INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 4, Issue 5 AUG 2015

Efficient Algorithms for Neighbor Node Discovery in Wireless Sensor Networks

[1] **CHAGANTI.B. N.LAKSHMI, Assoc.Professor M.Tech[Ph.D],**

[2] **MAMTA SHARMA, PG Scholar in S E (CSE),**

[1] ch.nagalakshmi08@gmai l.com, [2] sharmamamta1225@gmail.com

[1,2] Mahaveer Institute of science & Technology, Hyderabad, R.R Dist, Telangana –India

***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 prolongs the executing duration, and thus compromises their performance. In this paper, we propose a novel randomized protocol FRIEND, a pre-handshaking neighbor discovery protocol, to initialize synchronous full duplex wireless ad hoc networks. Byintroducing a pre-handshaking 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 FRIEND, and construct the first full duplex neighbor discovery protocol. Our theoretical analysis proves that FRIEND can decrease the duration of ND by up to 48% in comparison to the classical ALOHA-like protocols [3, 4]. In addition, we propose HD-FRIEND for half duplex networks and variants of FRIEND for multi-hop networks and duty cycled networks. Both theoretical analysis and simulation results show that FRIEND can adapt to various scenarios, and significantly decrease the duration of ND.**

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

I. INTRODUCTION

Wireless ad hoc networks have attracted a lot of interest 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, nodes in a wireless ad hoc network need to configure themselves through their own communication activities to form a reliable infrastructure during 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 existing deterministic [11] and multi-user detection-based [12] protocols, randomized protocols are most commonly used to conduct ND process in wireless networks [3–8]. In those protocols, 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 a synchronous system, 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 proven to be $1/n$). By reducing the ND problem to *Coupon Collector's Problem* [16], 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 by reducing the problem to *K Coupon Collector's Problem*.

However, the family of birthday protocols has a vital drawback. The probability of generating an idle slot is given by

$$p_0 = (1 - \frac{1}{n})^n.$$

When $n = 10$, $p_0 \approx 0.349$. When $n \to +\infty$, $p_0 \to 1/e \approx$ Therefore when the number of nodes is large, the probability that no node transmits in a slot is about 37%. We must point out that the probability that there is only one node transmitting is

$$p_1 = \binom{n}{1}\frac{1}{n}(1 - \frac{1}{n})^{n-1} \geq \frac{1}{e}.$$

The last inequality comes from the Lemma 2 which we will present in later sections. We can see that compared with the probability that a node successfully transmits its discovery message, the probability of idle slots is as large as it, and they contribute about 73% to the all possible scenarios. 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

This work was supported in part by the State Key Development Program for Basic Research of China (Grant No. 2012CB316201), in part by National Natural Science Foundation of China (Grant No. 61170236, 61272443, 61133006, 61073152, 61202024), in part by Shanghai Science and Technology fund (No. 12PJ1404900), in part by Shanghai Educational Development Foundation (Chenguang Grant No.12CG09), in part by the Natural Science Foundation of Shanghai (Grant No.12ZR1445000), and in part by Key Project of Educational Science Research of Shanghai of China (No. A1120). The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

G. Sun, F. Wu, X. Gao, and G. Chen are with the Department of Computer Science and Engineering at Shanghai Jiao Tong University, P.R. China (E-mail: sunguobao@sjtu.edu.cn, {fwu, gao-xf, gchen}@cs.sjtu.edu.cn). W. Wang is with the Department of Computer Science and Technology at Tongji University, P.R. China (E-mail: wwang@tongji.edu.cn).

* F. Wu is the corresponding author.

INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 4, Issue 5 AUG 2015

$1/(6e) \approx 0.06$.[1] Comparing the relatively small probability of collisions, the idle slot probability is unacceptably high. If we can effectively reduce the probability of idle slots, the neighbor discovery time will be tremendously reduced. Fortunately, with the development of full duplex wireless communication technology [1, 2], we can design more time-efficient protocols, i.e., protocols that consume less time, to cope with this issue if nodes can transmit and receive 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 conduct this pre-handshaking, 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 discovery message in a normal slot by transmitting an anonymous election signal and catch its neighbors' signals simultaneously. With different transmitting-receiving scenarios, we design an effective strategy for each node to determine how to behave in normal slots. Correspondingly, we assign the behaviors of each node in the normal slots to complete the ND process. On the other hand, the reception status feedback mechanism is ameliorated by using full duplex wireless radios. Originally in [6], a sub-slot is added after the normal slot, and receivers 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 receiver finds that two or more nodes are transmitting simultaneously, it will transmit a warning message immediately to inform other transmitters the failure of their transmissions.

