An Optimal Multi-round Multi-slot Hello-Reply Directional Neighbor Discovery Algorithm

Xinru Li, Zhongjiang Yan(✉), Mao Yang, Bo Li, and Hang Zhang

School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China
1069769987@qq.com, {zhjyan,yangmao,libo.npu}@nwpu.edu.cn, 80090385@qq.com

Abstract. To solve the problem that multi-round and multi-slot Hello-Reply scheme takes a long time to discover all neighbors due to its multiple parameters and difficulty in optimization, this paper proposes an optimal multi-round and multi-slot (o-MRMS) Hello-Reply algorithm, which theoretically proves and reduces the total time and round number of the algorithm for neighbor discovery, effectively reducing discovery time, and combined with the existing discovery protocol. Simulation results show that compared with a fixed slot number Hello-Reply algorithm, the total neighbor discovery time in the proposed optimal multi-round and multi-slot Hello-Reply algorithm is reduced by about 50%. Compared with the DANDi protocol [3], discovery efficiency increased with the node number, when the node number is 256, the discovery time reduced by approximately 25%. It can be combined with the existing neighbor discovery protocol, which verifies its feasibility and efficiency.

Keywords: Neighbor discovery · Directional antenna · Dynamic slot number

1 Introduction

The directional wireless Ad Hoc network combines the directional antenna and the wireless Ad Hoc network. It has the characteristics of strong independence, high damage resistance and high spatial reuse, and can optimize the transmission performance of the network as a whole. It is widely used in military field, disaster relief and other occasions. Neighbor discovery is the premise of network communication, and the efficiency of neighbor discovery will directly affect the network performance. At present, more and more network researchers focus on wireless networks. Most of the relevant literature researches directly or indirectly talk about neighbor discovery, and many neighbor discovery protocols are constantly proposed. With the purpose of "shortening the total time of neighbor discovery", this paper focuses on the time slot optimization of neighbor discovery Hello-Reply phase in directional wireless Ad Hoc network.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2021
Published by Springer Nature Switzerland AG 2021. All Rights Reserved
Y.-B. Lin and D.-J. Deng (Eds.): SGIoT 2020, LNICST 354, pp. 468–486, 2021.
https://doi.org/10.1007/978-3-030-69514-9_36

According to the neighbor discovery protocols designed by scholars for wireless network discovery in recent years, it can be classified according to the following five conditions: (1) neighborhood discovery range; (2) antenna mode; (3) message reply mode; (4) clock state; (5) transmission/listening mode selection.

According to different neighbor scopes that nodes can discover, neighbor discovery protocols can be divided into direct and indirect neighbor discovery protocols. Ref. [4–6,10] all adopt direct neighbor discovery, that is, nodes can only discover their 1-hop neighbors through information exchange. A typical indirect neighbor discovery algorithm is the algorithm based on Gossip, and Ref. [16] is proposed on the basis of Gossip algorithm.

According to type of antenna in the network, neighbor discovery protocols can be divided into omnidirectional antenna, directional antenna and hybrid antenna protocols [15]. Hybrid antenna pattern is divided into directional-omnidirectional pattern and omnidirectional-directional pattern, the directional-omnidirectional pattern and omnidirectional pattern is two of the most common antenna combinations in wireless network, such as Ref. [4,7] adopt directional antenna patterns, Ref. [8,10,17] involves the hybrid antenna pattern.

According to the way nodes respond to Hello messages, neighbor discovery protocols can be divided into Handshake-based and periodic poll response protocols. At present, Handshake-based methods account for the majority. In Ref. [4,6,9,13], two-way handshake are adopted, while in Ref. [14,17], three-way handshake are adopted.

According to clock state of nodes, neighbor discovery protocols can be divided into synchronous and asynchronous neighbor discovery protocols. Ref. [6–8,14,17] are all synchronous neighbor discovery algorithms. Nodes in the network need time synchronization. In asynchronous algorithm, the clock or interval sequence of different nodes is out of sync, so it is unnecessary to start neighbor discovery process at the same time, as shown in Ref. [11–13].

According to how nodes choose transmission or listening mode, neighbor discovery protocols can be divided into stochastic and deterministic neighbor discovery protocols. Ref. [4,8,17] are all stochastic algorithms, and certain probability is introduced in node state selection. Ref. [5] is a deterministic algorithm based on Disco protocol. Nodes select their own state according to pre-set rules. Neighbor discovery protocols in most literature can be divided into multiple categories simultaneously.

In the existing neighbor discovery protocol, it is also different in setting slot number per round in Hello-Reply phase. In SAND [1] and Q-SAND protocol [2], slot number is fixed. The slot number and discovery round are set in advance, and do not change with neighbor nodes. DANDi protocol [3] proposed a dynamic slot number adjustment algorithm, adopting the idea of exponential retreat, the initial slot number is 1. When collision slot number in this round is 0, discovery is completed. When collision slot number in this round is not 0, slot number in the next round is twice that of the current round, the round number is also constantly changing. Compared with the fixed slot number, the utilization rate of slot using the exponential retreat method is higher. Finding the same node requires less time slot, but there is still room for efficiency improvement.

In this regard, this paper proposes an optimal multi-round and multi-slot Hello-Reply algorithm, theoretically proves and deduces the total time and round number of neighbor discovery algorithm, effectively reducing the total neighbor discovery time, and combines the multi-round and multi-slot Hello-Reply algorithm with existing protocols to verify its feasibility and discovery efficiency. Combined with the simulation results, we give a simple fitting expression for discovery round number and time.

