PHED: Pre-Handshaking Neighbor Discovery Protocols in Full Duplex Wireless Ad Hoc Networks

Guobao Sun, Fan Wu, Xiaofeng Gao, and Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering
Shanghai Jiao Tong University, China
sunguobao@sjtu.edu.cn, {fwu, gao-xf, gchen}@cs.sjtu.edu.cn

***Abstract*—Neighbor Discovery (ND) is a basic and crucial step for initializing wireless ad hoc networks. A fast, precise, and energy-efficient ND protocol has significant importance to subsequent operations in wireless networks. However, many existing protocols have high probabilities to generate idle slots in their neighbor discovering processes, which extends the executing duration, and thus compromises their performance. In this paper, we propose a novel randomized protocol PHED, *Pre-Handshaking Neighbor Discovery Protocol*, to initialize synchronous full duplex wireless ad hoc networks. By introducing a pre-communication strategy to help each node be aware of activities of its neighborhood, we significantly reduce the probabilities of generating idle slots and collisions. Moreover, with the development of single channel full duplex communication technology [1, 2], we further decrease the processing time needed in PHED, and construct the first full duplex neighbor discovery protocol. Our theoretical analysis proves that PHED can increase the speed of ND by approximately 98% in comparison with the classical ALOHA-like protocols [3, 4]. In addition, we provide several numerical experiments exhibiting the effectiveness of PHED.**

***Index Terms*—Wireless Ad Hoc Networks, Neighbor Discovery, Full Duplex Technology, Randomized Algorithm**

I. INTRODUCTION

Wireless ad hoc networks have attracted a lot of interests from both academia and industry due to their wide range of applications. In many scenarios, nodes are deployed without the support of pre-existing infrastructures for communication. As a result, in a wireless ad hoc network nodes need to configure themselves through communication to form a reliable infrastructure as the initialization for further operations. For each node, the knowledge of its one-hop neighbors (the nodes it can directly communicate with) has significant importance to the upper layer protocols like MAC protocols, routing protocols, etc. Consequently, *Neighbor Discovery* (ND) is designed to discover a node's one-hop neighbors and thus is momentous and crucial for configuring wireless networks.

Compared with deterministic [9] and multi-user detection-based [10] algorithms, randomized algorithms are most commonly used to conduct neighbor discovery in wireless networks [3–8]. In those algorithms, each node transmits at different randomly chosen time instants to reduce the possibility of the collision with other nodes. Usually, researchers discuss ND protocols under synchronous systems, and focus on a clique with n nodes, e.g., the famous Birthday Protocols [3]. In birthday protocols, at each single slot every node independently chooses to transmit discovery message by probability p and listen by probability $1-p$ (the optimal value of p is proved to be $1/n$). By reducing the ND problem to *Coupon Collector's Problem* [12], Vasudevan *et al.* [4] proved that the upper bound of expected time of birthday protocol is neH_n, where H_n is the n-th Harmonic number. Many subsequent researches on ND are based on birthday protocols. For example, the authors in [4] proposed solutions to scenarios for unknown neighbor numbers, asynchronous systems, and systems with reception status feedback mechanisms. Zeng *et al.* [5] discussed the performance of birthday protocols with multipacket reception (MPR). You *et al.* [8] discussed discovery time's upper bound when nodes have a low duty-cycle.

However, the family of birthday protocols has a vital drawback. The probability of an idle slot is $p_0 = (1-\frac{1}{n})^n$. When $n = 10$, $p_0 \approx 0.349$. When $n \to +\infty$, $p_0 \to 1/e \approx 0.368$. Therefore when the number of nodes is large, the probability that no node transmits in a slot is about 37%. Furthermore, the probability of collisions also increases the iterations running in the protocols. For instance, two nodes transmitting simultaneously in a slot has a probability $1/(2e) \approx 0.184$, and three nodes transmitting simultaneously has a probability $1/(6e) \approx 0.06$.[1] If we can effectively reduce the probabilities of collisions and idle slots, the performance will be tremendously ameliorated. Fortunately, with the development of full duplex wireless communication technology [1, 2], we can design better protocols if nodes can transmit and listen simultaneously in a single slot.

Our key idea is twofold. On one hand, we introduce a pre-handshaking strategy to help each node be aware of activities of its neighborhood before normal transmissions, such that the system can have higher probabilities to avoid collisions and idle slots. To achieve this pre-communication, we add some tiny sub-slots before each normal slot. With the help of full duplex technology, at each sub-slot every node will decide whether to transmit the message in a normal slot by broadcasting an anonymous election signal and catch its neighbors' signals simultaneously. With different sending-receiving scenarios, we design an effective strategy for each node to

[1] Lemma 1 in Section III proves that the probability of n nodes transmitting in a slot is $1/(n!e)$.

determine how to behave in normal slots. Correspondingly, we design the behaviors of each node in the normal slots to complete the ND process. On the other hand, a new reception status feedback mechanism is designed by using full duplex wireless radios. Originally in [6], a sub-slot is added after the normal slot and listeners will give feedback signals to transmitters in this sub-slot. In our design this overhead can be eliminated by using full duplex nodes. If a listener finds that two or more nodes are transmitting simultaneously, it will transmit a warning message immediately to inform other transmitters the failure of this transmission.