Our contributions in this paper are listed as follows:

- We propose a novel ND protocol named FRIEND, which is a protocol based on pre-handshaking activities, in which pre-handshaking activities are inserted before each normal slot. In FRIEND 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, research on ND problem in wireless networks are based on half duplex nodes. The 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.
- We extend the discussion of FRIEND to multi-hop networks, and show that FRIEND still performs better than the ALOHA-like protocol.
- Furthermore, we propose another strategy named HD-FRIEND for nodes with half duplex radios, our theoretical analysis shows that HD-FRIEND decrease the time for ND process by approximately 36.7%.
- Finally, we discuss how to improve the ND process in

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

INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies
Volume 4, Issue 5 AUG 2015

low-duty-cycle networks, and propose methods to handle the problem of when to start and terminate ND when n is unknown to nodes.

The rest of this paper is organized as follows. Section II describes our model and assumptions. Section III introduces FRIEND and its theoretical analysis. Section V gives some discussion and extensions, and in Section VI we evaluate FRIEND by simulations. In Section VII we present related works. The paper concludes with our future works in Section VIII.

II. NETWORK MODEL AND ASSUMPTIONS

In this section, we introduce the network model and sev- eral assumptions, under which we will present our FRIEND protocol and corresponding analysis. These assumptions are reasonable in the research on ND and many former works, including the traditional ALOHA-like protocols, 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. The synchronization can be achieved by different techniques and many works have focused on this problem (e.g. [19, 20, 27]).
- All nodes are in a clique of size n.
- n is known to all nodes in the clique. Typically, n can be pre-configured on nodes before deploying, or calculated based on the density of the network. The pre-configured or calculated result does not need to be exactly accurate, because a small difference only has little influence on nodes decisions about transmission probabilities and can be ignored normally.
- Nodes use omnidirectional antennas, and all nodes have the same transmission range.
- No multipacket reception technique is used, i.e., for a node that is receiving, a collision occurs when two or more nodes simultaneously transmit packets to it 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. Hence 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. FRIEND: PRE-HANDSHAKING PROTOCOL

In this section, we present our novel protocol FRIEND 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 FRIEND. Moreover, 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 with least overhead.

INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 4, Issue 5 AUG 2015

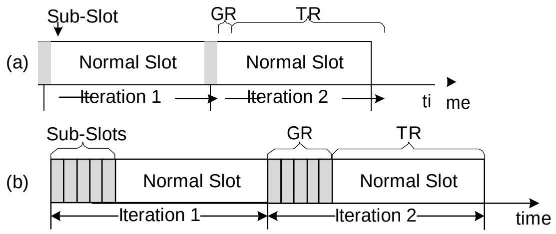

Fig. 1: The description of an iteration. GR is used for pre-handshaking, and TR is used for transmitting discovery messages. In (a), there is one sub-slot in GR, while in (b) there are multiple sub-slots in GR.

A. FRIEND 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 ([18] 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 FRIEND protocol to determine the action of a node in a slot. FRIEND is a distributed protocol and for each node the target is to discover all its neighbors after finite iterations. Assume that we are considering a clique of n nodes. We divide FRIEND into two sub-routines: FRIEND-GR and FRIEND-TR.

Let us describe the main idea of FRIEND-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 FRIEND-GR. Note that each node should run a copy of FRIEND-GR. To simplify our description, assume that we run FRIEND-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.

In FRIEND-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.

Algorithm 1 FRIEND-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**

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

In FRIEND-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.

INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 4, Issue 5 AUG 2015

Algorithm 2 FRIEND-TR (Neighbor Discovering)

```
if A plans to send M_d then
    A sends M_d and monitors the channel meanwhile.
    if A does not receive M_d during TR then
        A_f = 1.                  ▷ A will keep silent from now on
    else                          ▷ A receives M_d from other nodes
        Current iteration is invalid.
    end if
else                              ▷ A does not plan to send M_d
    A keeps listening.
    if A does not receive M_d during TR then
        Current iteration is invalid.
    else if A receives a single M_d then
        Record the ID in M_d.
        A_n = A_n − 1.   ▷ A records one of its neighbors.
    else                          ▷ There is a collision at A
        Current iteration is invalid.
    end if
end if
```

We will keep running FRIEND-GR and FRIEND-TR in turn until $A_n = 1$. Now we finish the description of FRIEND 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 FRIEND 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. $(1-\frac{1}{n})^{n-1} \geq \frac{1}{e}, \forall n = 2, 3, \ldots$

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 FRIEND, the probability that a node successfully transmits M_d in TR is bounded by*

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

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

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

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

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

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

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

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

$$p_2 = \sum_{k=2}^{n}\binom{n}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{n-k} \cdot k \cdot \frac{1}{2}(1-\frac{1}{2})^{k-1}$$
$$= \sum_{k=2}^{n}\binom{n}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{n-k}\frac{k}{2^k}$$

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

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

$$p_2 \geq \binom{n}{2}(\frac{1}{n})^2(1-\frac{1}{n})^{n-2}\frac{2}{2^2} = \frac{1}{4}(1-\frac{1}{n})^{n-1} \geq \frac{1}{4e} \qquad (3)$$

As a result, the theorem holds. The derivation of Inequality (1) 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 (1) 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: FRIEND-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 FRIEND-tGR ($t \geq 2$) with t sub-slots in GR and describe it in Alg. 3.