The rest of article is arranged as follows. In Sect. 2, we briefly introduce the system model and put forward the core issues to be solved in this paper. Section 3 introduces the multi-round and multi-slot Hello-Reply algorithm in this paper. After that, Sect. 4 conducts simulation and result analysis, and Sect. 5 summarizes this paper (Table 1).

2 System Model and Problem Modeling

Table 1. Symbol notation and meanings

Symbol notation	Symbol meanings
N	Total slave node number
S	Total slot number
R	Round number required by discovering all nodes
r	Current neighbor discovery rounds, $1 \leq r \leq R$
n_r	Slave node number in round r
s_r	Slot number in round r
n	Node number in the single round neighbor discovery
s	Slot number in the single round neighbor discovery
T	The total time required to discover all the nodes
t_h	The duration of the hello slot
t_r	The duration of the reply slot
N_E	The idle slot number after one round neighbor discovery
N_S	The success slot number after one round neighbor discovery
N_C	The collision slot number after one round neighbor discovery

2.1 System Model

In the Hello-Reply process of neighbor discovery, the master node establishes the connection between the two by sending hello message to slave node and receiving slave node's reply message. In the process of each Hello-Reply round, a Hello slot and several Reply slots form a frame. The frame length is determined by the Reply slot number N. Total rounds R of Hello-Reply is also a variable, which depends on slave node number to be found and Reply slot number in each round. Slot number per round is usually a series of discrete integers, for example, S could be $S = 2^k$, $S_1 < S_2 < \cdots < S_k < \cdots < S_K$, $1 \leq k \leq K$.

After the start of Hello-Reply in round r, the master node first transmits a Hello message to slave node in hello slot, which carrys the discovered slave node number and reply slot number s_r in this round. When the slave node receives the Hello message, it will check whether it has been found. If not, the slave node will randomly select a slot to reply Reply message in s_r reply slot, otherwise it will remain silent in this round.

After the slave node replies the Reply message, a reply slot may have three states:

Success: only one node selects this time slot to reply;

Idle: no node selects this time slot to reply;

Collision: two or more nodes select this time slot to reply.

When collision slot number is 0, it means that all slave nodes have been discovered by the master node, and the multiple rounds of Hello-Reply process neighbor discovery ends.

2.2 Problem Modeling

Assuming that hello slot duration is t_h and reply slot duration is t_r, the total time used by the Hello-Reply mechanism after round R can be expressed as

$$T = \sum_{x=1}^{R} (t_r s_x + t_h) = t_r \sum_{x=1}^{R} s_x + R t_h. \tag{1}$$

where t_h is one hello slot time required in round x, t_r is one reply slot time, and s_r is reply slot number required in round x. Our question is how to dynamically adjust the reply slot number per round to minimize time T needed to discover all neighbors.

3 Optimal Strategy

Theorem 1. *When reply slot number s_r per round is equal to remaining undiscovered slave node number n_r, the total time T required to find all slave nodes is minimum. At this point, the theoretical derivation of time T can be expressed as:*

$$\begin{cases} \text{Known } n_1 \text{ and } s_1 = n_1 \text{ (total number of slave nodes to be discovered)} \\ s_{r+1} = n_{r+1} = n_r - \sum\limits_{x=0}^{n_r} x \cdot Q_x(n_r, s_r) \\ \text{if } n_{r+1} = 0 \text{ , then } R = r \\ T = \sum\limits_{x=1}^{R} (t_r s_x + t_h) = t_r \sum\limits_{x=1}^{R} s_x + R t_h \text{ .} \end{cases} \tag{2}$$

By fitting, time can be approximately expressed as

$$\begin{aligned} FittingTime &= round(2.702 \times N - 4.963) \times t_r \\ &\quad + round(0.045 \times N + 5.2728) \times t_h \text{ .} \end{aligned} \tag{3}$$

where, "round" means round off.

In the case that reply slot number s_r per round is equal to remaining undiscovered slave nodes number n_r, the theoretical derivation of round number R required to find all neighbors is as follows:

$$R = round\left(\frac{\lg(1/N)}{\lg(1-1/e)}\right). \tag{4}$$

Through data fitting, round number can be approximately expressed as

$$R = round(0.4581 \times (\lg N)^2 + 3.561 \times \lg N + 1.01)\ . \tag{5}$$

where, "round" means round off.

In single-round neighbor discovery problem, if slave node number to be found is known, the maximum slot utilization and relationship between slot number and node number can be obtained by changing reply slot number S, that is, when $S = N$, the maximum slot utilization can be obtained. By popularizing the node discovery in a single round, we can get the value of reply slot number s_r in each round in the multi-round neighbor discovery, which can maximize total utilization rate of slot and minimize node discovery time, thus, the round number needed to find all nodes and the required minimum time can be derived theoretically.

It can be proved that in the single-round neighbor discovery, when reply slot number S is equal to slave node number N to be found, utilization rate of slot is the highest, and the maximum slot utilization decreases from 1 to $\frac{1}{e}$ with the node number increasing, as shown in Lemma 1. Extended to multi-round neighbor discovery, it can be proved that when reply slot number s_r per round is equal to the remaining undiscovered slave nodes number n_r, total slot utilization rate is the highest and node discovery time is the minimum, as shown in Lemma 2, and the theoretical derivation expression of round number required for discovery of all nodes and the minimum time required can be obtained. For the convenience of representation, it can be simulated and fitted, as shown in Lemma 3 and Lemma 4.

