Neighborhood Discovery Algorithm in Wireless Local Area Networks Using Multi-beam Directional Antennas

Jin Wang, Wei Peng and Song Liu
College of Computer, National University of Defense Technology, Changsha, Hunan, China
Email: jinwang1206@163.com, wpeng@nudt.edu.cn, liusong13a@nudt.edu.cn

Abstract. Neighborhood discovery is an important step for Wireless Local Area Networks (WLAN) and the use of multi-beam directional antennas can greatly improve the network performance. However, most neighborhood discovery algorithms in WLAN, based on multi-beam directional antennas, can only work effectively in synchronous systems but not in asynchronous systems. And collisions at AP remain a bottleneck for neighborhood discovery. In this paper, we propose two asynchronous neighborhood discovery algorithms: asynchronous hierarchical scanning (AHS) and asynchronous directional scanning (ADS) algorithms. Both of them are based on three-way handshaking mechanism. AHS and ADS reduce collisions at AP to have a good performance in a hierarchical way and directional way respectively. In the end, the performance of the AHS and ADS are tested on OMNeT++. Moreover, it is analyzed that different application scenarios and the factors how to affect the performance of these algorithms. The simulation results show that AHS is suitable for the densely populated scenes around AP while ADS is suitable for that most of the neighborhood nodes are far from AP.

1. Introduction

Neighborhood discovery in WLAN refers to the process where a given node determines that it can communicate directly to the neighbors [1]. Neighborhood discovery, which is used to discover its one-hop neighboring nodes in the network initialization, is the first step in the network self-configuration process. The information about one-hop neighboring nodes is very important for MAC protocols [2], routing protocols [3] and topology control algorithms.

In traditional wireless network using omnidirectional antennas, a certain broadcast scheme could be used to establish connections among neighboring nodes. However, recent studies have shown that using multi-beam directional antennas can increase the throughput of wireless networks [4]. Using multi-beam directional antennas in WLAN has many advantages, such as increasing network capacity, extending transmission distances and reducing energy consumption. However, the limited beam width of multi-beam directional antennas poses many challenges to the design of neighborhood discovery algorithms due to the following reasons: 1) The limited radial range of beamwidth covers only a fraction of the azimuth, and the limitation requires the neighbor discovery scheme to be repeated in different directions to cover the whole azimuth. 2) Neighboring nodes must know when and where to point their directional beams to discover each other.

According to the features of multi-beam directional antennas, neighbor discovery algorithms can be divided into three categories: omnidirectional transmission and directional reception (OD), directional transmission and directional reception (DD) and directional transmission and omnidirectional reception (DO) [5]. Most neighbor discovery algorithms assume directional transmission and directional reception or directional transmission and omnidirectional reception. Some neighbor discovery algorithms for Ad Hoc Networks are discussed in [5], [6]. [7]-[9] assume successful
communication between a pair of nodes can occur only when one node's transmitting beam points in the direction of the other node's receiving beam. And [5], [6], [10] assume directional transmission and omnidirectional reception. However, these algorithms are not applicable for WLAN. Although a few algorithms have been proposed in [11], [12] for WLAN, most of them are discussed in synchronous scenarios. Synchronous algorithms need GPS or other timing devices in networks, which can cause increased costs of networks. Also occasions without timing devices are still common in practice.

In this paper, we propose two asynchronous neighbor discovery algorithms, asynchronous hierarchical scanning algorithm (AHS) and asynchronous directed scanning neighbor (ADS) in the wireless local area networks. The AP needs priori-knowledge about the number of neighboring nodes. No clock synchronization is required between nodes. The nodes in the network are equipped with the same multi-beam directional antennas, and the algorithm adopts directional transmission and omnidirectional reception to complete the neighbor discovery process. AHS algorithm and ADS algorithm reduce the probability of conflict in AP to a certain extent, and the two algorithms are suitable for different application scenarios.

The rest of the paper is arranged as follows. In Section II, we describe current research related to neighbor discovery algorithms using multi-beam directional antennas. Section III presents the system model. Section IV gives a description of two asynchronous algorithms, AHS and ADS, in detail and the synchronous neighbor discovery algorithm proposed in [6] is briefly reviewed. Section V compares the simulation results for all the three algorithms and analyses the influence of parameters on performance and the application scenarios of AHS and ADS algorithm. Finally, we present the conclusions and future work in Section VI.