In FRIEND-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 FRIEND-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.

FRIEND-tGR is very similar to FRIEND-GR except in two aspects. The first is from Line 1 to 5, in which we put t sub-slots in GR to achieve a higher probability of successful transmissions. The other one is at Line 18, in which FRIEND-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.

Algorithm 3 FRIEND-tGR (Multiple Pre-HandShaking)

```
if A_t = t then                          ▷ FRIEND-tGR has run t times.
    A will keep silent in TR and exit.
else                                     ▷ Still processing in t sub-slots
    A_t = A_t + 1.
end if
if A_f = 1 then                          ▷ A has successfully sent M_d before.
    A will keep silent in TR and exit.
end if
A decides to send M_s by probability 1/A_n.
if A sends an M_s then
    if A does not receive M_s during GR then
        A will transmit M_d in TR;
    else                                 ▷ A receives M_s from other nodes
        A will transmit M_d in TR by probability 1/2.
    end if
else                                     ▷ A does not send an M_s
    if A does not receive M_s during GR then
        Call FRIEND-tGR and exit.
    else                                 ▷ A receives M_s from other nodes
        A will keep silent in TR.
    end if
end if
```

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

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

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

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

$$p_0 = (1-\frac{1}{n})^n \cdot P_t \geq \frac{P_t}{e}(1-\frac{1}{n}) \quad (4)$$

The other two scenarios are just the same as the proof in Theorem 1. According to the Inequality (2), (3) and (4),

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

Similarly, for simplicity we get

$$P_{t+1} = \frac{P_t}{e} + \frac{5}{4e} \quad (5)$$

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 (5) 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 FRIEND-3GR. We can get the lower bound of P_3 due to Theorem 1 and 2 as follows:

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

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.

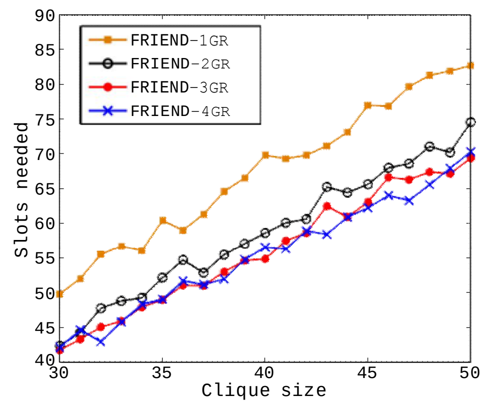

Fig. 2: Comparison of FRIEND-tGR

We can also compute the probability with other numbers of sub-slots in GR.

$$\lim_{k\to+\infty} P_2 = \frac{1}{e^3} + \frac{5}{4e^2} + \frac{5}{4e} \approx 0.679$$

$$\lim_{k\to+\infty} P_4 = \frac{1}{e^5} + \frac{5}{4e^4} + \frac{5}{4e^3} + \frac{5}{4e^2} + \frac{5}{4e} \approx 0.720$$

It is obvious that more sub-slots used for GR require more accurate synchronization techniques. To make the trade-off, it is feasible to determine that there are three sub-slots in GR. Our simulation for different numbers of sub-slots in GR also proved this. In Fig. 2, we can see that FRIEND-3GR has almost the same performance as FRIEND-4GR, but has less overhead and requirement of synchronizing techniques.

Now we discuss the expected value and upper bound of slots needed to discover all n nodes by FRIEND-3GR.