3.1 Neighbor Discovery of Single Round

Lemma 1. *In the single-round neighbor discovery problem, when reply slot number S is equal to slave node number N to be found, slot utilization is the highest, and the maximum slot utilization gradually decreases from 1 to $\frac{1}{e}$ as the node number increases. When $n \to \infty$, slot utilization approaches $\frac{1}{e}$, and when $n \to 1$, slot utilization approaches 1.*

Given the slot number S and the node number N, the probability of any slot being selected by the node and successfully sending reply packet can be shown, that is, the utilization rate of slot. By changing slot number and analyzing slot utilization rate expression, the relationship between slot number S and node number N can be obtained when slot utilization is at its maximum.

We use n to represent the node number to be discovered and S to represent the slot number available for the Reply message in the Hello-Reply process. When the master node sends Hello packet, the slave node receiving Hello packet will choose any slot to reply. There are three states of slot: success, collision and idle, among which the successful slot is also called discovery slot, so the probability of any slot is discovery slot $P(n, s)$ is

$$P(n,s) = \binom{n}{1}\left(\frac{1}{s}\right)\left(1-\frac{1}{s}\right)^{n-1} . \tag{6}$$

$\frac{1}{s}$ means the possibility that a node randomly chooses one of s slots.

From $\frac{\partial P(n,s)}{\partial s} = 0$ and $\frac{\partial P(n,s)}{\partial s^2} < 0$, when $s = n$ and $n > 1$, $P(n, s)$ gets the maximum. Substitute $s = n$ into $P(n, s)$:

$$P(n,s)|_{s=n} = n \cdot \frac{1}{n} \cdot (1-\frac{1}{n})^{n-1} = (1-\frac{1}{n})^{n-1} . \tag{7}$$

When $n \to \infty$, let $\frac{1}{n} = -x$, get:

$$\lim_{n\to\infty} (1-\frac{1}{n})^{n} \times (1-\frac{1}{n})^{-1} = \lim_{x\to 0} (1+x)^{-\frac{1}{x}} = \frac{1}{e} . \tag{8}$$

When $n \to 1$,

$$\lim_{n\to 1} (1-\frac{1}{n})^{n-1} = \lim_{n\to 1} e^{(n-1)\ln(1-\frac{1}{n})} = e^{\lim\limits_{n\to 1}(n-1)\ln(1-\frac{1}{n})} = e^{0} = 1 . \tag{9}$$

That is, when slot number is equal to node number to be discovered, the probability of any slot to be discovery slot is the highest, and when n approaches ∞, the probability of any slot to be discovery slot is equal to $\frac{1}{e}$. As n approaches 1, the probability of any slot to be discovery slot gradually increases to 1.

Let $num(n, s)$ represent the maximum node number that is found in one round when node number to be discovered is n and slot number is s. $num(n, s)$ is expressed as

$$num(n,s) = \begin{cases} n & \text{if } n \le s \\ s-1 & \text{if } n > s . \end{cases} \tag{10}$$

In the round of Hello-Reply containing s slots, the probability of discovering one node among n nodes is

$$Q_1(n,s) = C_s^1 P(n,s) L(n-1, s-1) . \tag{11}$$

$L(n-1, s-1)$ means the probability that all other $(s-1)$ slots are not discovery slot, $P(n, s)$ means the probability that any slot is discovery slot, and C_s^1 is the way that this slot is selected. In one round of Hello-Reply, the probability of finding x nodes $(1 \le x \le num(n, s))$ is expressed as

$$Q_x(n,s) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } n \le s \;\&\; x = n-1 \\ C_s^x \left[\prod\limits_{k=0}^{x-1} P(n-k, s-k)\right] & \\ \times L(n-x, s-x) & Otherwise . \end{cases} \tag{12}$$

where, $L(n-x, s-x)$ represents the probability that no slot is discovery slot in the remaining $(s-x)$ slots, which can be expressed as:

$$L(n-x, s-x) = Q_0(n-x, s-x) = 1 - \sum_{j=1}^{num(n-x,s-x)} q_j(n-x, s-x) \ . \quad (13)$$

The expected node number to be found in an average round is:

$$E[n] = \sum_{x=0}^{n} x \cdot Q_x(n, s) \ . \quad (14)$$

The slot utilization of one round can also be expressed as

$$\eta_s = \frac{\sum_{x=0}^{n} x \cdot Q_x(n, s)}{s} \ . \quad (15)$$

The relationship between slot utilization and the probability that any slot is a discovery slot:

The probability that any one slot can successfully discover a node is $P(n, s)$, then discovery node number per round is $P(n, s) \times s$, and slot utilization rate is $\frac{P(n,s)\times s}{s} = P(n, s)$. Therefore, the probability that any one slot is a discovery slot is the slot utilization rate. In the same way, when the slot number is equal to node number to be discovered, slot utilization rate is the largest, and when n is large enough, the probability that any one slot is a discovery slot is equal to $\frac{1}{e}$.

3.2 Neighbor Discovery Efficiency of Multiple Rounds

Lemma 2. *In the multi-round neighbor discovery, when reply slot number s_r per round is equal to the remaining undiscovered slave node number n_r, the total utilization rate of slot is the highest and node discovery time is the shortest.*

Neighbor discovery often cannot end in one round. In the case of multiple rounds, r represents the total round number required, n_r represents the node number to be discovered corresponding to round r, and s_r represents the slot number required for round r. The node number that can be found in round r can be expressed as

$$E[n_r] = \sum_{x=0}^{n_r} x \cdot Q_x(n_r, s_r) \ . \quad (16)$$

Then the slot utilization rate in round r is

$$\eta_{sr} = \frac{\sum_{x=0}^{n_r} x \cdot Q_x(n_r, s_r)}{s_r} \ . \quad (17)$$