2. Related Work
A lot of research on neighbor discovery algorithms has been conducted by [5]-[11]. Many algorithms assume directional transmission and directional reception. For example, in [7], Zhang extends the 1-way discovery algorithm to 2-way algorithms, and first proposes three random neighbor discovery algorithms, completely random or probabilistic algorithm with directional transmit and receive (CRA-DD), with directional transmit and omnidirectional receive (CRA-DO) and omnidirectional transmit and omnidirectional receive (CRA-OO). But for scan-based algorithms, the paper ignores the collision effect when the beamwidth is large. In [8], the author considers the collision case that more than one node exist in one directional beam and proposed a new Improved Scan-based algorithm (I-SBA) based on the SBA algorithm. The algorithm increases the detection probability when the number of neighbors is small, and reduces the collision probability when the number of neighbors is large. In [13], a new Scan-Based Algorithm with Probabilistic Idle (SBA-PI) is proposed. Each node chooses its mode in every time slot among transmission, reception or idle with probability $P_t$, $P_r$ and $P_i$ respectively. SBA-PI can address the high collision problem caused by high node density. These algorithms are synchronous, and need a Global Positioning System or other methods to achieve clock synchronous.

There have been some asynchronous neighbor discovery algorithms. In [14], Khatibi proposes an asynchronous pure directional quorum-based neighbor discovery algorithm for the first time. All nodes can find all their potential neighbors in a relatively small time interval. In [15], Feng uses the “Problem of Coloring Balls” model to analyze the one-way handshake-based asynchronous neighbor discovery algorithm, and proposes a hybrid handshake-based asynchronous algorithm. The feedback scheme is used which allows a better performance. In [9], a multi-carrier asynchronous neighbor discovery algorithm named 3-way MC-NDA is proposed. The algorithm avoids the failure of data transmission due to unnecessary collision and shortens the discovery delay by adding ACK packet. Compared with the two-way neighbor discovery algorithm, it overcomes the disadvantage that the node density has a great influence on the performance of algorithm.

Also many algorithms which assume omnidirectional transmission and directional reception are proposed. The packets sent by the node can be received when the omnidirectional receiving node is within the beam range of the directional sending node. In [10], the author presents a random or probabilistic protocol that called “birthday protocol”, which saves energy during the deployment of the node and makes neighbor discovery more efficient. In [5], a synchronous Direct-Discovery algorithm is proposed. In each time slot, a node selects directional transmission or omnidirectional reception with
probability $P_t$ and $1 - P_t$, and the direction angle is randomly selected. When a certain number of nodes are analyzed, the optimal $P_t$ that can maximize the probability of the node discovering its neighbors within a given amount of time. [9] proposes a synchronous neighbor discovery algorithm based on the four-way handshake mechanism, in which the nodes are in the transmit, receive or idle mode with probability of $P_t$, $P_r$ and $P_i$. The algorithm can discover not only one-hop neighbors, but also two-hop neighbors. And the ACK packet contains the information of the node itself and its neighbors previously discovered by the node. In [6], a synchronous neighbor node discovery algorithm is proposed in WLAN, which solves the conflict by using a contention-based directional polling mechanism. In this paper, we propose two asynchronous neighbor discovery algorithms in WLAN.

3. System Model

3.1. Multi-beam Directional Antennas Model
In this paper, we assume that the multi-beam directional antenna is composed of a circular antenna array. The cover-age area is divided into $M$ ($1 \leq M$) non-overlapping sectors, the relationship between the angle for each sector and the number of sectors is shown in (1). The other assumptions about the antenna are described as fallows.

$$\theta = \frac{2\pi}{M} \tag{1}$$

1) The antenna can work in omnidirectional mode and directional mode, the antenna mode can be freely switched and the switching time is ignored.

2) The direction of the beam can be adjusted in the directional mode by changing the antenna array factor, and the time required for the antenna to switch the beam direction can be ignored.

3) The antenna works in half-duplex mode and can only be in omnidirectional mode or directional mode at a certain time.

4) The side lobe gain can be ignored compared to the main lobe.

5) The beam angle of the omnidirectional mode is $\theta = 360^\circ$, the directional mode defines two kinds of beam angle $\theta$ and $\theta / 2$. The transmission power $P$ remains constant, and the variation of the beam angle leads to the variation of the range and coverage area of the antenna, as shown in Figure. 1. The relationship between the transmission distance $d$ and the beam angle $\theta$ is shown in (2), where $d_0$ is the transmission distance in the omnidirectional mode [6].

$$d = d_0 \frac{\sqrt{2.255}}{\theta} \tag{2}$$

![Figure 1](image-url) The transmission distance versus beam angle relationship.

3.2. Network Model
We assume that there is an access point (AP) with $N$ nodes around it in the network. Each node is equipped with the same multi-beam directional antenna, all nodes have the same transmission power, and the AP can communicate with all neighboring nodes. The node can adjust the working mode of the
antenna, and the node in the omnidirectional receiving state adjusts the antenna to omnidirectional mode for receiving information. The node in the omnidirectional transmission state adjusts the antenna to omnidirectional mode for sending information. There are two directional transmission states of the nodes, namely the directional transmission state A and the directional transmission state B respectively, corresponding to the directional mode of beam angle $\theta$ and $\theta/2$. Other features of these nodes are shown as follows.

