

Received November 26, 2018, accepted December 9, 2018, date of publication December 14, 2018, date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886779

# An Efficient Neighbor Discovery Scheme for Mobile WSN

OMAR A. SARAEREH<sup>1</sup>, IMRAN KHAN<sup>2</sup>, AND BYUNG MOO LEE<sup>3</sup>

<sup>1</sup>Electrical Engineering Department, The Hashemite University, Zarqa 13133, Jordan

<sup>2</sup>Department of Electrical Engineering, University of Engineering and Technology at Peshawar, Peshawar 814, Pakistan

<sup>3</sup>School of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, South Korea

Corresponding author: Byung Moo Lee (blee@sejong.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education (grant number: NRF-2017R1D1A1B03028350).

**ABSTRACT** Mobile low-duty-cycle wireless sensor network (MLDC-WSN) is a new type of WSN that has emerged in recent years. It can deploy nodes on moving objects and use the object mobility to capture a wide range of dynamic environmental information. It has diverse applications and has the ability of mobility and gets into sleep for a long time. However, the mobility and sleeping features of MLDC-WSN nodes result in dynamic network topology changes, which restricts discovering their neighbor nodes quickly. Thus, the optimal distribution decision is not established by the nodes. To solve this problem, in this paper, a novel selectively proactive neighbor discovery algorithm is proposed. The proposed algorithm enables the nodes when they wake up to search their neighbor nodes, which prevent the long-time waiting delay in the traditional passive neighbor discovery mechanism. Moreover, the proposed algorithm can further reduce the delay and acquire the accurate neighbor discovery results by quickly determining the next neighbor set at the next moment due to the prediction of the movement speed and distance of neighbors. The analytical and simulation results show that the proposed algorithm can effectively reduce the network energy consumption and delay and can determine all the neighbor nodes quickly and accurately as compared with the existing algorithms.

**INDEX TERMS** WSN, energy efficient nodes, delay, network optimization, adaptive topology.

## I. INTRODUCTION

The Mobile Low-Duty-Cycle Wireless Sensor Network (MLDC-WSN) is a new type of self-organizing network that has emerged in recent years. Like traditional WSNs, it consists of a large number of nodes with limited energy, communication range, storage capacity, and computing power. It is mainly deployed in a harsh environment that humans cannot enter and is engaged in long-term monitoring or tracking tasks [1], [2]. However, it is significantly different from WSN: the nodes in MLDC-WSN use a low duty cycle (LDC) protocol and go to sleep for a long-time to save energy and they have less active period compared with the sleeping period, whereas the traditional WSN nodes do not have LDC feature and therefore, its nodes awake for longer time causing more energy consumption and battery usage [3]. Therefore, the topology changes of the MLDC-WSN will be more frequent, causing the neighbor nodes to change from time to time.

Neighbor discovery is an important operation in the network. Nodes in the network need to quickly discover each other to form a network. Since nodes need to perform neighbor discovery in the network, a quick neighbor discovery with less energy consumption is an important scheme for the network to work normally and prolong its lifecycle [4], [5]. In MLDC-WSN, the node adopts a low-duty cycle mode of operation (i.e., the node has more sleep period than an active period) and neighbor discovery between nodes require the node to be simultaneously awakened. The state, which takes a long time for the node to wait for the discovered neighbor to wake up, causes a huge delay. In addition, the movement of the node when it wakes up will cause the network topology to change constantly so that the neighbors of the node will change from time to time. Therefore, it is very difficult to find neighbors with low-latency and power consumption.

At present, the typical work is to use the node to actively wake up to discover neighbors, but this method requires the

node to wake up many times, which consumes more energy. How to reduce the number of active wake-ups, reduce energy consumption, and complete neighbor discovery quickly is still a difficult problem to be solved.

To solve the above problem, this paper proposed a new low-energy selectively proactive neighbor discovery (SPND) algorithm. In the proposed SPND algorithm, the node can combine the movement model of the nodes in the network, and according to the division of the set, selectively wake up at the moment when the neighbor wakes up, and confirm with the neighbor node. For example, node  $i$  already has its own set of neighbors at a certain moment. After a period of movement of nodes in the network, some nodes are no longer neighbor of  $i$ . At the next moment, node  $i$  simply needs to actively wake up to discover its neighbor's nodes. In this way, the delay and energy consumption are significantly reduced.

The rest of the paper is organized as follow. Section II presents the related work. Section III provides the system model and problem description. Section IV describes the algorithm design. Section V gives the simulation results and analysis while Section VI concludes the paper.

## II. RELATED WORK

According to the certainty of the wake-up scheduling mode, the neighbor discovery algorithm can be divided into probabilistic neighbor discovery and deterministic neighbor discovery. In the deterministic neighbor discovery algorithm, according to whether the node actively wakes up to discover the neighbor, it can be further divided into active neighbor discovery and passive neighbor discovery.

### A. PROBABILISTIC NEIGHBOR DISCOVERY

The probabilistic neighbor discovery algorithm uses a random wake-up scheduling method. By setting the probability that the node is in each time slot, a lower average discovery delay can be obtained, but in the worst case, the delay may be very large. McGlynn and Borbash [6] proposed a typical probabilistic neighbor discovery algorithm, Birthday, which uses the probabilistic principle of birthday paradox, that is, in the collection of randomly selected  $n$  individuals, when  $n$  is large, the probability that someone has a birthday on the same day will be great. Therefore, the node selects the state of the current slot node (including sleep, transmission, and listening) with a certain probability. By carefully setting the probability values of the various states, the nodes have a high probability of discovering each other for a continuous period of time. The algorithm can achieve a good balance between energy consumption and discovery delay in the neighbor discovery process, but it cannot limit the worst-case discovery delay. In response to this problem, You *et al.* [7] proposed the ALOHA-Like algorithm, which analyzes the expected time of any node to discover  $n - 1$  neighbor nodes, and has a high probability to limit the worst-case discovery delay.

### B. DETERMINISTIC NEIGHBOR DISCOVERY

In deterministic neighbor discovery, the node adopts a fixed wake-up scheduling mode, that is, the node is in a

sleep or wake state and periodically repeats according to a predetermined sleep wake-up schedule.

#### 1) PASSIVE NEIGHBOR DISCOVERY

The passive neighbor discovery algorithm means that the node wakes up to complete the neighbor discovery according to the predetermined wake-up time. A typical passive neighbor discovery algorithm is the torus quorum system (TQS) neighbor discovery algorithm proposed by [8]. The algorithm arranges the node work cycle into a  $t \times w$  matrix with a working period length of  $n = t \times w$ . The node selects all the elements of one of the columns  $c$ , and then selects  $w/2$  elements from any of the selected  $c+(i = 1, 2, \dots, w/2)$  columns as the wake-up time slot of the node. TQS can achieve better energy consumption delay indicators, but it does not apply to the case where each node adopts different wake-up scheduling modes. In order to solve this problem, Zheng *et al.* [9] implemented an asynchronous wake-up protocol (AWP) algorithm that does not require time slot alignment, in which nodes can flexibly set the duty cycle according to their different roles in the network. AWP simplifies the neighbor discovery problem to the minimum vertex cover problem of a graph, while the minimum vertex cover problem of the solution graph is a centralized approach, but the centralized approach is not applicable in node-distributed wireless sensor networks.

For the shortcomings of the TQS algorithm and the AWP algorithm, Dutta and Culler [10] proposed the Disco algorithm. In the Disco algorithm, each node selects two prime numbers as its working period, and the node is in a sleep state most of the time. Each node has an independent counter. Once the node counter can divide any one of its prime working cycles, the node is in the awake state. According to the remainder theorem [11], the wake-up slots of two nodes must be periodically overlapping or partially overlapping. The Disco algorithm can ensure that two nodes in the neighbor range can discover each other within a certain time limit.

In order to solve the problem of asynchronous and synchronous node wake-up scheduling mode, Kandhalu *et al.* [12] proposed the U-connect neighbor discovery algorithm. The U-connect algorithm selects the prime number  $q$  as its basic duty cycle, and then constructs  $q \times q$  grid matrix ( $T = q \times q$ ) in consecutive  $T$  time slots, and selects a column and a row in the matrix. Half of the time slot is used as the wake-up time slot of the node. When the number of time slots following the line is less than half, the return to the first column of the line continues to be selected. The U-connect algorithm combines the ideas of the remainder theorem and the TQS algorithm. The energy consumption-delay index of the U-connect algorithm is about 1.5 times that of the optimal theoretical system. Which is superior to the TQS algorithm and the Disco algorithm in terms of energy consumption delay equalization.