It can be known from the derivation in Sect. 3.1 that the probability that any one slot is a discovery slot is the slot utilization rate, so the slot utilization rate in round r can also be expressed as

$$P(n_r, s_r) = n_r \times \frac{1}{s_r}\left(1 - \frac{1}{s_r}\right)^{n_r - 1} . \tag{18}$$

Then the slot utilization of multiple rounds of neighbor discovery is

$$\eta = \frac{\sum_{r=1}^{R} s_r \times P(n_r, s_r)}{\sum_{r=1}^{R} s_r} . \tag{19}$$

where $\sum_{r=1}^{R} s_r \times P(n_r, s_r)$ represents the total number of nodes, denoted by N, then the above formula can be written as

$$\eta = \frac{N}{\sum_{r=1}^{R} s_r} . \tag{20}$$

where, N is the total node number, and it is a fixed value. From single-round neighbor discovery process, when $s_r = n_r$, the slot utilization $P(n_r, s_r)$ is the largest, the discovery of same node number requires fewer slots. For multiple rounds Neighbor discovery, for the same reason, when the total node number to be discovered N is equal, when each round $s_r = n_r$, the total slot number required to discover all nodes is the least, the slot utilization is the highest, and the required discovery time is the shortest.

It is known from Sect. 3.1 that when $s_r = n_r$, the slot utilization rate of each round is the largest, and the maximum slot utilization maximum value is 1, the minimum value is $\frac{1}{e}$. As the node number increases, the slot utilization rate gradually decreases and approaching $\frac{1}{e}$. That is, in the slot utilization rate found by multiple rounds of neighbors, the slot utilization rate $P(n_r, s_r)$ of each round satisfies

$$\frac{1}{e} \le P(n_r, s_r) \le 1 . \tag{21}$$

Assuming the total slave node number is ∞, that is $n_1 = \infty$, then

$$\eta = \frac{\frac{1}{e}\sum_{r=1}^{R} s_r k_r}{\sum_{r=1}^{R} s_r} . \tag{22}$$

$$e \ge k_r = \frac{P(n_r, s_r)}{1/e} \ge 1 (r = 1, 2, \cdots, R), 1 = k_1 < k_2 < \cdots < k_r = e . \tag{23}$$

Therefore, the efficiency of neighbor discovery in multiple rounds

$$\frac{1}{e} < \eta < 1 \ . \tag{24}$$

In actual applications, the total slave node number tends not to tend to ∞. In this case, the actual multi-round node discovery efficiency will be larger.

3.3 Neighbor Discovery Round of Multiple Rounds

Lemma 3. *When the reply slot number s_r per round is equal to remaining undiscovered slave node number n_r, the theoretical derivation of the round number required to find all nodes can be expressed as $R = round\left(\frac{\lg(1/N)}{\lg(1-1/e)}\right)$ and can be fitted as $R = round\,(0.4581 \times (\lg N)^2 + 3.561 \times \lg N + 1.01)$.*

Let the total node number to be discovered be N, and slot number s_r and node number to be discovered n_r of each round remain the same, that is $n_r = s_r$. As can be seen from Sect. 3.1, when n_r of each round is large enough, the slot utilization rate of each round is $\frac{1}{e}$, then the node number that is found in the *rth* round is s_r/e, which can also be expressed as n_r/e, and the remaining node number to be found is $n_r(1-1/e)$. After R rounds, the remaining nodes number to be found is $N(1-1/e)^R$. If all nodes are discovered after neighbor discovery ends, it should satisfy

$$N(1-1/e)^R < 1 \ . \tag{25}$$

that is

$$R > \frac{\lg(1/N)}{\lg(1-1/e)} \ . \tag{26}$$

Since round number R is an integer, the above formula can be rounded to obtain

$$R = round\left(\frac{\lg(1/N)}{\lg(1-1/e)}\right) = round(5.02 \times \lg N) \ . \tag{27}$$

where, "round" means round off.

In order to facilitate the representation and application of neighbor discovery time and round number, the simulation data of discovery time and round number obtained by simulation is fitted.

As the round number grows slowly with nodes number, it is generally in a step shape. In order to facilitate fitting, Node number N is the logarithm to base 10, total round number is represented by R, and the linear fitting expression of $\lg N$ and round number R is

$$R = 4.771 \times \lg N + 0.3413 \ . \tag{28}$$

The quadratic fitting expression is

$$R = 0.4581 \times (\lg N)^2 + 3.561 \times \lg N + 1.01 \ . \tag{29}$$

3.4 Neighbor Discovery Time of Multiple Rounds

Lemma 4. *When reply slot number in each round is equal to node number to be found in this round* n_r*, hello slot duration is* t_h *and reply slot duration is* t_r*, then the minimum time required by discovering all nodes can be fitted as*

$$\begin{aligned} FittingTime = & \, round(2.702 \times N - 4.963) \times t_r \\ & + round(0.045 \times N + 5.2728) \times t_h \ . \end{aligned} \tag{30}$$

When the slot number s_r in each round changes with the remaining node number n_r, and $n_r = s_r$ is maintained, the node number to be found in the next round and slot number required are

$$s_{r+1} = n_{r+1} = n_r - \sum_{x=0}^{n_r} x \cdot Q_x(n_r, s_r) \ . \tag{31}$$

n_r and s_r are the node number to be discovered and the required slot number corresponding to the rth round.

When $n_{r+1} = 0$, it indicates that node number to be found in the next round is 0, that is, all nodes have been discovered, and r at this time is the round number R required to find all nodes.

Suppose the duration of the hello slot is t_h, the duration of the reply slot is t_r, the time required to pass the R round is

$$T = \sum_{x=1}^{R} (t_r s_x + t_h) \ . \tag{32}$$

The pseudo-code is used to represent the above discovery time calculation process. The algorithm is as follows.

