Analysis on Efficient Neighbor Discovery in Wireless Sensor Networks

#Dr.Ch.Srinivasa Rao[1], Professor, Computer Science Department
#Voruganti sreevani[2],M.Tech, CSE Department, E mail: srivani0597@gmail.com
# Christu Jyothi Institute of Science and Technology,Warangal, T.S ,INDIA

Abstract

The Neighbor Discovery is a process of identifying the nearest node. Neighbor discovery is an important first step in the initialization of a wireless adhoc networks. Fast and efficient discovery of all neighboring nodes by a node new to a neighborhood is critical to the deployment of wireless adhoc networks. Different than the conventional ALOHA-type random access discovery schemes, this paper assumes that all nodes in the neighborhood simultaneously send their unique on-off signatures known to the receive node. In this paper, we analyze several algorithms for neighbor discovery in wireless networks. Starting with the setting of a single-hop wireless network of 'n' nodes, we propose two types of algorithms, one is $\Theta(n \ln n)$ ALOHA like neighbor discovery algorithm when nodes cannot detect collisions, and second an order-optimal $\Theta(n)$ receiver feedback-based algorithm when nodes can detect collisions. Our algorithms do not require nodes to have a priori estimate of the number of neighbors (i.e., n) and also don't require synchronization between nodes. Our algorithms allow nodes to begin execution at different time instants and, to terminate neighbor discovery process till that node discovering all their neighbors. We finally show that receiver feedback can be used to achieve a $\Theta(n)$ running time, even when nodes cannot detect collisions. We then analyze the neighbor discovery problem in a general multi-hop setting. Here, we establish an upper bound of $O(\Delta \ln n)$ on the running time of the ALOHA-like algorithm, where 'Δ' denotes the maximum node degree in the network and 'n' is the total number of nodes. We also establish a lower bound of $\Omega(\Delta + \ln |E|)$ on the running time of any randomized neighbor discovery algorithm, where $|E|$ denotes the total number of edges in the network. Our result thus implies that when $|E| = \Omega(n)$, the ALOHA-like algorithm is at most a factor $\min(\Delta, \ln n)$ worse than the optimal.

1. Introduction:

Wireless ad hoc networks and sensor networks are typically deployed without any communication infrastructure and are required to "configure" themselves upon deployment. For instance, immediately upon deployment, a node has no knowledge of other nodes in it transmission range and needs to discover its neighbors in order to communicate with other nodes in the network.

Neighbor discovery algorithms can be classified into two categories, viz.

1. randomized
2. deterministic.

In randomize neighbor discovery, each node transmits at randomly chosen times and discovers all its neighbors by a given time with high probability . In deterministic neighbor discovery, on the other hand, each node transmits according to a predetermined transmission schedule that allows it to discover all its neighbors by a given time with probability one. In distributed settings, determinism often comes at the expense of increased running time in the particular case of neighbor discovery, typically requires unrealistic assumptions such as node synchronization and a priori knowledge of the number of neighbors

Neighbor discovery is non-trivial due to several reasons:

1) Neighbor discovery needs to cope with collisions. Ideally, a neighbor discovery algorithm needs to minimize the probability of collisions and therefore, the time to discover neighbors.

2) In many practical settings, nodes have no knowledge of the number of neighbors, which makes coping with collisions even harder.

3) When nodes do not have access to a global clock, they need to operate asynchronously and

still be able to discover their neighbors efficiently.
4) In asynchronous systems, nodes can potentially start the neighbor discovery process at different times and consequently, may miss each other's transmissions.
5) Furthermore, when the number of neighbors is unknown, nodes do not know when or how to terminate then neighbor discovery process.

In this paper, we present neighbor discovery algorithms that comprehensively address each of these practical challenges. Unlike existing approaches that make unrealistic assumptions such as a priori knowledge of the number of neighbors or clock synchronization among nodes, we propose neighbor discovery algorithms that:
P1 do not require nodes to have *a priori* knowledge of the number of neighbors,
P2 do not require synchronization among nodes,
P3 allow nodes to begin execution at different time instants, and
P4 enable each node to detect when to terminate the neighbor discovery process.