Our contributions in this paper are listed as follows:

- We design a novel ND protocol named PHED which stands for ***P**re-**H**andshaking N**E**ighbor **D**iscovery*, in which pre-handshaking activities are inserted before normal communication. In PHED we avoid the vital drawback of the traditional birthday protocols and reduce the probabilities of collisions and idle slots. Other existing protocols based on birthday protocols can be ameliorated easily with our design, such as the ones proposed in [5, 8].
- To the best of our knowledge, we are the first to consider the issue of ND with full duplex technology. For such a long time, researches of the ND problem in wireless networks are based on half duplex nodes. The import of full duplex technology enables nodes to transmit and receive simultaneously, which can be utilized to accelerate the ND process. Along with the emergence of full duplex technology, we can optimistically predict the transition from half duplex nodes to full duplex nodes, which implicates the significance of our design.

The rest of this paper is organized as follows. Section II describes our model and assumptions. Section III introduces PHED and its theoretical analysis. In Section IV we evaluate PHED by simulation. In Section V we present related works. The paper concludes with our future works in Section VI.

II. Network Model and Assumptions

In this section, we introduce the network model and several assumptions, under which we will present our PHED protocol and corresponding analysis. These assumptions are reasonable in the research of the ND and many former works are also based on the similar assumptions [3–5, 8]. Our assumptions are listed as follows:

- Each node has a unique ID (e.g., the MAC address).
- Time is identically slotted and nodes are synchronized on slot boundaries.
- All nodes are in a clique of size n and n is known to all nodes in the clique.
- Nodes use omnidirectional antennas, and all nodes have the same transmission range.
- No MPR technique is used, i.e., a collision occurs when two or more nodes simultaneously transmit in a slot.
- Nodes can listen and transmit on the same channel simultaneously.
- Nodes can distinguish between collisions and idle slots.

We also neglect possible errors caused by fading. So for two nodes A and B, if A transmits without collisions in a slot and B is within the transmission range of A, then B can receive the packet without any error.

III. PHED: Pre-Handshaking Protocol

In this section we present our novel protocol PHED based on the assumptions in Section II, and analyze its performance theoretically. Firstly in Subsection III-A we add one tiny sub-slot before each normal slot and complete our design for the pre-handshaking process. Next in Subsection III-B we extend our idea for the pre-handshaking process by introducing more sub-slots before the normal slot and design the corresponding variation of PHED. Additionally in Subsection III-C we discuss in detail how many sub-slots should be used for the pre-handshaking process to achieve the best performance. Moreover, in Subsection III-D we extend our discussion to the situation when n is unknown to nodes. Finally, in Subsection III-E we give the extension of PHED for multi-hop networks.

A. PHED with Single Sub-Slot for Pre-Handshaking

As mentioned in Section I, for each normal slot we insert a sub-slot before it to perform the pre-handshaking process. We name this combination as an *iteration*. (It can also be considered as a “big slot”.) Let *GR* be the greeting process and *TR* be the transmission process in one iteration. Note that the length of a sub-slot can be as short as 1 bit since we do not care what a node transmits and only need to know whether the signals exist or not. The authors in [4] also adopted this assumption. Let M_s be such kind of messages, which means an anonymous election signal with short duration. The normal slot is used to exchange discovery messages which may contain nodes' IDs or MAC addresses. The size of a sub-slot is significantly smaller than that of a normal slot ([14] mentioned that the size of a slot can be about 10 Bytes.) and thus the overhead caused by sub-slots is almost negligible. We define this kind of discovery messages as M_d.

Fig. 1 illustrates the combination of sub-slots and normal slots. In Fig. 1 (a), we insert one sub-slot for one normal slot while in Fig. 1 (b) we insert multiple sub-slots before one normal slot to further increase the probability of successful transmissions and we will mention it in Subsection III-B.

We are now ready to present our PHED protocol to determine the action of a node in a slot. PHED is a distributed protocol and for each node the target is to discover all its

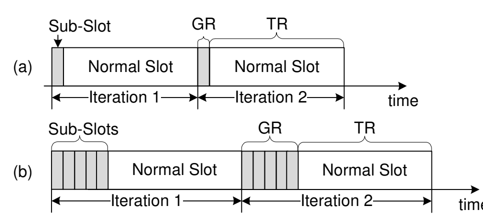

Fig. 1: The description of an iteration.

neighbors after finite iterations. Assume that we are considering a clique of n nodes. We divide PHED into two subroutines: PHED-GR and PHED-TR.