1) Each node has a unique ID.

2) All nodes and the AP work in half-duplex mode. At any time, the AP can be in omnidirectional receiving state, omnidirectional transmission state, or directional transmission state. The nodes can be either in omnidirectional receiving state or directional transmission state, but not both.

3) The AP and one node can communicate directly, required AP and the node to be distributed in omnidirectional receiving state and directional transmission state or in omnidirectional receiving state and omnidirectional transmission state, and the receiving node is within the beam range of the transmitting node. At the same time, the receiving node can receive a complete packet should be ensured. In other words, the node in receiving state cannot obtain information through parts of a data packet. Figure. 2 shows an example of successful communication between node A and node B.

4) Data packets cannot be recovered when a node receives packets from two or more neighboring nodes at the same time, and that is called collisions. Figure. 3 shows an example of a conflict where it receives packets from node A and node C at the same time.

5) The clock is asynchronous between nodes. All packets have a fixed duration.

4. Neighbor Discovery Algorithm
In this section, we first give a brief description of the synchronous neighbor discovery algorithm (hereinafter referred to as IL algorithm) proposed in [6]. Then, we describe our proposed neighbor discovery algorithms, AHS and ADS. In these two algorithms, the AP makes use of three-way handshaking mechanism to find neighboring nodes and both of them can efficiently reduce collision. AHS uses a hierarchical scanning mechanism to find neighboring nodes in different regions. These regions are divided according to the communication radius of the AP. ADS finds neighbors more
quickly using a directional scanning method, in which it divides the communication area into multiple sectors. For ease of description, the symbols are defined as follows.

N: the number of neighboring nodes around the AP, and there are N + 1 nodes totally in the network.

ϕ: the beam direction angle of multi-beam directional antennas.

τp: the packet transmission time.

td: the delay of packet propagation.

nA, nB: the number of beams corresponding to the directional transmission state A and the directional transmission state B respectively.

4.1. Synchronization Neighbor Discovery Algorithm
The IL algorithm assumes time is slotted and all nodes are perfectly synchronized on time slots, each time slot is divided into L mini-slots, as shown in Figure 4.

Step 1, at the beginning of each slot, that is at time (a), and after all users from the previously formed beam are resolved, the AP selects a new direction to send polling message PM.

Step 2, time (b) is the beginning of mini-slot. And the AP turns to omnidirectional receiving state at the beginning of time (b). If the AP successfully receives an ACK, it happens that the AP only receives an ACK from neighboring nodes (for example, node j).

Step 3, the RACK containing the ID of node j and AP is transmitted at time (c). If the AP receives more than one ACK, there is collision and AP does not transmit anything.

Repeat the step 2 and step 3 until all neighboring nodes in the beam range are resolved. In the mini-slot, if the AP receives an ACK, that is there are still unsolved neighboring nodes.

The other nodes are in omnidirectional receiving state at time (a) (for example, node j). If the polling message PM from the AP is received and the RACK containing its own ID has not been previously received, then send an ACK to the AP at time (b), the ACK contains ID of node j. The node j changes to omnidirectional receiving state at time (c). If a RACK containing ID of node j from the AP is received, the node j sends RACK no longer. Otherwise, the ACK is sent with probability p at the beginning of the next mini-slot. Repeat the above process until all the beams have been traversed.

4.2. Asynchronous Hierarchical Scanning Algorithm
The Asynchronous Hierarchical Scanning algorithm (AHS) is based on the three-way handshake mechanism to complete the neighbor discovery process. For the AP, the algorithm is divided into three phases.

Stage 1, the AP is in omnidirectional transmission state, and sends the polling message PM, which is identified by m1 and contains the ID of the AP. After sending PM, the AP enters omnidirectional receiving state and continues for T1 time. If an ACK is successfully received from the neighboring nodes during the period, then parse the contents of ACK. And after the end of the omnidirectional receiving state, the AP enters the omnidirectional transmission state and transmits the RACK. The ID of the neighboring nodes included in the received ACK and the ID of the AP itself are added to the RACK.

Stage 2, the AP converts to directional transmission state A, randomly selects a beam direction angle ϕ, and direction-ally sends PM. The PM is identified by m2 and contains the ID of the AP. After sending PM, the AP enters omnidirectional receiving state and continues for T2 time. If the ACK is successfully received, then analyze it. At the end of omnidirectional reception state, the AP turns into
directional transmission state A, and the beam direction angle is still \( \varphi \). Then directly sends the RACK.