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

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

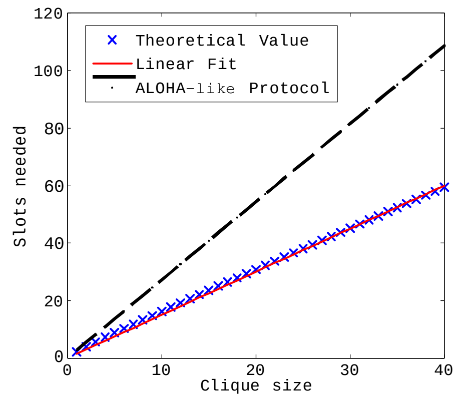

Fig. 3: The comparison among the exact values of FRIEND, the linear fitting, and the performance of [4] (ALOHA-like Protocol).

with parameter P_3 with $k = n - i$ (There are $n - i$ nodes to be discovered in epoch i). Hence,

$$E[T] = \sum_{k=1}^{n} E[T_k] = \sum_{k=1}^{n} \frac{1}{P_3} \approx 1.5n \qquad (6)$$

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. 3 shows the expected values of time slots needed to discover all nodes in different sizes of cliques in FRIEND. We can see that the linear fitting is quite close to the theoretical values of FRIEND 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 FRIEND-3GR and FRIEND-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 (6). 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:

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

On the other hand, the following equation holds:

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

Furthermore, Chernoff bounds point out that:

$$P(X < (1-\delta)tp) < e^{-tp\delta^2/2}, 0 < \delta \leq 1 \qquad (7)$$

The formal proof of this inequality can be found in [16]. Then we substitute $\delta = 1 - n/tp$ into (7):

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

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. ■

IV. HD-FRIEND: FRIEND FOR HALF DUPLEX RADIOS

For nodes with half duplex radios, although nodes cannot be aware of other nodes' actions during their own transmissions, we can still use the similar strategy to reduce the probability of generating idle slots. We name it as *HD-FRIEND*, which means the half duplex counterpart of FRIEND.

Similarly, there should be at least two sub-slots in one iteration, which is the same as the case in FRIEND. However, there should be one more sub-slot that is used for transmitting feedback signals, because radios are half duplex and cannot notify collisions during reception. As a result, there are three sub-slots in one iteration, the first one is used to conduct greeting process (*GR* sub-slot), and the second one is used for transmission of discovery message (*TR* sub-slot). Feedback signals are transmitted in the third sub-slot (*FB* sub-slot). We then assign different actions for different nodes that may choose to transmit or receive in one iteration. Initial settings are the same as they are in Section III, so we omit them.

In GR sub-slots, each node runs Alg. 4 to determine nodes that may transmit in TR sub-slot.

Algorithm 4 HD-FRIEND-GR (Half Duplex)

if $A_f = 1$ **then** ▷ A has successfully sent M_d.
    A will keep silent in TR (as well as FB) and exit.
end if
Node A decides to send M_s by probability $1/A_n$ and keep listening by probability $1 - 1/A_n$.
if A sends M_s **then** ▷ A hopes to send M_d in TR.
    A will transmit M_d in TR;
else ▷ A does not send M_s
    if A does not receive M_s during GR **then**
        A will transmit M_d in TR by probability $1/A_n$;
    else ▷ A receives M_s from other nodes
        A will keep silent in TR.
    end if
end if

We can see that the main difference in GR from the algorithm above. If a node intends to transmit in TR, it will send M_s in GR to notify other nodes, and send M_d in TR regardless of other nodes' actions. Receiving nodes behave the same way as they are in FRIEND.

Then in TR sub-slot, every node runs Alg. 5 to determine actions in FB sub-slot.

For a transmitting node, it will send its discovery message during TR and keep listening in FB to get feedback. While for a receiving node, it will keep listening in TR to determine whether to send a feedback signal to notify the failure of the transmission to transmitting nodes.

After TR, nodes enter into FB sub-slot and the transmitting node will be aware of whether its transmission is successful. Each node runs Alg. 6 in FB sub-slots.

In the FB sub-slot, if a receiving node detects collision, it will broadcast a feedback signal. As a result, transmitting nodes knows that their transmissions are failed. On the other hand, if the transmitting node does not receive the feedback signal, it knows that the transmission is successful, and it is time for it to keep silent during the remaining process of ND.

We can use similar method to analyze the performance of HD-FRIEND. A theorem is as follows:

Algorithm 5 HD-FRIEND-TR (Half Duplex)

```
if A plans to send M_d then
    A sends M_d.
    A will keep listening in FB.
else                                    ▷ A does not plan to send M_d
    A keeps listening.
    if A does not receive M_d during TR then
        Current iteration is invalid.
    else if A receives a single M_d then
        Record the ID in M_d.
        A_n = A_n − 1.   ▷ A records one of its neighbors.
    else                                ▷ There is a collision at A
        A will send a feedback signal in FB.
    end if
end if
```

Algorithm 6 HD-FRIEND-FB (Half Duplex)

```
if A transmitted in TR then
    A keeps listening in FB.
    if A receives the feedback signal. then
        Current iteration is invalid.
    else
        A_f = 1.
    end if
else                                    ▷ A received in TR
    if A plans to send a feedback signal in FB then
        Send the feedback signal.
    end if
end if
```