#### 2) ACTIVE NEIGHBOR DISCOVERY

There are many neighbor discovery algorithms, but they are based on node passive wake-up for neighbor discovery.

Chen *et al.* [13] proposed a neighbor discovery algorithm Q-connect that actively wakes up nodes. The algorithm analyzes the energy consumption of the node from the sleep state to the awake state. Such energy consumption cannot be ignored and thus the Q-connect algorithm subdivides the node time slot into several subslots, and actively wakes up the broadcast beacon message. In the fourth part, the device wakes up to receive the beacon message. The Q-connect algorithm can improve the energy efficiency of the network, but the nodes find a large average delay.

In order to make full use of beacon messages to reduce the discovery delay, Qiu *et al.* [4] proposed an active neighbor discovery algorithm Nihao. The Nihao algorithm is based on the principle of “Talk More Listen Less” (TMLL). The nodes are in the “listening” (awake) state in the first  $m$  time slots of one cycle ( $m \times n$ ), and every  $m$  times slots in the whole cycle. Enter the status of “say” (actively wake up to send a beacon message), that is, send a total of  $n$  beacon messages. In the Nihao algorithm, the beacon message can be actively awake during the sleep state of the node, thus causing the nodes to discover each other in less time, reducing the discovery delay.

Kindt *et al.* [14] proposed an active neighbor discovery algorithm Griassdi. In the Griassdi algorithm, nodes periodically wake up and send beacon messages, and wakeups are sent independently of beacon messages. After a node wakes up to receive the beacon message of other nodes in the certain time slot, it adjusts the time when the next beacon message is sent according to the next wake-up time of other nodes, so as to quickly discover each other with other nodes.

This part of the algorithm requires the node to send a beacon message at part of a time slot. Chen *et al.* [15] proposed a node that does not need to send a beacon message to actively wake up to find the neighbor group-based discovery (GBD) algorithm. In the GBD algorithm, nodes that found each other are in a group. The nodes in the group selectively share the sleep and wake times of their existing neighbors to the neighbor nodes in the newly discovered joining group according to the spatio-temporal characteristics of the mobile low-duty cycle network. The newly discovered node can quickly obtain the wake-up time table of the surrounding nodes, and actively wake up at the moment when the surrounding nodes wake up, thereby judging whether the two nodes are neighbors. For networks with high node density, Chen *et al.* [15] improved the GBD algorithm and proposed the advanced group-based discovery (AGBD) algorithm. In the AGBD algorithm, the node follows a selection mechanism based on the spatio-temporal characteristics of the mobile network. This mechanism can reduce the unnecessary active wake-up of the node part, but may miss the neighbor node. The advantage of the GBD algorithm is that the average discovery delay is low, and the node can actively wake up to discover neighbors when the potential neighbor nodes wake up. The discovery delay of increasing network low-duty cycle characteristics is reduced. However, the GBD algorithm has two shortcomings: i) The node selectively actively wakes up for neighbor discovery, and cannot limit the worst-case maximum discovery.

ii) Probability-based selective forwarding of nodes reduces the reliability of nodes discovering neighbors.

There are many other state-of-the-art neighbor discovery algorithms such as WiFlock [16], Acc [17], Group-based [10] and Extended Quorum Systems (EQS) [18]. In the WiFlock algorithm [16], all nodes in the range of one-hop are regarded as one group. By uniformly adjusting the distribution of the nodes' active slots in the group, the nodes outside the group can be found more quickly, thereby speeding up the neighbor discovery process. In the Acc [17] algorithm, the node selects some additional slots to be active. Each slot owns a slot gain value at the current slot, and the slot gain is decided by temporal diversity and spatial similarity. The slot gains of selected slots must be larger than the gains of other slots. The main limitation of Acc is that the calculation of temporal diversity requires information about many neighbors. Otherwise, it will cause the additional active slots to be noneffective. As a result, Acc performs poorly when node density in the network is low. The EQS method [15] uses quorum graph theory to remove the unnecessary working slots to improve energy efficiency. The Group-based algorithm [18] adds a small amount of additional active slots, used to recommend and verify the neighbor relationship, which greatly speeds up neighbor discovery. However, Group-based is not very efficient in the selection of verification nodes.

The above are some typical neighbor discovery algorithms for low-duty-cycle networks [19]. It can be found that the probabilistic neighbor discovery algorithm can achieve lower network energy consumption and find most neighbors in a smaller time. However, it is impossible to limit the maximum discovery delay of the entire network in the worst case. In deterministic neighbor discovery, the passive neighbor discovery algorithm tends to delay large, active neighbor discovery algorithms can achieve smaller discovery delays, but require nodes to actively wake up or actively send beacon messages, and the network consumes a lot of energy. This paper model the advantages and disadvantages of these algorithms and proposes an optimized solution.

### III. SYSTEM MODEL AND PROBLEM DESCRIPTION

This section will give a description of the network model of MLDC-WSN and the research questions in this paper, as well as the definition of related terms.

#### A. RELATED DEFINITION

- Communication Radius [20]: The node communicates with a certain power to guarantee the farthest distance that the message is received by other nodes.
- The probability of discovery of a node [21]: The probability that two nodes within the communication range can discover each other.
- The discovery ratio of nodes [21]: The ratio of the number of neighbor nodes discovered by any node of the network to the number of nodes in all communication ranges.

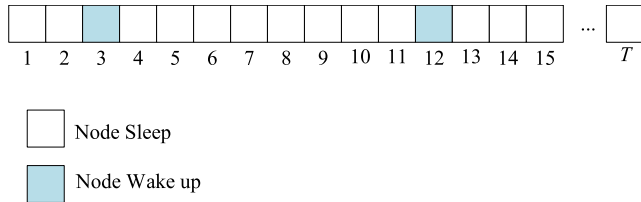


FIGURE 1. Duty cycle.

- Found delay [21]: The time interval from when the node moves to the time within the communication range to when they discover each other.

**B. NETWORK MODEL**

Suppose a wireless sensor network  $G_{net} = \{V, E\}$  with  $n$  mobile nodes, where  $V$  represents the set of nodes in the network, and  $E$  represents the information of whether the nodes in the network are connected. The node in  $G$  net adopts the low-duty cycle working mode to define the working state of the mobile node as sleep-activated state, as shown in Figure 1, that is, the node is in sleep state most of the time, and the timer is turned off. All other functional modules are only in a small amount of time, such as the 3rd and 12th time slots are in the active state, sensing data and communication. The node can actively wake up to send data packets at any time, but only receive data packets at the scheduled wake-up time.

In order to consider the situation where the network distribution is relatively uniform, the nodes of the network adopt the Random Waypoint mobile model. In the Random Waypoint moving model [22], each mobile node randomly selects a direction as the target direction and then selects a range moving at a constant velocity  $v$  of  $[0, v_{max}]$ , where  $v_{max}$  is the maximum speed at which the node can move. The selection of direction and speed between nodes is relatively independent. The nodes in the network move within a certain time period  $\Delta t$ , and the distance that the node moves in a  $\Delta t$  period is  $[0, v_{max} \times \Delta t]$ .

**C. PROBLEM DESCRIPTION**

Node  $i$  and node  $j$  in the network can meet each other at a certain time  $t$ .

1) Within a certain timer period  $\Delta t$ , node  $i$  and node  $j$  are within the communication range, assuming that the communication radius of the nodes in the network is consistent with  $R$ .

2) Node  $i$  and node  $j$  are both awake at  $\Delta t$ , and  $\Delta t$  is not less than the minimum time required for the node to achieve neighbor discovery.

