On Neighbor Discovery in Wireless Networks With Directional Antennas

Sudarshan Vasudevan, Jim Kurose, Don Towsley

Department of Computer Science,
University of Massachusetts,
Amherst, MA 01003
{svasu,kurose,towsley}@cs.umass.edu *

UMass Computer Science Techincal Report 04-53

Abstract

We consider the problem of neighbor discovery in static wireless ad hoc networks with directional antennas. We propose several probabilistic algorithms in which nodes perform random, independent transmissions to discover their one-hop neighbors. Our neighbor discovery algorithms are classified into two groups, viz. *Direct-Discovery Algorithms* in which nodes discover their neighbors only upon receiving a transmission from their neighbors and *Gossip-Based Algorithms* in which nodes gossip about their neighbors' location information to enable faster discovery. We first consider the operation of these algorithms in a slotted, synchronous system and mathematically derive their optimal parameter settings. We show how to extend these algorithms for an asynchronous system and describe their optimal design. Analysis and simulation of the algorithms show that nodes discover their neighbors much faster using gossip-based algorithms than using direct-discovery algorithms. Furthermore, the performance of gossip-based algorithms is insensitive to an increase in node density.

The efficiency of a neighbor discovery algorithm also depends on the choice of antenna beamwidth. We discuss in detail how the choice of beamwidth impacts the performance of the discovery process and provide insights into how nodes can configure their beamwidths.

1 Introduction

Wireless ad-hoc networks, particularly, static ad-hoc networks such as sensor networks and community mesh networks, have generated tremendous amount of interest recently. Sensor networks have applications such as surveillance and tracking [16], environmental observation [2], habitat monitoring [7], and health monitoring [13], while mesh networks [8] enable nodes to connect home networks together forming a community ad-hoc network. A characteristic requirement of these ad-hoc networks is that they be "self-configuring", i.e., that a large number of wireless nodes organize themselves to efficiently perform the tasks required by the application after they have been deployed. Examples of self-configuration include construction of routing paths, clustering, and formation of minimum weight spanning trees. "Self-configuring" ad-hoc networks are very attractive since they reduce the cost of installation and allow for building large scale systems.

*This work was supported by the National Science Foundation under the grant EEC-0313747 001.

In this paper, we consider an aspect of self-configuration in wireless ad-hoc networks referred to as **neighbor discovery**. After nodes are deployed, they need to discover their one-hop neighbors. Knowledge of one-hop neighbors is essential for almost all routing protocols, medium-access control protocols and several other topology-control algorithms such as construction of minimum-energy spanning trees. Neighbor discovery is, therefore, a crucial first step in the process of self-organization of a wireless ad-hoc network. Ideally, nodes should discover their neighbors as quickly as possible as rapid discovery of neighbors often translates into energy efficiency, since nodes have to spend less energy discovering neighbors. Also, rapid discovery allows for other protocols (such as topology control, medium access and routing protocols) to quickly start their execution. We emphasize that the focus of this paper is on neighbor discovery alone and not how the discovered neighbor information is used by topology control algorithms [11, 6, 15], medium access protocols [3, 1] and routing algorithms [4].

There has been earlier work on neighbor discovery in wireless networks [9, 5]. In these papers, the authors present algorithms for neighbor discovery in wireless networks where nodes have omni-directional antennas and operate in a synchronous fashion. Our work differs from the existing work in two important ways. First, we address the problem of neighbor discovery when nodes have directional antennas. Second, we also consider the case in which the neighbor discovery algorithms operate asynchronously. Directional antennas offer many advantages over omni-directional antennas such as increased spatial reuse, increased transmission range and increased capacity. However, discovery of neighbors becomes harder since nodes must control the direction of their antennas in order to transmit or receive data packets from their neighbors. Thus the efficiency of neighbor discovery algorithms using directional antennas depends not only on how often nodes transmit and receive but also on antenna properties such as their direction and beamwidth. In this paper, we propose several probabilistic neighbor discovery algorithms in which nodes perform random independent transmissions in different directions to discover their one hop neighbors. The goal of these neighbor discovery algorithms is to maximize the probability of discovery of neighbors within a given amount of time. We consider both synchronous and asynchronous algorithms and their optimal design. While the algorithm in [9] can be made asynchronous in the manner described in this paper, synchronization is a requirement for the algorithm described in [5].

