

## Research Article

# GLT: Grouping Based Location Tracking for Object Tracking Sensor Networks

**Ammar Hawbani,<sup>1</sup> Xingfu Wang,<sup>1</sup> Saleem Karmoshi,<sup>1</sup> Hassan Kuhlani,<sup>1</sup>  
Aiman Ghannami,<sup>1</sup> Adili Abudukelimu,<sup>1</sup> and Rafia Ghoul<sup>2</sup>**

<sup>1</sup>University of Science and Technology of China, Hefei, China

<sup>2</sup>Hunan University, Hunan, China

Correspondence should be addressed to Xingfu Wang; wangxfu@ustc.edu.cn

Received 21 December 2016; Revised 16 March 2017; Accepted 9 April 2017; Published 18 May 2017

Academic Editor: Pierre Leone

Copyright © 2017 Ammar Hawbani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The use of wireless sensor networks (WSN) in tracking applications is growing rapidly. In these applications, the nodes detect, monitor, and track a target, object, or event. In this paper, we consider the problem of tracking mobile objects in wireless sensor networks (WSN). We present a novel tracking model, named Grouping based Location Tracking (GLT), scaling well with the number of nodes and the number of mobile objects. GLT is based on the Grouping Hierarchy Structure, GHS. In GHS, nodes are partitioned into groups (not clusters) according to their maximum covered region (MCR) such that each group contains a number of nodes and a number of leaders. GLT consists of two tiers. The first tier, which is called the Notification Tree (NT), enhances the activation mechanism, the data cleaning mechanism, and the energy balancing mechanism. On the other hand, the second tier, which is called the Hierarchical Spanning Tree (HST), supports the data reporting mechanism and the lifetime prolonging mechanism. Simulations results show that GLT reduces the communication node selections overhead without diminishing object tracking accuracy and achieves a significant energy consumption reduction and network lifetime extension compared with the state-of-the-art approaches.

## 1. Introduction

In contrast to sophisticated surveillance technologies such as RADARS, which are in fact reliable, robust, and accurate but expensive, WSN enable a cheap technology which do not rely on any centralized infrastructure [1, 2]. WSN is a collection of nodes, which are of low cost, low power, and small size, carrying out the task of sensing, performing simple data processing, and communicating wirelessly over a short distance. The node contains three main subsystems: (1) the sensing subsystem contains one or more physical sensor devices and one or more analog-to-digital converters as well as the multiplexing mechanism to share them. (2) The processing subsystem executes instructions pertaining to sensing, communication, and self-organization. (3) The communication subsystem contains the transmitter and receiver for sending or receiving information [3–5].

Location tracking has many applications in our daily life in different areas, for example, military intrusion detection,

vehicular networks, and habitat monitoring [6]. These applications must periodically collect the sensory data (locations, velocities, and trajectories) and use it to reconstruct the overall status of the monitored area through data aggregation. When the sensor nodes detect an event, they record it and rely on a distributed routing protocol to send the relevant information toward a base station or sink. The target events must be detected with an acceptable degree of accuracy. In addition, the event report should be received by the sink in a very short time, especially for time-critical events [7].

In object tracking sensor networks, power consumption is a key issue, which directly affects network lifetime. The power can be conserved in object tracking sensor networks by switching a sensor node to sleep when there is no object in the node's sensing range and turn on a sensor node when an object is to enter the node's sensing range [8]. Energy conservation can be achieved using different methods: prediction and nonprediction [1]. In prediction-based

tracking models [9–11], the nodes are woken up on demand following the target path. The previous active nodes predict the activation zone in which not all the nodes can effectively detect the target but only a subset of them. On the other hand, in nonprediction tracking, the nodes are periodically put to sleep in order to conserve energy and they wake up after a period of time to monitor the sensing region and collect the object movement information [1, 8]. Nevertheless, prediction and nonprediction strategies have shortcomings such as missing target trajectory and duplicated packet reporting and the predictions may not be entirely correct. This leads to loss of object, and large amount of energy may be wasted in recovering the object. Additionally, in object tracking sensor networks, the data aggregation model, nodes activation model, and data reporting model play an essential role that could maximize the lifetime of network [12–15]. Those models should be designed to reduce the communication overhead in the network taking in to consideration the hardware limitations of nodes.

With these motivations, we propose GLT, a model to detect movements of a target using Grouping Hierarchy Structure (GHS) [16], which does not fall into existing tracking categories and is, to the best of our knowledge, the first of its kind. GLT is a two-tier tracking and reporting tree. The first tier is the NT. This tier is designed to achieve three goals. (1) Activation mechanism: it notifies the nodes in the Notification Tree to wake up and collect the data. (2) Data cleaning mechanism: when the object is detected in an overlapped region, NT prevents the redundant sensed data during data aggregation. (3) Energy balancing mechanism: the nodes in the NT consume approximately the same amount of energy. On the other hand, the second tier of GLT is HST (Hierarchical Spanning Tree). HST is designed to work as two mechanisms: data reporting mechanism and lifetime prolonging mechanism. HST and NT are working together to achieve their best performance in terms of network lifetime, tracking accuracy, traffic reduction, and transmission delay reduction.

The major contributions of this paper include the following: (1) Notification Tree: we proposed NT to wake up the nodes in the activation zone in order to reduce the overhead of the proxy node. Moreover, NT reduces the communication overhead during data aggregation. (2) Hierarchical Spanning Tree: to reduce the communication overhead from the source node (aggregation node) to the sink, we proposed HST, which is based on GHS [16]. (3) GLT: it is a combination of NT and HST to ensure tracking accuracy and to minimize the data reporting delay. (4) Find a closed form for the number of deployment patterns for  $n$  nodes. The remainder of this paper is organized as follows. In Section 2, the related work is presented. Grouping structure, NT, and HST are explained in Section 3. In Section 4, we have given an analysis for GHS and NT; moreover, we have provided a mathematical model for grouping of GHS, which is designed to compute the average communication cost and network deployment patterns. The simulation results are shown in Section 5. Finally, Section 6 concludes this work.

## 2. Related Work

Tracking mobile object using WSN is a hard problem, since the nodes have limited power and computational capability. In recent years, tracking mobile objects in WSN has gained a great deal of attention. It is classified into three categories: tree-based tracking [17–20], cluster-based tracking [12–15, 21], and prediction-based tracking [9–11].

In case of prediction-based tracking [9–11], the next location of mobile object is calculated using distributed structure or using the historical data or the information of object. During each defined time step, only some nodes near the predicted location are activated while the other nodes stay in sleep mode to save energy. Different approaches with different types of measurements have been used in target tracking systems, such as Angle of Arrival (AOA), Received Signal Strength (RSS), Time of Arrival (TOA), and the Time Difference of Arrival (TDOA). Additionally, Variational Filter (VF), Kalman Filter, Extended Kalman Filters (EKF) [9], Distributed Kalman Filter, Particle Filter, and data mining [10, 11] are the most commonly used models to predict the next location of object.

The prediction-based models have a shortcoming such as missing the trajectory of object and the predictions may not be entirely correct. This leads to loss of object, and large amount of energy may be wasted in the recovery mechanism for the object. In tree-based target tracking [17–20], nodes in a network organized in a hierarchical tree or represented as a graph in which vertices represent sensor nodes and edges link between nodes that can directly communicate with each other. Scalable Tracking Using Networked Sensors (STUN) [17] is a tree-based tracking algorithm that scales well to large numbers of sensors and moving objects by using hierarchy. It utilizes drain-and-balance (DAB) method to build an efficient tracking hierarchies implemented in 1D and 2D sensor network topologies. The total communication cost of STUN is reduced in [18] by extending the location part. In [19] the authors improved about 9.9% of energy consumption compared with shortest path tree (SPT). They utilized heuristic object tracking algorithm that formulates the problem as 0/1 integer programming. STUN has a drawback; the drain-and-balance tree does not replicate physical sensor network as it is in the logical tree; hence an edge may consist of multiple communication hops.

In cluster-based tracking [12–15, 21], the nodes are organized into clusters, with each cluster consisting of a cluster head (CH) and several neighboring sensors (members). The cluster-based tracking is further divided into static cluster-based tracking and dynamic cluster-based tracking. In static cluster-based target tracking, the clusters are formed when the nodes are deployed and the clustering nodes and CHs are defined before target tracking. However, in the dynamic clustering, the sensor nodes are from different clusters at different times according to the movement of the target. Clustering-based networks suffer from more traffic due to cluster members and cluster heads intercommunication. Additionally, more energy will be dissipated when the nodes, which belong to multiple clusters, detect one target. This leads to redundant data transmissions from cluster heads to the

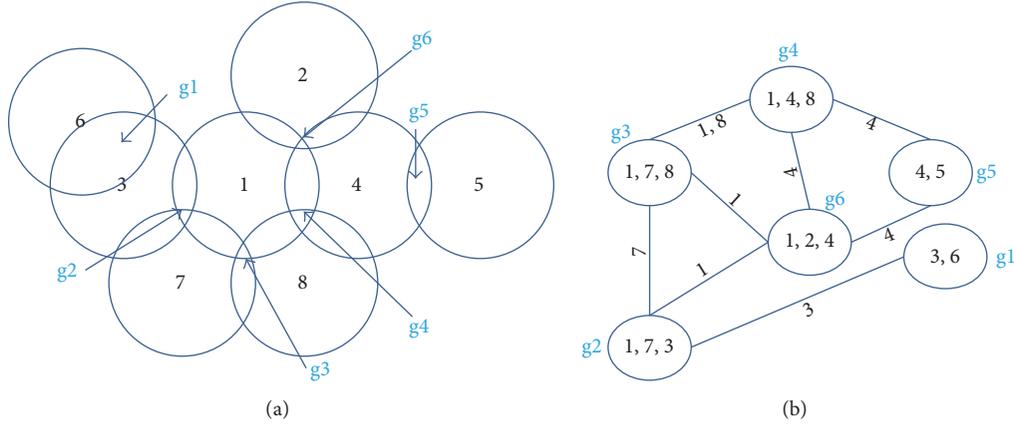


FIGURE 1: Random network and corresponding groups. (a) 8 randomly deployed nodes (the arrows indicate the maximum covered areas). (b) The groups according to maximum covered areas.

sink. This issue is avoided in [12] (CAICS) and this is why it archives better energy saving compared to DLC [13, 14] and HMC [15, 21]. In DLC, the packets are transmitted to the sink over a long distance since each cluster head aggregates the sensed data from its cluster members and directly transmits it to the sink. This issue is managed in HMC; HMC adopts a hierarchical cluster heads model and the data can be transmitted to the sink hop by hop through multiple intermediate cluster heads. In [12] a coordinated and adaptive information collecting strategy (CAICS) for target tracking WSN is proposed. CAICS uses a joint distance weighted measurement to estimate the information utility of sensing nodes. CAICS achieves a very good performance in terms of number of control packets and energy saving. However, due to the frequent selection of aggregation node in CAICS, the transmission delay is increased as the number of nodes increases. In addition, when the velocity of the target is getting higher the transmission delay in CAICS will be increased.

### 3. Proposed GLT Model

GLT is a tree-based tracking model. It consists of two tiers. The first tier, the activation zone (*Notification Tree*), enhances the activation mechanism, data cleaning mechanism, and the energy balancing mechanism. The second tier is called Hierarchical Spanning Tree. It supports data reporting mechanism and lifetime prolonging mechanism. The first and second tiers are based on the GHS [16]. Before going any further, we first explain GHS, which is explained comprehensively in [16].

**Definition 1** (neighbor list). For any node  $s_a \in S$ , its neighbor list is defined as a set of nodes  $nl(s_a) = \{s_b \mid s_b \in S \text{ and } \sqrt{(x_{s_a} - x_{s_b})^2 + (y_{s_a} - y_{s_b})^2} < r_a + r_b\}$ , where  $r_a$  and  $r_b$  are the sensing ranges for  $s_a$  and  $s_b$ , respectively;  $S$  is the set for all sensor nodes in the network.  $nl(S) = \{nl(s_i) \mid s_i \in S\}$  is an array of neighbor lists.

**Definition 2** (group). A set of nodes, noted as  $G^k = \{s_0, \dots, s_j\} \forall s_i \in S$ , is said to be a group of size  $k$  (i.e., contains

$k$  nodes) if and only if  $G^k = \{s_0, \dots, s_k\}$  occurs exactly  $k$  times as a subset in the array  $nl(S)$ . The array of all groups in the network is denoted by  $G^*$ . For example, the six nodes in Figure 2 are partitioned into three groups  $g_1 = G^3 = \{s_1, s_2, s_6\}$ ,  $g_2 = G^3 = \{s_1, s_3, s_5\}$ , and  $g_3 = G^3 = \{s_1, s_3, s_4\}$ .