Theorem 6. *When there are n nodes in a clique and all nodes run HD-FRIEND, the probability Q_1 that a node successfully transmits M_d in TR is bounded by*

$$Q_1 \ge \frac{1}{e} + \frac{1}{e^2}\left(1 - \frac{1}{n}\right) \approx 0.503.$$

Proof: There are two cases where there is one and only one node transmitting its M_d in TR. If there is only one node transmitting M_s in GR, then the node is surely to be discovered. We denote this probability as p_1^r and it is easy to see that

$$p_1^r = \binom{n}{1}\frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} = \left(1 - \frac{1}{n}\right)^{n-1} \ge \frac{1}{e}.$$

If there are no nodes transmitting M_s in GR, all nodes will reconsider their actions. We denote the probability that there is one node transmitting in TR in this case as p_2^r, and we can similarly get

$$p_2^r = \left(1 - \frac{1}{n}\right)^n \cdot p_1^r \ge \frac{1}{e^2}\left(1 - \frac{1}{n}\right).$$

According to the equation

$$Q_1 = p_1^r + p_2^r,$$

the theorem holds. ■

Furthermore, we can insert more sub-slots in GR and extend HD-FRIEND-GR to HD-FRIEND-*t*GR, by using the same method as FRIEND-*t*GR. We denote the probability that a node successfully transmits in TR as Q_t fot the case where there are t sub-slots in GR. The proof of the following theorem is similar as it is in Section III, and thus we omit it.

Theorem 7. *Q_{t+1} is bounded by*

$$Q_{t+1} \ge \frac{Q_t}{e}\left(1 - \frac{1}{n}\right) + \frac{1}{e}$$

We can also get the upper bound of Q_t.

Theorem 8. $\lim_{t \to +\infty} Q_t = \frac{1}{e-1} \approx 0.582.$

The results show that our strategy can still decrease the neighbor discovery time by up to 36.7%, although nodes only have half duplex radios. In Sec. VI, we will present the simulation result for HD-FRIEND.

V. Discussion

In this section, we discuss some issues related to the FRIEND and their corresponding analysis results. Although the following discussions are based on FRIEND, they still hold for HD-FRIEND, due to the reason that FRIEND and HD-FRIEND are based on the similar pre-handshaking strategy.

A. Unknown Number of Neighbors

In this subsection, we discuss the situation when n is unknown to nodes. The basic idea is similar to [4].

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.

Now we determine the expected time needed in this case, the total time is given by

$$E[T] = \sum_{m=1}^{\lceil \log_2 n \rceil} 1.5 \cdot 2^m$$

Since we know that $\sum_{m=1}^{\lceil \log_2 n \rceil} 2^m = 2n - 2$, the total time is

$$E[T] \approx 3(n-1)$$

Comparing the result with Theorem 4, we can observe that the lack of knowledge of n results in about a factor of two slowdown when n is relatively large. We will show the simulation results of unknown n case in Section VI.

B. Multi-Hop Networks

There is an obvious obstacle we must face when extending FRIEND to the multi-hop case. In a clique, a node i can determine whether its transmission is successful or not by itself, by keeping listening during its transmission. Nevertheless, the potential hidden terminals may cause that not all nodes within i's transmission range receive i's message correctly.

To make sure that transmitting nodes can get correct feedback signals, we can modify FRIEND to let receivers take the responsibility. Now, transmitters do not detect collisions themselves and nodes that are receiving 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.

The existence of hidden terminals has a great impact on FRIEND's performance. This is predictable because that each node belongs to several different cliques, and a node's action in a clique may affect nodes in other cliques. However, according to simulation we show that FRIEND still has much better performance than the ALOHA-like protocol, which we will see in Section VI.

C. Low-Duty-Cycle Scenario

The basic drawback of the ALOHA-like protocol also exists in low-duty-cycle wireless networks. In [8], the performance of ALOHA-like protocol is analyzed when nodes' duty cycle is smaller than 1. In this case, the probability of generating an idle slot is still given by

$$p_0 = (1 - \frac{1}{n})^n \approx \frac{1}{e},$$

which indicates that the basic idea of our FRIEND can also be utilized here. Since we have conducted extensive analysis of FRIEND, we will omit the further discussion here.

We note that the performance of FRIEND can be optimized furthermore. When nodes all choose to be silent in GR, there may be some number of nodes are not in receiving mode, but in dormant mode. As a result, each node can decrease its estimation of n (e.g., set $n = \frac{1}{2}n$ after each round), and thus improve the probability of transmitting. However in this paper we do not focus on designing the strategy of estimating n, and leave it to our future work.