Only when the above two conditions are met between nodes, then node  $i$  and node  $j$  can achieve mutual discovery in  $[t, t + \Delta t]$ . In MLDC-WSN, the neighbor discovery delay of a node refers to the time interval  $\delta$  from the time  $t_0$  at which two nodes can communicate with each other, to the time  $t + \Delta t$  at which they complete mutual discovery, which

can be expressed as:

$$\delta = t + \Delta t - t_0 \tag{1}$$

The probability  $P_{ij}$  of discovery of a node is the probability that the two nodes  $i$  and  $j$  in the communication range can discover each other.  $P_{ij}$  has a value range of  $[0,1]$ .  $P_{ij}(t) = 0$  means that the nodes cannot discover each other in  $t$  time,  $P_{ij}(t) = 1$  means that the nodes can definitely discover each other in  $t$  time. The ratio  $P(t)$  is used to represent the ratio of neighbor nodes discovered by node  $i$  in time  $t$  to neighbors in all communication ranges. Therefore, the neighbor discovery problem in MLDC-WSN can be summarized as the minimization of discovery delay  $\delta$  with minimum energy consumption and the maximization of discovery ratio  $P^i$  in minimum time, i.e.:

$$\{\min \delta, \max P^i\} \tag{2}$$

where  $\delta = t + \Delta t - t_0$  referring to Equation (1), since  $\Delta t$  is usually constant and cannot be adjusted, there is only  $t - t_0$  that can be optimized by the algorithm. In existing algorithms that achieve lower discovery latency, it is often not possible to bring  $P$  close to one [23]. This paper proposes a new node active neighbor discovery algorithm SPND, which can achieve fast neighbor discovery with low energy consumption. Section IV provides the proposed algorithm to achieve fast neighbor discovery with low energy consumption, and Section V provides simulation results showing that the proposed algorithm consumes less energy compared to other algorithms.

**IV. ALGORITHM DESIGN**

This section will give the basic idea and detailed design of the SPND algorithm, and theoretically compare the SPND algorithm with the existing neighbor discovery algorithm. Considering the fact that the traditional algorithm utilizes the active wakeups of the node to reduce the neighbor discovery delay, the SPND algorithm focuses on the network mobility model to reduce the number of active wakeups of the node, thereby achieving active neighbor discovery with low energy consumption.

**A. BASIC IDEA OF ALGORITHM**

In the SPND algorithm, for the sensor network just deployed, each node performs neighbor discovery in a distributive manner. For the characteristics of the mobile sensor network, the neighbor discovery of the node is divided into 2 steps.

- Step 1: According to the group-based neighbor discovery algorithm, the initial neighbor nodes set is constructed according to the principle that the neighbor nodes in the newly joined group share their neighbor set information to other nodes in the group. Then the other nodes can quickly obtain the wake-up time of the potential neighbor node. After this, they actively wake up for neighbor discovery at that time.

- Step 2: It combines the movement model of the node to selectively specify the active wake-up time of the node. The node actively wakes up to discover the neighbor set at that



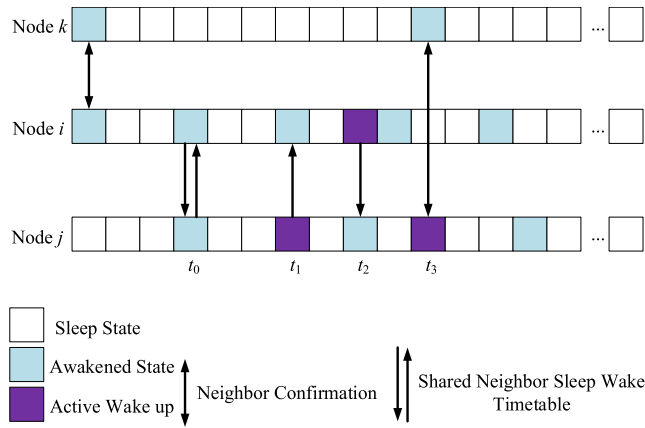


FIGURE 2. Build an initial neighbor collection.

moment. According to the principle that: after a period of movement of a node in the network, the node selects only those neighbor nodes that may be active in the communication range after the mobile to actively wake up the neighbor discovery. It is ensured that the nodes in the neighboring range perform neighbor discovery according to the established sleep awake state. The node that is definitely moving out of the neighbor range is excluded from the neighbor set.

**B. DETAILED DESIGN OF THE ALGORITHM**

The SPND algorithm is divided into 2 phases. The network has just been deployed, and the initial set of nodes is built as the first stage. Optionally, the node is actively awake, and the node actively wakes up to find the neighbor set at that moment as the second phase. The nodes in the network perform the first-stage neighbor discovery process distributivity, and then the nodes switch from the first phase to the second phase completely asynchronously.

**1) BUILD AN INITIAL NEIGHBOR COLLECTION**

After the MLDC-WSN is deployed, first, at time  $t$ , any node  $i$  in the network distributivity generates its own neighbor set  $G_t^i = \{i\}$ . When the number of nodes in it  $G_t^i$  is 1, node  $i$  will send a broadcast message at its own wake-up time to discover the neighbors until the first neighbor node  $k$  is added to the set  $G_t^i$ . At this time, the number of nodes in the set  $G_t^i = \{i, k\}$  of the node  $i$  is 2. When  $G^i \geq 2$ , as shown in Figure 2, node  $i$  proceeds to the next neighbor discovery as follows:

1) Each node in the network broadcasts a message at the moment of waking up to inform the network of its existence. As shown in Figure 2, at a certain time  $t_0$ , the node  $i$  and the  $j$  are in communication range, and received each other's broadcast messages. That is, node  $i$  and  $j$  are mutually discovered as neighbor nodes and can acquire each other's sleep wake-up timetable.

2) Node  $j$  will actively wake up at the next wake-up time  $t_1$  of node  $j$ , and send the sleep wake-up timetable information of the neighbor nodes already located in  $G^j$  to node  $i$ . Node  $i$  then sends sleep wake-up timetable information for neighbor

**Algorithm 1** Initial Neighbors Discovery

```

1: for (each node  $i$  in the network) {
2:   while ( $t < T$ ) {
3:     if (node  $i$  wakes up to find node  $h$ ) {
4:       node  $i$  and node  $h$  share  $G$ ;
5:     } else {node  $i$  sleep}
6:     if (node  $h$ 's neighbor wakes up) {
7:       Node  $i$  actively wakes up;
8:       Node  $i$  is confirmed with node  $k$ ;
9:     } } }

```

nodes (e.g., node  $k$ ) already in  $G^j$  to node  $j$  at the next wake instant  $t_2$  of node  $j$ .

After receiving the neighbor information of node  $i$ , node  $j$  will actively wake up at the next wake-up time  $t_3$  of the neighbor node of node  $i$  (such as node  $k$ ), and then send a message to confirm whether the node is the neighbor of node  $j$ . In the same way, node  $i$  confirms whether the neighbor of node  $j$  is the neighbor of node  $i$ .

The specific algorithm is shown in Algorithm 1.

Whenever the neighbor set  $G^i$  of the node  $i$  joins the new neighbor node, the loop performs the step of confirming with the neighbor of the newly joined node, that is, step 6 to 8 in the Algorithm 1, until all the neighbor information received is confirmed. Complete the initial phase set construction of the first phase.

**2) SELECTIVELY SPECIFY WAKE TIME**

After the node constructs the initial neighbor set, it enters the second stage of neighbor discovery work, that is, when there is a node moving around node  $i$ , the node  $i$  is selectively designated to actively wake up for neighbor discovery. The node in the network adopt the Random Waypoint moving model, that is, the maximum distance of each movement of the node is  $\Delta S = v_{\max} \times \Delta t$ , and the direction is random. In the network, it is assumed that the communication radius of the nodes is consistently  $R$ . The neighbor set of node  $i$  at a certain time  $t$  is  $G_t^i$ , after a period of  $\Delta t$  movement, the new neighbor set of node  $i$  is  $G_t^i + \Delta t$ . As shown in Figure 3, according to the network model, the node in the range of  $[0, R - \Delta S)$  from the node  $i$  in  $G_t^i$  is definitely in  $G_t^i + \Delta t$ , and when the distance from the node  $i$  is  $[R + \Delta S, +\infty)$ , the nodes in the range are definitely not in  $G_t^i + \Delta t$ . The nodes in the range  $[R - \Delta S, R + \Delta S)$  may be in  $G_t^i + \Delta t$ . Therefore, for node  $i$ , after the  $\Delta t$  time shift, the next time neighbor discovery is performed, and only nodes with a range of  $[R - \Delta S, R + \Delta S)$  can be considered.