First of all, we set the time slot length of both sending request packet and receiving reply as 1. After simulation, the time and round number needed for node discovery of different numbers are obtained, and node discovery time is fitted with known data. We can get linear fitting time at this point is

$$T_1 = 2.747 \times N + 0.3098 \ . \tag{33}$$

N is total node number.

The fitting time T_1 at this point includes slot used by Hello package and slot used by Reply package. Since each round the primary node sends a Hello packet, neighbor discovery round number is the slot number used by the Hello packet.

Subtract round number from the total discovery time to get the time used without Hello package, and then conduct the fitting again. At this point, the fitting discovery time is

$$T_2 = 2.702 \times N - 4.963 \ . \tag{34}$$

The total fitting discovery time expression is

$$FittingTime = T_2 \times t_r + (T_1 - T_2) \times t_h \ . \tag{35}$$

Algorithm 1. DiscoveyTime (Notes, Slots) : For a given n and S, when slot number per round is equal to node number to be discovered, the round number and time required by discovering all nodes and the slot number per round.

Input: $N_{initial}$, t_r, t_h
Output: SumTime, Slots[], Round

begin procedure
Notes[1] = $N_{initial}$
/*The array format stores the number of slots required per round*/
Slots[1] = Notes[1]
x = 1
/*Round number in a while loop*/
while ($Notes[x] \neq 0$) **do**
    /*Node number to be found in the next round = node number to be found in the current round - node number already found in the current round*/
    /*round off*/
    Notes[x+1] = Notes[x] - round($\sum_{j=0}^{Notes[x]} k \times q_j(Notes[x], Slots[x])$)
    Slots[x+1] = Notes[x+1]
    x = x + 1
end while
Round = x - 1 /*Get round number needed*/
SumTime = 0
for r = 1:Round **do**
    /*Calculate the total time*/
    SumTime = SumTime + (Slots(r) * t_r + t_h)
end for
end procedure

By substituting the specific expressions of T_1 and T_2, it can be obtained that:

$$\begin{aligned} FittingTime = & round(2.702 \times N - 4.963) \times t_r \\ & + round(0.045 \times N + 5.2728) \times t_h \ . \end{aligned} \tag{36}$$

where: t_r is the length of a reply slot and t_h is the length of a hello slot.

3.5 Application of Multi-round and Multi-slot Hello-Reply Algorithm

In order to verify the application of Hello-Reply algorithm in this paper, this section applies the multi-round and multi-slot Hello-Reply algorithm on the basis of Q-SAND protocol [2]. In Hello-Reply phase of the Q-SAND protocol, a fixed pattern with 5 rounds of neighbor discovery and 5 reply slots per round is used. The DANDi protocol [3] adopts the idea of exponential retreat and the initial number of slots is 1. When collision slot number in this round is 0, node discovery is completed; when collision slot number in this round is not 0, the slot number in the next round is twice that of the current round, and the node discovery round is not set. When using the dynamic slot number selection algorithm in this article, the reply slot and node discovery round numbers are set as follows: The first round of node discovery reply slots in the Hello-Reply phase is set to 5, which is consistent with Q-SAND, in each subsequent round of node discovery, the slot number is determined by the previous round of node discovery, that is, the total node number is estimated by the successful number, idle, and collision slots for node discovery, thereby determining the next round of response number of slots. Another way to set slot number is to set slot number in the first Hello-Reply round to 1, consistent with DANDi, and the slot number per subsequent round is decided by the exact total node number minus the discovered node number. The node discovery round number is not set in advance. When all nodes are discovered, the node discovery process ends.

The node estimation method adopts Chen estimation method [18]. That is, in a round of node discovery with the slot number L, after the node discovery is completed, there are N_E idle slots and N_S successful slots, and the probability of N_C collision slots is:

$$P(N_E, N_S, N_C) = \frac{L!}{N_E!N_S!N_C!} p_e^E p_s^S p_c^C \ . \tag{37}$$

Among them, p_e, p_s, p_c respectively represent the idle, successful and collision slot probability when a node is discovered, and its expression is:

$$p_e = (1 - (1/L))^n \ . \tag{38}$$

$$p_s = (n/L)(1 - (1/L))^{n-1} \ . \tag{39}$$

$$p_c = 1 - p_s - p_e \ . \tag{40}$$

For a node discovery round with a slot number of L, the posterior distribution probability can be obtained. Under the environment of n nodes to be discovered, when there are N_E idle slots and N_S successful slots, collision slot number N_C can be obtained, and its expression is as follows:

$$\begin{aligned} P(n|N_E, N_S, N_C) = {} & \frac{L!}{N_E!N_S!N_C!} \times \left[\left(1 - \frac{1}{L}\right)^n\right]^{N_E} \\ & \times \left[\frac{n}{L}\left(1 - \frac{1}{L}\right)^{n-1}\right]^{N_S} \times \left[1 - \left(1 - \frac{1}{L}\right)^n - \frac{n}{L}\left(1 - \frac{1}{L}\right)^{n-1}\right]^{N_C} . \end{aligned} \tag{41}$$

According to the posterior distribution, when the above formula reaches the maximum, the estimated value of the node is the best.

The pseudo codes for node estimation and slot number selection are as follows.