D. Initiating and Terminating ND Process

In this part, we will discuss the case when nodes start the process of ND at different time instants instead of the previous discussion, which is based on the assumption that nodes start at the same start instant. In addition, we will address the issue about when to terminate the ND process when n is unknown to nodes.

1) Initialization: To simplify the discussion, we assume that the maximum offset between any two nodes in the clique is δ. As a result, if the node which is the earliest one to start ND begins at time slot t, all nodes will begin at $[t, t + \delta]$.

In comparison with Subsection V-A, we add δ slots to each phase, i.e., now phase i lasts $\lceil 1.5 \cdot 2^i \rceil + \delta$ slots. Hence, all nodes will stay in the i-th phase for at least $\lceil 1.5 \cdot 2^i \rceil$ time slots, and all nodes can be discovered with high probability at the $\lceil \log_2 n \rceil$-th phase.

2) Termination: We use the similar strategy in [4] to determine when to terminate the ND process. An extra time slot is added to the end of each phase, i.e., phase i lasts $\lceil 1.5 \cdot 2^i \rceil + 1$ slots. In the last slot of a phase, nodes that have not successfully transmitted their discovery messages will broadcast signals in this slot. Therefore, the process of ND can be terminated once no signals can be detected in the last slot of a phase.

VI. Performance Evaluation

A. Simulation Setup

Simulations of the performance comparison were implemented using MATLAB. We simulate the random actions that nodes may choose to take in a slot, according to the corresponding probabilities. Our simulations include various settings of the sizes of the cliques. In a clique where all nodes are within communication range of each other, we simulate the discovery process in a clique of 3 nodes to 100 nodes, considering the usual settings of wireless networks. It can be seen from the previous sections that the more nodes are deployed in a clique, the better FRIEND's performance will be. In terms of the multi-hop case, we put 200 nodes to a 300m×300m 2D plane. Nodes are put into the plane according to a uniform distribution, and they all have the same transmission range 50m. We can know that the average number of neighbors for a certain node is about 18.

We compare FRIEND-3GR with the ALOHA-like protocol with the feedback mechanism proposed in [4]. Furthermore, we show the simulation results of HD-FRIEND-3GR, and the results for the unknown n scenario. The advantage of having a feedback mechanism has already been shown in [4, 6]. Thus we will not compare FRIEND-3GR 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

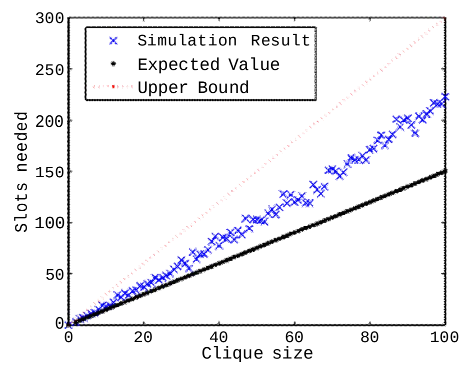

Fig. 4: Neighbor Discovery Time in Clique for FRIEND-3GR

1) Validation of Theoretical 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. 4 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. 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 simulation 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 FRIEND-3GR with the ALOHA-like protocol (In this paper, we regard the ALOHA-like protocol as the birthday protocol with a feedback mechanism proposed in [4].). 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. 5 shows the comparison between two protocols with different sizes of cliques. We can see that FRIEND-3GR

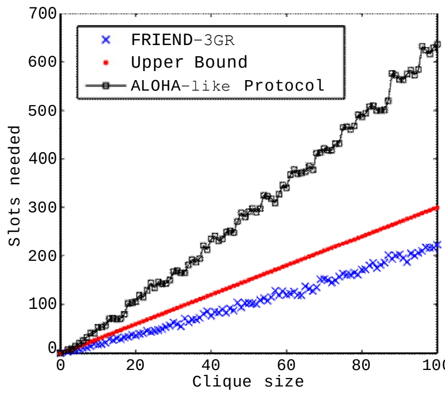

Fig. 5: Comparison of Neighbor Discovery Time in Clique

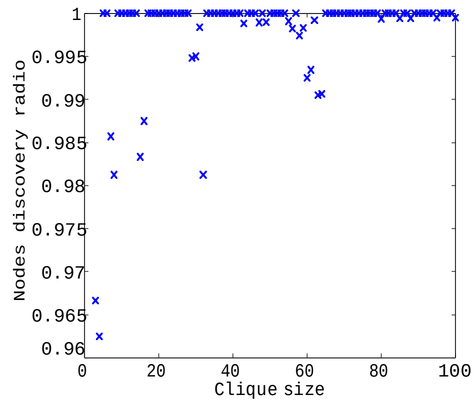

Fig. 7: Nodes Discovery Ratio of FRIEND-3GR (Unknown n) in Clique

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 FRIEND-3GR 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 FRIEND-3GR 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 protocols' performance by comparing their consumption of time slots.