After that, the next beam direction angle is selected in the counterclockwise direction based on the beam direction angle \( \varphi \), and repeats the above process until all the \( n_A \) beam is traversed.

Stage 3, the AP is converted to directional transmission state B, randomly selects a beam direction angle \( \varphi \), the discovery process is similar to stage two. The relationship between the duration time \( T_i \) of AP and the selected maximum of random integer \( k_i \) by neighboring nodes after receiving PM identified as \( m_i \) is as shown in (3).

\[
T_i = t_p \times 2 + t_d \times (k_i + 1) (i = 1, 2, 3)
\]  

(3)

For neighboring nodes around the AP, for example, expressed in node j, node j is in omnidirectional receiving state. When the PM is received, node j does not receive the RACK from the AP or receives the RACK from the AP but the RACK does not contain the ID of node j. And if the PM is identified as \( m_i \), then randomly selects an integer \( R \) in the interval \([0, k_3]\). After \( R \times t_d \) time, node j converts to directional transmission state A, then adds the ID of node j and the AP to the ACK and sends the ACK in the direction of receiving the PM. If the received PM is identified as \( m_2 \) or \( m_3 \), the algorithm process is similar to the process above. After sending the ACK, node j changes to omnidirectional receiving state. When the RACK from the AP is received, then judges whether or not the RACK contains its own ID. If the ID is included, it indicates that the AP has discovered node j, and node j no longer sends an ACK when it receives the PM from the AP again.

For the AP, the variable \( mode_{AP} \) indicates the state of the AP. \( mode_{AP} \) has four values, \( omniTrait \), \( omniRec \), \( dirA \) and \( dirB \), which indicates that the AP is in the omnidirectional transmission state, the omnidirectional receiving state, the directional transmission state A and the directional transmission state B. After the AP sends PM in the omnidirectional state, the directional transmission state A and the directional transmission state B, the duration time of the omnidirectional receiving state is respectively \( T_1 \), \( T_2 \) and \( T_3 \). The neighboring nodes around the AP are represented by j. The algorithm process at AP is as follows.

1) After the node starts, it enters to the omnidirectional sending state, \( mode_{AP} = omniTra \):
   a) Send PM identified as \( m_1 \);
   b) \( mode_{AP} = omniRec \), and continue for \( T_1 \) time;
   c) \( mode_{AP} = omniTra \), if the ACK is successfully received from j in the omnidirectional reception period, then add the ID of j into the RACK and send the RACK;
2) \( mode_{AP} = dirA \), randomly selects a beam direction angle \( \varphi \);
   a) FOR \(( i = 1; i < n_A; i++)\)
     i) Send PM identified as \( m_2 \) at direction angle \( \varphi \);
     ii) \( mode_{AP} = omniRec \), and continues for \( T_2 \) time;
     iii) \( mode_{AP} = dirA \), if the ACK is successfully received from j in the omnidirectional reception period, then add the ID of j into the RACK. If the channel is idle, send the RACK at the direction angle \( \varphi \), and select the next beam direction angle \( \varphi \); in counterclockwise order;
     iv) IF \( i = n_A \)
     v) \( mode_{AP} = dirB \);
3) \( mode_{AP} = dirB \), randomly select a beam direction angle \( \varphi \);
   a) FOR \(( j = 1; j < n_B; j++)\)
     i) Send PM identified as \( m_3 \) at direction angle \( \varphi \);
     ii) \( mode_{AP} = omniRec \), and continue for \( T_3 \) time;
     iii) \( mode_{AP} = dirB \), if the ACK is successfully received from j in the omnidirectional reception period, then add the ID of j into the RACK. If the channel is idle, send the RACK at the direction angle \( \varphi \), and select the next beam direction angle \( \varphi \); in counterclockwise order;
     iv) IF \( j = n_B \), the algorithm executes once.

For neighboring nodes around the AP, we take node j as an example. And set a boolean variable \( isFound \) for node j. When the ACK from node j is received by the AP and the ID of j is included in the AP’s RACK, \( isFound \) is set to true, otherwise false. The variable \( mode_j \) indicates the state of j. \( mode_j \) has three values, \( omniRec \), \( dirA \) and \( dirB \). Also we set a variable k, and k has three values as \( k_1 \), \( k_2 \) and
$k_3$, respectively the maximum value of random integers is chosen when $j$ receives PM identified as $s_1$, $s_2$ and $s_3$. The algorithm process for node $j$ is described as below.

1) After the node starts, it enters the omnidirectional receiving state, $mode_j = \text{omniRec}$.