To the best of our knowledge, our work provides the first solution to the neighbor discovery problem that satisfies each of the properties **P1-P4**. Our approach is to start with a single hop wireless network in which nodes are synchronized and know exactly how many neighbors they have. As we will see, the analysis in such a simplistic setting yields several valuable insights about the neighbor discovery problem. These insights allow us to progressively relax each of the simplifying assumptions leading to a complete and practical solution to the neighbor discovery problem in a multi-hop network setting.

2.Literature survey

In neighbor discovery the authors propose a synchronous ALOHA-like neighbor discovery algorithm. More recently, an asynchronous, randomized neighbor discovery algorithm has been proposed. A feedback based neighbor discovery algorithm designed to operate in fading environments. However, the performance of these algorithms is not well-understood, even in the case of single hop networks. Further, each of these algorithms require *a priori* estimates of node density and do not address the issue of termination of neighbor discovery.

Keshavarzian propose a novel, deterministic neighbor discovery algorithm. However, nodes need to be synchronized with each other and need to know the maximum number of neighbors, n, *a priori*. Furthermore, the neighbor discovery needs n2 time slots to discover all the neighbors.

Neighbor discovery algorithms using a multi-user detection approach have been proposed. However, these algorithms require synchronization between nodes and also require each node to know the signatures of every other node in the network. An interesting approach to neighbor discovery based on group testing has been proposed.

There have been numerous proposals for neighbor discovery when nodes have directional antennas. In general, these solutions propose antenna scanning strategies for efficient neighbor discovery. However, none of these proposals address the practical challenges considered in this paper. Further, the analysis in this paper can be used to provide a more rigorous understanding of the directional neighbor discovery problem.

There exists a large body of literature addressing the RFID tag identification problem, where a designated tag reader needs to determine the IDs of tags in its range. *Prima facie*, the tag identification problem bears resemblance to the neighbor discovery problem. However, there are several crucial differences from the tag identification problem which make neighbor discovery more challenging. First, existing tag identification algorithms typically assume that the number of tags is known *a priori*. Second, the transmissions of the tags can be more easily controlled due to the presence of the tag reader which functions as a *master node* during the discovery process. Finally, the tag identification algorithms do not address the hidden terminal problem present in multi-hop wireless networks.

- We first study the ALOHA-like neighbor discovery algorithm proposed in a single-hop Wireless network of ‘n’ nodes. We show that its analysis reduces to that of the **Coupon Collector’s** *Problem* and that each node discovers all its neighbors in Θ(n ln n)* time (with high probability).
- receiver feedback can be used even when nodes Cannot detect collisions and propose a novel algorithm that achieves a Θ (n) running time.
- We next show that absence of an estimate of the number of neighbors, n, results in slow down of no more than a factor of two, compared to when nodes know *n*.
- We further show that lack of synchronization among nodes results in at most a factor of two slow down in the algorithm performance from the case when nodes are synchronized.

- We then describe how neighbor discovery can be accomplished even when nodes start execution at different time instants. Furthermore, when nodes
- ach node has a path to every other node in the network, our result implies that the ALOHA-like algorithm is at most a factor min(Δ, ln n) worse than the optimal.

3. Problem definition

Let G = (V,E) represent a static multi-hop wireless network, where V denotes the set of nodes and E denotes the set of directed edges in G. Throughout this paper, we assume that |V | = n. We do not impose restrictions on the specific definition used to determine edges between node pairs. However, a common example for the definition of an edge is that an edge exists between nodes i and j if they are within the transmission range of each other. The transmission range might be defined as that distance below which the signal-to noise ratio (SNR) exceeds a fixed threshold γ, allowing node i to transmit at a fixed rate to node j.