In this paper, we present several probabilistic neighbor discovery algorithms, both synchronous and asynchronous. Our neighbor discovery algorithms can be classified into two groups, viz. *Direct-Discovery Algorithms* in which a nodes discovers its neighbor only when it successfully hears a transmission from that neighbor and *Gossip-Based Discovery Algorithms* in which nodes gossip about each others' location information to speed up discovery. Some of the important contributions of our work are:

1. A simple mathematical model to derive the optimal parameter settings for synchronous direct-discovery and gossip-based algorithms.

2. A simulation-based performance comparison of the gossip-based and the direct-discovery algorithms, demonstrating that nodes discover their neighbors significantly faster using the gossip-based algorithm than using the direct-discovery algorithm. Interestingly, we also see that while the performance of direct-discovery algorithm degrades as node density increases, the gossip-based algorithm remains insensitive to an increase in node density.

3. A detailed study of how the performance of the gossip-based algorithm varies with the fraction of nodes with location information. An interesting feature of our gossip-based algorithm is that it can operate even when only a fraction of nodes have location information. Simulations show that the performance of the gossip-based algorithm degrades gracefully to that of the direct discovery algorithm as the fraction of nodes with location information decreases.

4. Extension of our synchronous discovery algorithms to their asynchronous counterparts and derivation of their optimal parameter settings.

5. A discussion of how nodes should configure their beamwidths in order to maximize the number of discovered neighbors in a given amount of time.

The rest of the paper is structured as follows. In Section 2, we describe our model and list various assumptions. Section 3 describes a direct-discovery algorithm and its analysis. We next present the gossip-based algorithm in Section 4. In Section 5, we extend the discovery algorithms to operate asynchronously. In Section 6, we discuss how the choice of beamwidth affects neighbor discovery. Finally, we conclude in Section 7.

2 Model and Assumptions

We make the following assumptions about the wireless network:

1. **Unique Node IDs:** Each node is distinguishable by a unique identifier such as a MAC address.

2. **Static Nodes:** Nodes are assumed to be static i.e., non-mobile.

3. **Radio model:** Each node is equipped with a radio transceiver that enables the node to transmit and receive signals. At any given time, a node can either be in transmit or receive mode, but not both. All nodes have a fixed transmission power.

4. **Antenna Model:** Each node is equipped with a directional antenna with beamwidth θ $(0 < \theta \leq 2\pi)$. We assume that each antenna is steerable, i.e., each node can point its antenna in any desired direction. Nodes can use their antennas for directional transmission and/or directional reception.

5. **Antenna Pattern:** We approximate the antenna pattern as a **circular sector** with angle θ and radius equal to the transmission/reception range. In reality, the directional antenna pattern consists of a mainlobe which is the direction of maximum radiation or reception and several smaller backlobes arising due to inefficiencies in antenna design. For simplicity, we ignore backlobes from our discussion in this paper.

6. **Synchronous and Asynchronous Algorithms:** We propose both synchronous and asynchronous algorithms for neighbor discovery. For a synchronous discovery algorithm, we assume that time is slotted and nodes are perfectly synchronized on time slots. The length of each time slot is equal to the duration of a packet. In case of an asynchronous algorithm, nodes need not be synchronized.

7. **Collision Model:** Collisions occur if a node simultaneously receives transmissions from two or more of its neighbors. While a receiving node can detect collisions, the transmitting node cannot. No partial recovery of the collided packets is possible.

The goal of the neighbor discovery process is to have each node in the network rapidly discover all of its one-hop neighbors.

3 Direct-Discovery Algorithms

The neighbor discovery algorithms described in this section are called *Direct-Discovery Algorithms*. In these algorithms, a node must receive at least one successful transmission from its neighbor in order for it to discover that neighbor. When a node successfully receives a transmission from a neighbor, it records the Angle-Of-Arrival (AOA) information of the received signal along with the identity of the neighbor. Alternately, if nodes are capable of determining their location using GPS or any other locating mechanism, then the location information associated with the neighbor is recorded. The AOA or location information of the neighbors is essential for future directional transmission/reception to the neighboring nodes, once discovery is completed. We emphasize that requiring nodes to be capable of providing either AOA or location information is *not* a constraint imposed by our neighbor discovery algorithms, since this information is required for future directional transmission/reception.