2) When the PM sent from the AP is received, if isFound is false, then
   a) IF the identification from PM $s_i = s_1$
      i) $m = \text{random}(0, k_1)$;
      ii) After waiting for $m \times t_d$ time, $mode_j = \text{dirA}$, if the channel is idle, send ACK to the AP in the direction of receiving the PM;
   b) ELSE IF the identification from PM $s_i = s_2$
      i) $m = \text{random}(0, k_2)$;
      ii) After waiting for $m \times t_d$ time, $mode_j = \text{dirA}$, if the channel is idle, send ACK to the AP in the direction of receiving the PM;
   c) ELSE IF the identification from PM $s_i = s_3$
      i) $m = \text{random}(0, k_3)$;
      ii) After waiting for $m \times t_d$ time, $mode_j = \text{dirB}$, if the channel is idle, send ACK to the AP in the direction of receiving the PM;
   d) After the ACK is sent, $mode_j = \text{omniRec}$;

3) When receives the RACK sent by the AP, if isFound is false and the RACK contains the ID of the node $j$, then set isFound = true.

4.3. Asynchronous Directional Scanning Algorithm

The Asynchronous Directional Scanning Algorithm (ADS) is also based on the idea of three-way handshake of TCP protocol, and utilizes the direction transmission state $B$ of the multi-beam directional antenna to fulfill the neighbor discovery process. At the beginning of ADS, the AP is in the directional transmission state $B$, randomly selects a beam direction angle $\varphi$ to send the polling message PM, and then selects the next beam direction angle to send the PM in the counterclockwise order until the PM is transmitted in all directions of the directional transmission state $B$. Where PM contains the ID of AP and $n_i$, $n_i$ indicates the number of times that the AP sends PM. For example, if the AP sends PM for the first time, then the value of $n_i$ is $1$, and the maximum value of $n_i$ is the number of beams when the AP is in directional transmission state $B$, namely $n_i = n_B$.

Then the AP turns into omnidirectional receiving state, where the duration of the omnidirectional reception is $T$, and the relationship between $T$ and the maximum value $k$ of the selected integer after the neighboring nodes receive the PM is shown in (4). If the reply packet ACK from the neighboring nodes is received during the period, then the node’s ID in the ACK and the direction of the arrival (DOA) of ACK are recorded. At the end of omnidirectional receiving state, the AP is converted to the directional transmission state $B$, the beam direction angle $\varphi$ is randomly selected, and sends RACK. The information about the ID of neighboring nodes received at the direction angle $\varphi$ and the ID of the AP are placed in the RACK. And then selects the next beam direction angle in the counterclockwise order to send a RACK until all beam directions are polled.

$$T = t_p \times 2 + t_d \times (k + 1)$$ (4)

The neighboring nodes around the AP are in the omnidirectional receiving state, node $j$ is used as an example. When the polling message PM from the AP is received, node $j$ parses the PM and gets the value of $n_i$. Then node $j$ randomly selects the integer $R$ in the interval $[0, k]$, after $(R + n_B - n_i) \times t_d$ time, sends the ACK in the direction of receiving the PM, where ACK contains the ID of the node $j$ and the ID of the AP. When the RACK from the AP is received, node $j$ judges whether or not the RACK contains the node $j$’s own ID. If the ID is included, it indicates that the node $j$ has been found by the AP.

For the AP, we set the variable $n_l$ to indicate that the number of times that the AP sends the PM. The variable $mode_{Ap}$ indicates the state of the AP. $mode_{Ap}$ has two values, omniRec and dirB, which indicates that the AP is in the omnidirectional receiving state and the directional transmission state $B$. The neighboring nodes around the AP are denoted by $j$. The algorithm process is as follows.
1) After the node starts, it enters the directional transmission state B, \( mode_{AP} = \text{dirB} \), randomly selects a beam direction angle \( \varphi \);
   a) FOR \( (n_i = 1; n_i < n_B; n_i + +) \)
   i) If the channel is idle, send the PM contains \( n_i \) and the ID of AP at the direction angle \( \varphi \), and select the next beam direction angle \( \varphi \) in the counterclockwise order;
   ii) IF \( n_i = n_B \), it enters the omnidirectional receiving state;
2) \( mode_{AP} = \text{omniRec} \), and continue for \( T \) time. If an ACK from node \( j \) is received, the ID and the DOA of \( j \) are recorded;
3) At the end of the omnidirectional receiving state, \( mode_{AP} = \text{dirB} \), randomly select the beam direction angle \( \varphi \);
   a) FOR \( (n_i = 1; n_i < n_B; n_i + +) \)
   i) Add the ID of node \( j \) received at the angle \( \varphi \) and the ID of the AP to the RACK. If the channel is idle, send a RACK at the direction angle \( \varphi \) and select the next beam direction angle \( \varphi \) in counterclockwise order;
   b) IF \( n_i = n_B \), the algorithm executes once.