In a sensor network, nodes can roughly estimate the distance between nodes that are neighbors by the RSSI signal strength. For the node  $i$  that has acquired the neighbor set  $G_t^i$ , as shown in Figure 4(a), the neighbor set  $G_t^i$  can be divided into two sub-sets  $G_t^{i1}$  and  $G_t^{i2}$  according to the distance between the nodes. Among them, the node  $k$  in the set  $G_t^{i1}$  has the property that the distance  $l_{ik}$  between node  $k$  and  $i$  is in the range  $[0, R - \Delta S)$ . The node  $h$  in the set  $G_t^{i2}$  has the

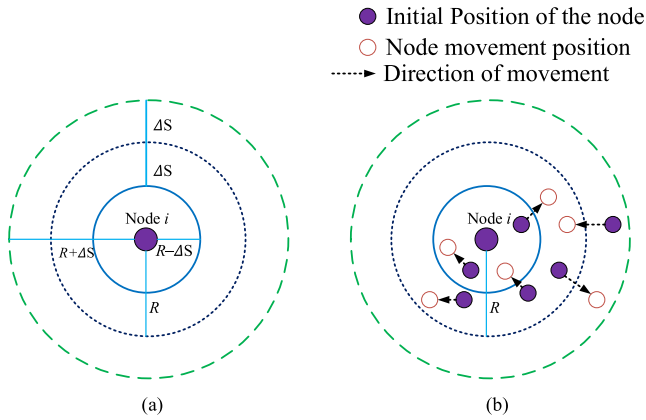


FIGURE 3. Node movement illustration. (a) Node  $i$  illustration. (b) Node movement around node  $i$ .

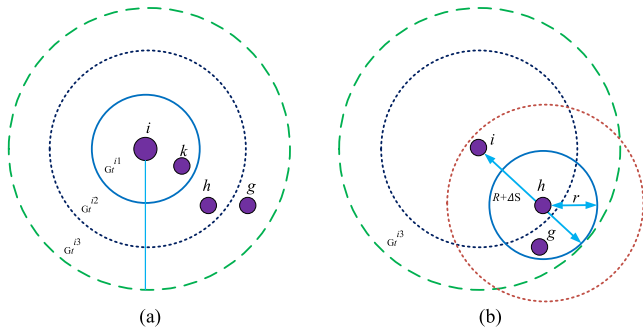


FIGURE 4. Neighbor set division. (a) Subset partition. (b) Getting nodes in the set  $G_i^3$ .

property that the distance  $l_{ih}$  between the node  $h$  and  $i$  is in the range  $[R - \Delta S, R + \Delta S)$ .

For node  $g$  in the range of  $[R, R + \Delta S)$ , it can be obtained by node  $h$ . As shown in Figure 4(b), let  $r = R + \Delta S - l_{ih}$ , the node  $h$  is in the set  $G_i^2$ , and has its own neighbor set  $G_h^h$ , then the information of the neighbor nodes in the set  $G_h^h$  and the distance of the node  $h$  in the range  $[0, r]$  needs to be send to the node  $i$ , and the neighbor set  $G_i^3$  of the node  $i$  is constructed. The specific steps for such neighbor set generation is shown in Algorithm 2.

After node  $i$  acquires the three sets  $G_{i1}$ ,  $G_{i2}$  and  $G_{i3}$ , it only takes  $G_i \text{ part} = G_{i2} + G_{i3} - G_{i1}$ . The wake-up moment of the node actively wakes up to find the neighbor's confirmation. All the nodes in  $G_{i1}$  and the nodes identified as neighbors in the set  $G_i \text{ part}$  are added to the set  $G_i + \Delta t$  to complete the neighbor discovery process in the second stage.

C. THEORETICAL ANALYSIS

For the discovery delay, SPND is compared with the existing typical algorithms, and the energy consumption and discovery probability of the SPND algorithm are analyzed. Firstly, it is theoretically proved that the SPND algorithm discovery delay is better than the existing passive wake neighbor discovery algorithm, and then quantitatively compares the expected values of the nodes to find all neighbor nodes.

Algorithm 2 Neighbour Set Partitioning

```

1: for (each node  $i$  in the network) {
2:   if (the neighbor set of node  $i$  is not empty) {
3:     for (any node  $j$  in the set  $G_i$ ) {
4:       if (distance between node  $i$  and  $j < R - \Delta S$ ) {
5:         Node  $j$  joins the set  $G_i^1$ ;
6:       } else if ( $R - \Delta S < \text{distance between node } i \text{ and } j < R + \Delta S$ ) {
7:         Node  $j$  joins set  $G_i^2$ ;
8:       } }
9:   if (distance between neighbor node  $k$  of node  $h$  and node  $h$  in  $G_i^2 < r$ ) {
10:    Node  $k$  joins the set  $G_i^3$ ;
11:  }
12:  if ( $G_i^{\text{part}}$  node  $l$  wake up) {
13:    Node  $i$  actively wakes up with node  $l$  to confirm;
14:  } }

```

1) DISCOVERY DELAY ANALYSIS

Assuming a pair of neighbor nodes  $i$  and  $j$  that have been mutually discovered, the neighbor set of node  $i$  contains node  $k$ , and the following analysis node  $j$  finds the delay of node  $k$ . To simplify the problem model referred in Equation (2), assume that node  $j$  and node  $k$  just move to the communication range before the start of a cycle. Let set  $C$  be the possible discovery time set of node  $j$  and  $k$ . When node  $i$  and node  $j$  perform neighbor discovery with passive wake-up, then there is  $C = \tau_j \cap \tau_k$ , that is, the minimum discovery delay of node  $j$  and  $k$  is  $\min C$ . When the SPND algorithm is used to construct the initial neighbor set of nodes, node  $j$  can acquire the wake-up time of node  $k$  in advance through node  $i$ , and actively wake up for neighbor discovery, which will increase the wake-up time slot in the working mode  $\tau_j$  of node  $j$ . The new working mode of node  $j$  is  $\tau'_j$ , then  $C' = \tau'_j \cap \tau_k$ . Since  $\tau_j \subseteq \tau'_j$ , there is  $C \subseteq C'$ . That is, the discovery delay of the selective active wake neighbor discovery algorithm is less than or equal to the discovery delay of the traditional passive wake neighbor discovery algorithm.

Without loss of generality, this paper selects the Disco algorithm as the object of quantitative comparison. Like other passive wake-up algorithms U-connect and Searchlight, each node in the network in the Disco algorithm plans its own sleep wake-up schedule before network deployment. After the network is deployed, the node wakes up according to the schedule to discover neighbors. Its characteristics are that it can limit the maximum delay  $\delta_{\max}^{jk}$  found by node  $j$  and  $k$ , that is, when node  $j$  and  $k$  are within the communication range  $t > \delta_{\max}^{jk}$ , they can definitely find each other. When  $t < \delta_{\max}^{jk}$ , the probability that node  $j$  and  $k$  can find each other is  $p = f(j, k, t)$ . That is, the probability that node  $j$  and  $k$  find each other in  $t$  time can be expressed as:

$$P_{jk}(t) = \begin{cases} f(j, k, t), & t \in [0, \delta_{\max}^{jk}) \\ 1, & t \geq \delta_{\max}^{jk} \end{cases} \quad (3)$$

Then the probability that node  $j$  uses Disco algorithm to find all  $n-1$  neighbor nodes in  $t$  time can be expressed as  $P_D^j(t) = \prod_{k=0}^{n-1} P_{jk}(t)$ . Its density function can be expressed as:

$$P_D^j(t) = \sum_{k=0}^{n-1} \left[ P_{jk}(t)' \prod_{h=0, h \neq k}^{n-1} P_{jk}(t) \right] \quad (4)$$

Calculate node  $j$  by probability density function and find the expected time of all neighbor nodes as:

$$\begin{aligned} t_D^j &= E_D^j(t) = \int_0^{\delta_{\max}^j} t P_D^j(t) dt \\ &= \int_0^{\delta_{\max}^j} t \sum_{k=0}^{n-1} \left[ P_{jk}(t)' \prod_{h=0, h \neq k}^{n-1} P_{jk}(t) \right] dt \end{aligned} \quad (5)$$

where  $\delta_{\max}^j$  is the maximum delay of all neighbor nodes found by node  $j$ .