We first describe the synchronous neighbor discovery algorithms. We assume that time is slotted with each slot of duration equal to the length of a packet. The discovery algorithms are probabilistic in nature, i.e., in each time slot a node transmits with a certain transmission probability p_t and listens with probability $1 - p_t$.

We first describe a direct-discovery algorithm when nodes have directional transmitter with beamwidth, θ and an omni-directional receiver. Subsequently, we discuss extensions to other antenna models.

3.1 Directional Transmission and Omni-directional Reception

3.1.1 Algorithm Operation

All nodes execute the following direct-discovery algorithm. At the beginning of each time slot, a node transmits in a random direction with transmission probability p_t and listens for transmissions with probability $1 - p_t$. The goal is to find the optimal p_t that maximizes the probability of the node discovering its neighbors within a given amount of time.

3.1.2 Analysis

For simplicity, we assume a clique of k nodes, i.e., k wireless nodes within transmission range of each other. Consider a random node (call it node i) which has k-1 neighbors. We know from our collision model that if only one station transmits to node i in a given time slot, then i discovers that node. If two or more stations transmit simultaneously, i does not discover any of the nodes.

Under the assumption that transmission events are independent, the probability that node i discovers node j in a given time slot, $j = 1, 2, \ldots, k-1$ is:

$$p_{i,j} = \frac{\theta}{2\pi} p_t (1 - \frac{\theta}{2\pi} p_t)^{k-2} (1 - p_t)$$

where p_t is the probability that j transmits in the time slot and $\frac{\theta}{2\pi}$ is the probability that j's transmit beam covers i.

The probability that node i discovers node j within t time slots is then given by:

$$P_{i,j}(t) = 1 - (1 - p_{i,j})^t \tag{1}$$

We are interested in maximizing the probability of node i discovering a neighbor j within t time slots by a proper choice of p_t. Since all nodes are in a clique, the probability of node i discovering any other neighbor is also maximized by the same choice of p_t. In fact, the probability of any node in the clique discovering its neighbor within time t is exactly as (1). Hence, the optimal p_t is the same for all the nodes.

From (1), we note that maximizing $P_{i,j}(t)$ is equivalent to maximizing the probability of node i discovering node j in a given time slot, $p_{i,j}$ by a proper choice of p_t. On differentiating (1) and equating it to 0, we find the optimal p_t to be :

$$p_t = \frac{(2 + (k-1)\frac{\theta}{2\pi}) - \sqrt{(2 + (k-1)\frac{\theta}{2\pi})^2 - 4k\frac{\theta}{2\pi}}}{\frac{k\theta}{\pi}} \tag{2}$$

For large k,

$$p_t \approx \frac{2\pi}{k\theta}$$

Intuitively, the probability of discovering a neighbor is maximized when nodes transmit in a probabilistic round-robin fashion i.e., each node transmits once every $\frac{\theta k}{2\pi}$ time slots. The multiplicative factor of $\frac{2\pi}{\theta}$ in the expression for the optimal p_t is due to the spatial reuse offered by using a directional antenna of beamwidth θ.

An alternative objective function to the one considered in this section is to maximize the fraction, F, of neighbors discovered within a given amount of time. We will see in Section 3.1.4 that the optimal p_t derived in (2) also maximizes F.

3.1.3 Practical Considerations

In practice, a node will not have exact information about the number of neighbors it has. How then should nodes choose their transmission probability, p_t? p_t could be chosen based on some estimate of the number of neighbors. One such estimate is the expected number of neighbors $\overline{k}$ of a node. This estimate is easily available since the density of the network can be "wired" into the nodes prior to deployment. The expected number of neighbors of a node is given by: $\overline{k} = \gamma\pi r^2$, where γ is the density of the wireless network measured in nodes per unit area and r is the transmission radius of the node. When $k > \overline{k}$, we overestimate the transmission probability leading to more collisions, while underestimation occurs when $k < \overline{k}$ thereby under-utilizing the channel and missing opportunities to discover neighbors. In Figure 1, we plot $p_{i,j}$ as a function of the estimation error $\overline{k} - k$, when $k = 20$ and $\theta = 60^o$. We observe that $p_{i,j}$ is maximized when there is no estimation error and decreases as the error increases either due