Let us describe the main idea of PHED-GR: the pre-handshaking process. At the beginning of a sub-slot, each node should determine its action in the following normal slot. The purpose is to find a subset of nodes in the network to send M_d without collisions. Alg. 1 describes the detail of PHED-GR. Note that each node should run a copy of PHED-GR. To simplify our description, assume that we run PHED-GR on node A. Recall that M_s is the election signal and M_d is the discovery message. Define A_f as a flag variable to indicate whether A has successfully sent M_d. If $A_f = 0$ then A has to send M_d successfully in one of the following iterations, else A will keep silent and only receive messages. Initially $A_f = 0$. Define A_n as the number of undiscovered neighbors of A. Initially A_n should be $n-1$ and we let $A_n = n$ for the simplicity of later discussion.

Algorithm 1 PHED-GR (Pre-Handshaking)

1: **if** $A_f = 1$ **then** ▷ A has successfully sent M_d.
2: A will keep silent in TR and exit.
3: **end if**
4: Node A decides to send M_s by probability $1/A_n$ and keep listening by probability $1 - 1/A_n$.
5: **if** A sends M_s **then** ▷ A hopes to send M_d in TR.
6: **if** A does not receive M_s during GR **then**
7: A will transmit M_d in TR;
8: **else** ▷ A receives M_s from other nodes
9: A will transmit M_d in TR by probability $1/2$.
10: **end if**
11: **else** ▷ A does not send M_s
12: **if** A does not receive M_s during GR **then**
13: A will transmit M_d in TR by probability $1/A_n$;
14: **else** ▷ A receives M_s from other nodes
15: A will keep silent in TR.
16: **end if**
17: **end if**

In PHED-GR, each node decides to send M_s by probability $1/A_n$ or keep silent by probability $1 - 1/A_n$. (The values of probabilities are chosen to be optimal according to [3].) Next we face two cases:

1) If A sends an M_s (Line 5-10), it implies A hopes to send M_d in TR.
   a) At this moment, if A does not receive M_s during GR, it means A wins the election and will definitely send M_d in the following TR.
   b) If A receives M_s. It means there exist other candidates within A's direct communication range. Therefore A can only send M_d by probability $1/2$. (We will explain the reason of setting probability $1/2$ after the proof of Lemma 1.)
2) If A does not send M_s (Line 11-17), it implies that A hopes to keep silent in the following TR.
   a) At this moment, if A does not receive M_s in GR, it means no nodes decide to send M_d in TR. A will reconsider sending M_d by probability $1/A_n$.
   b) If A receives M_s. It means that there are nodes intending to transmit and thus A will keep silent.

When PHED-GR is finished, we enter the TR and start the process of neighbor discovering. Next we run PHED-TR: the neighbor discovering process, and the detailed description is shown in Alg. 2.

Algorithm 2 PHED-TR (Neighbor Discovering)

1: **if** A plans to send M_d **then**
2: A sends M_d and monitors the channel meanwhile.
3: **if** A does not receive M_d during TR **then**
4: $A_f = 1$. ▷ A will keep silent from now on
5: **else** ▷ A receives M_d from other nodes
6: Current iteration is invalid.
7: **end if**
8: **else** ▷ A does not plan to send M_d
9: A keeps listening.
10: **if** A does not receive M_d during TR **then**
11: Current iteration is invalid.
12: **else if** A receives a single M_d **then**
13: Record the ID in M_d.
14: $A_n = A_n - 1$. ▷ A records one of its neighbors.
15: **else** ▷ There is a collision at A
16: Current iteration is invalid.
17: **end if**
18: **end if**

In PHED-TR, there are two scenarios:

1) If A sends M_d, A will meanwhile check the existence of other signals (Line 1-7).
   a) If A does not receive M_d during TR, it means that A's transmission is successful. Consequently A will keep silent during the rest of ND process.
   b) If A receives M_d from other nodes, it means that the current transmission is failed.
2) If A does not send M_d, A will check the number of transmitters (Line 8-18).
   a) If A does not receive M_d during TR, it implies that no nodes send M_d in TR. Therefore the current iteration is invalid.
   b) If A receives a single M_d during TR, it means that there is one node successfully transmitting its M_d. A will record the ID in M_d and decrease the value of A_n by 1.
   c) If there is a collision at A, it means that the current transmission is failed.

We will keep running PHED-GR and PHED-TR in turn until $A_n = 1$. Now we finish the description of PHED and start the discussion about the performance of this protocol.