When the SPND algorithm is used to construct the initial neighbor set of nodes, node  $i$  is passively awake, and the probability  $P_S^i(t)$  of the first neighbor node  $j$  in the communication range is found in time  $t$  as:

$$P_S^i(t) = 1 - \prod_{j=0}^{n-1} (1 - P_{ij}(t)) \quad (6)$$

Its probability density function is:

$$P_S^i(t) = \sum_{k=0}^{n-1} \left[ P_{ij}(t)' \prod_{k=0, k \neq j}^{n-1} (1 - P_{ik}(t)) \right] \quad (7)$$

Calculate node  $i$  by probability density function to find the expected time of all neighbor nodes as:

$$\begin{aligned} E_S^i(t) &= \int_0^{\delta_{\min}^i} t P_S^i(t) dt \\ &= \int_0^{\delta_{\min}^i} t \sum_{j=0}^{n-1} \left[ P_{ij}(t)' \prod_{k=0, k \neq j}^{n-1} (1 - P_{ik}(t)) \right] dt \end{aligned} \quad (8)$$

where  $\delta_{\min}^i$  indicates the delay of node  $i$  finding the first neighbor node in the communication range. After discovering the first neighbor node, node  $i$  adds it to the neighbor set and shares the sleep wake timetable of the neighbor set and shares the sleep wake timetable of the neighbor node with the neighbors in the set. Thereafter, node  $i$  will actively wake up with potential neighbor nodes until it finds nodes within all communication ranges. Therefore, node  $i$  finds that the expected time  $t_S^i$  of all  $n$  neighbor nodes satisfies the following conditions:

$$t_S^i \leq E_S^i(t) + 2 \max(T_h), \quad h \in [0, n-1] \quad (9)$$

where  $\max(T_h)$  is the maximum interval between two consecutive wake-up times of node  $h$ . After node  $i$  finds the first neighbor node with the expected time of  $E_S^i(t)$ , it needs at most  $2\max(T_h)$  time to perform neighbor confirmation and share its neighbor set information. Section V will compare the expectation of the Disco algorithm with the delay  $t_D^i$  and the expected discovery delay  $t_S^i$  of the SPND algorithm through simulation experiments.

## 2) ANALYSIS OF THE NUMBER OF WAKE-UPS AND ACTIVE WAKE-UPS

The wake-up time slot used by the node for neighbor discovery mainly consists of two parts: a time slot that wakes up on the established sleep wake-up timetable and a time slot that actively wakes up. In the SPND algorithm, after the node  $j$  acquires the neighbor of the node  $i$  through the neighbor node  $i$ , by increasing the wakeup time slot in the working mode  $\tau_j$  of the node  $j$  for neighbor discovery, the discovery delay can be reduced and the discovery probability can be increased. In order to reduce the energy consumption caused by the increased wake-up time slot, this paper designs a selective node active wake-up time.

Assuming that the nodes in the network are uniformly distributed and the density is  $\rho$ , that is, the network has  $\rho$  nodes in the range of unit area, there are average  $N^i = \pi R^2 \rho$  nodes in the range of distance  $R$  around the node  $i$ . In the existing node active wake-up algorithm, node  $i$  needs to actively wake up to find all potential neighbor nodes in the set  $G_t^{i1}$ ,  $G_t^{i2}$  and  $G_t^{i3}$ , that is, the average active wake  $N_t^i = \pi(R + \Delta S)^2 \rho$  times. In this design, node  $i$  only needs to actively wake up to find all potential neighbor nodes in the set  $G_t^{i2}$  and  $G_t^{i3}$ , that is, the average number of active wakes is:

$$N_t^{i'} = \left( \pi(R + \Delta S)^2 - \pi(R - \Delta S)^2 \right) \rho \quad (10)$$

Node  $i$  can reduce the number of active wakes about  $\varphi$  times,  $\varphi$  is expressed as:

$$\varphi = \frac{N_t^i - N_t^{i'}}{N_t^i} \times 100\% = \frac{(R - \Delta S)^2}{(R + \Delta S)^2} \quad (11)$$

Among them,  $\Delta S$  is the maximum distance that a node in the network moves at one time. It can be seen that the number of waking times that node  $i$  can reduce is a considerable part when  $\Delta S$  is not zero.

## 3) ENERGY CONSUMPTION ANALYSIS

The energy consumption  $E_n$  of a node in WSN mainly includes three parts: the energy consumption of the processor module  $E_p$ , the energy consumption of the communication module  $E_c$ , and the energy consumption of the sensing module  $E_s$ . The processor module has three working states, which are run, idle, and sleep. The communication module has six states, namely, send, receive (recv), off, idle, sleep, and channel detection evaluation (CCA/ED). The sensing module has only two states: on and off. When evaluating network neighbors to find energy consumption, the sensing module can be considered to be independent of the other two modules. Therefore, when evaluating the SPND algorithm energy consumption  $E_{n-SPND}$ , the energy consumption of the processor module and the communication module is mainly considered:

$$E_{n-SPND} = E_p + E_c \quad (12)$$

In the time slot when then node wakes up and actively wakes up, the processor may be in the running state or idle state.

The running state processor needs to consume energy  $E_{p-run}$  in a time slot, and the idle state part function is suspended, and the energy consumption  $E_{p-idle}$  is relatively small. The communication module may be in the transmit, receive or channel detect evaluation state. When the node is in the sleep state, the processor module and the communication module are also in a sleep state. At this time, most of the modules of the node are closed, and the system power consumption  $E_{p-sleep}$  and  $E_{c-sleep}$  are the smallest.

According to [12], the energy consumption  $E_{c-CCA/ED}$  of the node for channel detection evaluation in an awake state time slot is substantially equal to the energy consumption  $E_{c-send}$  and  $E_{c-recv}$  required for transmitting and receiving data. That is, in the time slot of the node waking up and actively waking up, the energy consumed by the communication module to consume  $E_c$  is certain, and the processor module has different energy consumption according to the difference between the running state and the idle state, but both are larger than the processor module energy consumption  $E_{p-sleep}$  in which the node is in a sleep state.

Therefore, this paper defines the  $E_{n-SPND}$  energy model of the SPND algorithm which is the product of the number of times the node is awake and actively awake, and the energy consumption  $E_n$  of the node waking up in each time slot. Then add the energy consumption of the node in sleep state, i.e:

$$\begin{aligned}
 E_{n-SPND} &= (E_p + E_c) + N_S(E_{p-sleep} + E_{c-sleep}) \\
 &= N(E_{p-run} + E_{p-idle}) + NE_c + N_S(E_{p-sleep} + E_{c-sleep}) \\
 &= N_{run}E_{p-run} + N_{idle}E_{p-idle} + NE_c \\
 &\quad + N_S(E_{p-sleep} + E_{c-sleep}) \tag{13}
 \end{aligned}$$

Among them,  $N$  is the number of nodes waking up and active waking, and  $N_S$  is the number of time slots in which the node is asleep.  $N_{run}$  indicates the number of times the processor module in  $N$  in the running state, and  $N_{idle}$  indicates the number of times the node processor module in  $N$  is idle.  $E_c$  indicates the average value of the energy consumption of the communication module for transmitting and receiving data or channel detection.  $E_{c-sleep}$  indicates the energy consumption of the communication module for sleep in a time slot.

#### 4) PROBABILITY ANALYSIS

In the existing active wake-up algorithm, the GBD algorithm can implement neighbor discovery with better delay and energy consumption. This paper compares the probability that node  $i$  finds all neighbors with the algorithm. In the GBD algorithm, the node  $i$  judges whether to actively wake up at the wake-up time of the node  $k$  according to the distances  $l_{ik}$  and  $l_{ij}$  between the nodes, where  $k$  is the neighbor of the node  $j$ . The probability that compute node  $k$  is the neighbor

of node  $i$  is [22]:

$$P_{j,ik}(l_{jk}, l_{ij}) = \begin{cases} \frac{1}{\pi} \arccos \left( \frac{l_{jk}^2 + l_{ij}^2 - R^2}{2l_{jk}l_{ij}} \right), & l_{jk} + l_{ij} > R \\ 1, & l_{jk} + l_{ij} \leq R \end{cases} \tag{14}$$

To determine whether node  $i$  actively wakes up to find node  $k$  is by setting different thresholds for  $P_{j,ik}(l_{jk}, l_{ij})$ . The algorithm only judges whether to actively wake up for neighbor discovery by probability. This may cause node  $i$  to actively wake up to discover nodes that are not in the range of neighbor, or miss the node in the neighbor range by actively waking up, the probability that  $P^i(t)$  is small for node  $i$  to discover all neighbor nodes in  $t$  time.

In the SPND algorithm, among the three sub-sets  $G_t^{i1}$ ,  $G_t^{i2}$  and  $G_t^{i3}$  of node  $i$ , the node of  $G_t^{i1}$  is still the neighbor of node  $i$  at  $t + \Delta t$ , which can be found through passive wake-up. The nodes in  $G_t^{i2}$  and  $G_t^{i3}$  may be neighbors of node  $i$  at  $t + \Delta t$ . It can be found through active wake-up, that is, node  $i$  can discover all neighbor nodes in three sub-sets  $G_t^{i1}$ ,  $G_t^{i2}$  and  $G_t^{i3}$ . It should be noted that in the actual network situation, due to the packet loss rate, the node movement may be frequent or fast, and the sub-set  $G_t^{i3}$  may not be able to obtain all the nodes in the  $[RR + \Delta S)$  range, causing the node  $i$  cannot find all neighbor nodes at  $t + \Delta t$ . However, the existence of these assumptions only affects the performance of the SPND algorithm and does not change the reliability of the SPND algorithm.