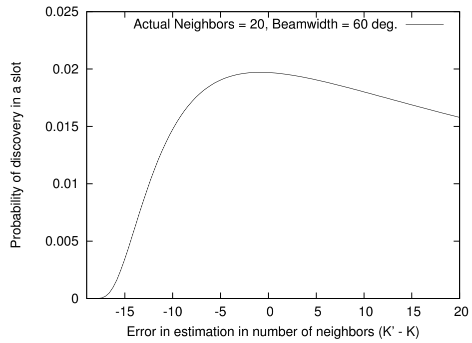


Figure 1: Effect of estimation error on discovery probability

to underestimation or overestimation. We also see that an overestimation of the number of neighbors results in a larger $p_{i,j}$ than an underestimate (a similar observation was made in [9]). Similar behavior was observed for other choices of k and θ. The key observation, however, is that discovery can still be achieved even if there is an error in estimating the number of neighbors and that performance degrades gracefully with increasing error.

3.1.4 Validation of Model

In deriving the optimal p_t in Section 3.1.2, we made two simplifying assumptions about the discovery process. First, our analysis was based on the assumption that all nodes belong to a single clique. In reality, network topologies are arbitrary and multi-hop. Second, our model ignores the spatial correlation among nodes in calculating the probability of discovery. Our model asumes that the probability of a node i discovering another node j in a time slot is independent of another node k discovering node j in the same time slot. In order to validate these assumptions, we compute the expected fraction of neighbors discovered by a node within time t using our model assumptions and compare it with the results obtained using simulation. A node i, which has $k-1$ neighbors, discovers m of them in t time slots in one of the following two ways:

1. i discovers $m-1$ neighbors in the first $t-1$ time slots and another one of the remaining $k-m$ neighbors in the t^{th} time slot; or

2. i discovers m neighbors in the first $t-1$ time slots and none of the remaining $k-m-1$ neighbors in the t^{th} time slot

Hence, the probability that node i discovers m neighbors within t slots, denoted by $P_i(m,t)$ is given by the following recurrence:

$$P_i(m,t) = P_i(m-1,t-1)(k-m)p_s + P_i(m,t-1)[1-(k-m-1)p_s]$$

where p_s is the probability of a successful transmission from a given neighbor to node i in a given time slot and is given by:

$$p_s = \frac{\theta p_t}{2\pi}(1-\frac{\theta p_t}{2\pi})^{k-2}(1-p_t)$$

The boundary conditions of the recurrence relation are:

$$P_i(m,t) = 0, m > t$$

$$P_i(0,0) = 1$$

The expected fraction of neighbors discovered by node i within time t is given by:

$$F = \frac{\sum_{n=1}^{min(t,k-1)} nP_i(n,t)}{k-1} \tag{3}$$

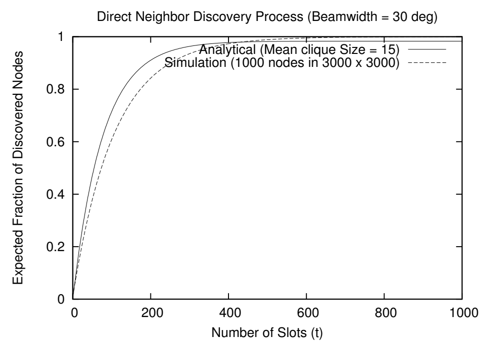


Figure 2: Validation of Analysis

It is difficult to obtain a closed form expression for the fraction F and hence, we solve equation (3) numerically. We find that the value of p_t that maximizes fraction F is exactly as given in equation (2). This is not surprising since the value of p_t given in equation (2) maximizes the probability of a successful transmission in a time slot and hence the probability of successfully discovering a neighbor within a given amount of time. Intuitively, this p_t should also maximize the expected number of neighbors discovered within a given amount of time.

In order to validate our model assumptions, we compare the results obtained using equation (3) with simulation results. The comparison is shown in Figure 2. The simulation scenario consists of 1000 nodes each with a transmission range (r) of 200 m and a beamwidth of 30^o uniformly distributed in a square with area $9 \times 10^6 m^2$. The node density is $\gamma = \frac{1000}{9 \times 10^6}$ nodes/m^2. Each node thus has on average $\overline{k} = \gamma \pi r^2 = 14$ neighbors. The transmission probability, p_t, is obtained from equation (2) by substituting $k = 15$, for both our simulation and the analytical model in equation (3). From Figure 2, we observe a good match between our analytical results and simulations. Similarly good matches were obtained for other values of γ. This validates the model assumptions used to obtain the optimal p_t.