Algorithm 2. Node estimation and slot number selection.

```
Round = 1, Slots[1] = 5, After a round, get N_E, N_S and N_C
if N_C = 0 then
    Node discovery complete.
else
    Calculate the maximum value of n for P(n|N_E, N_S, N_C).
    Round = Round + 1
    /*Remaining node number to be discovered, and rounded up*/
    Slots[Round] = [n - N_S]
end if
```

4 Simulation and Analysis

In the Sect. 3, in order to facilitate the representation and application of the discovery round number and discovery time of multiple round nodes, it is fitted according to the simulation data. In this section, compare the dynamic slot selection method in this paper with the fixed slot number per round, that is, the slot number always takes the initial slave node number to be found. For example, when the total node number is 100, slot number in each round is 100, and specific round number is determined by the neighbor discovery situation. Finally, the Hello-Reply proposed in this paper is applied to the Q-SAND protocol to verify its neighbor discovery efficiency. This paper uses C++ simulation platform.

4.1 Neighbor Discovery Rounds Simulation

Firstly, fitting round number at different node counts is compared with actual rounds obtained by simulation to verify rationality of the fitting formula. The results are shown in Fig. 1.

As shown in Fig. 1, in the case that the slots number s_r is fixed, except that the number of slots s_r in the first round is equivalent to the number of nodes n_r, the rest of the slot each round number compared with the node number is larger, so while the slot utilization rate each round is low, but the node number can be found in each round is more than $s_r = n_r$, so neighbor discovery need less number of rounds, but the overall trend is consistent with the $s_r = n_r$, with the increase of the nodes number and ladder shaped growth. In the case of $s_r = n_r$, when the node number N is small, the rounds number fitting value and simulation value is appropriate, when N is large, there will be an error value of 1 in the round number of some nodes, and compared with those of linear quadratic has higher accuracy, namely when the node number is equal to the slot number per round, the node number and round relations can be expressed as follows:

$$R = 0.4581 \times (\lg N)^2 + 3.561 \times \lg N + 1.01 \ . \tag{42}$$

4.2 Neighbor Discovery Time Simulation

Let $t_r = 1$ and $t_h = 1$. Compare the fitting slots and simulation slots to verify the rationality of the fitting formula. The results are shown in Fig. 2.

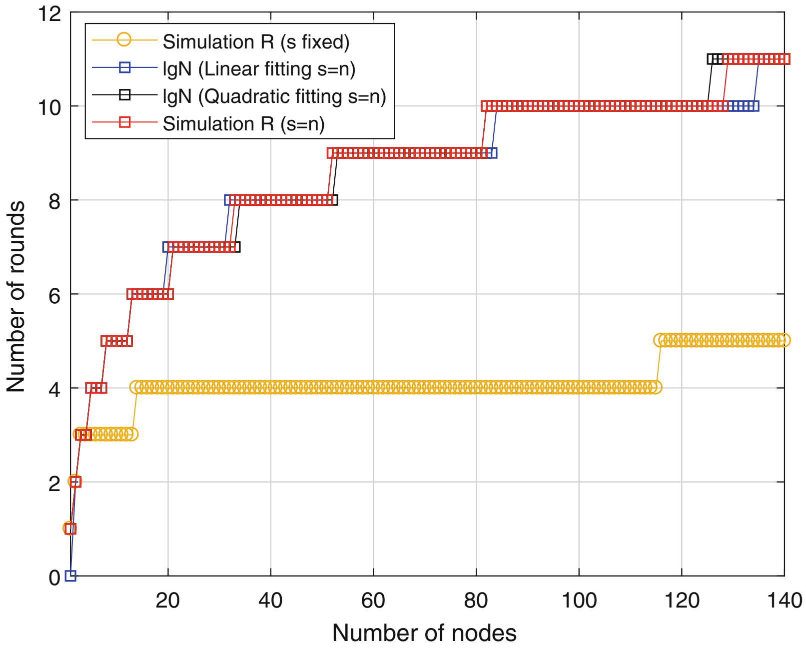

Fig. 1. Number of fitting rounds and simulation rounds at different nodes number

As shown in Fig. 2 that when slot number per round is equal to node number to be found per round, the node discovery time is far less than the number of fixed slot, and the performance is improved by about 50%. When the slot number is fixed, when the node number increases to a certain value, the discovery round number will increase by one round, causing the node discovery time to jump, as shown in the figure with the node number 110 and 120. When $s_r = n_r$ in each round, when $N = 20$, error rate of fitting time is the largest, the maximum error rate is 0.0587. According to the theoretical derivation in Sect. 4.2, slot utilization ratio is equal to $\frac{1}{e}$, which is obtained under the premise of $n \to \infty$. When N is small, there will be a certain error, and when N is small, node discovery time is small, and a small error will also cause a large error rate. Therefore, when N is small, error rate of fitting and simulation results is high. When N is greater than 90, error rate is basically stable below 0.01, that is, the fitting time formula has high accuracy.