- When nodes can detect collisions, we propose an order optimal neighbor discovery algorithm that employs feedback from receiving nodes and allows each node to discover all its neighbors in Θ(n) time (with high probability) interestingly, we find that

  do not know n, we propose a provably correct termination condition that allows each node to terminate neighbor discovery after discovering all its neighbors with high probability.
- Finally, the general multi-hop wireless network setting. Here, we establish an upper bound of O(Δ ln n) for the running time of the ALOHA like algorithm, where ‘Δ’ is the maximum node degree in the network and n denotes the total number of nodes. Under a mild technical assumption, we establish a lower bound of Ω (Δ+ ln |E|) on the running time for any number randomized neighbor discovery algorithm, where |E| denotes the total of edges in the network. When |E| =Ω(n), as is the case when e

In addition, we make the following assumptions about the multi-hop wireless network:

- **Node IDs:** We assume that the nodes have locally unique identifiers i.e., no two neighbors of a Given node have the same identifier. The identifier could be the MAC address of the node, the Location of the node, or a random bit string that is long enough to ensure that it is locally Unique with a sufficiently large probability.
- **Radio Model:** Each node is equipped with a radio transceiver that allows a node to either transmit or receive messages, but not both simultaneously.
- **Collision Model:** When two or more nodes, each of which has a directed edge to a common receiver, transmit concurrently, a collision occurs at the receiver. When a collision occurs, we assume that no partial recovery of packets is possible at the receiving node.

The collision model, although idealized, will later allow us to obtain a deep understanding of the neighbor discovery problem and yields valuable insights for designing practical neighbor discovery algorithms.

- **Symmetric Edges:** Edges between nodes are assumed to be symmetric i.e., if (i, j) ∈ E, then (j, i) ∈ E.

Problem Definition: n nodes are deployed over an area without prior knowledge about the graph G. We say that a node i discovers node j by time t if i receives at least one message from node j by time t. Our goal is to propose efficient algorithms that allow each node i ∈ V to discover all nodes j such that (i, j) ∈ E.

4. Analysis

4.1. Aloha-Like Neighbor Discovery Algorithm:

The ALOHA-like algorithm is a randomized algorithm that operates as follows. In each slot, a node independently transmits a *DISCOVERY* message announcing its ID, with probability p_{xmit} , and listens with probability $1 - p_{xmit}$. A discovery is made in a given slot only if exactly one node transmits in that slot. It has been shown in that the optimal value of p_{xmit} that maximizes the rate of discovery of neighbors is 1/n, where n denotes the clique size. However, the crucial question of how long it takes to discover all the neighbors when nodes transmit with p_{xmit} = 1/n was not addressed in which we proceed to analyze next.

A) Neighbor Discovery As Coupon Collector's Problem:

We first describe how the neighbor discovery problem maps into the classical *Coupon Collector's Problem*. First, we observe that the probability of a successful transmission by node 'i' in a given slot equals

$$p = p_{xmit}\left(1 - p_{xmit}\right)^{n-1} = \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \approx \frac{1}{ne} \quad (1)$$

Note that p is the same for each node i, $1 \leq i \leq n$.

The process of neighbor discovery can be then be treated as a coupon collector's problem in the following manner. Consider a coupon collector C drawing coupons with replacement from an urn consisting of n distinct coupons, each coupon corresponding to a distinct node in the clique. In each slot, C draws one of the n coupons (i.e. discovers a given node) with probability p, and draws no coupon (i.e., detects an idle slot or a collision) with probability 1 − np. It is easy to see that when C collects n distinct coupons, this can be interpreted as each node in the clique having discovered all of its 'n – 1' neighbors.

We are now ready to derive E [W], where W is a random variable that denotes the time required for each node to discover all its neighbors. Let random variable 'W' denote the number of slots needed for each node to discover all its n−1 neighbors. The neighbor discovery process can be thought of as consisting of a sequence of *epochs*, each epoch consisting of one or more slots. Let Wm denote the length of epoch m, $0 \leq m \leq n - 1$, that starts when the mth node is discovered and ends when the m + 1-st node is discovered. Thus, in the mth epoch there are n − m nodes yet to be discovered, each of which has a probability p of being discovered in a given slot. It is easy to see that the epoch length, Wm, is geometrically distributed with parameter (n − m)p. Thus, noting that W = W0 + . . . +Wn−1, we get

$$E[W] = \sum_{m=0}^{n-1} E[W_m] = \frac{1}{(n-m)p} = \frac{1}{p}\sum_{m=1}^{n}\frac{1}{m} \approx neH_n$$

$$E[W] = ne(\ln n + \Theta(1)) = ne\ln n + O(n) = \Theta(n\ln n) \quad (2)$$

where H_n denotes the nth Harmonic number, i.e., $H_n = \ln n + \Theta(1)$. Therefore,

In Appendix A, we obtain an upper bound on the error introduced in p_s due to the approximation in (1) and show that the error goes to 0, for large n. In other words, E[W] → ne, as n → infinity.