**Definition 3** (associated groups). The associated groups for  $s_j$  are defined as a set of groups  $G^*(s_j) = \{G^k \mid G^k \in G^* \text{ and } s_j \in G^k\}$ . For example, in Figure 2, the associated groups for each node are  $G^*(s_1) = \{g_1, g_2, g_3\}$ ,  $G^*(s_2) = \{g_1\}$ ,  $G^*(s_3) = \{g_2, g_3\}$ ,  $G^*(s_4) = \{g_3\}$ ,  $G^*(s_5) = \{g_2\}$ , and  $G^*(s_6) = \{g_1\}$ , where  $g_1 = G^3 = \{s_1, s_2, s_6\}$ ,  $g_2 = G^3 = \{s_1, s_3, s_5\}$ , and  $g_3 = G^3 = \{s_1, s_3, s_4\}$ .

#### 3.1. Structure Models

**3.1.1. Network Model.** Network's nodes are divided into groups (*Definition 2*) such that the all nodes are represented by  $G^* = \{G^i, G^j, G^k, \dots\}$ , where  $i, j$ , and  $k$  are integers (indicating the number of nodes in  $G^i, G^j$ , and  $G^k$ , resp.). Consider a graph  $G = (V, E, \mu)$  to represent the network, where  $V = G^* = \{G^i, G^j, G^k, \dots\}$ ,  $E$  is the edge set such that  $\forall G^i \in G^* \& G^j \in G^* E = \{s_e \mid s_e \in G^i \& s_e \in G^j\}$  and  $\mu$  indicates the weight function where  $\mu : E \rightarrow \mathbb{Z}^+$  such that  $\forall s_e \in E : \mu(s_e) = e$ . The edge is a node that belongs to two groups also known as leader node. An example of this network model is shown in Figure 1: (a) shows the deployed nodes; (b) shows the network model, which contains the groups (vertices) and the edges (leader nodes).

**3.1.2. Activation Zone Model.** Given a set of mobile objects  $O_m = \{o_0, o_1, o_2, \dots, o_m\}$ ; at a given time, each  $o_u \in O_m$  moves in the field randomly and should be tracked by one node  $s_j \in S$ . To continuously track  $o_e$ , the node  $s_j$  set up an activation zone (*Notification Tree*). This tree is rooted at  $s_j$  and contains all nodes in  $G^*(s_j)$ . The nodes in  $G^*(s_j)$  are partitioned into subtrees and these subtrees are attached to the Notification Tree  $NT(s_j)$ . The nodes in  $G^*(s_j)$  are partitioned into subtrees as follows: given graph  $G_j = (V_j, E_j)$ , where  $V_j = \{s_a \mid s_a \in G^*(s_j)\}$  and  $E_j \subseteq [V_j]^2$ , take  $s_j$  as cut vertex in

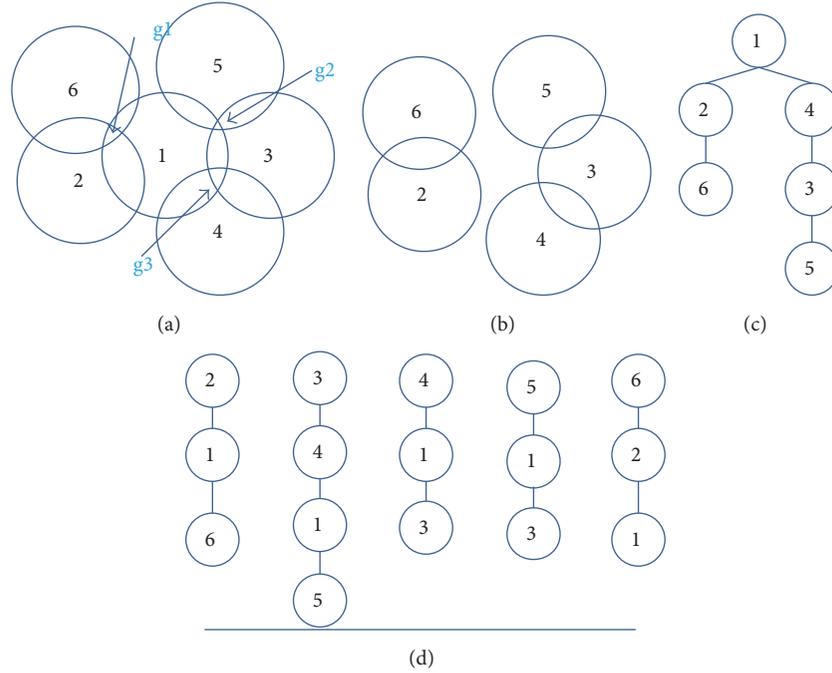


FIGURE 2: Notification Tree or the activation zone for each node. (a) 6 random nodes; (b)  $C_1 = \{c_{1,0}, c_{1,1}\}$ ; (c)  $NT(s_1)$ ; (d)  $NT(s_2), NT(s_3), NT(s_4), NT(s_5),$  and  $NT(s_6)$ .

$G_j$  such that  $C_j = G_j - s_j = \{c_{j,0}, c_{j,1}, \dots, c_{j,i}\}$ , where  $c_{j,i}$  is a component or a subgraph in  $G_j$ . Each subgraph  $c_{j,i} \in C_j$  builds a subtree  $t_{j,i}$ , such that all subtrees are  $T_j = \{t_{j,0}, t_{j,1}, \dots, t_{j,i}\}$ . All the trees in  $T_j$  are attached to  $NT(s_j)$ . For example, in Figure 2, to build the  $NT(s_1)$ , we take  $s_1$  as a cut vertex; the generated subgraphs are  $C_1 = \{c_{1,0}, c_{1,1}\}$ , where  $c_{1,0} = \{s_2, s_6\}$  and  $c_{1,1} = \{s_3, s_4, s_5\}$  (see Figure 2(b)). Thus,  $NT(s_1)$  contains two subtrees. The first subtree contains the nodes  $s_2$  and  $s_6$ , while the second subtree contains  $s_3, s_4$ , and  $s_5$ . Finally, two subtrees are attached to the root node  $s_1$  as shown in Figure 2(c). The notifications trees of other nodes  $NT(s_2), NT(s_3), NT(s_4), NT(s_5)$ , and  $NT(s_6)$  are shown in Figure 2(d).

**3.1.3. Object Tracking Model.** We define three states for each node: active, idle, and sleep. In active state, the node collects the sensory data and reports it to the aggregation node, while in idle state, the node gets ready to track object and waits for an activation from the currently selected proxy node. In the sleep state, no sensory data could be collected. We assume that the nodes support two types of messages (i.e., control packets): (1) **INSERT** control packet: this control packet is sent from the currently selected proxy node (the node that detects the object). The main role of this control packet is to let the receiver nodes change their state from sleep to idle and be ready for the object when the object crosses their sensing ranges. The other role of this control packet is to let all nodes in the activation zone know that there is a proxy node that has been selected, and they should not be activated until the proxy node send them **DELETE** packet. (2) **DELETE** control packet: this control packet is sent from the proxy node to let the nodes in its NT know that the object is out of its range

currently, and one of NT nodes should be selected as a new proxy node. All the receiver nodes should change their state from idle to sleep excluding the new selected proxy node.

We consider that any  $o_u \in O_m$  has only one proxy node  $s_j \in S$  at any given moment. When the object  $o_u$  moves in  $k$ -covered area ( $k > 1, k \in \mathbb{Z}$ ), the proxy node can be selected upon three main parameters: (1) the residual energy of node (i.e., the node with more energy wins), (2) the distance to object (i.e., the nearest node wins), and (3) the current state of node (i.e., busy or free). For  $s_j \in S$ , we assume that it has a Sensed Objects Table denoted by  $SOT(s_j)$  to save the information of the sensed object (i.e., current position, image, etc.) and a Notifications Messages Table denoted by  $NMT(s_j)$  to store the notification sent/received to/from other nodes (see Figure 3). When  $o_u \in O_m$  is detected by  $s_j, s_j$  will create a new entry in  $SOT(s_j)$  for  $o_u$  if and only if  $s_j$  has no notification record received from other  $s_x \in G^*(s_j)$ . When  $s_j$  creates a new entry in  $SOT(s_j)$  it will send an **INSERT** notification message to its activation zone  $G^*(s_j)$  via its Notification Tree  $NT(s_j)$ .  $s_j$  will notify all nodes in its activation zone and ask them to add the notification message to their  $NMT$  tables; hence the other nodes in  $G^*(s_j)$  will not sense  $o_u$  until  $s_j$  sends a **DELETE** notification message indicating that  $o_u$  has gone out of its range. After receiving **DELETE**, all nodes in  $G^*(s_j)$  will delete the notification messages and  $o_u$  will enter any other node in  $G^*(s_j)$ . The tracking model is explained in the main objectives of NT below.

### 3.2. The Two Tiers

**3.2.1. First Tier: The Notification Tree.** In this subsection, we introduce the first tier of GLT. We first explain how to build

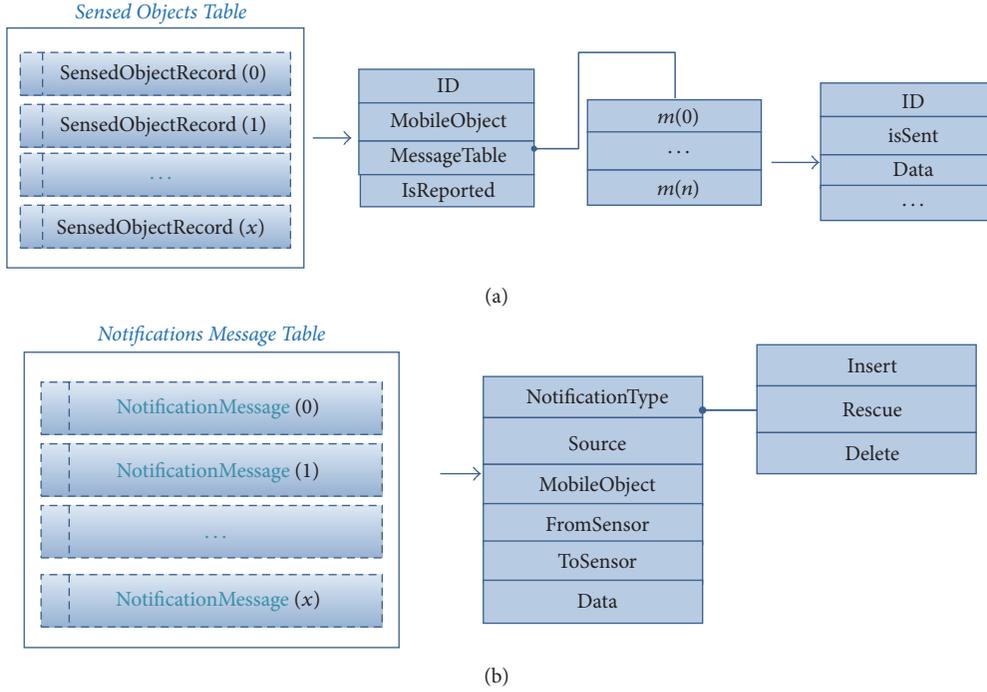


FIGURE 3: Basic storage tables: (a) Sensed Objects Table. (b) Notification Message Table.

the NT and then explain its main goals. Note that each vertex is a group of nodes according to Definition 2. Let the vertex set of subgraph  $c_{j,i}$  be denoted by  $V_{j,i}$ ; the process to build the subtree  $t_{j,i}$  for the subgraph  $c_{j,i}$  is as follows:

- (1) Select a vertex  $v_{\min} \in V_{j,i}$  with the minimum degree  $\delta(G_{j,i})$ .
- (2) Assume that  $v_{\min} = \{s_0, s_1, \dots, s_p\}$ . Select a node  $s_b \in v_{\min}$  ( $b \leq p$ ,  $s_b$  is not a leader) to be the root of  $t_{j,i}$  and then link them; that is, if we select  $s_0$  to be the root then the linking operation will be as follows:  $s_0$  is the parent of  $s_1$ ,  $s_1$  will be the parent of  $s_2, \dots$ , and  $s_{p-1}$  will be the parent of  $s_p$ ;  $L_k = \{s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_p\}$ .
- (3) Select a leader node  $s_l$  from  $v_{\min}$  and then link it to the tail of  $L_k$ :  $L_i = \{s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_p \rightarrow s_l\}$ . If  $L_k$  is empty then  $l_i$  will be the root node of  $t_{j,i}$ .
- (4) Mark  $v_{\min}$  as visited. Then select  $v_z$  as an incident of  $v_{\min}$  such that  $l_i \in v_z$  and  $l_i \in v_{\min}$ .
- (5) Repeat the steps from (2) to (4) for  $v_z$  until all nodes are marked.