We denote the probability that a node successfully transmits its M_d without collisions in TR as P_1. We now begin to analyze the expected time needed to discover all nodes with high probability with two lemmas.

Lemma 1. *When all nodes independently transmit by probability $1/n$, the probability that k nodes transmit simultaneously in a single slot is given by $p_k = \frac{1}{k!e}$ while $n \to +\infty$.*

Proof: Since nodes choose their actions independently, the probability that k nodes transmit simultaneously in a slot with clique size n is given by $p_{n,k} = \binom{n}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{n-k}$. When $n \to +\infty$, we use Poisson distribution to replace Binomial distribution. Hence, $p_k = \lim_{n\to+\infty} p_{n,k} = \frac{e^{-\lambda}\lambda^k}{k!}$ with $\lambda = n \cdot \frac{1}{n} = 1$. Thus the result holds. ■

From Lemma 1 we can see that the probability that 3 or more nodes transmit simultaneously in a sub-slot is so small that it is acceptable to ignore it and assume that there are only 2 nodes transmitting when the collision occurs to simplify the design of PHED since it is hard and also unnecessary to infer the exact number of transmitting nodes, which explains the Line 9 in Alg. 1.

Lemma 2. $(1-\frac{1}{n})^{n-1} \geq \frac{1}{e}, \forall n = 2, 3, \dots$

This lemma is just the same as Lemma 1 in [4].

We then use these two lemmas to evaluate the probability of a successful discovery in an iteration.

Theorem 1. *When there are n nodes in a clique and all nodes run PHED, the probability that a node successfully transmits M_d in TR is bounded by*

$$P_1 \geq \frac{1}{e^2}(1-\frac{1}{n}) + \frac{5}{4e} \tag{1}$$

Furthermore, when $n \to +\infty$,

$$P_1 \geq \frac{1}{e^2} + \frac{5}{4e} \tag{2}$$

Proof: We analyze different events which may occur in GR. If no one sends M_s in GR, all nodes will reconsider their actions. The successful event's (only one node transmits in TR) probability is

$$p_0 = (1-\frac{1}{n})^n \binom{n}{1}\frac{1}{n}(1-\frac{1}{n})^{n-1} = (1-\frac{1}{n})^{2n-1} \tag{3}$$

If there is exactly one node sending a signal in GR, no collisions will occur in TR. Therefore the probability is

$$p_1 = \binom{n}{1}\frac{1}{n}(1-\frac{1}{n})^{n-1} = (1-\frac{1}{n})^{n-1} \tag{4}$$

If there are at least two nodes transmitting signals in GR, each node will transmit its M_d with probability $1/2$. Thus the successful event's probability is

$$\begin{aligned} p_2 &= \sum_{k=2}^{n}\binom{n}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{n-k}\cdot k \cdot \frac{1}{2}(1-\frac{1}{2})^{k-1} \\ &= \sum_{k=2}^{n}\binom{n}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{n-k}\frac{k}{2^k} \end{aligned} \tag{5}$$

Obviously, $P_1 = p_0 + p_1 + p_2$. Together with Lemma 2, we can get the following inequalities.

$$p_0 = (1-\frac{1}{n})^{2n-2}(1-\frac{1}{n}) \geq \frac{1}{e^2}(1-\frac{1}{n}); \quad p_1 \geq \frac{1}{e}; \tag{6}$$

$$p_2 \geq \binom{n}{2}(\frac{1}{n})^2(1-\frac{1}{n})^{n-2}\frac{2}{2^2} = \frac{1}{4}(1-\frac{1}{n})^{n-1} \geq \frac{1}{4e} \tag{7}$$

As a result, the theorem holds. The derivation of Inequality (2) is trivial hence we omit it. ■

According to Theorem 1, $P_1 \geq 0.572$ when $n = 10$ and when $n = 20$, $P_1 \geq 0.584$. Note that $\frac{1}{e^2} + \frac{5}{4e} \approx 0.595$. For simplicity, we will regard the Inequality (2) as an equation in our later discussion, i.e., $P_1 = 0.595$.

We can see that the probability is significantly improved in comparison with the probability $1/e$ derived in [4].

B. Recursive Protocol: PHED-tGR

To further improve the successful transmission probability, we introduce more sub-slots in GR before TR in one iteration.

In Subsection III-A, the probability of an idle slot is $(1-\frac{1}{n})^{2n} \approx \frac{1}{e^2} \approx 0.135$. It is still too high in practice, although we have significantly reduced it. Thus we add more sub-slots to reduce this probability. We now give PHED-tGR ($t \geq 2$) with t sub-slots in GR and describe it in Alg. 3.

In PHED-tGR, A_t is the local counter for each node to identify the current sub-slot in GR. Initially $A_t = 0$, and after one round of PHED-tGR, A_t will increase by 1. The maximum value of A_t is t. Because of the synchronization assumption, in each node the local A_t remains the same in each round.