B) Sharp Concentration Around Mean

We next show that 'W' is sharply concentrated around its mean, we make use of the Poisson approximation to the binomial distribution. In Appendix B, we derive the sharp concentration result without relying on Poisson approximation. Let $N_i(t)$ be a random variable that denotes the number of successful transmissions by node 'I' in the first 't' slots. It is easy to see that $N_i(t)$ Binomial(t, p). Using the Poisson approximation (assuming large t and small p),

$$P(N_i(t)=k)=\frac{e^{-\lambda}\lambda^k}{k!}$$

Where $\lambda = tp$. Let $E_i(t)$ denote the event that node i is not discovered in t slots. Therefore,

$$P(\mathcal{E}_i(t)) = P(N_i(t)=0) = e^{-tp}$$

Substituting from (1) into the above equation yields

$$P(\mathcal{E}_i(t)) = e^{-\frac{t}{ne}}$$

Therefore,

$$P(\neg\mathcal{E}_i(t)) = 1 - e^{-\frac{t}{ne}}$$

We are interested in the probability that all n nodes are discovered by time t, i.e.

$$.\ P\left[\neg(\cup_{i=1}^{n}\mathcal{E}_i(t))\right].$$

$$P\left[\neg(\cup_{i=1}^{n}\mathcal{E}_i(t))\right] = P\left[\cap_{i=1}^{n}(\neg\mathcal{E}_i(t))\right]$$

We next show that {Ei(t)}s can be treated as an independent set of events.

Lemma 1: For $1 \le i \le n$, and for any set of indices {j1, . . . jk} not containing i,

Proof:

$$P\left[\mathcal{E}_i(t)\middle|\cap_{\ell=1}^{k}\mathcal{E}_{j_\ell}(t)\right] \approx P\left(\mathcal{E}_i(t)\right)$$

$$\begin{aligned} P\left[\mathcal{E}_i(t)\middle|\cap_{\ell=1}^{k}\mathcal{E}_{j_\ell}(t)\right] &= \frac{P\left[\mathcal{E}_i(t)\cap(\cap_{\ell=1}^{k}\mathcal{E}_{j_\ell}(t))\right]}{P\left[\cap(\cap_{\ell=1}^{k}\mathcal{E}_{j_\ell}(t)\right]} \\ &= \frac{(1-(k+1)p)^t}{(1-kp)^t} \end{aligned}$$

Using the approximation 1 + x ≈ ex in the above equation yields

$$P[\mathcal{E}_i(t)|\cap_{\ell=1}^{k}\mathcal{E}_{j_\ell}(t)] \approx \frac{e^{-t(k+1)p}}{e^{-tkp}} = e^{-\frac{t}{ne}} = P(\mathcal{E}_i(t))$$

From Lemma 1 and (3), it follows that

$$P[\neg(\cup_{i=1}^{n}\mathcal{E}_i(t))] = (1-e^{-\frac{t}{ne}})^n \approx e^{-ne^{\frac{-t}{ne}}}$$

Therefore,

$$P(W>t) = 1 - P[\neg(\cup_{i=1}^{n}\mathcal{E}_i(t))]$$

Letting t = ne(ln n + c), for some $c \in \Re$, we conclude

$$P(W>t) = 1 - e^{-ne^{-(\ln n + c)}} = 1 - e^{-e^{-c}}$$

In other words, W = Θ(*n* ln *n*) w.h.p.

4.2 Order-Optimal Neighbor Discovery Using Feedback

We now describe a Θ(n) neighbor discovery algorithm that exploits feedback from receiving nodes. As we will see, feedback to a transmitting node allows it to determine if it has been discovered by other nodes, which then allows it to stop transmitting. We first describe a Θ(n) neighbor discovery algorithm employing feedback when nodes can detect collisions, i.e., each node can distinguish between a collision and an idle slot. Subsequently, we describe how feedback can be exploited to achieve a Θ(n) algorithm even when nodes cannot detect collisions. Throughout this section, we assume that all the n nodes are arranged in a clique.