The nodes around the AP is assumed to be node \( j \). We set a boolean variable isFound, and isFound is set to true when node \( j \) receives RACK contains the ID of \( j \) from the AP, otherwise false. The variable \( mode_j \) indicates the state of \( j \). \( mode_j \) has two values, omniRec and dirB. The algorithm process for node \( j \) is described as below.

1) The node \( j \) enters omnidirectional receiving state after it starts, \( mode_j = \text{omniRec} \);
2) When the PM sent by AP is received, if isFound is false, then
   a) \( n = n_i \);
      i) \( m = \text{random}(0, k) \);
      ii) Waiting for \( (m + n_B - n) \times t_d \) time, after that, if the channel is idle, \( mode_j = \text{dirB} \), then send the ACK to the AP at the direction of receiving PM;
      iii) After the ACK is sent, \( mode_j = \text{omniRec} \);
3) When the RACK sent by AP is received, if isFound is false and the RACK contains the ID of the node \( j \), then
   a) isFound = true;

5. Evaluation

5.1. Experiment Setup

In this paper, OMNeT++ 5.1.1 is used to simulate the experiment, and we construct an AP-based network. The nodes in the network are equipped with multi-beam directional antennas. The nodes are randomly distributed in a square with an area of \( 2000 \text{m} \times 2000 \text{m} \). The AP is located in the center of the square. The size of the network varies from 20 nodes to 200 nodes. The transmission rate of the packet namely the transmission delay is certain, and the packet transmission delay is also a fixed value. The parameters used in the simulation experiment are shown in Table 1.

We assume the total number of nodes in the network is 20, 50, 100, 200, that is, the number of neighboring nodes \( N = 19, 49, 99, 199 \) and an AP. In these four scenarios, compare the performance of the asynchronous hierarchical scanning algorithm (AHS), the asynchronous directional scanning algorithm (ADS) and the synchronization algorithm (IL). In the AHS algorithm, the corresponding maximum value of random numbers in the optional integer interval is \( k_1, k_2 \) and \( k_3 \) when the neighboring nodes receive the PM identified as \( s_1, s_2 \) and \( s_3 \). In the ADS algorithm, the maximum random integer that the neighboring nodes can select when receiving the PM is \( k \). In the IL algorithm, the mini-slots and the probability of the ACK returned by neighboring nodes is represented by \( L \) and \( P \).

We adjust the values of \( L \) and \( P \) to make the algorithm as optimal as possible in the case of different number of neighboring nodes.
5.2. Performance Results
When the number of neighboring nodes is certain, the 50 kinds of topological structures of neighboring nodes distributed randomly are generated, and the AHS, ADS and IL algorithm are executed respectively. The time required for all neighboring nodes to be found is compared with the statistical average. When the number of neighboring nodes N is 19, the k_1, k_2 and k_3 of the AHS algorithm is 5, 5, 5, the k of the ADS algorithm is 20, the L and P of the IL algorithm are 6 and 0.5. When number of neighboring nodes N is 49, the k_1, k_2 and k_3 of the AHS algorithm is 5, 5, 5, the k of the ADS algorithm is 50, the L and P of the IL algorithm are 6 and 0.5. When the number of neighboring nodes N is 99, the k_1, k_2 and k_3 of the AHS algorithm is 5, 10, 10, the k of the ADS algorithm is 100, the L and P are 10 and 0.2. When the number of neighboring nodes N is 199, the k_1, k_2 and k_3 of the AHS algorithm is 5, 10, 10, the k is 200, and the L and P are 10 and 0.2. The comparison of algorithm performance is shown in Figure. 5. When the number of neighboring nodes is small, the ADS performs better than AHS and IL, and the AHS requires less time to find neighboring nodes than IL. When the number of neighboring nodes is large, the performance of AHS and ADS is better than that of IL, and it takes less time for ADS than AHS to find all neighboring nodes. When almost 75% of neighbors are found of the total number of neighboring nodes, the AHS and ADS performance is similar. With the increase of the number of neighboring nodes, the difference in performance of AHS, ADS and IL algorithm is more obvious, and two asynchronous neighbor discovery algorithms proposed in this paper are more advantageous.

Table 1 Experimental Parameters

| Parameter Name                  | Parameter Name                  |
|---------------------------------|---------------------------------|
| area                            | 2000 meters × 2000 meters       |
| number of Neighbor Nodes        | 20, …, 200                      |
| packet transmission delay       | 60us                            |
| packet propagation delay        | 30us                            |
| omnidirectional transmission radius | 150m                           |
| beam angle of directional mode A | 45°                             |
| beam angle of directional mode B | 22.5°                           |

Neighboring nodes are distributed randomly, and the neighboring nodes are scattered around the AP, that is, the nodes in the area near the AP are few, and most nodes in the area far away from the AP. The experimental results show that ADS is more suitable for such scenes compared to AHS.