The main objectives of NT are as follows.

(1) *Activation Mechanism.* In GLT, the activation zone is the nodes in the NT only. When any object  $o_u \in O_m$  moves within the range of  $s_j \in S$ , all nodes in  $NT(s_j)$  will be in idle state except  $s_j$  that will be in active state. That is to say, the root of  $NT(s_j)$  (namely,  $s_j$ ) will be in active state, while the other nodes will be in idle state. When  $o_u$  moved out the range of  $s_j$ , it will move to one node, say  $s_p$ , in  $NT(s_j)$ .  $s_p$  will be selected as new proxy node and will be activated by the last proxy  $s_j$ , while the other nodes in  $NT(s_j)$  will

be in the idle state. After selecting  $s_p$  as proxy node,  $o_u$  will be tracked by  $NT(s_p)$ . Unlike GLT, in the prediction-based protocols [1, 8], the previous active nodes predict the activation zone in which not all nodes can detect the target but only a subset of them. On the other hand, in the nonprediction-based protocols which define a schedule or a plan for nodes activation and/or sleeping, following the object trajectory, some of the nodes are woken up to execute the sensing and the communication tasks for a scheduled time, and then they change their states to the sleep state. Both of prediction-based protocols and nonprediction-based protocols may contain some prediction errors that may lead to more energy dissipation. Although a recovery mechanism is provided [10], the recovery mechanism consumes much energy. In GLT, there is no need for recovery mechanism since the NT is not prediction based.

(2) *Data Cleaning Mechanism.* NT provides a dynamic data cleaning mechanism by avoiding the redundant sensed data. When the proxy node is selected, it will send an INSERT message to its children and its children will pass the message to their children downward until all nodes in the activation zone are notified and change their state to idle. However, when the nodes received the message, they will get ready for the mobile target and will not start sensing and collecting the location of the target. When the target goes out the range of current proxy, the proxy node will send a DELETE notification to all nodes in the activation zone (the nodes in the NT of the current proxy). When any node received a DELETE notification, it will not execute the DELETE command immediately. Nevertheless, it will check first whether the object is currently in its range. When the object moves in  $k$ -covered area ( $k > 1$ ,  $k \in \mathbb{Z}$ ), the proxy node can be selected

upon three main parameters: the residual energy of node (i.e., the node with more energy wins), the distance to object (i.e., the nearest node wins), and the status of the node (i.e., busy or free).

(3) *Energy Balancing Mechanism*. When there is a notification message (i.e., INSERT, DELETE) to be sent from  $s_j$  to its associated groups (i.e., activation zone  $G^*(s_j)$ ),  $s_j$  will not send it to all nodes in  $G^*(s_j)$  separately. However,  $s_j$  sends the notifications to its children in  $NT(s_j)$ . Additionally, the needed energy per notification will be distributed among all the nodes in  $G^*(s_j)$ . To clarify the importance of the Notification Tree, we use the First-Order Model [16] to evaluate the consumed energy by each node during intercommunications. The First-Order Model is explained in (see [16] for symbols used in the equations)

$$E_{Tx}(d, k) = \begin{cases} (k \cdot E_{elec}) + (k \cdot \epsilon_{fs} \cdot d^2) & d < t_0 \\ (k \cdot E_{elec}) + (k \cdot \epsilon_{mp} \cdot d^4) & d \geq t_0 \end{cases} \quad (1)$$

$$E_{Rx}(k) = k \cdot E_{elec}.$$

When  $s_j$  has a notification to be sent to  $G^*(s_j)$ , without using NT, it will dissipate the energy of  $s_j$  according to (2).  $T_N$  is the total energy needed to send the notifications message to all nodes in  $G^*(s_j)$ . In (2),  $T_N$  is consumed by node  $s_j$ :

$$T_N(E_{Tx}(d, k)) = |G^*(s_j)| * \begin{cases} (k \cdot E_{elec}) + (k \cdot \epsilon_{fs} \cdot d^2) & d < t_0 \\ (k \cdot E_{elec}) + (k \cdot \epsilon_{mp} \cdot d^4) & d \geq t_0, \end{cases} \quad (2)$$

where  $|G^*(s_j)|$  is the number of nodes in  $G^*(s_j)$ . By using NT, the energy, which is needed to let all nodes in  $G^*(s_j)$  know that  $s_j$  detects the object, is distributed among the nodes in  $G^*(s_j)$  as shown in

$$T_N(E_{Tx}(d, k)) = \sum_{\forall s_i \in G^*(s_j)} \begin{cases} (k \cdot E_{elec}) + (k \cdot \epsilon_{fs} \cdot d^2) & d < t_0 \\ (k \cdot E_{elec}) + (k \cdot \epsilon_{mp} \cdot d^4) & d \geq t_0. \end{cases} \quad (3)$$

**3.2.2. The Second Tier: The Hierarchical Spanning Tree (HST).** In this section, we explain the second tier of GLT, which is called Hierarchical Spanning Tree (HST). The data reporting mechanism and the lifetime prolonging mechanism are the two main objectives of this tier. The first tier achieves data collection process. However, the length of the first tier is not enough to reach the sink due to the limited height of NT. On the contrary, the height of HST expands to reach all nodes in the network. Nevertheless, the NT tree plays an important role in the data reporting mechanism; NT excludes the nodes, which have lower energy than threshold; this is an important role, which enhances the lifetime prolonging mechanism of HST. We first show how to construct the HST; second, we explain the reporting mechanism using HST. Finally, we show

how HST and NT cooperate to achieve network lifetime prolonging.

(1) *Construction Process of HST*. Here we explain how to use the Notification Tree and Notification Spanning Tree to build the Hierarchical Spanning Tree (HST); see Figure 4. The process of building HST tree is based on NT. For the convenience, let  $HST = \{L_0, L_1, L_2, \dots, L_x\}$  be the levels of the tree,  $1 \leq x \leq n/2$ , where  $n = |S|$  is the number of nodes in the network. Each level contains a set of nodes such that  $L_i = \{s_0, s_1, \dots\}$ . We assume that the root node (sink node) is located in the first level of the tree; for example, in Figure 4,  $L_0 = \{s_{15}\}$ . For the nodes already existing in HST we mark them by AIT. The process of building of HST tree runs from the sink to the leaves (top-down) as follows.

- (1) For the root node, it will be added to  $L_0 = \{s_r\}$ , and then for each node in the Notification Tree  $NT(s_r) = \{s_r \rightarrow s_{r+0} \rightarrow s_{r+1} \rightarrow \dots\}$ ; if  $s_{r+i}$  is not marked by AIT, then add  $s_{r+i}$  to  $L_1$  and  $s_{r+i}$  will be a child of  $s_r$ . For example, in Figure 4,  $L_0 = \{s_{15}\}$ ,  $NT(s_{15}) = \{15 \rightarrow 14 \rightarrow 16 \rightarrow 17 \rightarrow 27 \rightarrow 25 \rightarrow 13\}$  (all nodes in  $NT(s_{15})$  are not marked by AIT; thus, all of them will be added to  $L_1$ , and  $s_{15}$  will be the parent of them).
- (2) For the other nodes in other levels, the processing method is the same as the first level. The only difference is that the first level contains only a node, namely, the root.

(2) *Data Reporting Mechanism Using HST*. In GLT, NT aggregates the data and the proxy node will start the forwarding process. Any node in the path from the proxy to the sink forwards the collected data to its parent in HST or its parent in NT. The parent in HST is the default. If the energy of the parent in HST is greater than the threshold value, then the parent in HST is selected. Otherwise, the parent in NT is selected (see Algorithm 1).

(3) *Lifetime Prolonging Mechanism*. From GLT structure, each node may have two parents; one parent for NT and the other for HST. The routing mechanism is based on HST tree and NT tree; two of them are working together in order to maximize the lifetime of the network. Although the main task of NT is to avoid the data redundancy and duplication, it plays an important role in maximizing the lifetime of network. When the HST parent of the reporter node has a lower power (i.e., lower than the threshold value), the reporter node should switch its parent according to the switch-operation. The switch-operation of the parent is managed by NT, which is entirely based on the remaining energy of the nodes in NT. The node with the greatest energy will be selected as parent for the reporter node. For example, in Figure 4, NT will switch the routing path from node 10 to node 20 if node 8 is busy or it has a lower battery power.

The HST is the default tree for reporting the data to the sink; however, it should be updated according to energy distribution and consumption requirements. The energy threshold value for the nodes in NT will be reduced per switch-operation. With the assistance of NT, the nodes

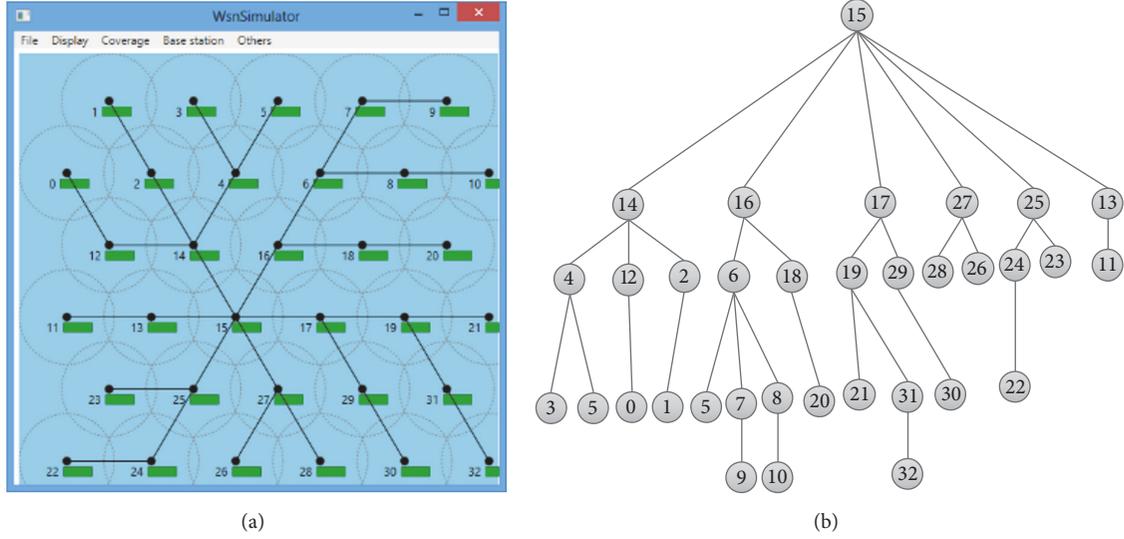


FIGURE 4: Hierarchical Spanning Tree, assuming that node 15 is the sink node.

**Input: Reporter Node****Output: The data of Reporter Node is forwarded to its parent**

```

(1) DataReport(ReporterNode, pkg)
(2) {
(3)   ParentInHST ← ReporterNode.ParentSensor;
(4)   if (ParentInHST.BatteryPercentage > ParentInHST.Energythreshold)
(5)     {
(6)       ReporterNode.SendData(ParentInHST, pkg);
(7)       ParentInHST.ReceiveData(ReporterNode, pkg);
(8)       DataReport(ParentInHST, pkg); // recursive call
(9)     }
(10)  else
(11)  { // select a node from the NT of ReporterNode
(12)    ParentInNT ← SwitchOperation(ReporterNode, ParentInHST); // see Algorithm 2.
(13)    ReporterNode.SendData(ParentInNT, pkg);
(14)    ParentInNT.ReceiveData(ReporterNode, pkg);
(15)    DataReport(ParentInNT, pkg); // recursive call
(16)  }
(17) }

```

ALGORITHM 1: Data reporting.

in HST consume their energy as evenly as possible (see Algorithm 2).