A)Collision Detection-based Neighbor Discovery

The collision detection-based algorithm the key idea behind the algorithm is as follows. We divide each slot into two sub-slots. Upon successful reception of a *DISCOVERY* message in the first sub-slot, each receiving node transmits bit "1" to the source of the message in the second sub-slot. Collision detection allows the source to detect that its transmission was successfully received by all other nodes, and hence, it "drops out" of neighbor discovery, i.e., stops transmitting.

The remaining nodes (henceforth, called *surviving nodes*) then increase their transmission probabilities in the next slot. As we will see, allowing nodes which have been discovered to drop out and the surviving nodes to increase their transmission probabilities, yields a significant improvement over the ALOHA-like algorithm. Note that since a single bit suffices for the feedback, the second sub-slot is much shorter than the first.

Algorithm 1 Collision Detection-Based ND(i,n)

```
$b \leftarrow 0$ //Number of neighbors discovered by node $i$
$flag \leftarrow 0$ //Has node $i$ been discovered by other nodes?
$NbrList \leftarrow [\,]$ //List of neighbors of node $i$
loop
  $p_{\text{xmit}} \leftarrow 1/(n-b)$
  if (($flag = 0$) and (Bernoulli($p_{\text{xmit}}$) = 1)) then
    Transmit $DISCOVERY(i)$ in first sub-slot
    if energy detected in second sub-slot then
      $flag \leftarrow 1$ //"Drop out"
    end if
  else
    if successful reception in first sub-slot then
      Transmit bit "1" in second sub-slot
      $NbrList[b{+}{+}] \leftarrow DISCOVERY.source$
    end if
  end if
end loop
```

B) Asynchronous collision detection-based algorithm:

We next describe the asynchronous collision detection-based algorithm, which is presented in Algorithm 2. Here, each transmission is of a fixed duration τ and is followed by a *feedback period* of duration σ. Let κ = τ + σ. We further assume that a node in the receive mode can always detect if the wireless channel is *busy* or *idle*.

As shown in below Figure, the timeline for an asynchronous collision detection-based algorithm consists of (i) *Unsuccessful Busy Periods*, during which two or more nodes transmit concurrently; (ii) *Feedback Periods* immediately following message transmissions; (iii) *Idle Periods* during which no transmissions occur; and (iv) *Successful Busy Periods* during which exactly one transmission occurs. As in the synchronous algorithm, the key idea is to allow each node that has been discovered by its neighbors to drop out and let the surviving nodes increase their transmission rate.

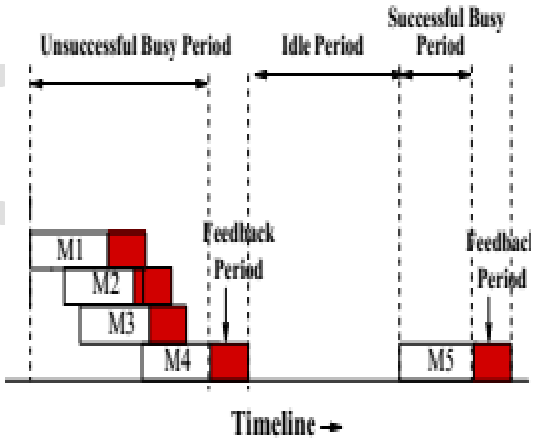

Fig. 1. Timeline of asynchronous collision detection-based algorithm

Algorithm 2 Asynch. Collision Detection-based ND(i,n)

```
$b \leftarrow 0$ //Number of neighbors discovered by node $i$
$flag \leftarrow 0$ //Has node $i$ been discovered by other nodes?
$NbrList \leftarrow [\,]$ //List of neighbors of node $i$
loop
  $\lambda \leftarrow 1/(2\kappa(n-b))$
  Alarm($TIMEOUT$,Exp($1/\lambda$)) //Set Listen duration
  try:
    loop
      Listen for DISCOVERY messages
      if collision detected then
        Transmit bit "1" at the end of busy period
      else
        $NbrList[b{+}{+}] \leftarrow DISCOVERY.source$
      end if
    end loop
  end try
  catch TIMEOUT:
    if ($flag = 0$) then
      Transmit $DISCOVERY(i)$
      if feedback period idle then
        $flag \leftarrow 1$ //"Drop out"
      end if
    end if
  end catch
end loop
```

4.3)The Multi-Hop Network Case:

The neighbor discovery problem has been in the context of single hop networks. We are now ready to consider the more general multi-hop network setting. In this we derive lower and upper bounds for the neighbor discovery problem in the network case.