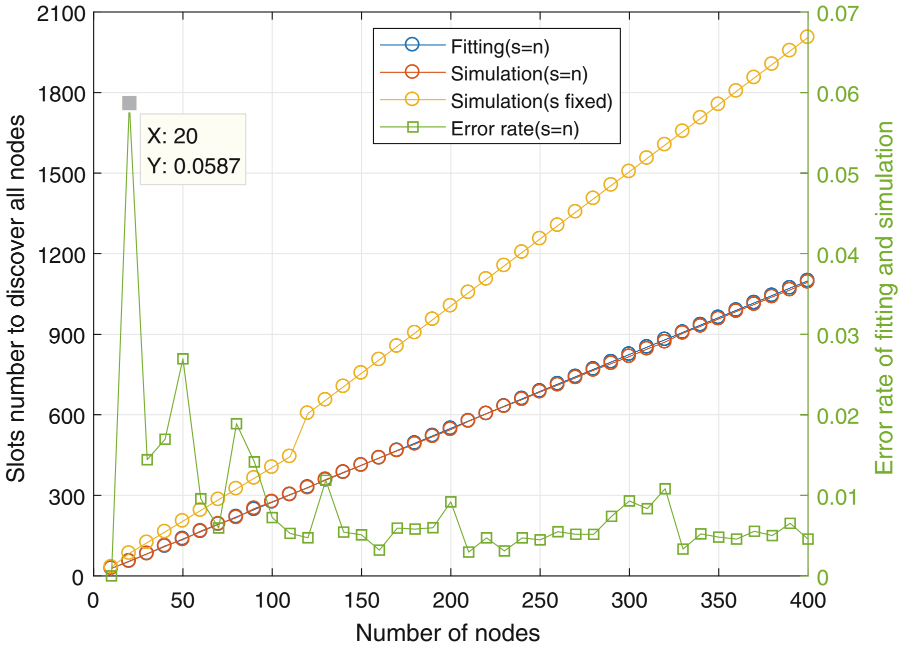

Fig. 2. Comparison of fitting time and simulation time and error rate for different number of nodes ($t_h = 1$, $t_r = 1$)

4.3 Application of Multi-round Multi-slot Hello-Reply Algorithm in Q-SAND

In this section, the application of the proposed multi-round and multi-slot Hellow-Reply algorithm in Q-SAND is simulated and verified, and compared with the Q-SAND protocol and DANDi protocol. The simulation parameters and network scenario settings in the protocol are shown in Table 2 and Table 3 respectively.

The node position is randomly set in a certain region. When neighbor discovery is conducted, the randomly generated nodes are required to form a connected network topology. If there are random isolated nodes, or if there is more than one network topology, the resulting network topology is discarded and does not count in the statistical results.

Under the same simulation parameters and network scenes, the node number is changed to simulate the neighbor discovery time in network.

Figure 3 shows the simulation discovery time with fixed slot number, DANDi, and variable slot number under Q-SAND protocol for different node number. To ensure the universality of the results, we randomly generate 20 kinds of network topologies for the same node number, and average the simulation results.

Table 2. Protocol simulation parameters

The stage of the node	Parameter	Value
FastScan mode	t_{switch}	3 ms
Hone-In mode	t_{HoneIn}	1.5 ms
	h	12
Hello-Reply mode(Q-SAND)	N_{slots}	5
	N_{rounds}	5
	t_{slots}	1.5 ms
Token Passing mode	$t_{GoToFastScan}$	1.5 ms

Table 3. Network scene parameter setting

Network scene setting	Value
Node distribution region	500 * 500
Maximum single hop communication distance	100
Number of sectors per node	6
Network simulation times with same nodes number	20

As can be seen from Fig. 3, compared with the original Q-SAND protocol with fixed slot, the node discovery time of Q-SAND protocol using the slot number variable algorithm proposed in this paper is reduced by about 50%, which is consistent with the neighbor discovery time in Hello-Reply phase in Sect. 4.2. Node discovery efficiency of the network is greatly improved. Meanwhile, when the node number is accurate, compared with DANDi, when the total node number in network topology is small, the number of neighbor node will also be small, the discovery time is both similar. As the neighbor node number increased, advantage of time-slot number variable algorithm presented in this paper gradually emerged, and discovery time is lower than DANDi protocol, and the gap increases gradually. When the node number in topology is 256, compared with DANDi protocol, the neighbor discovery time of slot number variable algorithm in this paper is reduced by approximately 25%. When node estimation is introduced, due to the large impact of estimation error when the node number is relatively small, the o-MRMS neighbor discovery time is slightly larger than DANDi, and with the node number increasing, the o-MRMS node discovery time is gradually smaller than DANDi protocol.

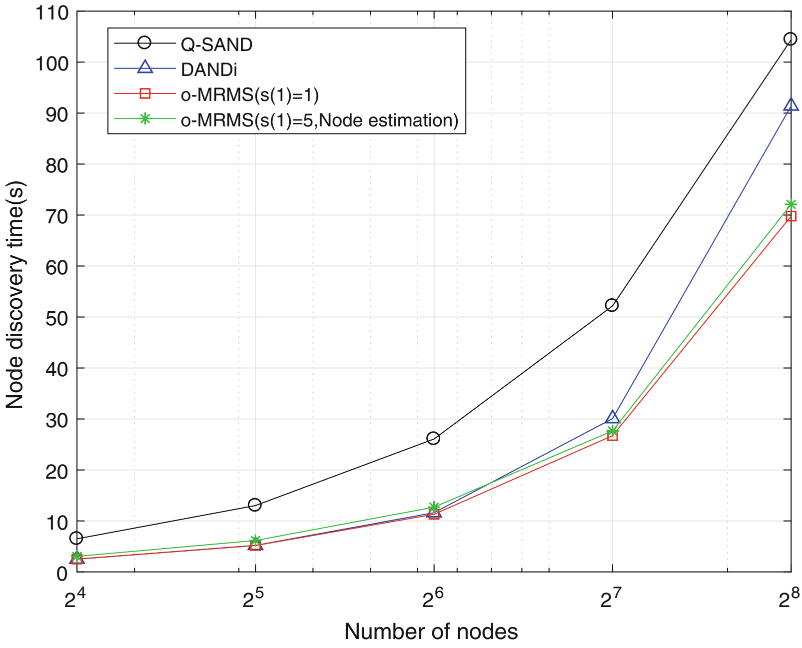

Fig. 3. Comparison of discovery time of various time slot selection methods under Q-SAND protocol

5 Summary and Future Outlook

In this article, we have conducted study on the neighbor discovery. Theoretical deduction proves that when the reply slots number per round is equal to the node number to be discovered, the slot utilization is the highest and the node discovery time is the shortest. Through simulation, the fitting expression of node discovery round and node discovery time is given, and the fitting expression is compared with the simulation data to verify the accuracy of the fitting expression. At the same time, we combine the proposed o-MRMS algorithm with existing protocols to verify its feasibility. Compared with Q-SAND protocol, node discovery efficiency is improved by about 50%. Compared with the DANDi protocol, which node discovery to be more efficient, when the network node number is greater than 250, efficiency can be increased by more than 25%, and the greater the node number, the greater the efficiency increase.