PHED-tGR is very similar to PHED-GR except in two aspects. The first is from Line 1 to 5, in which we put t

Algorithm 3 PHED-tGR (Multiple Pre-HandShaking)

1: **if** $A_t = t$ **then** ▷ PHED-tGR has run t times.
2: A will keep silent in TR and exit.
3: **else** ▷ Still processing in t sub-slots
4: $A_t = A_t + 1$.
5: **end if**
6: **if** $A_f = 1$ **then** ▷ A has successfully sent M_d before.
7: A will keep silent in TR and exit.
8: **end if**
9: A decides to send M_s by probability $1/A_n$.
10: **if** A sends an M_s **then**
11: **if** A does not receive M_s during GR **then**
12: A will transmit M_d in TR;
13: **else** ▷ A receives M_s from other nodes
14: A will transmit M_d in TR by probability $1/2$.
15: **end if**
16: **else** ▷ A does not send an M_s
17: **if** A does not receive M_s during GR **then**
18: Call PHED-tGR and exit.
19: **else** ▷ A receives M_s from other nodes
20: A will keep silent in TR.
21: **end if**
22: **end if**

sub-slots in GR to achieve a higher probability of successful transmissions. The other one is at Line 18, in which PHED-tGR invokes itself recursively to utilize the remaining sub-slots in GR. By using this recursive strategy, we can further reduce the probability of idle slots.

We denote the successful event's occurrence in PHED-tGR as P_t and now we analyze the performance of PHED-tGR.

Theorem 2. *P_{t+1} is bounded by*

$$P_{t+1} \geq \frac{P_t}{e}(1-\frac{1}{n}) + \frac{5}{4e} \tag{8}$$

where P_1 is given by Theorem 1.

Proof: If there are $t+1$ sub-slots in GR, we again analyze different events which may occur in GR. If no one sends signal in GR, all nodes will invoke Alg. 3 recursively. Thus the successful event's probability is

$$p_0 = (1-\frac{1}{n})^n \cdot P_t \geq \frac{P_t}{e}(1-\frac{1}{n}) \tag{9}$$

The other two scenarios are just the same as the proof in Theorem 1. According to the Inequality (6), (7) and (9),

$$P_{t+1} \geq \frac{P_t}{e}(1-\frac{1}{n}) + \frac{5}{4e} \tag{10}$$

Similarly, for simplicity we get

$$P_{t+1} = \frac{P_t}{e} + \frac{5}{4e} \tag{11}$$

as $n \to +\infty$. ■

We then point out the upper bound of P_t.

Theorem 3. $\lim_{t\to+\infty} P_t = \frac{5}{4(e-1)} \approx 0.727$

This result can be derived by using the Equation (11) trivially, hence we omit the proof.

We can see that the probability of a successful transmission in a slot is increased by approximately 98% compared with the probability 0.368 in the algorithm proposed in [4].

C. Proper Number of Sub-Slots

We have proved that the probability of a successful transmission can be significantly increased if there are sufficient sub-slots for nodes to detect other nodes' actions. Nevertheless it is impossible to introduce infinite sub-slots in GR, we now discuss how to select a proper number of sub-slots in GR.

Let us consider the Algorithm PHED-3GR. We can get the lower bound of P_3 due to Theorem 1 and 2 as follows:

$$P_3 \geq \frac{1}{e^4}(1-\frac{1}{k})^3 + \frac{5}{4e^3}(1-\frac{1}{k})^2 + \frac{5}{4e^2}(1-\frac{1}{k}) + \frac{5}{4e} \tag{12}$$

where k stands for the number of nodes to be discovered at the current iteration. We can get $\lim_{k\to+\infty} P_3 \approx 0.710$. It is quite close to the optimal value so it is feasible to introduce only three sub-slots before TR. Now we discuss the expected value and upper bound of slots needed to discover all n nodes.

Theorem 4. *By using PHED-3GR and PHED-TR, the expected value of slots needed to discover all nodes with high probability is $1.5n$.*

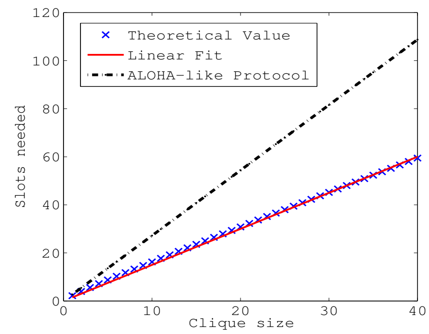

Fig. 2: The comparison among the exact values of PHED, the linear fitting, and the performance of [4] (ALOHA-like Protocol).