## 4. Analysis of GLT and HST

**4.1. The  $k$ -Covered Regions.** To analyze the GLT, we assume that the nodes are deployed systemically (i.e., using the deployment methods in [3, 4]). With considering that in all nodes deployment schemes there will be regions covered by more than one node, such regions are called  $k$ -covered regions (covered by  $k$  nodes). Without managing internotification between the nodes, the  $k$ -covered regions will be monitored by multiple nodes, which lead to duplicated packets. For example, in Figure 5(b), suppose the object is appearing

in the region marked as  $4$ -covered, which means the four nodes will sense and report the same data for the same object in the same region. NT provides a solution to avoid the packet duplication and data redundancy. Nevertheless, sometimes it sends unnecessary notifications. The number of INSERT notification messages depends on the number of  $k$ -covered regions, so here we can find the number of  $k$ -covered regions where  $k = 0, 1, 2, \dots$ ; here  $k$  is the number of sensors in a group. A group  $G^k \in G^*$  contains  $R(G^k) = 1 + k(k-1)$  regions (e.g., Figure 5(a)). We can count  $R(G^k)$  easily by considering the number of regions in  $G^k$  with no repetition; that is to say, among  $R(G^k)$  there will be one area which is  $k$ -covered,  $k$  areas are  $1$ -covered,  $k$  areas are  $2$ -covered  $\dots$ , and  $k$  areas are  $k-1$  covered. Let  $C_{j,k}$  be the notation for  $j$ -covered region in

**Input: ReporterNode and ParentInHST**

**Output: select a new parent for ReporterNode (maximum energy node in NT tree of ReporterNode).**

```

(1) SwitchOperation(ReporterNode, ParentInHST)
(2) {
(3)   NTnodesWithSameLevlOFParentInHST = NT(ReporterNode);
(4)   if(NTnodesWithSameLevlOFParentInHST.Count = 1)
(5)   {
(6)     newParentNode ← NTnodesWithSameLevlOFParentInHST[0];
(7)     newParentNode.Children.Add(ReporterNode);
(8)     ReporterNode.ParentSensor = newParentNode;
(9)     ParentInHST.Children.Remove(ReporterNode);
(10)    if (newParentNode.Energythreshold > 0)
(11)    newParentNode.Energythreshold ← newParentNode.Energythreshold - 1;
(12)  }
(13) else
(14) {
(15) newParentNode ← NTnodesWithSameLevlOFParentInHST[0];
(16) ∀(sen ∈ NTnodesWithSameLevlOFParentInHST)
(17) {
(18)   if (sen.BatteryPercentage > newParentNode.BatteryPercentage)
(19)   {
(20)     newParentNode ← sen;
(21)   }
(22) }
(23) newParentNode.Children.Add(ReporterNode);
(24) ReporterNode.ParentSensor ← newParentNode;
(25) ParentInHST.Children.Remove(ReporterNode);
(26) if (newParentNode.Energythreshold > 0)
(27) newParentNode.Energythreshold ← newParentNode.Energythreshold - 1;
(28) }
(29) }

```

ALGORITHM 2: Lifetime prolonging mechanism.

$R(G^k)$  and let  $\Phi(C_{j,k})$  be the number of  $C_{j,k}$  in  $R(G^k)$ . Then

$$\begin{aligned}\Phi(C_{0,k}) &= 0, \\ \Phi(C_{k,k}) &= 1;\end{aligned}\quad (4)$$

generally,

$$\Phi(C_{x,k}) = k, \quad \text{where } (k-1 \geq x \geq 1). \quad (5)$$

Thus,

$$\begin{aligned}R(G^k) &= \Phi(C_{0,k}) + \Phi(C_{1,k}) + \Phi(C_{2,k}) + \dots \\ &+ \Phi(C_{k,k}) = \sum_{i=0}^k \Phi(C_{i,k}) \\ &= 0 + k + k + \dots + 1 = 1 + k(k-1) \\ &= k^2 - k + 1.\end{aligned}\quad (6)$$

Furthermore, among  $R(s_j \in G^k)$  there will be one area which is  $k$ -covered, 1 area is 1-covered, 2 areas are 2-covered, 3 areas are 3-covered ..., and  $k-1$  areas are  $k-1$  covered; for example, see Figure 5(b). We denote to the  $k$ -covered regions

in the range of node  $s_j$  by  $c_{k,j}$  ( $k \geq 0$ ), and let  $\theta(c_{k,j})$  be the number of  $c_{k,j}$  regions; then for any  $s_j \in G^k$ ,  $\theta(c_{i,j}) = i$ , where  $(k-1 \geq i \geq 0)$ . We can obtain that  $\theta(c_{0,j}) = 0$  and  $\theta(c_{k,j}) = 1$ . It is easy to see that

$$\begin{aligned}R(s_j \in G^k) &= \theta(c_{0,j}) + \theta(c_{1,j}) + \theta(c_{2,j}) + \dots \\ &+ \theta(c_{k,j}) = \theta(c_{k,j}) + \sum_{i=0}^{k-1} \theta(c_{i,j}) \\ &= \theta(c_{k,j}) + 0 + 1 + 2 + 3 + \dots + (k-1) \\ &= \frac{k(k-1)}{2} + 1 = \frac{1}{2}k^2 - \frac{1}{2}k + 1.\end{aligned}\quad (7)$$

Therefore, we can get number of regions  $R(s_j \in G^*(s_j))$ , where  $G^*(s_j)$  is the set of associated groups of  $s_j$  such that  $G^*(s_j) = \{G^x \mid G^x \in G^* \text{ and } s_j \in G^x\}$ . Let  $Y = |G^*(s_j)|$ , then

$$R(s_j \in G^*(s_j)) \leq 1 + \left( \sum_{G^x \in G^*(s_j)} R(s_j \in G^x) \right) - Y. \quad (8)$$

Any object  $o_i \in M$  moves in a group  $G^k \in G^*$ ; its location should be detected within the group range with respect to

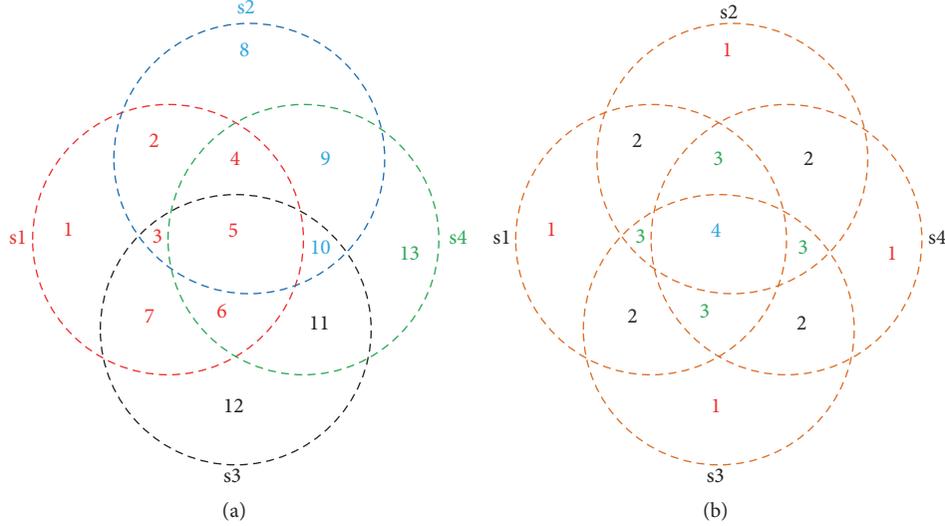


FIGURE 5:  $G^4$ , a group of 4 nodes. (a) The 13 subareas of group  $G^4$ . (b) The coverage degree of  $G^4$ .

the probability space  $\Omega(G^k \in G^*) = \{C_{1,k}, C_{2,k}, \dots, C_{k,k}\}$ . However, any object  $o_i \in M$  moves in the range of node  $s_j \in G^*(s_j)$ ; its location should be detected with respect to the probability space  $\Omega(s_j \in G^*(s_j)) = \{c_{1,j}, c_{2,j}, \dots, c_{k,j}\}$ . For networks deployment using [3],  $\Omega(s_j \in G^*(s_j)) = \{c_{1,j}, c_{2,j}, c_{3,j}\}$ . The Maximum Coverage Efficiency is shown in (9), while the Minimum Coverage Redundancy is shown in (10).

$$\text{MCE} = c_{1,j} = \frac{\pi r^2 - 2r^2(\pi - 3)}{\pi r^2} = 91\%, \quad (9)$$

$$\text{MCR} = c_{2,j} + c_{3,j} = \frac{2r^2(\pi - 3)}{\pi r^2} = 9\%, \quad (10)$$

$$c_{1,j} = 91\%,$$

$$c_{2,j} \approx 8\%, \quad (11)$$

$$c_{3,j} < 1\%.$$

**4.2. Optimal Coverage Pattern for NT and HST.** The size of NT is strongly related to the number of groups while the number of groups is related to the deployment pattern of network. In what follows, we will find the number of groups that can be created by  $n = |S|$  nodes. For the convenience, let  $x^i$  be a group that contains  $i$  nodes. We define the plug operation  $x^i \rightarrow j$  ( $i \geq j > 0$ ) as follows:  $x^i$ : a group of  $i$  nodes;  $\rightarrow j$ : plugging a new node to  $x^i$ . The new node should cross any  $j$ -covered region in the group  $x^i$ . The results of plug operation  $x^i \rightarrow j$  ( $i \geq j > 0$ ) are expressed as shown in (12). Figure 6(a) shows the set of patterns for  $n = 1, 2, 3, 4$ , while Figure 6(b) shows  $x^3 \rightarrow j$ .

$$x^i \rightarrow j \quad (i \geq j > 0) = \begin{cases} x^{i+1} & i = j, \\ x^i + x^{j+1} & j < i. \end{cases} \quad (12)$$

We use  $P(n) = \{p_0, p_1, p_2, \dots, p_m\}$  to be the distinct patterns set; each  $p_i \in P(n)$  is called a pattern of deployment  $p_r = \{x^i + x^j + \dots, x^k + x^v + \dots, \dots\}$   $r, i, j, k, v \geq 2$ . Thus, the set of distinct patterns of  $n = 1, 2, 3, 4$  is as follows:  $P(1) = \{x^1\}$ ,  $P(2) = \{x^2\}$ ,  $P(3) = \{x^2 + x^2, x^3\}$ , and  $P(4) = \{x^2 + x^2 + x^2, x^2 + x^3, x^3 + x^3, x^4\}$ . Let  $\Psi(n) = |P(n)|$  (the size of distinct patterns set); for example,  $\Psi(1) = 1$ ,  $\Psi(2) = 1$ ,  $\Psi(3) = 2$ , and  $\Psi(4) = 4$ . Let  $\mathcal{G}(n)$  be the number of groups of the pattern  $P(n)$ , so we have  $\mathcal{G}(1) = 1$ ,  $\mathcal{G}(2) = 1$ ,  $\mathcal{G}(3) = 3$ , and  $\mathcal{G}(4) = 8$ . To find the set of distinct patterns of  $P(n)$ , we have to execute a plug operation to  $P(n-1)$  which is denoted by  $P(n) \Rightarrow P(n-1)$ .  $\Rightarrow P(n-1)$  means applying the plug operation  $x^i \rightarrow j$  for each group in  $P(n-1)$ . For example, to find  $P(5)$ , we have to do a plug  $\Rightarrow P(4)$  as shown in Table 1. As shown in Figure 6(a),  $P(4) = \{x^2 + x^2 + x^2, x^2 + x^3, x^3 + x^3, x^4\}$ . Then using (14) we can get the patterns of  $P(5)$  as shown in Table 1. From Table 1, we can see that  $\Psi(5) = 9$  and  $\mathcal{G}(5) = 22$ .

The average communication cost of HST for the network that contains  $1, 2, 3, \dots, n$  nodes is  $\mathcal{A}(n) = \mathcal{G}(n)/\Psi(n)$ . Any network of  $n$  nodes ( $n > 1$ ) randomly or systemically deployed will match one pattern of the patterns sets  $P(n)$  ( $\Psi(n)$  is the number of all possibilities of nodes deployments). Since the number of hops depends on the number of groups of each pattern  $\mathcal{G}(n)$  (the nodes of any group can communicate with only one hop), then the average communication cost of the network, obviously, is  $\mathcal{A}(n) = [\mathcal{G}(n)/\Psi(n)]$ . Figure 7 shows the number of groups and patterns for network of size  $n = (1, 13)$ . In GHS, the deployment pattern of network plays an important role in terms of energy consumption and balancing. Both of HST and NT are based on GHS. To demonstrate the impact of deployment pattern, the relation between the number of deployment patterns and the energy consumption levels is shown in Figure 8. As shown in Figure 8, the network of eight nodes can be deployed in  $\Psi(8) = 98$  different deployment patterns. In order to maximize the lifetime of each node and to balance the energy consumption of the nodes, the average

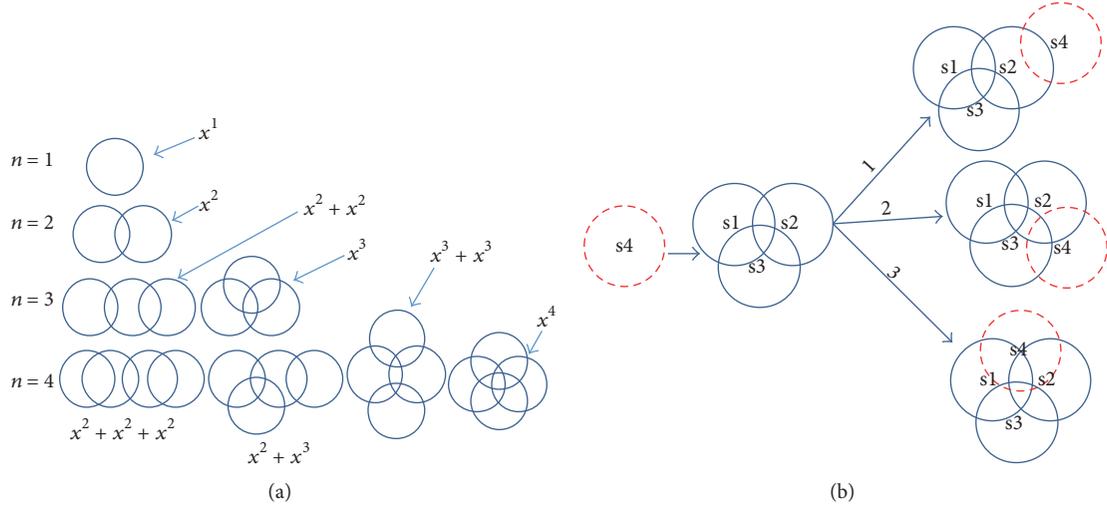


FIGURE 6: Plug operation example. (a) Patterns for  $n = 1, 2, 3, 4$ . (b)  $x^3 \rightarrow j$ . Add a new node  $s_j$  to the group  $x^3$ .