3.2 Other Antenna Models

We now consider the direct-discovery algorithms using two other antenna models:

1. Directional Transmitter and Directional Receiver

2. Omni-directional Transmitter and Directional Receiver

3.2.1 Directional Transmission and Directional Reception

The direct-discovery algorithm is quite similar to that described in Section 3.1. The only difference in this case is that nodes have directional receivers and in every time slot a node listens with probability $1 - p_t$ by pointing its receive beam in a random direction.

As shown in Figure 3, a node *A* successfully discovers another node *B*, only if the transmit and receive beams of *B* and *A* point to each other and no other node within *A*'s reception beam transmits to *A*.

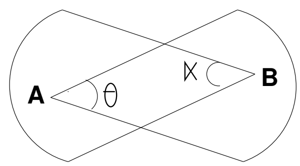


Figure 3: Directional Transmission and Reception

The probability that node i discovers a neighbor j in a time slot, j = 1,2,. . .,k - 1 is

$$p_{i,j} = \frac{\theta\alpha}{4\pi^2} p_t (1 - \frac{\theta\alpha}{4\pi^2} p_t)^{k-2} (1 - p_t)$$

Proceeding in the same manner as in Section 3.1, we obtain the optimal p_t to be:

$$p_t = \frac{(2 + (k-1)\frac{\theta\alpha}{4\pi^2}) - \sqrt{(2 + (k-1)\frac{\theta\alpha}{4\pi^2})^2 - 4k\frac{\theta\alpha}{4\pi^2}}}{\frac{2k\theta\alpha}{4\pi^2}}$$

For large k,

$$\therefore p_t \approx \frac{4\pi^2}{k\theta\alpha}$$

3.2.2 Omnidirectional Transmission and Directional Reception

In this antenna model, a node i successfully receives a transmission from a neighboring station j only if i's receive beam points to station j and no other station within i's receive beam transmits in the same time slot.

The probability that node i discovers a neighbor j, j = 1,2,. . .,k - 1 is same as the case of directional transmission and omni-directional reception:

$$p_{i,j} = \frac{\alpha}{2\pi} p_t (1 - \frac{\alpha}{2\pi} p_t)^{k-2} (1 - p_t)$$

The optimal p_t is therefore the same as derived in Section 3.1:

$$p_t = \frac{(2 + (k-1)\frac{\alpha}{2\pi}) - \sqrt{(2 + (k-1)\frac{\alpha}{2\pi})^2 - 4k\frac{\alpha}{2\pi}}}{\frac{k\alpha}{\pi}}$$

which for large k can be approximated as

$$p_t \approx \frac{2\pi}{k\alpha}$$

4 A Gossip-Based Neighbor Discovery Algorithm

The neighbor discovery algorithms described thus far belong to the family of *Direct-Discovery Algorithms*. In this section, we explore *Gossip-Based Discovery Algorithms* in which a node exploits location information and can learn indirectly about its neighbor's existence through its interaction with other neighbors.

4.1 Algorithm Operation

We assume that each node knows its location using a locating device such as GPS. The gossip-based algorithm operates almost exactly the same as the direct-discovery algorithm described in Section 3. Consider a node i. In each slot, i chooses a random direction and transmits with probability p_t with a fixed beamwidth, θ. The only difference is that i includes in its message the list of neighbors that it has discovered so far and their locations. When a node, m, receives a transmission from its neighbor i, m not only discovers node i but but also any information about nodes that i has discovered (including nodes that probably are neighbors of m).

We refer to our algorithm as a gossip-based neighbor discovery algorithm since it is analogous to gossip-style algorithms. Before the start of the algorithm, each node has a unique gossip viz. its identifier and location, which it wishes to communicate to all of its neighbors. During the execution of the algorithm, each node maintains a table of the gossip (discovered neighbors and their locations) it has accumulated so far. At each round, each node transmits this table with probability p_t in a random direction (with beamwidth θ).

The gossip-based discovery algorithm differs from the direct-discovery algorithm in two crucial ways. First, it allows a node to discover its neighbors indirectly (i.e., through some other neighbor). Second, it allows a node to discover multiple neighbors in one step. We will soon see that these differences help nodes discover their neighbors significantly faster than with a direct-discovery algorithm.

