

Novel Based Protocols for Time-Efficient Neighbor Discovery in Wireless Ad Hoc Networks

Divya S J

PG Student

Department of Computer Science and Engineering

Rajeev Institute of Technology

Hassan-573201

Email-id: divyasandesh87@gmail.com

Abstract— In wireless ad hoc networks, an important problem for each node is to discover its neighbor nodes so that the connectivity amongst nodes can be established. Neighbor Discovery (ND) is one of the first steps in configuring and managing a wireless network and crucial step for initializing wireless ad hoc 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.

A novel randomized protocol FRIEND, a pre-handshaking neighbor discovery protocol, to initialize synchronous full duplex wireless ad hoc networks. By introducing a pre-handshaking strategy to help each node be aware of activities of its neighborhood, it significantly reduces the probabilities of generating idle slots and collisions. Moreover, with the development of single channel full duplex communication technology, the protocol further decrease the processing time needed and construct the first full duplex neighbor discovery protocol.

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.

Our key idea is twofold. On one hand, we introduce a prehandshaking 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 subslot. 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.

II. NETWORK MODEL AND ASSUMPTIONS

In this section, we introduce the network model and several 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, 26]).
- 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.

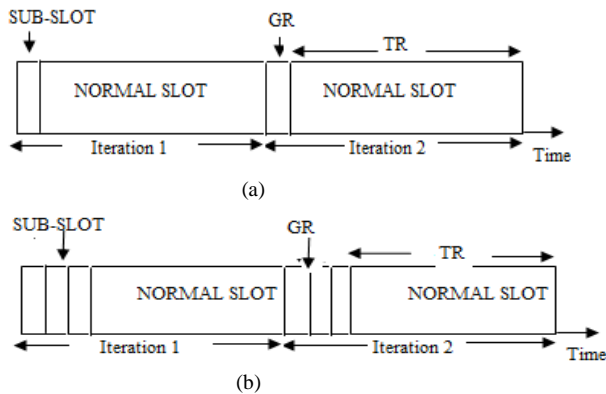


Fig 1: The description of iteration. GR is used for pre-handshaking, and TR is used for transmitting discovery messages. In (a), there is one sub-slot in GR, while in (b) there are multiple sub-slots in GR.

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

For each normal slot we insert a sub-slot before it to perform the pre-handshaking process. We name this combination as 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 probability.

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 prehandshaking 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 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 $A_n = n$.

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

```

The two cases are

- If A sends an M_s (Line 5-10), it implies A hopes to send M_d in TR.
 - 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.
 - 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 $1/2$.
- If A does not send M_s , it implies that A hopes to keep silent in the following TR.
 - 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 $1/A_n$.
 - If A receives M_s . It means that there are nodes intending to transmit and thus A will keep silent.

Algorithm 2 FRIEND-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

```

The two cases of above algorithm are

1. If A sends M_d , A will meanwhile check the existence of other signals
 - 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
 - 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 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 analyse 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 $p_k = 1/k!e$ while $n \rightarrow +\infty$.*

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.

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

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 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*

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

Furthermore, when $n \rightarrow +\infty$

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

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

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

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

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

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

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

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

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

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

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

According to Theorem 1, $P_1 = 0.572$ when $n = 10$ and when $n = 20$, $P_1 > 0.584$. Note that $1/e^2 + 5/4e \approx 0.595$ simply, 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. 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 slot.

Algorithm 3 FRIEND-tGR (Multiple Pre-Handshaking)

- 1: **if** $A_t = t$ **then** FRIEND-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** $A_t = 1$ **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  $1/An$ .
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  $M_s$  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 FRIEND-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
    