TABLE 1: The plug operation of  $\rightarrow P(4)$ .

$x^4$	$x^3 + x^3$	$x^2 + x^3$	$x^2 + x^2 + x^2$
$x^4 \rightarrow 1 = x^2 + x^4$	$x^3 \rightarrow 1 + x^3 = x^2 + x^3 + x^3$	$x^2 \rightarrow 1 + x^3 = x^2 + x^2 + x^3$	$x^2 + x^2 + x^2 \rightarrow 1 =$
$x^4 \rightarrow 2 = x^3 + x^4$	$x^3 \rightarrow 2 + x^3 = x^3 + x^3 + x^3$	$x^2 \rightarrow 2 + x^3 = x^3 + x^3$	$x^2 + x^2 + x^2 + x^2$
$x^4 \rightarrow 3 = x^4 + x^4$	$x^3 \rightarrow 3 + x^3 = x^3 + x^4$	$x^2 + x^3 \rightarrow 1 = x^2 + x^2 + x^3$	$x^2 + x^2 + x^2 \rightarrow 2 = x^2 + x^2 + x^3$
$x^4 \rightarrow 4 = x^5$		$x^2 + x^3 \rightarrow 2 = x^2 + x^3 + x^3$	
		$x^2 + x^3 \rightarrow 3 = x^2 + x^4$	
$x^2 + x^4$		$x^2 + x^3 + x^3$	
$x^3 + x^4$	$x^3 + x^3 + x^3$	$x^2 + x^2 + x^3$	$x^2 + x^2 + x^2 + x^2$
$x^4 + x^4$		$x^3 + x^3$	
$x^5$			

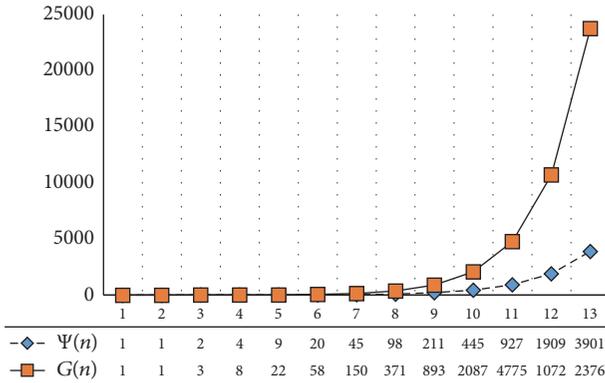


FIGURE 7: The number of patterns and groups for  $n = (1, 13)$ .

cost should be  $\mathcal{A}(8) = 3$ ; this indicates that the network of eight nodes should be divided into three groups. Among  $\Psi(8) = 98$  there are 28 patterns fit with the average cost and the other patterns lead to either more data redundancy or to more energy consumption. Let  $\ell(n)$  be the number of patterns that fit with the average cost as shown in Figure 9; it showed that only about 23% of patterns fit with average cost.

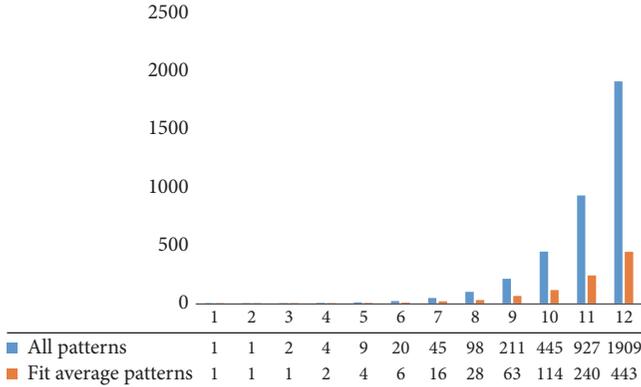


FIGURE 8: The energy consumption for all possible different deployment patterns of eight nodes  $\Psi(8) = 98$ .

The recursive equation of  $\Psi(n)$  is shown as

$$\Psi(n) = 2\Psi(n-1) + \left( \sum_{k=0}^{a(n-2)} \Gamma(a_k) \right). \quad (13)$$

$a(n)$  is number of partitions of  $n$  (the partition numbers) [22];  $\Gamma(a_k)$  is number of up-steps in the partition  $a_k$  (represented as weakly increasing lists) [23]. For example,  $\Psi(8) = 2\Psi(7) + (\sum_{k=0}^{a(6)} \Gamma(a_k))$ ,  $a(6) = 11$ , shown in Table 2. The sum of up-steps is 8. Then,  $\Psi(8) = 2\Psi(7) + 8 = 2 * 45 + 8 = 98$ .

FIGURE 9: The number of  $\Psi(n)$  and  $\ell(n)$  for  $n = (1, 12)$ .TABLE 2: The partitions of  $a(6)$ .

$k$	$a_k$	$\Gamma(a_k)$
0	$[1\ 1\ 1\ 1\ 1\ 1]$	0
1	$[1\ 1\ 1\ 1\ 2]$	1
2	$[1\ 1\ 1\ 3]$	1
3	$[1\ 1\ 2\ 2]$	1
4	$[1\ 1\ 4]$	1
5	$[1\ 2\ 3]$	2
6	$[1\ 5]$	1
7	$[2\ 2\ 2]$	0
8	$[2\ 4]$	1
9	$[3\ 3]$	0
10	$[6]$	0

## 5. Results and Discussion

### 5.1. Configurations and Network Topologies

**5.1.1. Energy Model.** We assume a simple model called (First-Order Radio Model). This model is discussed in [16]. As shown in (14) and (15), the sensor node will consume  $E_{Tx}(d, k)$  of energy to transmit  $k$ -bits packet over distance  $d$  and consume  $E_{Rx}(d, k)$  of energy to receive  $k$ -bits size message.

$$E_{Tx}(d, k) = \begin{cases} (k \cdot E_{elec}) + (k \cdot \varepsilon_{fs} \cdot d^2) & d < t_0 \\ (k \cdot E_{elec}) + (k \cdot \varepsilon_{mp} \cdot d^4) & d \geq t_0 \end{cases} \quad (14)$$

$$E_{Rx}(k) = k \cdot E_{elec}. \quad (15)$$

**5.1.2. Target Mobility Model.** We used Gauss–Markov Mobility Model [24, 25]. It was designed to adapt to different levels of randomness via one tuning parameter. Gauss–Markov Mobility Model generates the next position depending on the previous position and conserving the speed and direction.

Suppose the initial position of target is  $p(x_1, y_1)$ ; then the next position can be found according to

$$\begin{aligned} x_n &= x_{n-1} + s_{n-1} \cos(d_{n-1}), \\ y_n &= y_{n-1} + s_{n-1} \sin(d_{n-1}), \end{aligned} \quad (16)$$

where  $s_{n-1}$  and  $d_{n-1}$  are the speed and direction, respectively;  $x_{n-1}$ ,  $y_{n-1}$ , and  $x_n$ ,  $y_n$  are the old and the new positions of the target, respectively. The  $n$ th position, direction, and speed are calculated using Gauss–Markov model from the  $(n-1)$ th position, direction, and speed.  $s_n$  and  $d_n$  are calculated by

$$\begin{aligned} s_n &= \alpha s_{n-1} + (1 - \alpha) \hat{s} \sqrt{1 - \alpha^2} s_{x_{n-1}}, \\ d_n &= \alpha d_{n-1} + (1 - \alpha) \hat{d} \sqrt{1 - \alpha^2} d_{x_{n-1}}. \end{aligned} \quad (17)$$

$\hat{s}$  and  $\hat{d}$  are the values representing the main of speed and direction as  $n \rightarrow \infty$ .  $s_{x_{n-1}}$  and  $d_{x_{n-1}}$  are random variables from Gauss–Markov distribution.  $\alpha$  is the level of randomness  $1 \geq \alpha \geq 0$ .

**5.1.3. Simulations Parameters.** Since the GHS [16] is affected by the deployment pattern of nodes, in our simulation we choose four types of deployment schemes as shown in Figure 10. These networks have different number of groups. GLT does not have any MAC layer requirements except the support for broadcasts. The default parameters we used are shown in Table 3. For each configuration, the results are collected, averaged, and shown in their respective figures. The performance metrics we used are as follows: (1) Average Total Energy Consumption: the average energy (in joules) which is consumed by the sensor node for transmit/receiving data/control packets during the simulation time; (2) Average Traffic Load: the average number of generated control packets by NT and the number of routing packets, which are generated by HST during the simulation time; (3) End-to-End Delay: the average time from the moment the source node generates the data packet to the moment the sink received the data packet.

The Hierarchical Spanning Trees (HST) shown in Figure 10 are the default routing trees. However, the process of updating the HST tree is based on NT tree nodes (switch-operation). That is to say, when any node  $s_j \in$  HST requires a switch-operation (i.e., when the parent node of  $s_j$  has an energy lower than threshold), the new node  $s_k$  will be selected by NT to be the new parent of  $s_j$  (see Algorithms 1 and 2).

### 5.2. GLT and Grouping Structure Evaluation

**5.2.1. Grouping Structure and Notification Tree.** The number of control packets (i.e., INSERT and DELETE packets) depends on the size of NT tree, while the size of NT depends on the number of groups (the maximum covered regions in the field). Similarly, the number of groups is strongly related to the deployment scheme of nodes. The greater the size of NT is, the greater the number of control packets will be and the