Proof: We assume that the discovery process is divided into epochs, and each epoch consists of at least one slot. Epoch i starts when the i-th node is discovered and terminates when the $(i+1)$-th node is discovered. Let T_i denote the number of slots of epoch i and T_i is a geometrically distributed variable with parameter P_3 with $k = n - i$ (There are $n-i$ nodes to be discovered in epoch i). Hence,

$$E[T] = \sum_{k=1}^{n} E[T_k] = \sum_{k=1}^{n} \frac{1}{P_3} \approx 1.5n \tag{13}$$

where the last approximation comes from the result of the linear fitting since it is non-trivial to derive an exact upper bound of the summation. ■

Fig. 2 shows the expected values of time slots needed to discover all nodes in different sizes of cliques in PHED. We can see that the linear fitting is quite close to the theoretical values of PHED and the time used is significantly decreased in comparison with [4].

We next point out the upper bound of the time slots needed to discover all nodes with high probability.

Theorem 5. *By using PHED-3GR and PHED-TR, all nodes can be discovered in $3n$ slots with high probability.*

Proof: Since P_3 varies little as k changes, we regard P_3 as a constant $1/1.5 = 2/3$ for simplicity according to (13). Thus T is a sum of n independent and identically distributed Geometric random variables, and this distribution's parameter is $p = 2/3$. As a result, T is a negative binomial random variable with parameters n and $p = 2/3$.

The probability mass function is:

$$P(T=t) = \binom{t-1}{n-1} p^n (1-p)^{t-n}, t = n, n+1, \ldots \tag{14}$$

On the other hand, the following equation holds:

$$P(T > t) = P(X < n), X \sim \text{Binomial}(t, p) \tag{15}$$

Furthermore, Chernoff bounds point out that:

$$P(X < (1-\delta)tp) < e^{-tp\delta^2/2}, 0 < \delta \leq 1 \tag{16}$$

The formal proof of this inequality can be found in [12]. Then we substitute $\delta = 1 - n/tp$ into (16):

$$P(T > t) = P(X < n) < e^{\frac{-tp}{2}(1-\frac{n}{tp})^2} \tag{17}$$

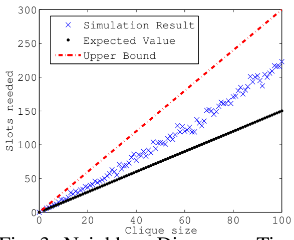

Fig. 3: Neighbor Discovery Time in Clique for PHED

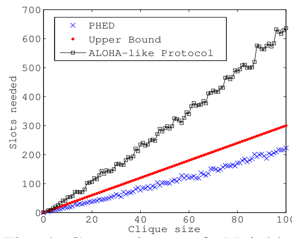

Fig. 4: Comparison of Neighbor Discovery Time in Clique

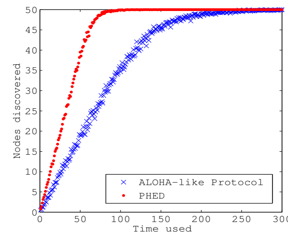

Fig. 5: Comparison of Discovered Node Numbers in Clique

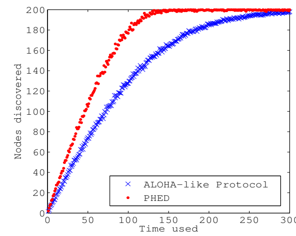

Fig. 6: Comparison of Discovered Node Numbers in Network

Therefore we can get $P(T > 3n) < e^{-\frac{n}{4}}$. It is clear that $e^{-\frac{n}{4}} \to 0$ for sufficiently large n. So the ND process can be finished in $3n$ slots with high probability. ■

D. Unknown Number of Neighbors

We now discuss the situation when n is unknown to nodes.

We divide the process of discovery into phases. In phase i, each node runs the protocol with parameter $n = 2^i$, which means that we assume there are 2^i nodes in phase i. This phase lasts $\lceil 1.5 \cdot 2^i \rceil$ slots. As a result, in the $\lceil \log_2 n \rceil$-th phase, each node regards the number of nodes as n and this phase lasts about $1.5n$ slots. This is just the expected value which we have derived in the Subsection III-C.

The total time needed is $E[T] = \sum_{m=1}^{\lceil \log_2 n \rceil} 1.5 \cdot 2^m$. Since we know that $\sum_{m=1}^{\lceil \log_2 n \rceil} 2^m = 2n - 2$, the total time is

$$E[T] \approx 3(n-1) \tag{18}$$

Hence, the lack of knowledge of n results in about a factor of two slowdown when n is relatively large.

E. Extension for Multi-Hop Networks

PHED can be extended to work in multi-hop wireless networks, in which we must compete with the hidden terminal problem. In PHED, transmitters do not detect the collision themselves and nodes that are in receiving mode will detect collisions. If a receiver has detected the collision in TR, it will send a feedback signal immediately. In this way transmitters will know their transmissions are failed.