#### V. SIMULATION RESULTS AND ANALYSIS

In this paper, the performance of the SPND algorithm is tested by simulation experiments, and the experimental data of the SPND algorithm is compared with the data of the existing algorithms. In order to reduce the impact of experimental errors, each set of experiments was performed 1000 times and the data were averaged. This paper builds a simulation platform using MATLAB simulator, and then implements the SPND algorithm and the simulation of the three comparison algorithms of Disco, GBD and Birthday on the platform. In the simulation, the network is assumed to be deployed in the range of  $1000 m \times 1000 m$ , consisting of 500 randomly moving low-duty nodes. All nodes move within the network deployment area, and the nodes move using the Random Waypoint mobile model. The default moving speed of the node is  $1 m/s$ , which is randomly selected within the range of  $1 \pm 0.3 m/s$ , that is, the maximum moving speed of the node is  $1.3 m/s$  and the minimum moving speed is  $0.7 m/s$ . The duty ratio of the node is set to 5%, which means that the node will have a time slot in the waking state every 20 time slots, and the other time slots are in the sleep state. The default communication radius of nodes in the network is  $100 m$ . The TTL of the node neighbor information in the network (that is, the time when the neighbor information fails) is set to 5000 time slots, and each time slot is 10ms. This setting is in line with the actual situation. For example, a CC2420 wireless sensor node



needs to transmit and receive 1 *B* data for about 0.032 *ms*. Therefore, when the network is completely asynchronous, the 10 *ms* time slot can be guaranteed to a maximum extent. Nodes can perform neighbor discovery work in a time slot that wakes up at the same time.

In this paper, the data of the network average discovery delay, the number of node wake-ups, the node energy consumption and the node discovery ratio within the defined time delay is measured respectively. The comparison between the SPND algorithm and other algorithms is carried out through the four sets of simulation experimental data.

### A. NETWORK AVERAGE DISCOVERY DELAY

This section measures the average discovery delay of the network for different network duty cycles and node movement speeds. Firstly, the discovery delay of all neighbor nodes found in the network is determined, and then the average of the delays found by all nodes is taken as the average discovery delay of the entire network. Set the network node moving speed to 1 *m/s*, the duty ratio is from 1% to 10%, and the discovery delay with different duty ratios is shown in Figure 5(a). Set the node duty ratio in the network to 3%, and move the speed from 1 *m/s* to 10 *m/s*. The discovery delay with different moving speed is shown in Figure 5(b).

It can be found from Figure 5(a) that as the duty ratio of the node increases, the node has more wake-up time slots for neighbor discovery, which can reduce the neighbor discovery delay of the node. When the node duty ratio reaches 0.05 or above, the SPND algorithm can achieve neighbor discovery delay within 10sec. It can be found from Figure 5(b) that as the moving speed of the node increases, the discovery delay of all algorithms decreases to some extent. This is because the node moving speed is increased, the time of the node in the communication range is reduced, and the number of neighbor nodes that the node can discover is less, so the node discovery delay is reduced, and the average discovery delay of the entire network is reduced.

### B. NODE WAKE UP

This section mainly measures the number of waking times that the node needs to find all neighbor nodes under different duty cycles and moving speeds. The number of waking times here includes the number of times the node wakes up passively according to the sleep-waking timetable and the number of times the node actively wakes up according to the neighbor set information, and the simulation experiment data is shown in Figure 6. It can be seen from Figure 6 that the average number of active wake-ups of nodes in the SPND algorithm accounts for the total wakeup of the nodes. The average number of times is less than 30%, and the additional energy consumption is better than the existing active wake neighbor discovery algorithm. It can be found from Figure 6(a) that as the node duty ratio increases, the total number of wake-ups of the node decreases, and the number of active wake-ups of the node remains substantially stable. This is because the change in the duty cycle of the node does

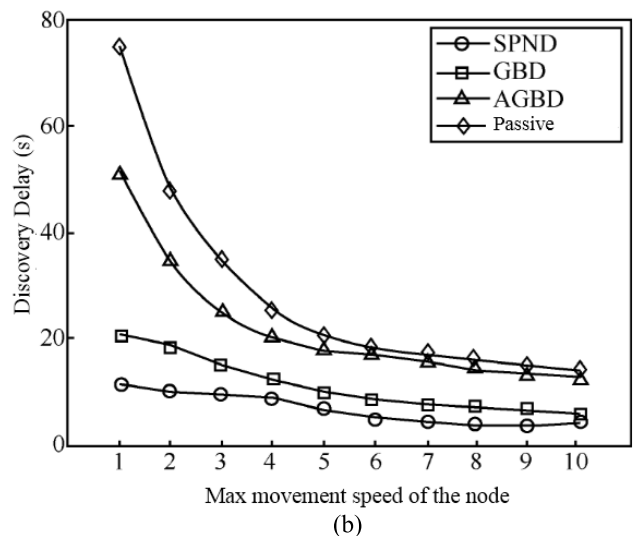
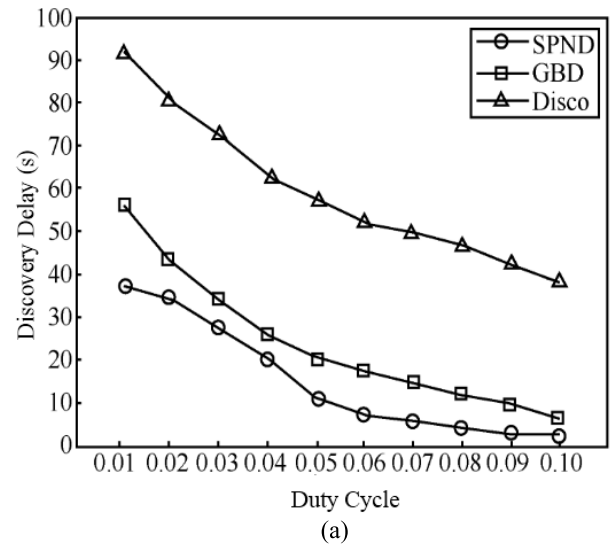


FIGURE 5. Network average discovery delay. (a) Discovery delay versus duty cycle. (b) Discovery delay with different moving speed.

not affect the number of neighbor nodes that the node needs to discover at a certain time, and the larger duty cycle causes the node to have more passive wake-up time, and therefore, the total number of wake-ups is reduced. It can be found from Figure 6(b) that when the moving speed of the node increases, the number of active wake-ups of the node will increase first and then remain flat, and the total number of wake-ups will also increase. This is because the increased speed of movement causes the set of neighbors around the node to change rapidly, and the node needs more active wake-up time to discover new neighbor nodes. When the node speed increases to a certain extent, the time of the node in the communication range is insufficient for the neighbor discovery work, and the number of active wake-ups of the node remains stable.

Figure 6(c) compares the wake-up time of the proposed SPND algorithm with the reference methods of Disco [10], GBD [15] and Birthday [16] algorithms. It is clear from

TABLE 1. Energy consumption analysis parameters.

Parameter	Value (J)
$E_{p-run}$	$1 \times 10^{-6}$
$E_{p-idle}$	$0.3 \times 10^{-6}$
$E_{p-sleep}$	$1 \times 10^{-9}$
$E_c$	$1.5 \times 10^{-5}$
$E_{c-sleep}$	$1 \times 10^{-6}$

Figure 6(c) that the proposed SPND algorithm has better wakeup time as compared with the reference methods which makes it more efficient for discovering neighbor nodes with less energy consumption and prolong node lifetime.

C. NODE ENERGY CONSUMPTION

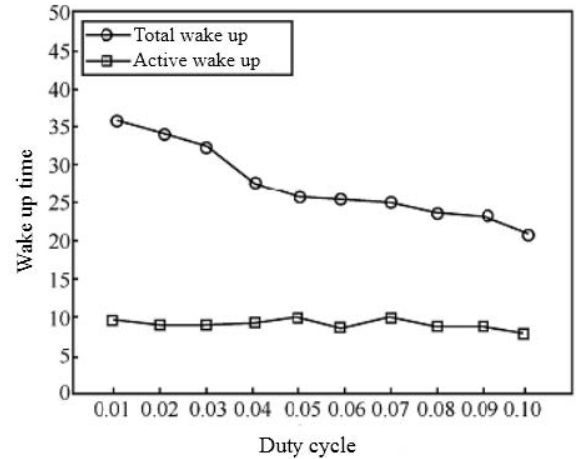
This section will show the network energy consumption of the SPND algorithm and compare it with the existing active neighbor discovery algorithm GBD. In the network energy consumption here, the simulation experiment considers the energy consumption of the processor module related to node neighbor discovery and the energy consumption of the communication module. The setting of each parameter value is shown in Table 1.