```

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

Theorem 2: P_{t+1} is bounded

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

Where P_1 is given by theorem by 1

Proof: If there are $t+1$ sub-slots in GR, we again analyse 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

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

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

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

Similarly for simplicity we get

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

As $n \rightarrow +\infty$

We then point out the upper bound of P_t

Theorem 3 :

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

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:

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

where k stands for the number of nodes to be discovered at the current iteration. We can get $\lim_{k \rightarrow +\infty} P_3 \approx 0.710$. It is quite close to the optimal value so it is feasible to introduce only three sub-slots before TR.

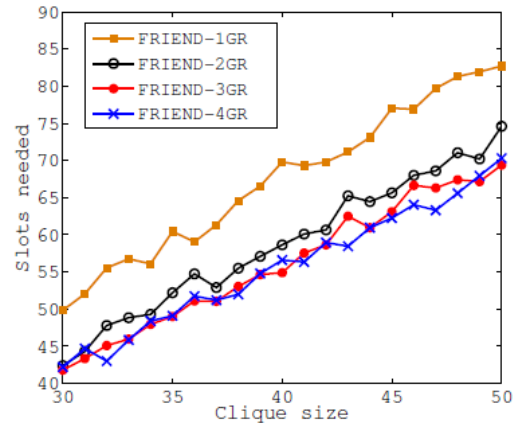


Fig 2: Comparison of FRIEND tGR

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

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

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

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 $(i+1)^{\text{th}}$ node is discovered. Let T_i denote the number of slots of epoch i and T_i is a geometrically distributed variable

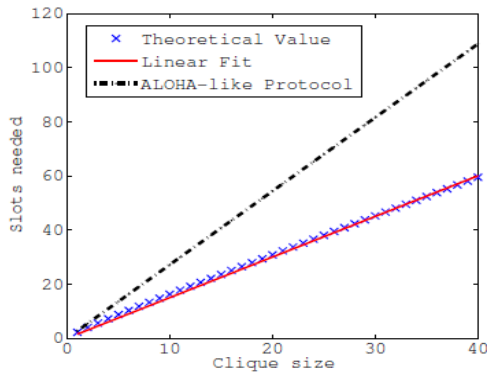


Fig 3: The comparison among the exact values of FRIEND, the linear fitting and the performance of ALOHA like protocol.

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

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

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:

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

On the other hand, the following equation holds:

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

Furthermore, Chernoff bounds point out that:

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

The formal proof of this inequality can be found in [16]. Then 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 $P(T > 3n) < e^{-(n/4)}$. It is clear that $e^{-(n/4)} \rightarrow 0$ for sufficiently large n . So the ND process can be finished in $3n$ slots with high probability

III.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)

```

1: if  $A_f = 1$  then //A has successfully sent  $M_d$ .
2:   A will keep silent in TR (as well as FB) 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:   A will transmit  $M_d$  in TR;
7: else //A does not send  $M_s$ 
8: if A does not receive  $M_s$  during GR then
9:   A will transmit  $M_d$  in TR by probability  $1 - A_n$ ;
10: else // A receives  $M_d$  from other nodes
11:   A will keep silent in TR.
12: end if
13: 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 know 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.

Algorithm 5 HD-FRIEND-TR (Half Duplex)

```

1: if A plans to send  $M_d$  then
2:   A sends  $M_d$ 
3:   A will keep listening in FB.
4: else // A does not plan to send  $M_d$ 
5:   A keeps listening.
6: if // A does not receive  $M_d$  during TR then
7:   Current iteration is invalid.

```

```

8: else if // A receives a single  $M_d$  then
9:   Record the ID in  $M_d$ .
10:  $A_n = A_n / 1$  // A records one of its neighbors.
11: else //There is a collision at A
12:   A will send a feedback signal in FB.
13: end if
14: end if

```

Algorithm 6 HD-FRIEND-FB (Half Duplex)

```

1: if A transmitted in TR then
2:   A keeps listening in FB.
3: if A receives the feedback signal then
4:   Current iteration is invalid.
5: else
6:  $A_f = 1$ .
7: end if
8: else // A received in TR
9: if A plans to send a feedback signal in FB then
10: Send the feedback signal.
11: end if
12: end if

```

IV. 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

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.

