navigation in delay-guaranteed wireless sensor networks," *Journal of Networks*, vol. 5, no. 6, pp. 658–665, 2010.
- [12] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*, pp. 111–124, ACM, 2004.
- [13] M. Ma and Y. Yang, "SenCar: An energy-efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1476–1488, 2007.
- [14] E. M. Saad, M. H. Awadalla, M. A. Saleh, H. Keshk, and R. R. Darwish, "A data gathering algorithm for a mobile sink in large-scale sensor networks," in *Proceedings of the 4th International Conference on Wireless and Mobile Communications (ICWMC '08)*, pp. 207–213, IEEE, 2008.
- [15] X. Zhang, L. Zhang, G. Chen, and X. Zou, "Probabilistic path selection in wireless sensor networks with controlled mobility," in *Proceedings of the International Conference on Wireless Communications Signal Processing*, pp. 1–5, IEEE, 2009.
- [16] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 231–240, 2008.
- [17] A. Kaswan, P. K. Jana, and M. Azharuddin, "A delay efficient path selection strategy for mobile sink in wireless sensor networks," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI '17)*, pp. 168–173, IEEE, 2017.
- [18] A. Kaswan, K. Nitesh, and P. K. Jana, "Energy efficient path selection for mobile sink and data gathering in wireless sensor networks," *AEU - International Journal of Electronics and Communications*, vol. 73, pp. 110–118, 2017.
- [19] L. Chen, X. Jianxin, X. Peng, and X. Kui, "An energy-efficient and relay hop bounded mobile data gathering algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 680301, 9 pages, 2015.
- [20] Y. Chen, X. Lv, S. Lu, and T. Ren, "A Lifetime Optimization Algorithm Limited by Data Transmission Delay and Hops for Mobile Sink-Based Wireless Sensor Networks," *Journal of Sensors*, vol. 2017, Article ID 7507625, 11 pages, 2017.
- [21] M. Mishra, K. Nitesh, and P. K. Jana, "A delay-bound efficient path design algorithm for mobile sink in wireless sensor networks," in *Proceedings of the 3rd International Conference on Recent Advances in Information Technology (RAIT '16)*, pp. 72–77, IEEE, 2016.
- [22] C. Cheng and C. Yu, "Mobile data gathering with bounded relay in wireless sensor network," *IEEE Internet of Things Journal*, pp. 1–16, 2018.
- [23] C. Zhu, S. Wu, G. Han, L. Shu, and H. Wu, "A tree-cluster-based data-gathering algorithm for industrial WSNs with a mobile sink," *IEEE Access*, vol. 3, no. 4, pp. 381–396, 2015.
- [24] J.-Y. Chang and T.-H. Shen, "An efficient tree-based power saving scheme for wireless sensor networks with mobile sink," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7545–7557, 2016.
- [25] O. Jerew and K. Blackmore, "Estimation of hop count in multi-hop wireless sensor networks with arbitrary node density," *International Journal of Wireless and Mobile Computing*, vol. 7, no. 3, pp. 207–216, 2014.
- [26] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 2001.



Hindawi

Submit your manuscripts at
www.hindawi.com