The simulation experiments respectively measured the average energy consumption of a neighbor discovery in any node in the network when the duty ratio is different from the node moving speed. The obtained data is shown in Figure 7.

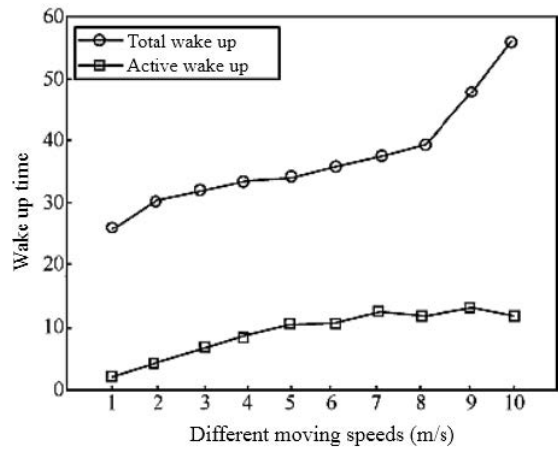
It can be found from Figure 7(a) that the node in the SPND algorithm can achieve a small energy consumption, and as the duty ratio increases, the energy consumption of the node for neighbor discovery can be reduced by a considerable part, while the energy consumption of the node in the GBD algorithm tends to be stable. It can be found from Figure 7(b) that the node energy consumption in the network increases due to the increase of the node’s moving speed, but the SPND algorithm can always achieve less network energy consumption than the GBD algorithm. Figure 7(c) compares the energy consumption of the proposed SPND algorithm with the reference methods [8], [10], and [15]. It can be seen from Figure 8 that the proposed SPND algorithm shows small energy consumption as the duty cycle increase and it has overall better energy consumption performance as compared with the reference methods GBD, Disco, and Birthday respectively.

D. NODE DISCOVERY RATIO WITHIN A DEFINED TIME DELAY

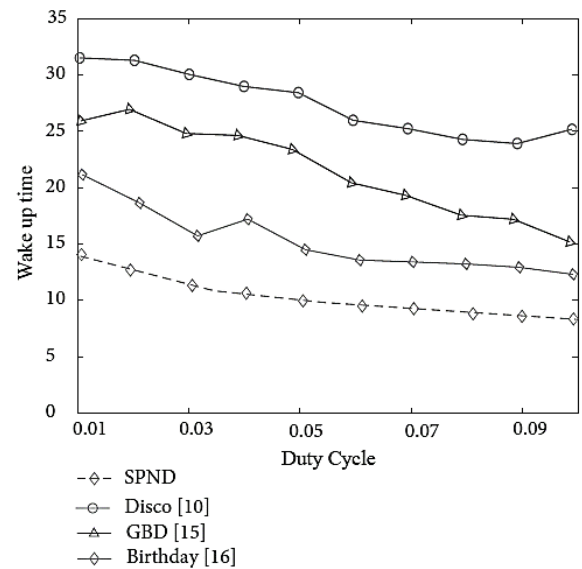
This section will show the node discovery ratio for a limited delay and compare the SPND algorithm with the classic Disco algorithm and Birthday algorithm. When the node duty



(a)

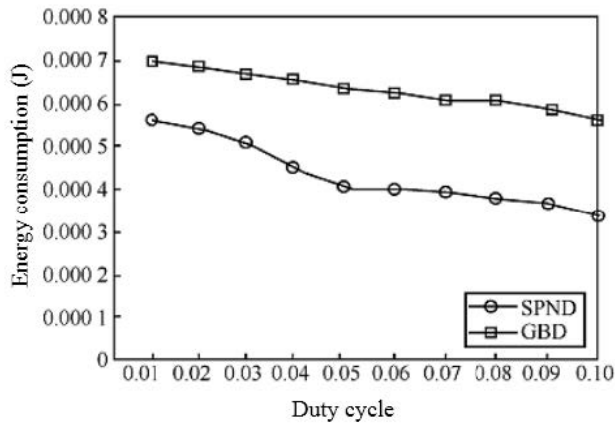


(b)

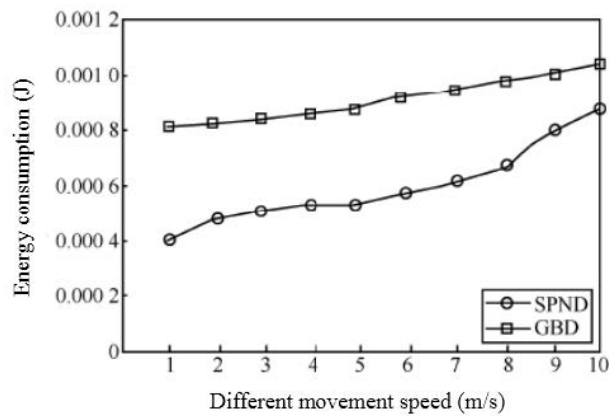


(c)

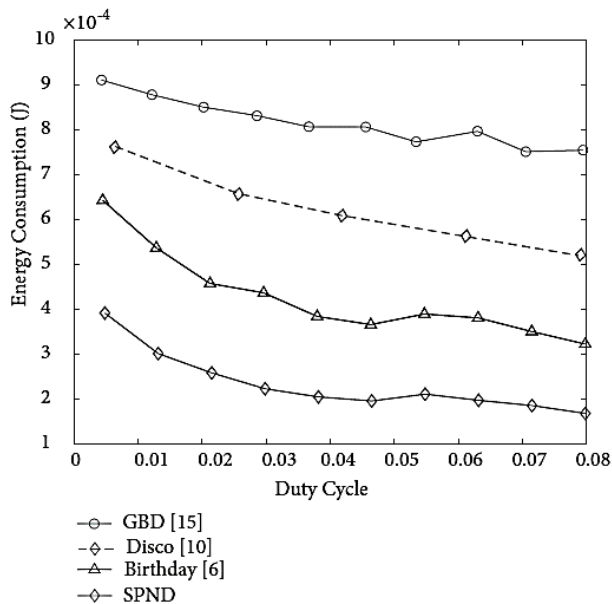
FIGURE 6. Node wake up analysis. (a) Number of wake-up times with different duty cycle. (b) Number of waking times at different moving speeds. (c) Number of wake-up times with different duty cycle by comparison with references [10], [15], and [16] methods.



(a)

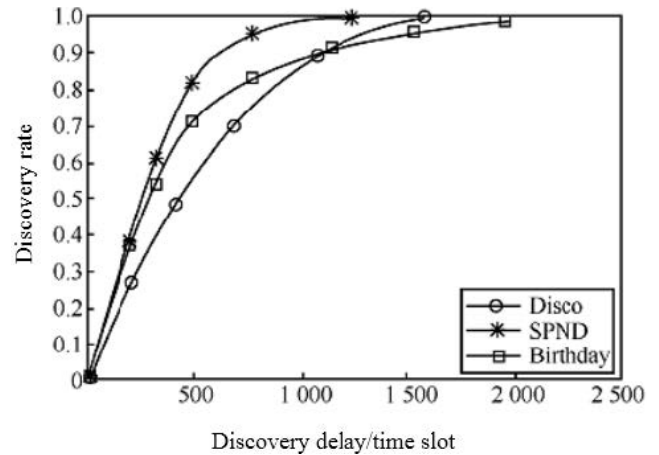


(b)



(c)

**FIGURE 7.** Average node energy consumption. (a) Energy consumption with different duty cycle. (b) Energy consumption with different moving speed (m/s). (c) Energy consumption with different duty cycle with reference [6], [10], and [15] comparison.