A).ALOHA-like Neighbor Discovery: Upper Bound Analysis

We begin by analyzing the ALOHA-like neighbor discovery algorithm. Recall that the wireless network is represented by a graph G = (V,E), where |V | = n.

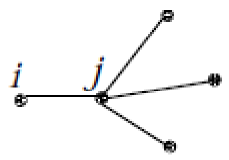

Fig. 2. Node i and node j with different number of neighbors.

B). Lower Bound Analysis:

In this we establish a lower bound for the running time of any randomized neighbor discovery algorithm. More formally, we show that, given an arbitrary input graph G = (V,E), any randomized algorithm requires Ω(Δ + ln |E|) time w.h.p., provided |E| → ∞ as n → ∞.

Since a node needs to receive at least one transmission from a neighboring node in order to discover the neighbor any distributed algorithm, randomized or not, has a running time of at least Ω(Δ). Thus, it suffices to show that when Δ = *o*(ln |E|)‡, any randomized neighbor discovery algorithm has a running time of Ω(ln |E|) w.h.p. This implies a lower bound of Ω (Δ+ ln |E|).

In establishing the lower bound, we assume nodes can detect collisions. Since collision detection can only help reduce the discovery time, the lower bound also applies to algorithms which assume that nodes cannot detect collisions.

1. **Uniform Randomized Algorithms:** We initially establish the lower bound for the class of *uniform randomized algorithms* i.e. each node uses the same algorithm. we assume that nodes do not know the IDs of their neighbors.
2. **Non-uniform Randomized Algorithms:** We next show that the lower bound established in the previous section also applies for the class of *non-uniform randomized algorithms*, where nodes may use different algorithms.

5. Conclusion

In this paper, we have presented efficient neighbor discovery algorithms for wireless networks that comprehensively address various practical limitations of the earlier approaches. Our neighbor discovery algorithms do not require estimates of node density and allow asynchronous operation. Furthermore, our algorithms allow nodes to begin execution at different times and also allow nodes to detect the termination of the neighbor discovery phase. A number of avenues for future work remain open. Our analysis shows a gap between the lower and upper bounds on the running time for neighbor discovery in the network case. Clearly, the quest for an order-optimal neighbor discovery algorithm remains an intriguing prospect. Of particular interest is the question of whether the feedback-based algorithms, which are order-optimal in the single-hop case, can be extended to the multi-hop network setting while outperforming the ALOHA like algorithm. Another direction of interest is the extension of the various algorithms and the analysis presented in this to wireless channel models that incorporate phenomena such as fading and shadowing.

7. References:

1. D. Angelosante, E. Biglieri, and M. Lops. Neighbor discovery in wireless networks: a multiuser-detection approach. In *Information Theory and Applications Workshop*, 2007.
2. S. A. Borbash, A. Ephremides, and M. J. McGlynn. An asynchronous neighbor discovery algorithm for wireless sensor networks. *Ad Hoc Networks*, 2007.
3. R. Khalili, D. Goeckel, D. Towsley, and A. Swami. Neighbor discovery with reception status feedback to transmitters. In *IEEE INFOCOM*, 2010.
4. K. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In *IASTED Distributed Sensor Networks*, 2008.
5. Z. Zhang and B. Li. Neighbor discovery in mobile ad hoc self configuring networks with directional antennas: algorithms and comparisons. *IEEE Transactions on Wireless Communications*, 2008.
6. D. Simplot-Ryl, I. Stojmenovic, A. Micic, and A. Nayak. A hybrid randomized protocol for RFID tag identification. *Sensor Reviews*, 2006.