We assume that the AP are located at the center of the network area, the neighboring nodes of the AP are distributed around the AP which is similar to the normal distribution pattern. The nodes in the area near the AP are more concentrated, that is, the number of nodes is larger, and the nodes in the area far from the AP are less. In this scenario, we construct 50 kinds of network topologies, and run the AHS and ADS algorithm respectively. According to the statistical average of the simulation results, the line graph of the relationship between the number of nodes found and the required time is drawn, as shown in Figure. 6 (a). Similarly, in this scenario, 50 network topologies with a total number of 100 nodes are constructed and the parameters are adjusted so that the time required for AHS and ADS algorithm to run once is roughly the same. Record the algorithm execution times and the number of neighboring nodes discovered by AP at the end of each algorithm execution. And plot the histogram that is the comparison of run times and the number of neighboring nodes found, as shown in Fig. 6 (b).
Figure 5 Performance comparison of algorithms. (a) Neighboring nodes in the network is 19. (b) Neighboring nodes in the network is 49. (c) Neighboring nodes in the network is 99. (d) Neighboring nodes in the network is 199.

Figure 6 Performance comparison of AHS and ADS. (a) The number of neighboring nodes found and the required time. (b) Run times and the number of neighboring nodes found.
When the number of nodes in the area near the AP is large, the AHS can find more nodes in the same run times than the ADS in the early stage of algorithm execution and the performance is better than that of the ADS. However, at the end of the algorithm execution, AHS takes more time to find the same number of neighboring nodes as ADS. When AHS and ADS spend roughly the same time for each execution, the two algorithms need to execute four times to find all neighbors. However, when the algorithms execute once, the AHS finds more nodes than the ADS. This is because the hierarchical scanning mechanism is used in AHS, and the PM is sent in the area near the AP first, and AP immediately transfers to the receiving state after sending a PM, which can quickly find closer neighbors. While ADS uses the directional scanning mechanism, the AP sends PM at all beam direction and then turns into receiving state, which causes ADS to find neighbors slowly at the beginning. Therefore, the AHS algorithm is more suitable for the scene that most nodes distributed in the area close to the AP and less nodes in the area away from the AP, it can quickly find neighbors. ADS is applicable to scenes that most nodes are in the area away from AP.

5.3. The Effect of Parameter $k$

We analyze the influence of parameters on AHS algorithm. When the number of neighboring nodes is fixed, this paper analyze the influence of the maximum value of random integers $k_1$, $k_2$ and $k_3$ can be selected by neighbors on the algorithm performance. The number of neighbors is 99, and we construct 20 different network topologies in which nodes are randomly distributed. The $k_1$, $k_2$ and $k_3$ are respectively $k_1 = 5$, $k_2 = 10$, $k_3 = 10$ and $k_1 = 10$, $k_2 = 10$, $k_3 = 10$ and $k_1 = 5$, $k_2 = 10$, $k_3 = 5$ and $k_1 = 5$, $k_2 = 15$, $k_3 = 10$. The line graph of the number of found neighbors and the time required is shown in Figure 7 with these four kinds of values, where the performance is best when the value is $k_1 = 5$, $k_2 = 10$, $k_3 = 10$. That is because nodes are randomly distributed, and the nodes in the omni-directional transmission range of AP are less, $k_1 = 5$ can avoid the conflict that the neighboring nodes in the range send ACK at the same time. More nodes are located in the range of directional beam of AP, $k_2$ and $k_3$ are 5 will increase the probability of conflict generated at AP, thereby the performance degrades, the value of $k_2$ and $k_3$ should be increased. While $k_2 = 15$ reduces the conflict probability, it increases the execution time of the algorithm, which makes the algorithm performance worse. Compared to the other two values, $k_2 = 10$ makes the performance better.

![Figure 7 Effect of parameter $k$ on AHS.](image1)

Then we analyze the influence of parameters on ADS algorithm. When the number of neighboring nodes is certain, the maximum value $k$ in the random integer interval chosen by the neighbors affects the performance of the algorithm. This paper assumes that the number of neighbors is 19, constructs 20 kinds of network topologies with nodes randomly distributed, $k$ value is 10, 20 or 30. The line graph of the number of found nodes and the time required with different $k$ is shown in Figure 8. Compared with the other two values, the algorithm performance is the best when $k$ is 20. When the $k$ value is 10, the
neighboring nodes send the ACK at the same time so that the probability of collision at the AP is large, and the times of algorithms executed to find all neighbors is more, which will take more time. When the k is 30, although the probability of collision at AP is reduced, the times of algorithms that need to be executed to find all neighbors is the same as the number of k values of 20. But it will take longer time to execute the algorithm once, so the algorithm performance is worse.