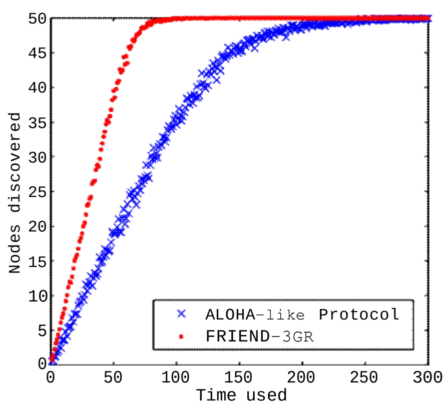

Fig. 6: Comparison of Discovered Node Numbers in Clique

Fig. 6 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 FRIEND-3GR while it costs about 200 slots in ALOHA-like protocol. These observations can also be found in Fig. 4 and Fig. 5.

3) Validation of n Unknown Case: The nodes discovery ratio of different clique sizes when FRIEND-3GR is deployed without the knowledge of n is shown in Fig. 7. All nodes can be discovered in the given time with high probability, which proves the correctness and effectiveness of the estimation mechanism proposed in Sec. V.

It is worthy noticing that this time the discovery process can be finished in the expected values with high probability, which is different to the previous simulation for FRIEND. This is mainly because that in the estimation mechanism, the $\lfloor\log_2 n\rfloor$-th phase is actually conducting the ND process with desired configuration. However, phases before the $\lfloor\log_2 n\rfloor$-th phase will also contribute to the discovering process. Consequently, here the discovery process can be finished in the corresponding expected values.

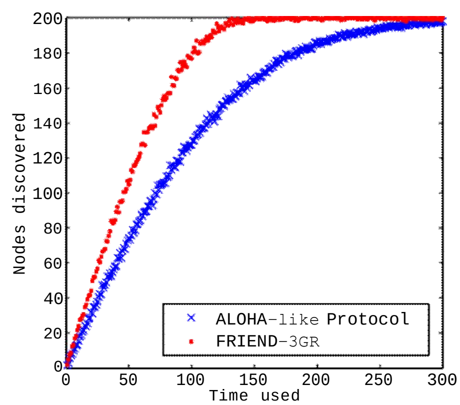

Fig. 8: Comparison of Discovered Node Numbers in Network

4) Comparison in Multi-Hop Networks: Fig. 8 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 FRIEND-3GR, whereas it takes about 300 slots to discover all nodes by the ALOHA-like protocol.

5) Half Duplex Networks: Due to the similar reason in Subsection III-C, we choose HD-FRIEND-3GR to simulate, and compare its performance with the ALOHA-like protocol with the feedback mechanism. In Fig. 9, we can see that even if nodes use half duplex radios, HD-FRIEND-3GR still decrease the time tremendously.

The definitions of a slot are different in HD-FRIEND-3GR and [4]. HD-FRIEND-3GR has three more sub-slots before the slot that is described in [4]. Nevertheless, since the length of sub-slots is short (as we have mentioned in Section III), we can compare two protocols' performance by comparing the number of slots.

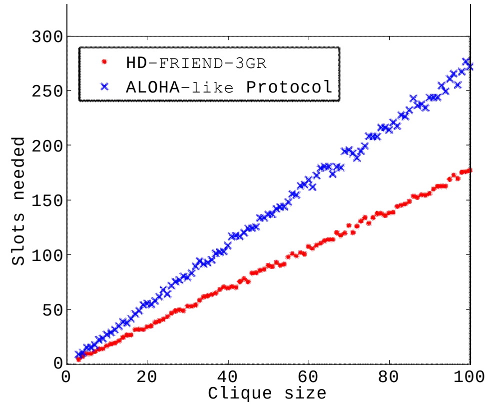

Fig. 9: Comparison of Neighbbor Discovery Time in Clique (HD-FRIEND-3GR)

VII. Related Work

A large number of works have focused on the problem of accelerating the process of ND in wireless networks and various protocols have been proposed to adapt to different situations [3–15]. Due to the space limitation we mainly introduce several works with close relationship with FRIEND. 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.

With the development of multipacket reception [23], Zeng *et al.* [5] extended the result of [4] to the multipacket reception situation where no collision occurs if and only if 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 *el al.* [26] proposed the similar result in P2P multipacket reception networks.

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 [17], 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. Sun *et al.* [24] proposed a pre-handshaking strategy, which significantly reduces the probability of idle slots, for full duplex wireless ad hoc networks. In [25], Sun *et al.* also discussed how the ALOHA-like protocol works in low duty cycle sensor networks with MPR technique.