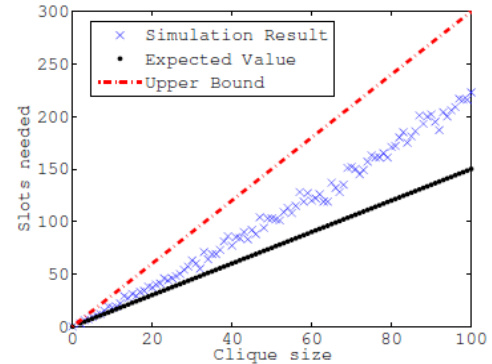


Fig 4: Neighbor Discovery Time in Clique for FRIEND-3GR

2) Comparison in Clique:

Similarly we analyze the performance of FRIEND-3GR 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. 5 shows the comparison between two protocols with different sizes of cliques. We can see that FRIEND-3GR

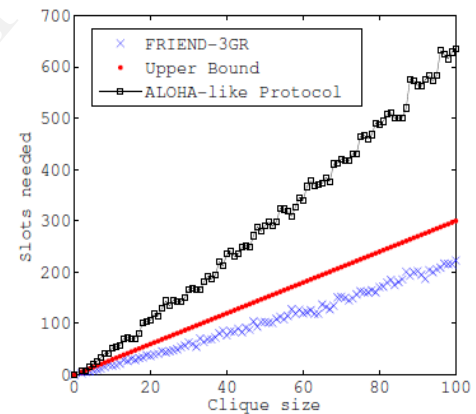


Fig. 5: Comparison of Neighbor Discovery Time in Clique

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 is one sub-slot after the normal slot. Because the duration of subslots is really short, we can still compare two protocols' performance by comparing their consumption of time slots.

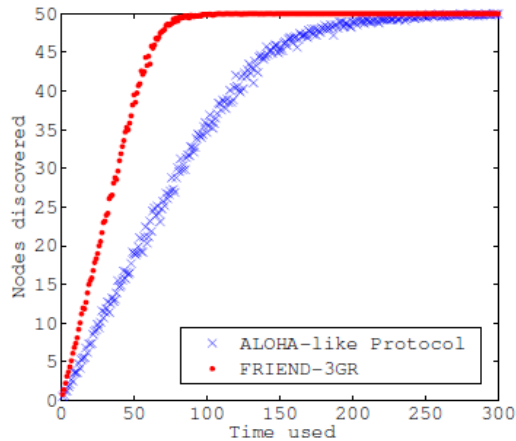


Fig. 6: Comparison of Discovered Node Numbers in Clique

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.

V. 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. Compared with existing deterministic and multi-user detection-based protocols, randomized protocols are most commonly used to conduct ND process in wireless networks. 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.

M. J. McGlynn and S. A. Borbosh [3] have proposed Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks. It uses 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 $1/n$. We address two problem associated with static ad hoc wireless network method of saving energy during a deployment of the nodes and efficient methods of performing adjacent neighbour discovery. To meet these goals we introduce a family of "birthday protocols" which use random independent transmission of discover adjacent nodes various mode of the birthday protocol are used to solve the two problems.

It provides a mathematical model and analysis of two modes of the protocol and is led to a third mode which is probabilistic analogue of the deterministic round robin scheduling algorithm.

This shows that by analysis and simulation that the birthday protocols are a promising tool for saving energy during the deployment of an ad hoc network as well as an efficient and flexible means of having the nodes discover their neighbours.

S. Vasudevan, D. Towsley, D. Goekel, and R. Khalili [4] have presented Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem. Neighbor discovery is one of the first steps in the self-organization of a large wireless

network. Neighbor discovery algorithms can be classified into two categories, viz. randomized or deterministic. In a randomized neighbor discovery algorithm, each node transmits at randomly chosen times and yet discovers all its neighbors by a given time with high probability. In a deterministic neighbor discovery algorithm, each node transmits according to a pre-determined transmission schedule that allows it to discover all its neighbors by a given time with probability one. Guaranteed neighbor discovery typically comes at the cost of increased running time and often requires unrealistic assumptions such as synchronization between nodes and priority knowledge of the number of neighbors. We, therefore, choose to investigate randomized neighbor discovery algorithms in this paper the neighbor discovery problem is non-trivial due to several reasons:

W. Zeng, X. Chen, A. Russell, S. Vasudevan, B. Wang, and W. Wei [5] have proposed Neighbor Discovery in Wireless Networks with Multipacket Reception. Neighbor discovery is one of the first steps in configuring and managing a wireless network. Most existing studies on neighbor discovery assume a single-packet reception model where only a single packet can be received successfully at a receiver. In this paper, motivated by the increasing prevalence of multipacket reception (MPR) technologies such as CDMA and MIMO, we study neighbor discovery in MPR networks that allow multiple packets to be received successfully at a receiver.