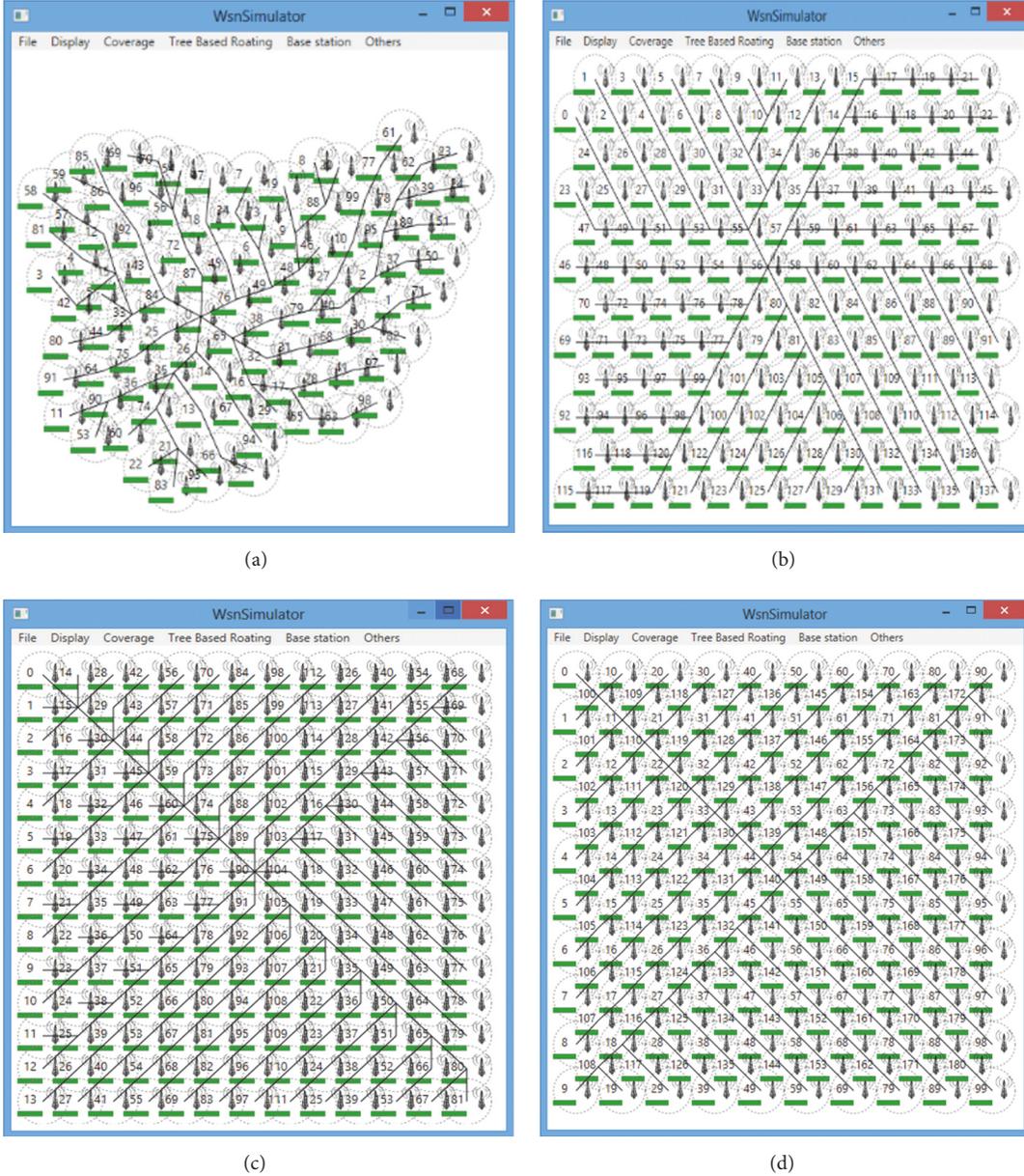


FIGURE 10: Default network topologies of simulation. (a) Net 1: Hierarchical Spanning Tree (HST) of 100 nodes deployed randomly. The sensing range is 25 m while the communication range is 50 m. The progress bar indicates the capacity of the battery (initially full). The sink node is 0. (b) Net 2: Hierarchical Spanning Tree (HST) of 138 nodes deployed using zigzag scheme [3]. The sensing range is 25 m and the communication range is 50 m. The progress bar indicates the battery capacity. The sink node is 56. (c) Net 3: Hierarchical Spanning Tree (HST) of 182 nodes deployed using grid scheme 1 [4]. The sensing range is 25 m and the communication range is 50 m. The progress bar indicates the battery capacity. The sink node is 90. (d) Net 4: Hierarchical Spanning Tree (HST) of 181 nodes deployed using grid scheme 2 [4]. The sensing range is 25 m and the communication range is 50 m. The progress bar indicates the battery capacity. The sink node is 140.

more the energy consumption will be. As shown in Figures 11(a) and 11(b), the number of groups affects the size of NT. When the number of groups is getting greater, the size of NT will be smaller. The size of NT plays an essential role that affects directly the number of control packets. To show this, we used different number of targets and different speeds (Figures 11(e) and 11(f)). When the speed is getting higher, the

number of control packets will be increased (see Figures 11(c) and 11(d)).

**5.2.2. Grouping Structure and HST.** Figure 12 shows the evaluation of HST during data routing process. The performance of HST is affected by the number of groups in the network, the size of the group, and the deployment scheme of nodes.

TABLE 3: Default parameters to evaluate the performance of GLT.

Parameter	Value (Section 5.2)	Value (Section 5.3)
Network size	$500 \times 500 \text{ m}^2$	$450 \times 450 \text{ m}^2$
Nodes	100, 138, 181, 182 (static nodes)	600 (static nodes)
Communication range	50 m	24 m
Sensing range (radius)	25 m	12 m
Data length for routing packet	1024 bits	256 bytes
Data length for control packet	256 bits	8 bytes
Initial energy	0.5 Joule	0.5 Joule
$E_{elec}$	50 nJ/bit	50 nJ/bit
$\epsilon_{amp}$	$0.001 \text{ pJ/bit/m}^2$	$0.001 \text{ pJ/bit/m}^2$
Network topology	Zigzag, random, grid	Random
Speed of mobile targets	1–10 m/sec	8 m/s
Data reporting frequency	1 second	0.3 second
Number of mobile targets	1–10	1
Simulation time	60–180 seconds	300 seconds
Sink	One static sink (inside)	One static

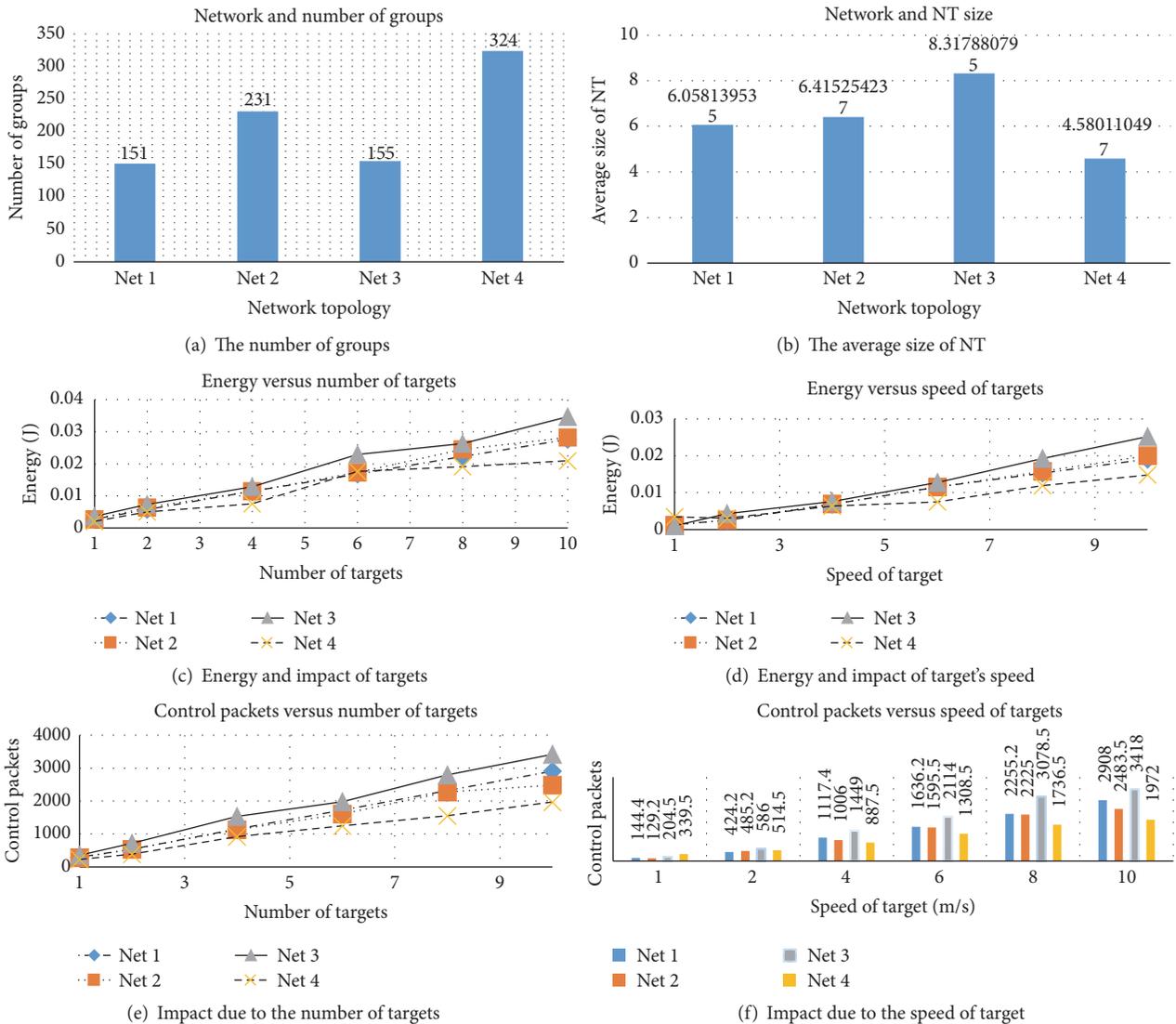


FIGURE 11: Impact of grouping structure and NT.

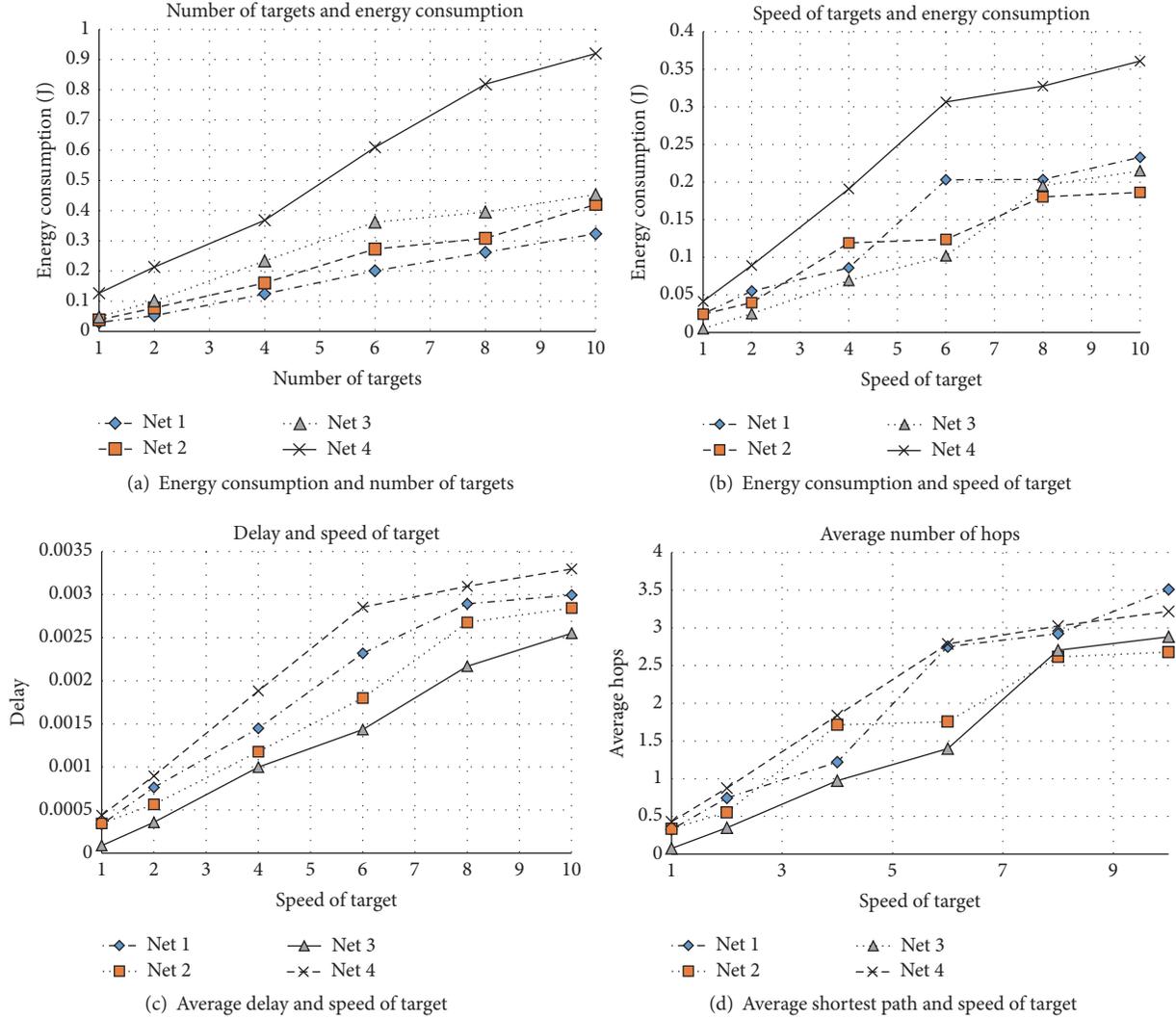


FIGURE 12: HST evaluation.

The evaluation shows that *Net 4* has the worst performance in terms of energy consumption, delay, and shortest path selection. However, *Net 4* shows the best performance in terms of energy consumption during the activating of nodes for tracking. HST shows that the number of groups in the network affects the average number of hops and delay. The greater the number of groups, the shorter the path to the sink and the lesser the energy consumption (see Figure 8).