Although the hidden terminal problem has a great impact on PHED's performance, in Section IV we will show by simulation that PHED still has much better performance in a multi-hop network than the ALOHA-like protocol.

IV. Performance Evaluation

A. Simulation Setup

Our simulation includes two kinds of settings: network and clique. The network setting is a 300m×300m 2D plane and nodes are put into the plane according to a uniform distribution. The nodes have the same transmission range 50m. There are various clique settings in our simulation. Considering the usual settings of wireless networks, we simulate the discovery process in a clique of 2 nodes to 100 nodes. It can be seen from the previous sections that the more nodes are deployed in a clique, the better PHED's performance will be. Furthermore, we also simulate the situation in a multi-hop network.

We compare PHED with the ALOHA-like protocol with the feedback mechanism proposed in [4]. The advantage of having a feedback mechanism has already been shown in [4, 6]. Thus we will not compare PHED with protocols which do not have such mechanisms. Each data point in the figures stands for an average result over 20 runs for accuracy.

B. Simulation Results

1) Validation of Theoretic Upper Bound: We now use simulation to validate the theorems stating that the expected value of time slots needed is $1.5n$ and the upper bound is $3n$.

Fig. 3 shows the number of slots needed to discover all nodes in different sizes of cliques. Three kinds of values are compared: the simulation results, the expected values and the upper bounds and we can see that the simulation results are larger than the corresponding expected values. This is mainly because when we simulate the discovery process, we regard a value as an output only when all nodes can be discovered in the time given in 20 consecutive runs. Nevertheless, the simulation results are still smaller than the upper bounds we derived, which proves the correctness of our derivation.

2) Comparison in Clique: Similarly we analyze the performance of PHED with the ALOHA-like protocol. For a certain clique size, a time threshold can be regarded as an exact value only when all nodes are discovered in consecutive 20 runs.

Fig. 4 shows the comparison between two protocols with different sizes of cliques. We can see that PHED significantly reduces the processing time, so as the upper bound estimation. When there are 100 nodes in a clique, it takes more than 600 slots to finish ND process by ALOHA-like protocol, whereas PHED only uses 300 slots to finish the process.

We must point out that the definitions of a slot are slightly different in these two protocols. In PHED there are three tiny sub-slots before the normal slot while in [4] there are one sub-slot after the normal slot. Because the duration of sub-slots is really short, (We have mentioned it in Section III.) we can still compare two protocol's performance by comparing their consumption of time slots.

Fig. 5 shows the trend of the number of discovered nodes in a clique with increasing number of iterations. We can see that ND is almost finished after 100 slots in PHED while it costs about 200 slots in ALOHA-like protocol. These observations can also be found in Fig. 3 and Fig. 4.

3) Comparison in Multi-Hop Network: We also evaluate the performance of PHED and the ALOHA-like protocol under the network environment. We put 200 nodes to the plane and the average number of neighbors for a certain node is about 18.

Fig. 6 shows that the number of discovered nodes in a network is increasing with the number of iterations. We can know that after 150 slots, almost all 200 nodes are discovered by PHED, whereas it takes about 300 slots to discover all nodes by the ALOHA-like protocol.

Due to the existence of the hidden terminal, the time needed to discover all nodes in a multi-hop network is much larger than the time needed when there is only one clique. Nevertheless, PHED still significantly reduces the time needed in comparison with the ALOHA-like protocol.

V. Related Work

A large number of works have focused on the problem of ND in wireless networks and various protocols have been proposed to adapt to different situations [3–11]. Due to the space limitation we mainly introduce several works with close relationship with PHED. Birthday protocols in [3] use a randomized strategy for nodes in a synchronous system to choose their actions in a slot independently and randomly. The authors proved that for a clique with n nodes, the optimal probability that a node transmits is $1/n$.

Vasudevan *et al.* [4] later pointed out that the expected time slots needed to finish ND process by using the birthday protocol in [3] is neH_n where H_n is the n-th Harmonic number. The authors also proposed protocols for more realistic situations where the size of a clique is unknown to nodes, a feedback mechanism is introduced into the system and the clocks of nodes are not identical, i.e., the system is asynchronous [7]. Basically, a factor of two slowdown is brought in if the size of a clique is unknown, while a factor of $\ln n$ slowdown is brought in if there are no feedback mechanisms.

Zeng *et al.* [5] extended the result of [4] to the MPR situation where no collision occurs iff there are no more than k $(k \geq 2)$ nodes transmitting simultaneously and proved that the expected time needed to discover all nodes is $\Theta(n \ln n/k)$. Ideally, if $k \geq n$, the discover time is shortened to $\Theta(\ln n)$. Similarly, the authors designed protocols for realistic situations in [4] and analyzed the upper bounds respectively.