More specifically, we design and analyze a series of randomized algorithms for neighbor discovery in MPR networks. We start with a simple Aloha-like algorithm that assumes synchronous node transmissions and the number of neighbors, n , is known. We show that the time for all the nodes to discover their respective neighbors is $\Theta(\ln n)$ in an idealized MPR network that allows an arbitrary number of nodes to transmit simultaneously. In a more realistic scenario, in which no more than k nodes can transmit simultaneously, we show that the time to discover all neighbors is $\Theta(n \ln n k)$.

When a node knows whether its transmission is successful or not (e.g., based on feedbacks from other nodes), we design an adaptive Aloha-like algorithm [8] that dynamically determines the transmission probability for each node, and show that it yields a $\ln n$ improvement over the simple Aloha-like scheme. Last, we extend our schemes to take into account a number of practical considerations, such as lack of knowledge of the number of neighbors and asynchronous algorithm operation, while resulting in only a constant or $\log n$ factor slowdown in algorithm performance. Neighbor discovery is one of the first steps in configuring and managing wireless networks. The information obtained from neighbor discovery viz. the set of nodes that a wireless node can directly communicate with, is needed to support basic functionalities such as medium access and routing. Furthermore, this information is needed by topology control and clustering algorithms to improve the performance and efficiency of the network. Due to its critical importance, neighbor discovery has received a lot of attention, and a number of studies have been devoted to this topic. Most studies however assume a single packet reception (SPR) model,

Zeng et al extended the result of [5] to the multipacket reception situation where no collision occurs if and only if there are no more than $k(k > 2)$ nodes transmitting

simultaneously and proved that the expected time needed to discover all nodes. Ideally, if $k > n$, the discover time is shortened. Similarly, the authors designed protocols for realistic situations in [4] and analyzed the upper bounds respectively.

Keshavarzian and E. Uysal-Biyikoglu (2004) [11] have presented Energy-efficient Link Assessment in Wireless Sensor Networks For energy constrained stationary wireless networks of sensor, selection of links with high quality rate helps to ensure reliable long-term operation During the implementation of a protocol targeting industrial applications of such systems, it was found that it is advantageous to acquire accurate information about the availability and quality of the RF communication links prior to the network topology formation. "Link assessment" as part of the initialization process, accomplishes this task by assessing a sufficient number of packets exchanged between neighbouring nodes. This paper introduces and analyzes two different approaches to link assessment: The first approach is a random nondeterministic scheme that allows for a probabilistic guarantee of collision-free packet exchange. An alternative method is described which employs 'constant-weight codes' and provides a deterministic guarantee of success. In particular, a special class of constant-weight codes, known as optical orthogonal codes, are considered. Since, these codes are cyclically permutable, they make the link assessment process simpler, and therefore they are preferred over other codes. We evaluate the performance of these methods based on their energy consumption, time duration, and implementation complexity.

The link assessment process includes discovery of all nodes and the available links between them, and grading the quality of these links. The latter can be achieved by estimating parameters such as packet success rate or signal strength, which may be determined by assessing a sufficient number of packets exchanged between neighboring nodes [11]. This information can then be used to make routing decisions and form a reliable multi-hop network topology. The paper is organized as follows: In the following section the network scenario and link assessment requirements and assumptions are described. a scheme where each node uses a random pattern is described and analyzed.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a pre-handshaking neighbour 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. Erdős and A. Rényi. "On a Classical Problem of Probability Theory". Magyar Tud. Akad. Mat. Kutató Int. Közl, 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. Poturlasim, 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. Hamäläinen. "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.

IJERT