**5.2.3. Grouping Structure and Target Tracking.** In GLT, the tracking quality is based on NT. The quality of tracking is based on the size of NT and its coverage around the current proxy node. Therefore, to guarantee the tracking process continuously, all the nodes in NT will be in the idle state in order to be ready for tracking. The size of NT tree is shown in Figure 11(b). In Figure 13, we deployed 20 objects in the network topology shown in Figure 10(a). The speed of each object is 5 m/second. The simulation time is 60 seconds. The NT tree ensures a very high performance in avoiding

data redundancy such that the reported messages to the sink node will not be duplicated; and at any given time, there will be one proxy node for each object. For example, as shown in Figure 13(a), in the time interval [1–6 s], target 1 will be tracked by node 18 even if the object is crossing an overlapping region. In the case of the object appearance for the first time in an overlapped area, there will be one duplicated message. For example, as shown in Figure 13(a) in second 25, the packet is reported by node 84 and node 87. After reporting the first packet, both nodes will discover that they have the same Notification Message Record in their NMT (see Figure 3). At this time, node 84 (with higher energy) will be selected as the proxy for object 1, while node 87 will send a DELETE notification message to its NT tree. Figure 13(b) illustrates the trajectory of mobile target 2.

**5.2.4. Grouping Structure and Network Lifetime.** To evaluate the performance of GLT, HST, and NT, we conducted a series of simulation experiments using our simulator in the

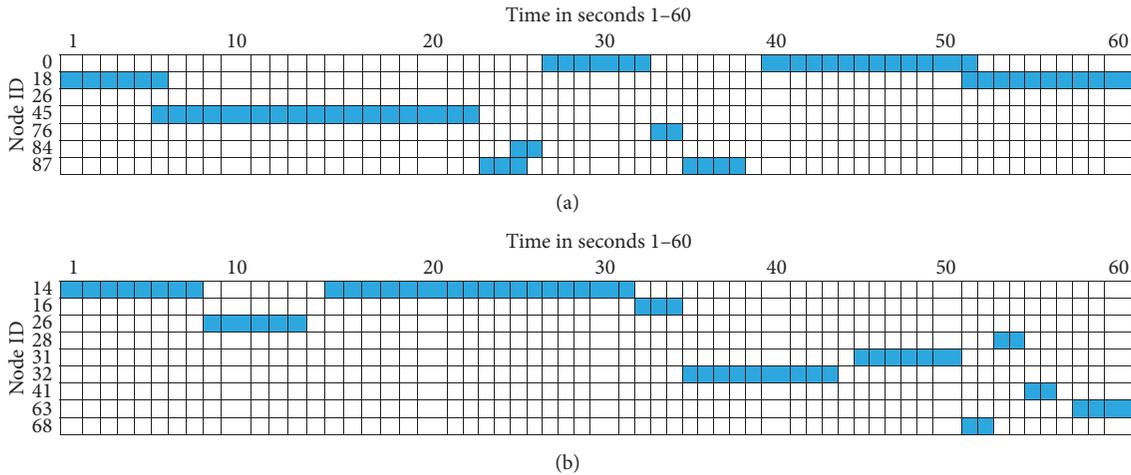


FIGURE 13: The nodes detected and reported the motion of objects 1 and 2. (a) The trajectory for object 1 within the network (Net 1): the speed of object is 5 m/1 s; the simulation time is 60 seconds. (b) The trajectory for target 2 within the network (Net 1): the speed of object is 5 m/1 s; the simulation time is 60 seconds.

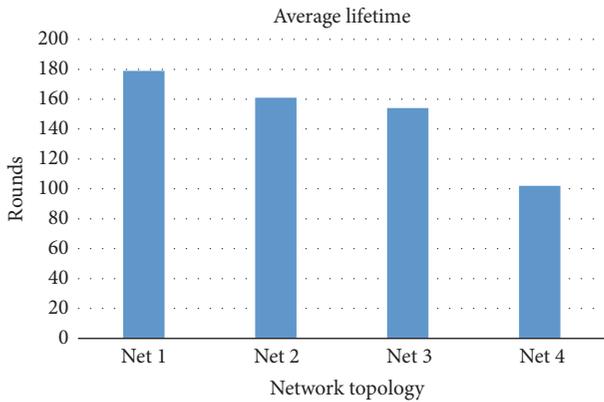


FIGURE 14: Network lifetime.

network topologies explained in Figure 10. We assume that the maximum lifetime of network is the number of rounds until the first node is dead. Here, one round means that each node in the network sends a packet of data with size of 1024 bits; that is to say, in each round there will be  $|S| \times 1024$  bits or  $|S|$  packets. Figure 14 shows the network lifetime of GLT with different topologies.

**5.3. Simulation Results.** This subsection focuses on comparing GLT results with selected existing clustering protocols CAICS [12], DLC [13, 14], and MHC [15, 21]. The default parameters used in this section are the same as [12]. Moreover, two scenarios are considered: (1) Scenario 1 (S1): the sink is located at (0, 0); (2) Scenario 2 (S2): the sink is located in the center of the field (200, 200). The measurement units are simulation time (seconds), transmission delay (milliseconds), energy consumption (joules), and target velocity (meters/second). GLT is implemented using C#. Please see more details about the source code and the toolkit in Supplementary Material available online at <https://doi.org/10.1155/2017/4509697>.

**5.3.1. Number of Packets.** In GLT, the packets include data routing, building HST, building NT, activation, and switch-operation. Figure 15 shows the number of packets for GLT, which is *Grouping Hierarchy Structure based* [16], and the number of packets for different clustering-based protocols (i.e., CAICS [12], DLC [13, 14], and MHC [15, 21]). In GLT, the number of control packets including HST packets and NT packets does not change per time. On the other hand, the number of activation packets, data routing packets, and switch-operation packets will change per time according to the mobility pattern of the target. In GLT, the location of sink affects the number of switch-operation packets, while it has no impact on the number of other control packets. When the sink is located in the center of the interested field, the lifetime of the network is getting longer, and the number of switch-operation packets is reduced. In contrast, when the sink's location is getting closer to the sides of the interested field, network lifetime is getting shorter and as a result the number of the switch-operation packets will increase in order to prolong the network's lifetime as long as possible.

**5.3.2. Energy Consumption.** In GLT, HST consumes most of the energy during data routing toward the sink. On the other hand, NT consumes a smaller amount of energy. Figure 16 shows the average energy consumption of the sensor nodes over different simulation times. GLT achieves better energy saving compared to DLC, HMC and CAICS. The nodes in NT will be activated to guarantee the tracking accuracy, but their collected data will be transmitted to the proxy node by one and only one hop. Although each node belongs to more than one group, each node has only one permanent NT tree. Therefore, the nodes in NT will transmit their collected data to the correct proxy even if the node is activated by more than one proxy node in the same time (in case of more than one object is found). In contrast, clustering-based protocols suffer from more traffic due to cluster members and cluster heads intercommunication. Additionally, more

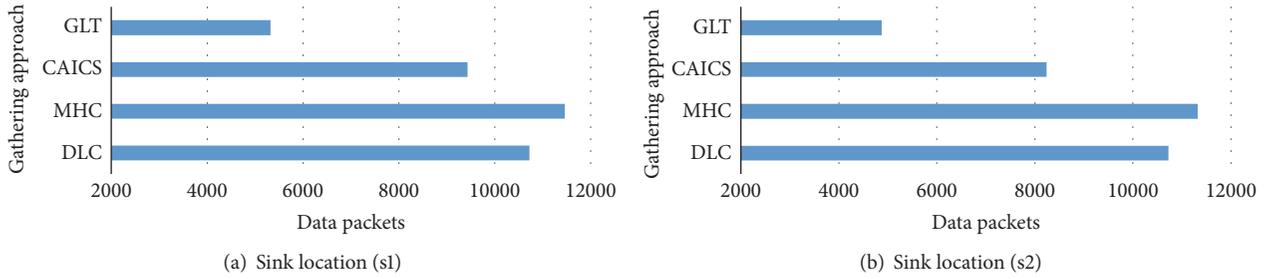


FIGURE 15: Comparison of the number of transmitted data packets.

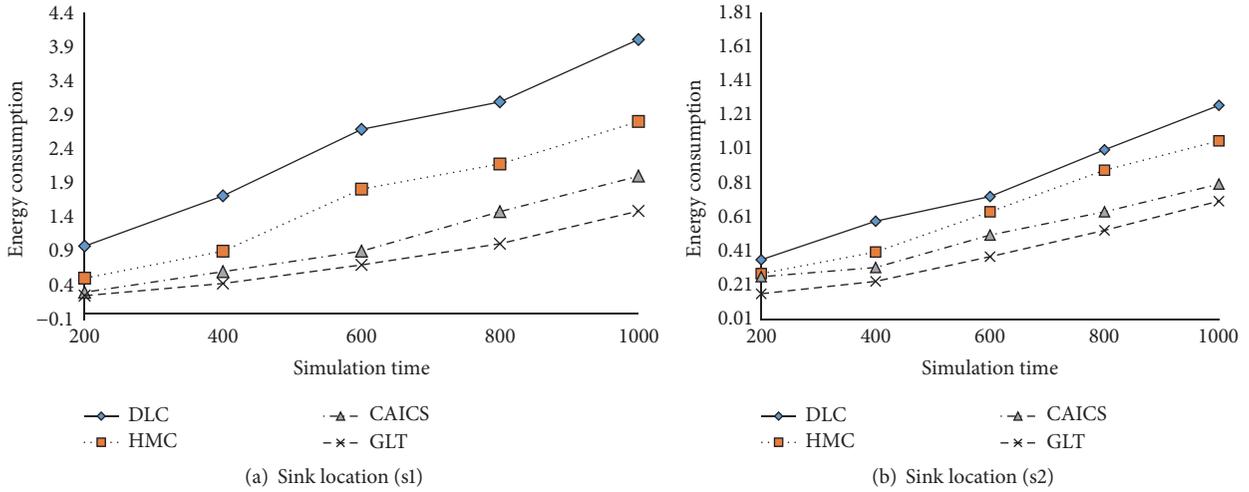


FIGURE 16: Comparison of energy consumption.

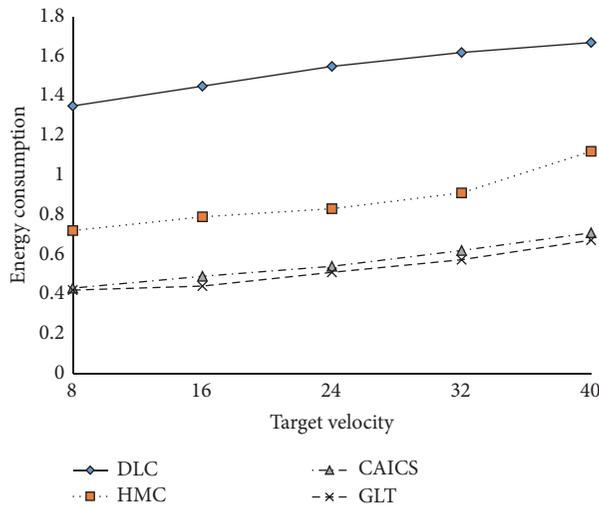


FIGURE 17: Energy consumption after 300 s versus the target velocity (Scenario 1).

energy will be dissipated when the nodes, which belong to multiple clusters, detect the same target. This leads to redundant data transmissions from cluster heads to the sink. This issue is avoided in [12] and this is the reason why CAICS archives better energy saving compared to DLC and HMC. In DLC, the packets are transmitted to the sink over a long distance, since each cluster head aggregates sensed data from its cluster members and directly transmits it to the sink.

This issue is solved in HMC; HMC adopts a hierarchical cluster heads model and the data can be transmitted to the sink hop by hop through multiple intermediate cluster heads. The sink location has an impact on the energy consumption of GLT, CAICS [12], DLC [13, 14], and MHC [15, 21]; the closer the sink to the center of the interested field is the shorter the path from source to the sink will be and the lesser the energy will be consumed (see Figure 16). Figure 17

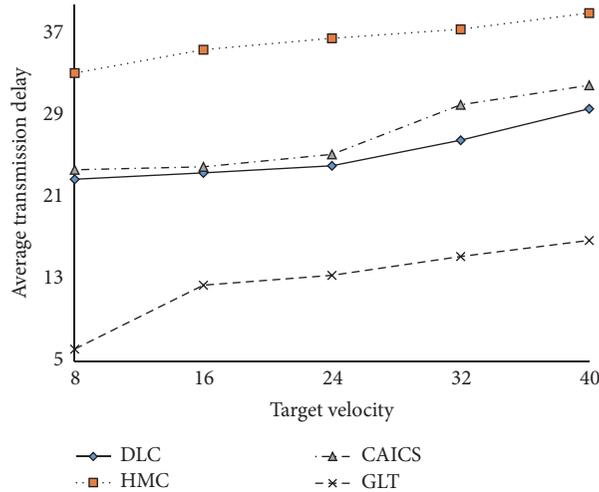


FIGURE 18: Average transmission delay versus the target velocity (Scenario 1).