6. Conclusion and Future Work
This paper proposes two algorithms as AHS and ADS to solve the problem of neighbor discovery with multi-beam directional antennas. AHS makes use of omnidirectional /directional transmission and omnidirectional receiving mode of antennas, and ADS adopts directional transmission and omnidirectional receiving mode. These two algorithms use similar back-off mechanism to reduce the collision probability at AP when more than one neighbor send ACK at the same time, which makes the algorithm more efficient. Also, the performance of AHS and ADS is compared and the applicable scenarios of the two algorithms are analyzed.

In the future works, we can construct the mathematical model of these two algorithms and theoretically analyze the relationship among the number of neighbors, the distribution of neighboring nodes and the parameters of the algorithm to make the algorithm performance better, and analyze the relationship between the number of neighbors and the number of times that the algorithm needs to run to find all neighboring nodes. It is also possible to record the number of nodes found in each beam direction during the neighbor discovery process and use this information to improve the performance of AHS and ADS algorithms, so that a more efficient neighbor discovery algorithm can be designed.

7. Acknowledgment
This work is supported by projects of Natural Science Foundation of China under grant no. 61601483 and 61379147.

8. References
[1] S. A. Borbash, A. Ephremides, and M. J. McGlynn, “An asynchronous neighbor discovery algorithm for wireless sensor networks,” Ad Hoc Networks, vol. 5, no. 7, pp. 998–1016, 2007.
[2] Bao, Lichun, GarciaLunaAceves, and J. J., “A new approach to channel access scheduling for ad hoc networks,” International Conference on Mobile Computing and Networking, vol. 63, no. 1, pp. 210–221, 2001.
[3] Z. Zhang, “Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas,” in IEEE International Conference on Communications, 2005, pp. 3386–3390 Vol. 5.
[4] S. Yi, Y. Pei, and S. Kalyanaraman, “On the capacity improvement of ad hoc wireless networks using directional antennas,” in ACM International Symposium on Mobile Ad Hoc NETWORKING & Computing, 2003, pp. 108–116.
[5] S. Vasudevan, J. Kurose, and D. Towsley, “On neighbor discovery in wireless networks with directional antennas,” in IEEE Joint Conference of the IEEE Computer and Communications Societies, 2005, pp. 2502–2512 vol. 4.
[6] G. Astudillo and M. Kadoch, “Neighbor discovery and routing schemes for mobile ad-hoc networks with beamwidth adaptive smart antennas,” Telecommunication Systems, pp. 1–11, 2017.
[7] Z. Zhang and B. Li, “Neighbor discovery in mobile ad hoc selfconfig-uring networks with directional antennas: algorithms and compari-sons,” IEEE Transactions on Wireless Communications, vol. 7, no. 5, pp. 1540–1549, 2008.
[8] H. Cai, B. Liu, L. Gui, and M. Y. Wu, “Neighbor discovery algorithms in wireless networks using directional antennas,” in IEEE International Conference on Communications, 2012, pp. 767–772.
[9] S. Zhao, Y. Liu, T. Yang, Z. Feng, Q. Zhang, and C. Gao, “3-way multi-carrier asynchronous neighbor discovery algorithm using directional antennas,” wireless communications and networking conference, pp. 1–6, 2016.

[10] M. J. Meglynn and S. A. Borbash, “Birthday protocols for low en-ergy deployment and flexible neighbor discovery in ad hoc wireless networks,” in ACM Interational Symposium on Mobile Ad Hoc NETWORKING and Computing, MOBIHOC 2001, October 4-5, 2001, Long Beach, Ca, Usa, 2001, pp. 137–145.

[11] I. Koutsopoulos and L. Tassiulas, “Fast neighbor positioning and medium access in wireless networks with directional antennas ,” Ad Hoc Networks, vol. 11, no. 2, pp. 614–624, 2013.

[12] E. Felemban, R. Murawski, E. Ekici, and S. Park, “Sand: Sectored-antenna neighbor discovery protocol for wireless networks,” in IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2010, pp. 1–9.

[13] B. Liu, B. Rong, R. Q. Hu, and Y. Qian, “Neighbor discovery algorithms in directional antenna based synchronous and asynchronous wireless ad hoc networks,” Wireless Communications IEEE, vol. 20, no. 6, pp. 106–112, 2013.

[14] S. Khatibi, M. Dehghan, and M. A. Poormina, “Quorum-based pure directional neighbor discovery in self-organized ad hoc networks,” in International Symposium on Telecommunications, 2010, pp. 476–481.

[15] F. Tian, B. Liu, H. Cai, H. Zhou, and L. Gui, “Practical asynchronous neighbor discovery in ad hoc networks with directional antennas,” IEEE Transactions on Vehicular Technology, vol. 65, no. 5, pp. 3614–3627, 2016.