You *et al.* [8] extended the result of [4] to the situation when the duty cycle of nodes is not 1, i.e., some nodes may be dormant at a certain time instant. By reducing the problem to the generalization of the classical Coupon Collector's Problem [13], the authors proved that when the duty cycle is $1/2$, the upper bound is $ne(\log_2 n + (3\log_2 n - 1)\log_2 \log_2 n + c)$ with a constant c and the lack of knowledge of n results in a factor of two slowdown as well in a clique.

Many papers have focused on the feasibility of designing a practical full duplex wireless radio. Choi *et al.* [1] proposed a method named antenna cancellation to avoid self-interference. However, this technique requires three antennas, which makes it unattractive in comparison with a 3-antenna MIMO system with higher throughput. This method also suffers from the constraint of bandwidth seriously, which makes it not feasible for wideband signals such as WiFi.

Jain *et al.* [2] overcame the drawbacks of [1] and proposed a novel mechanism which is called balun cancellation, in which a balun circuit is used to create inverse signals to achieve the cancellation of self-interference. This method requires only two antennas and has no bandwidth constraints theoretically. Furthermore, the authors had made an experimental device which supports the signal channel full duplex communication. Though some realistic conditions make the device not as perfect as it is in theory, it is still safe to say that the single channel full duplex technology is promising and thus our work utilizes it to accelerate the ND process.

VI. Conclusion and Future Work

In this paper, we proposed a pre-handshaking neighbor discovery protocol PHED by adding pre-handshaking sub-slots before the traditional slots. Furthermore, we applied the full duplex technology and used it to conduct pre-handshaking with new feedback mechanisms. We analyzed the expected value and upper bound of ND processing time theoretically, and validated our analysis by simulation compared with the ALOHA-like protocol proposed in [4]. Both theoretical analysis and numerical experiments proved that PHED significantly decreases the time needed to finish the ND process.

In the future, we would like to evaluate the performance of PHED by test-bed experiments. We also want to consider more realistic models, e.g., nodes with MPR techniques, nodes with low duty cycles and asynchronous models.

References

[1] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. "Achieving Single Channel, Full Duplex Wireless Communication". In *Proc. of ACM MobiCom*, 2010.

[2] M. Jain, J. I. Choi, T. M. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. "Practical, Real-time, Full Duplex Wireless". In *Proc. of ACM MobiCom*, 301-312, 2011.

[3] M. J. McGlynn and S. A. Borbash. "Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks". In *Proc. of ACM MobiHoc*, 137-145, 2001.

[4] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili. "Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem". In *Proc. of ACM MobiCom*, 181-192, 2009.

[5] W. Zeng, X. Chen, A. Russell, S. Vasudevan, B. Wang, and W. Wei. "Neighbor Discovery in Wireless Networks with Multipacket Reception". In *Proc. of ACM MobiHoc*, 3:1-10, 2011.

[6] R. Khalili, D. Goeckel, D. Towsley, and A. Swami. "Neighbor Discovery with Reception Status Feedback to Transmitters". In *Proc. of IEEE INFOCOM*, 2010.

[7] S. A. Borbash, A. Ephremides, and M. J. McGlynn. "An Asynchronous Neighbor Discovery Algorithm for Wireless Sensor Networks". Elsevier Ad Hoc Networks, 5(7): 998-1016, 2007.

[8] L. You, Z. Yuan, P. Yang, and G. Chen. "ALOHA-Like Neighbor Discovery in Low-Duty-Cycle Wireless Sensor Networks". In *Proc. of IEEE WCNC*, 749-754, 2011.

[9] A. Keshavarzian and E. Uysal-Biyikoglu. "Energy-Efficient Link Assessment in Wireless Sensor Networks". In *Proc. of IEEE INFOCOM*, 2004.

[10] D. Angelosante, E. Biglieri, and M. Lops. "Neighbor Discovery in Wireless Networks: A Multiuser-Detection Approach". In *Information Theory and Applications Workshop*, 46-53, 2007.

[11] Z. Zhang and B. L., "Neighbor Discovery in Mobile Ad Hoc Self-Configuring Networks with Directional Antennas: Algorithms and Comparisons". In *IEEE TWC*, (7)5:1540-1549, 2008.

[12] R. Motwani and P. Raghavan. "Randomized Algorithms". Cambridge University Press, 1995.

[13] P. Erdös and A. Rényi. "On a Classical Problem of Probability Theory". Magyar Tud. Akad. Mat. Kutato Int. Kozl, 1961.

[14] L. Kong, L. Fu, X. Liu and M. Wu. "Accelerating Initialization for Sensor Networks". In *Proc. of IEEE GLOBECOM*, 2009.