In wireless self-organizing network neighbor discovery, in order to improve the efficiency of discovery, often need to dynamically change the Reply time slot number in Hello-Reply phase, but there are so many ways to change, so how to choose is a question. Through this paper, reader can get an optimal selection method for slot number. Through the node number in the network, we can calculate the reply slot number each round and obtain the optimal way for dynamic change of slot, so as to improve the efficiency of system. Under the condition of known node number, this paper gives the fitting formula of discovery round number and total discovery time, can do not need to simulation, only for simple operation, convenient to calculate round and time for discovering all nodes roughly, so as to serve as a reference.

For the node discovery time fitting formula, there will be some error when the node number is small. Whether the accurate expression of neighbor discovery time can be obtained is an issue we need to consider in the future.

Acknowledgement. This work was supported in part by Science and Technology on Avionics Integration Laboratory and the Aeronautical Science Foundation of China (Grant No.201955053002), the National Natural Science Foundations of CHINA (Grant No.61871322, No.61771392, No.61771390, and No.61501373), and Science and Technology on Avionics Integration Laboratory and the Aeronautical Science Foundation of China (Grant No.20185553035).

References

1. Felemban, E., Murawski, R., Ekici, E., et al.: SAND: sectored-antenna neighbor discovery protocol for wireless networks. In: 2010 7th Annual IEEE Communications Society Conference on Sensor. Mesh and Ad Hoc Communications and Networks (SECON), Boston MA, pp. 1–9 (2010)
2. Gammarano, N., Schandy, J., Steinfeld, L.: Q-SAND: a quick neighbor discovery protocol for wireless networks with sectored antennas. In: 2018 Ninth Argentine Symposium and Conference on Embedded Systems (CASE), Cordoba, pp. 19–24 (2018)
3. Gammarano, N., Schandy, J., Steinfeld, L.: DANDi: dynamic asynchronous neighbor discovery protocol for directional antennas. In: VIII Brazilian Symposium on Computing Systems Engineering, pp. 16–23 (2018)
4. Ji, D., Wei, Z., Chen, X., et al.: Radar-communication integrated neighbor discovery for wireless ad hoc networks. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, pp. 1–5 (2019)
5. Gao, M., Shen, R., Mu, L., et al.: An anti-collision neighbor discovery protocol for multi-node discovery. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, pp. 1–5 (2019)
6. Yang, A., Li, B., Yan, Z., et al.: A bi-directional carrier sense collision avoidance neighbor discovery algorithm in directional wireless ad hoc sensor networks. Sensors **19**(9), 2120–2138 (2019)
7. Liu, B., Rong, B., Hu, R., et al.: Neighbor discovery algorithms in directional antenna based synchronous and asynchronous wireless ad hoc networks. IEEE Wirel. Commun. **20**(6), 106–112 (2013)
8. Mir, Z., Jung, W., Ko, Y.: Continuous neighbor discovery protocol in wireless ad hoc networks with sectored-antennas. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangiu, pp. 54–61 (2015)
9. Cai, H., Liu, B., Gui, L., et al.: Neighbor discovery algorithms in wireless networks using directional antennas. In: 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, pp. 767–772 (2012)
10. Mir, Z., Ko, Y.: Self-adaptive neighbor discovery in wireless sensor networks with sectored-antennas. Comput. Stand. Interfaces **70**, 1–16 (2020)

11. Murawski, R., Felemban, E., Ekici, E., et al.: Neighbor discovery in wireless networks with sectored antennas. Ad Hoc Netw. **10**(1), 1–18 (2012)
12. Wang, Y., Liu, B., Gui, L.: Adaptive scan-based asynchronous neighbor discovery in wireless networks using directional antennas. In: WCSP 2013: Proceedings of the International Conference on Wireless Communications and Signal Processing, Piscataway, N.J., pp. 1–6. IEEE (2013)
13. Tian, F., Hu, R., Qian, Y, et al.: Pure asynchronous neighbor discovery algorithms in ad hoc networks using directional antennas. In: 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, pp. 498–503 (2014)
14. Li, J., Peng, L., Ye, Y., et al.: A neighbor discovery algorithm in network of radar and communication integrated system. In: 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, pp. 1142–1149 (2014)
15. Dai, H., Ng, K., Li, M., et al.: An overview of using directional antennas in wireless networks. Int. J. Commun. Syst. **26**(4), 413–448 (2013)
16. Astudillo, G., Kadoch, M.: Neighbor discovery and routing schemes for mobile ad-hoc networks with beam width adaptive smart antennas. Telecommun. Syst. Model. Analy. Des. Manag. **66**(1), 17–27 (2017)
17. Zhang, Z., Li, B.: Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: algorithms and comparisons. IEEE Trans. Wirel. Commun. **7**(5), 1540–1549 (2008)
18. Chen, W.T.: An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA. IEEE Trans. Automat. ENCE Eng. **6**(1), 9–15 (2009)