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 of [1, 2] 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.

Besides the works which are aimed at accelerating the process of ND, there are also many other researches that discuss other issues about ND. For instance, in [21, 28], the authors proposed the formal system definition of secure neighbor discovery when there are adversary nodes in the environment. Furthermore, they proposed a secure neighbor discovery protocol when the system model satisfies certain conditions. The authors in [22] discussed the issue of energy consumption of ND process. Since our main concern is how to finish the ND process in shortest time, we will not introduce them in detail.

VIII. Conclusion and Future Work

In this paper, we proposed a pre-handshaking neighbor discovery protocol FRIEND by adding pre-handshaking subslots 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 simulations proved that FRIEND significantly decreases the time needed to finish the ND process. Furthermore, we discussed some implementation issues and extensions of FRIEND, and showed that the half duplex counterpart of FRIEND, i.e., HD-FRIEND, also significantly decreases time consumption.

In the future, we would like to evaluate the performance of FRIEND by test-bed experiments. We also want to consider more realistic models, e.g., nodes with multipacket reception 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] X. An, R. Venkatesha Prasad, and I. Niemegeers. “Impact of Antenna Pattern and Link Model on Directional Neighbor Discovery in 60 GHz Networks”. In *IEEE TWC*, (10)5:1435-1447, 2011.
[10] R. Cohen and B. Kapchits. “Continuous Neighbor Discovery in Asynchronous Sensor Networks”. In *IEEE TON*, (19)1:69-79, 2011.
[11] A. Keshavarzian and E. Uysal-Biyikoglu. “Energy-Efficient Link Assessment in Wireless Sensor Networks”. In *Proc. of IEEE INFOCOM*, 2004.
[12] 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.
[13] 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.
[14] N. Karowski, A. Viana, and A. Wolisz. “Optimized Asynchronous Multi-Channel Neighbor Discovery”. In *IEEE INFOCOM*, 2011.
[15] G. Jakllari, W. Luo, and V. Krishnamurthy. “An Integrated Neighbor Discovery and MAC Protocol for Ad Hoc Networks Using Directional Antennas”. In *IEEE TWC*, 2007.
[16] R. Motwani and P. Raghavan. “Randomized Algorithms”. Cambridge University Press, 1995.
[17] P. Erdös and A. Rényi. “On a Classical Problem of Probability Theory”. Magyar Tud. Akad. Mat. Kutato Int. Kozl, 1961.
[18] L. Kong, L. Fu, X. Liu and M. Wu. “Accelerating Initialization for Sensor Networks”. In *Proc. of IEEE GLOBECOM*, 2009.
[19] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu. “Tiny-Sync: Tight Time Synchronization for Wireless Sensor Networks”. In *ACM TOSN*, Jun. 2007.
[20] K. B. Rasmussen, S. Capkun, and M. Cagalj. “SecNav: Secure Broadcast Localization and Time Synchronization in Wireless Networks”. In *Proc. of ACM MobiCom*, 2007.
[21] M. Poturalskim, P. Papadimitratos, and J. Hubaux. “Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility”. In *Proc. of ASIACCS*, 2008.
[22] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hännikäinen, and Timo D. Hämäläinen. “Energy-Efficient Neighbor Discovery Protocol for Mobile Wireless Sensor Networks”. In *Ad Hoc Networks*, 7(1):24-41, 2009.
[23] J. Jeon and A. Ephremides. “Neighbor Discovery in a Wireless Sensor Network: Multipacket Reception Capability and Physical-Layer Signal Processing”. In *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010.
[24] G. Sun, F. Wu, X. Gao, and G. Chen. “PHED: Pre-Handshaking Neighbor Discovery Protocols in Full Duplex Wireless Ad Hoc Networks”. In *Proc. of IEEE GLOBECOM*, 2012.
[25] G. Sun, F. Wu, and G. Chen. “Neighbor Discovery in Low-Duty-Cycle Wireless Sensor Networks with Multipacket Reception”. In *Proc. of IEEE ICPADS*, 2012.
[26] L. You, X. Zhu, and G. Chen. “Neighbor Discovery in Peer-to-Peer Wireless Networks with Multi-Channel MPR Capability”. In *Proc. of IEEE ICC*, 2012.
[27] Y. Wang, G. Pan, and Z. Huang. “Direct Multi-Hop Time Synchronization with Constructive Interference”. In *Proc. of IPSN*, 2012.
[28] R. Stoleru, H. Wu, and H. Chenji. “Secure Neighbor Discovery and Wormhole Localization in Mobile Ad Hoc Networks”. In *Ad Hoc Networks*, (10)7:1179-1190, 2012.