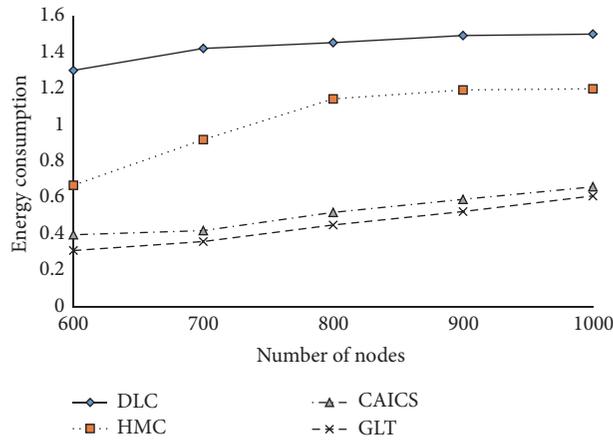


FIGURE 19: Average energy consumption of data gathering versus the node number.

shows that in GLT, CAICS, DLC, and MHC, the speed of the object has an impact on the consumed energy; higher speed of target always leads to greater energy consumption. Our proposed approach is influenced by the target’s speed due to the increment in the number of NT’s control packets. However, HST is not affected by the target’s speed since the number of sensed data packets is not affected by the target’s velocity. Like CAICS, which suffers from the frequent selection of aggregation nodes, GLT consumes more energy due to frequent transmission of control packets in NT. Unlike CAICS, the transmission delay is not getting longer in GLT. The frequent selections of aggregation nodes in CAICS affect both of the transmission delay and the energy consumption. In GLT, the transmission delay is not affected or increased due to the permanent structure of NT. Nevertheless, NT will increase the transmission delay during the switch-operation. The impact of target velocity on the average transmission delay is shown in Figure 18.

The size of the network (i.e., the number of nodes in the network) has an impact on the transmission delay and power

consumption in all four approaches (i.e., GLT, CAICS, DLC, and MHC). In case of GLT, the larger the size of the network, the greater the number of groups. Thus, the size of NT and HST will be greater. On the other hand, in cluster-based networks, the number of cluster members will be increased in each cluster, which leads to more nodes involved in sensing and tracking. The average energy consumption and average transmission delay of data gathering versus the nodes number are shown in Figures 19 and 20, respectively.

**5.3.3. Network Lifetime.** We assume that the maximum lifetime of the network is the number of rounds until the first node dies in the network. Here, one round means that each node in the network sends a packet of data with size of 2048 bits. That is to say, in each round there will be  $|S| \times 2048$  bits or  $|S|$  packets. In each round, every node in the network (600 nodes) is elected as an aggregation node and transmits the aggregated data to the sink either directly or via multihops. In GLT, the network lifetime is managed by the switch-operation, which is a combination between the

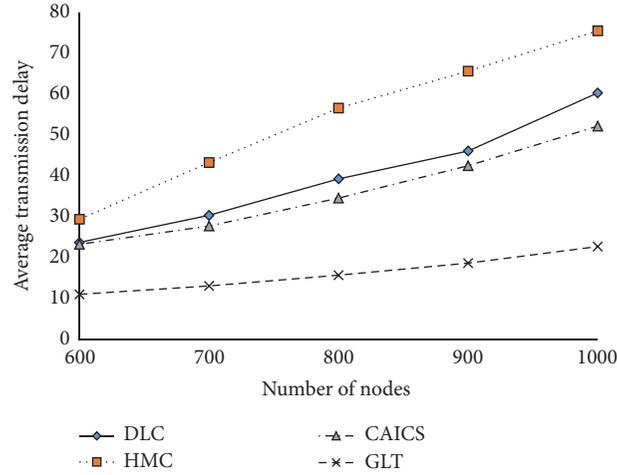


FIGURE 20: Average transmission delay of data gathering versus the node number.

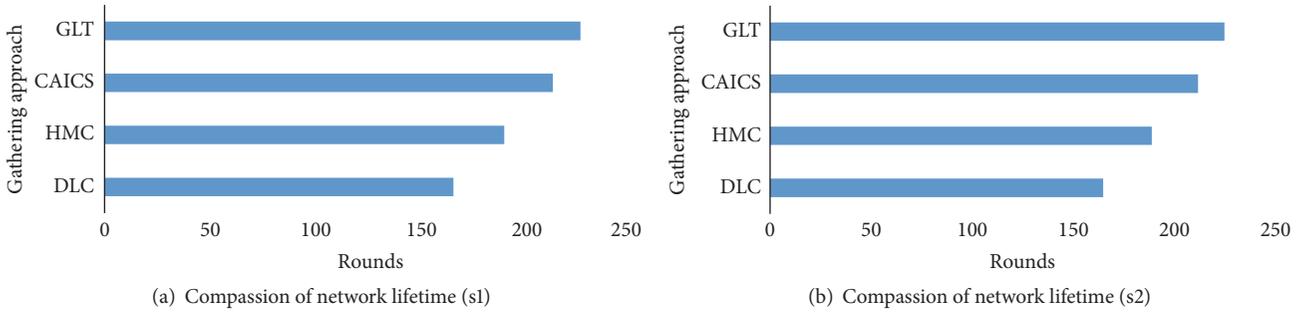


FIGURE 21: Network lifetime.

HST and NT trees. In each round, during data routing, the NT and HST select the node with maximum residual energy. CAICS selects the node which has the most residual energy and this is the reason why the network lifetime is maximized compared to DLC and MHC. However, CAICS suffers from frequent selection of aggregation nodes. In DLC and MHC the packets are transmitted over a long distance and that is the reason why their lifetime is shorter compared to GLT and CAICS. The network lifetime is shown in Figure 21. (a) shows a shorter lifetime compared to (b), since the location of the sink is further from the center of the interested field. Generally, the closer the sink to the center of the interested field, the shorter the path to the sink.

## 6. Conclusion

The system of this article is still an ongoing project and there are some issues under resolving. One of the main issues is the energy dissipation in NT tree, which is affected by the overlapped regions. We are investigating to combine the geometric and machine deep learning to identify the overlapped regions of WSN networks in order to clarify the overlapped regions for each node in the network. In this paper, the Grouping based Location Tracking (GLT) algorithm is proposed, which is based on the Grouping

Hierarchy Structure (GHS) [16]. GLT algorithm ensures a high performance by the support of NT tree and HST tree to aggregate and report the data to the sink node using the minimum number of hops considering the energy balancing. The data reporting process of GLT is totally based on the reporting algorithm HST. On the other hand, the NT tree is managing the notifications between the nodes to avoid the duplication messages and data redundancy.

The number of groups and the number of deployment patterns in GHS affect the communication overhead. We have tried to find a closed form for  $\mathcal{E}(n)$  but we failed due to the complicated combinations of  $\mathcal{E}(n)$ . Nevertheless, one contribution of this work is finding the number of deployment patterns. The readers are very welcome to solve the issue of  $\mathcal{E}(n)$ ; in the future, we will try to find closed form for  $\mathcal{E}(n)$ .

## Disclosure

Ammar Hawbani and Xingfu Wang are first authors.

## Conflicts of Interest

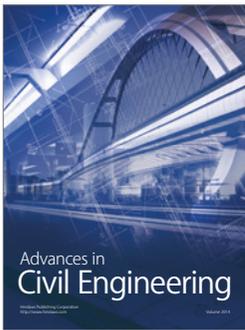
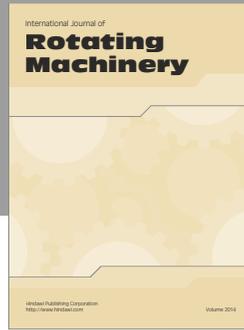
The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This paper is supported by the “Fundamental Research Funds for the Central Universities no. WK2150110007” and by the National Natural Science Foundation of China (nos. 61472382, 61472381, 61572454, and 61272472).

## References

- [1] O. Demigha et al., “On energy efficiency in collaborative target tracking in wireless sensor network: a review,” *IEEE Communications Surveys & Tutorials*, vol. 99, pp. 1–13, 2012.
- [2] A. Hawbani, X. Wang, H. Kuhlani et al., “Sink-oriented tree based data dissemination protocol for mobile sinks wireless sensor networks,” *Wireless Networks*, 2017.
- [3] A. Hawbani et al., “Zigzag coverage scheme algorithm & analysis for wireless sensor networks,” *Network Protocols and Algorithms*, vol. 5, no. 4, pp. 19–38, 2013.
- [4] A. Hawbani et al., “Grid coverage algorithm & analyzing for wireless sensor networks,” *Network Protocols and Algorithms*, vol. 6, no. 4, pp. 1–18, 2014.
- [5] L. Zhao, A. Al-Dubai, X. Li, G. Chen, and G. Min, “A new efficient cross-layer relay node selection model for wireless community mesh networks,” *Computers & Electrical Engineering*, 2017.
- [6] G. Sharma et al., “Near-optimal location tracking using sensor networks,” in *Proceedings of Distributed Processing Symposium Workshops (IPDPSW '14)*, *IEEE International*, pp. 737–746, Phoenix, AZ, USA, 2014.
- [7] C. T. Calafate, C. Lino, A. Diaz-Ramirez, J. Cano, and P. Manzoni, “An integral model for target tracking based on the use of a WSN,” *Sensors*, vol. 13, no. 6, pp. 7250–7278, 2013.
- [8] W.-H. Liao, K.-C. Chang, and S. P. Kedia, “An object tracking scheme for wireless sensor networks using data mining mechanism,” in *Proceedings of the Network Operations and Management Symposium (NOMS '12)*, pp. 526–529, IEEE, April 2012.
- [9] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, “Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1886–1896, 2009.
- [10] T.-S. Chen, J.-J. Chen, and C.-H. Wu, “Distributed object tracking using moving trajectories in wireless sensor networks,” *Wireless Networks*, 2015.
- [11] S. Samarah, M. Al-Hajri, and A. Boukerche, “A predictive energy-efficient technique to support object-tracking sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 656–663, 2011.
- [12] J. Feng, B. Lian, and H. Zhao, “Coordinated and adaptive information collecting in target tracking wireless sensor networks,” *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3436–3445, 2015.
- [13] H. Chen, H. Mineno, and T. Mizuno, “Adaptive data aggregation scheme in clustered wireless sensor networks,” *Computer Communications*, vol. 31, no. 15, pp. 3579–3585, 2008.
- [14] G.-Y. Jin et al., “Dynamic clustering for object tracking in wireless sensor networks,” in *Ubiquitous Computing Systems (Lecture Notes in Computer Science)*, vol. 4239, pp. 200–209, Springer-Verlag, Berlin, Germany, 2006.
- [15] Q. Mamun, S. Ramakrishnan, and B. Srinivasan, “Selecting member nodes in a chain oriented WSN,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, April 2010.
- [16] A. Hawbani, X. Wang, S. Karmoshi, L. Wang, and N. Husaini, “Sensors grouping hierarchy structure for wireless sensor network,” *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 650519, 2015.
- [17] H. T. Kung and D. Vlah, “Efficient location tracking using sensor networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, vol. 3, pp. 1954–1961, New Orleans, La, USA, March 2003.
- [18] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, “Efficient in-network moving object tracking in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, 2006.
- [19] C.-T. Lee et al., “An efficient object tracking algorithm in wireless sensor networks,” in *Proceedings of the Advances in Intelligent Systems Research (JCIS '06)*, 2006.
- [20] S. P. M. Tran et al., “Oco: optimized communication & organization for target tracking in wireless sensor networks,” in *Proceedings of Sensor Networks, Ubiquitous, and Trustworthy Computing. IEEE International Conference on*, vol. 1, pp. 428–435, IEEE, 2006.
- [21] X. Wang, J.-J. Ma, S. Wang, and D.-W. Bi, “Cluster-based dynamic energy management for collaborative target tracking in wireless sensor networks,” *Sensors*, vol. 7, no. 7, pp. 1193–1215, 2007.
- [22] <https://oeis.org/A000041>.
- [23] <https://oeis.org/A058884>.
- [24] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [25] S. Sharma and S. K. Jena, “Data Dissemination protocol for mobile sink in wireless sensor networks,” *Journal of Computational Engineering*, vol. 2014, Article ID 560675, 10 pages, 2014.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