**FIGURE 8.** Limit of discovery.

**TABLE 2.** Comparison of algorithm performance.

Algorithm	Discovery Delay	Network Energy Consumption	Active Wake Up	Discovery Ratio Convergence
SPND	Very Small	Very Small	Smaller	Faster
GBD	Smaller	Larger	Larger	Faster
Disco	Larger	Smaller	-	Faster
Birthday	Larger	Smaller	-	Slower

ratio is 0.05 in the network and the node moving speed is 3m/s, the discovery ratio of the node-defined delay is shown in Figure 8. It can be seen from Figure 8 that when the defined delay is small, the SPND algorithm, like the Birthday algorithm, finds that the ratio rises faster and the Disco algorithm rises more slowly. After a certain ratio, the Birthday algorithm has a slower rise, and the SPND algorithm can maintain a faster rate of rising like the Disco algorithm. In the end, the SPND algorithm can achieve discovery ratios around 1300 time slots. Convergence to 1, about 200-time slots faster than the Disco algorithm.

**E. EXPERIMENTAL SUMMARY**

By comparing the performance of SPND algorithm with other algorithms, this paper compares the four aspects of network average discovery delay, node wake-up times, node energy consumption and node discovery ratio within limited time delay. The simulation experiment data shows that the neighbor discovery delay of the SPND algorithm is reduced by about 19% compared with the GBD algorithm, and the network energy consumption is reduced by about 30%. In addition, in terms of the discovery ratio of the defined delay, the SPND discovery ratio converges to 1 faster than the Disco algorithm by 13%. The algorithm performance pairs are shown in Table 2.

Among them, the Disco and Birthday algorithms are passive wake-up algorithms, and the nodes will not actively wake-up. Table 3 compares the computing time

TABLE 3. Comparison of algorithm computing time.

Algorithm	Computing Time (s)
SPND	30.21
GBD [15]	54.42
Disco [10]	52.71
Birthday [6]	49.13

of the proposed SPND algorithm with the reference methods [6], [8], [10], and [15]. It is clear from the results that the proposed SPND took less computing for neighbor node discovery as compared to GBD, Disco and Birthday algorithms.

## VI. CONCLUSION

In this paper, a new low-energy selective proactive wake-up fast neighbor discovery (SPND) algorithm is proposed for the problem that nodes in the low-duty-cycle wireless sensor network actively wake up to find that the neighbors have higher energy consumption. After constructing the initial neighbor set of the node by using the shared neighbor information method, the algorithm can selectively perform the active wake-up time of the designated node by dividing the neighbor subset of the node in the next step to realize the neighbor discovery of the node with low energy consumption. Theoretical analysis and experiments show that the SPND algorithm can achieve a smaller discovery time delay than Disco and GBD algorithms, while the network energy consumption is much better than those of Disco and GDB. In the active neighbor discovery algorithm, the node actively wakes up according to the neighbor information and can communicate with the potential neighbor node when waking up. Therefore, the requirement for node clock synchronization in the network is high. However, after a period of work, the node's clock will drift. Therefore, the next step is to study the asynchronous network when the node clock will drift.

## REFERENCES

- [1] A. Razaque and K. M. Elleithy, "Low duty cycle, energy-efficient and mobility-based border node—MAC hybrid protocol for wireless sensor networks," *J. Signal Process. Syst.*, vol. 81, no. 2, pp. 265–284, 2015.
- [2] S. Guo, Y. Yang, and C. Wang, "DaGCM: A concurrent data uploading framework for mobile data gathering in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 610–626, Mar. 2016.
- [3] F. Wang and J. Liu, "On reliable broadcast in low duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 767–779, May 2012.
- [4] Y. Qiu, S. Li, X. Xu, and Z. Li, "Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.
- [5] T. Meng, F. Wu, and G. Chen, "Code-based neighbor discovery protocols in mobile wireless networks," *IEEE Trans. Netw.*, vol. 24, no. 2, pp. 806–819, Apr. 2016.
- [6] M. J. McGlynn and S. A. Borbosh, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2001, pp. 137–145.
- [7] L. You, Z. Yuan, P. Yang, and G. Chen, "ALOHA-like neighbor discovery in low-duty-cycle wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Cancun, Mexico, Mar. 2011, pp. 749–754.

- [8] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," *Mobile Netw. Appl.*, vol. 10, nos. 1–2, pp. 169–181, 2005.
- [9] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 35–45.
- [10] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 71–84.
- [11] I. Niven and H. S. Zuckerman, *An Introduction to the Theory of Numbers*, 5th ed. Hoboken, NJ, USA: Wiley, 1991.
- [12] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 350–361.
- [13] H. Chen, W. Lou, Z. Wang, and F. Xia, "On achieving asynchronous energy-efficient neighbor discovery for mobile sensor networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 4, pp. 553–565, Oct./Dec. 2018.
- [14] P. H. Kindt, D. Yunge, G. Reinert, and S. Chakraborty, "Griassdi: Mutually assisted slotless neighbor discovery," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2017, pp. 93–104.
- [15] L. Chen et al., "Group-based neighbor discovery in low-duty-cycle mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 1996–2009, Aug. 2016.
- [16] A. Purohit, B. Priyantha, and J. Liu, "WiFlock: Collaborative group discovery and maintenance in mobile sensor networks," in *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Chicago, IL, USA, Apr. 2011, pp. 12–14.
- [17] D. Zhang et al., "Acc: Generic on-demand accelerations for neighbor discovery in mobile applications," in *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst.*, Toronto, ON, Canada, Nov. 2012, pp. 6–9.
- [18] D. Zhang, T. He, F. Ye, R. K. Ganti, and H. Lei, "EQS: Neighbor discovery and rendezvous maintenance with extended quorum system for mobile sensing applications," in *Proc. Int. Conf. Distrib. Comput. Syst.*, vol. 16, Jun. 2012, pp. 72–81.
- [19] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1448–1459, 3rd Quart., 2014.
- [20] T. Huang, H. Chen, Y. Zhang, and L. Cui, "EasiND: Effective neighbor discovery algorithms for asynchronous and asymmetric-duty-cycle multi-channel mobile WSNs," *Wireless Pers. Commun.*, vol. 84, no. 4, pp. 3031–3055, 2015.
- [21] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Sep. 2002.
- [22] Y. Qu, J. Fang, and S. Zhang, "Nearest neighbor nodes and connectivity of three-dimensional wireless sensor networks with Poisson point field," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, Chengdu, China, Jul. 2010, pp. 269–272.
- [23] J. Liang, X. Zhou, and T. Li, "Energy saving proactive neighbor discovery algorithm in mobile low-duty-cycle wireless sensor network," *J. Commun.*, vol. 39, pp. 45–55, Apr. 2018.



**OMAR A. SARAEREH** received the B.S. degree in telecommunication engineering from Mutah University, Jordan, in 1999, the M.Sc. degree in digital communication systems in U.K., and the Ph.D. degree in electrical and electronic engineering/mobile communications from Loughborough University, Loughborough, U.K., in 2005. From 2001 to 2005, he was a Member of Staff with the Centre for Mobile Communication Research, Loughborough University. He has over 12 years of academic and practical experience in electrical engineering, mobile communications, various antennas design, fabrication, and measurements, radiation hazards and health effects, and wireless communications. He was a high-level consultant and a turnkey solution originator in countless business and charitable sectors and an international public speaker and a trainer on a variety of business and people management topics. He is currently an Associate Professor with the Department of Electrical Engineering, The Hashemite University, Jordan. He has published many papers in various international journals and conferences.





**IMRAN KHAN** received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Engineering and Technology at Peshawar, Pakistan, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. He is also a Faculty Member with the University of Engineering and Technology at Peshawar. He has published various research publications in well-reputed international journals, with the IEEE, Elsevier, Springer, MDPI, and

Wiley. His research interests include 5G, massive MIMO, mmWave, wireless communications, WSN, NOMA, D2D, the IoT, and communication networks. He has reviewed many research papers as a reviewer and is a member of the IEEE ComSoC.



**BYUNG MOO LEE** received the Ph.D. degree in electrical and computer engineering from the University of California at Irvine, Irvine, CA, USA, in 2006. He has 10 years of industry experience, including research positions at the Samsung Electronics Seoul R&D Center, Samsung Advanced Institute of Technology, and Korea Telecom R&D Center. He is currently an Assistant Professor with the School of Intelligent Mechatronics Engineering, Sejong University, Seoul, South Korea. Dur-

ing his industry experience, he participated in the IEEE 802.16/11, Wi-Fi Alliance, and 3GPP LTE standardizations, and also participated in Mobile VCE and Green Touch Research Consortiums where he made numerous contributions and filed a number of related patents. His research interests are in the areas of wireless communications, signal processing, and machine learning applications.

Dr. Lee served as the Vice Chairman of the Wi-Fi Alliance Display MTG, in 2015–2016.

...