4.2 Analysis

Similar to our earlier analyses, our goal is to find the transmission probability that maximizes the probability of discovering a neighbor. For our analysis, we assume that nodes have a directional transmitter and omni-directional receiver. Extension of our anlaysis to other antenna models is straightforward and is not considered here.

Consider a node i that has k-1 neighbors numbered $1, 2, \ldots, k-1$. Our goal is to find the optimal p_t that maximizes the probability of discovering a given neighbor within time t.

Let $P_{i,j}(t)$ denote the probability that node i "discovers" node j within t time slots. Node i can "discover" node j in one of the following two ways :

1. *directly* by successfully receiving a transmission from node j. Let $D_{i,j}(t)$ denote the probability that node i successfully receives one or more transmission from node j in t time slots.

2. *indirectly* by receiving a transmission from a node m which has itself discovered j at an earlier time slot either directly or indirectly. Let $I_{i,j}(t)$ denote the probability that node i discovers node j indirectly by time t.

We derive $P_{i,j}(t)$ based on the assumption that the probability of indirect discovery between a given pair of nodes is independent of the probability of direct discovery between any other pair of nodes.

$$P_{i,j}(t) = D_{i,j}(t) + (1 - D_{i,j}(t))I_{i,j}(t) \tag{4}$$

$$D_{i,j}(t) = 1 - (1 - p_{i,j})^t$$

where $p_{i,j}$ denotes the probability of a successful transmission from node j to node i in any time slot and is expressed as:

$$p_{i,j} = \frac{\theta}{2\pi}p_t(1 - \frac{\theta}{2\pi}p_t)^{k-2}(1 - p_t)$$

Since $p_{i,j}$s are the same for all node pairs, i and j, we simply denote the probability of a successful transmission from a given node to another node as p_s.

$I_{i,j}(t)$ is defined by the following recurrence:

$$I_{i,j}(t) = I_{i,j}(t-1) + (1 - I_{i,j}(t-1))A_{i,j}(\{i\}, t) \tag{5}$$

where $A_{i,j}(\{i\}, t)$ denotes the probability that i discovers j indirectly in the t^{th} slot, given that i has not discovered j indirectly by $t-1$ slots. In general, for a set S of nodes, $A_{i,j}(S, t)$ is defined as:

$$A_{i,j}(S,t) = \textstyle\sum_{m \neq i,j} D_{m,j}(t-1)p_s +$$

$$\sum_{m \notin S}(1 - D_{m,j}(t-1))I_{m,j}(S, t-1)p_s \tag{6}$$

Thus, $A_{i,j}(S, t)$ denotes the probability that i discovers j indirectly in the t^{th} slot given that none of the nodes in set S has discovered j indirectly by $t-1$ slots.

$I_{i,j}(S, t)$, in turn, is given by the following recurrence:

$$I_{i,j}(S,t) = I_{i,j}(S,t-1) + (1 - I_{i,j}(S,t-1))A_{i,j}(S \cup \{i\}, t) \tag{7}$$

The boundary conditions of the recurrence are given by:

$$I_{i,j}(1) = 0;\ I_{i,j}(S,1) = 0\ \forall S$$

From (5), we see that i discovers j indirectly by time t either by discovering j indirectly:

1. by time $t-1$, or

2. exactly in the t^{th} time slot, given that it did not discover j indirectly by time $t-1$. This happens if a node m

other than i or j has discovered j, directly or indirectly, by time $t-1$ and successfully transmits to node i in the t^{th} time slot with probability p_s. $A_{i,j}(\{i\}, t)$ denotes the probability of this event.

Note that, for a given t, $I_{i,j}(S,t)$ is smaller than $I_{i,j}(t)$, as none of the nodes in S has discovered j indirectly by time t, which reduces the probability that i discovers j indirectly through any of the nodes in S by time t. We also note that $I_{i,j}(S,t)$ keeps becoming smaller as t becomes smaller. This is because, from (7), we see that the set S of nodes that have not discovered j indirectly, keeps growing as t becomes smaller. Hence, both $A_{i,j}(S,t)$ and $I_{m,j}(S,t)$ become smaller.

We can solve (5) numerically for different values of p_t and t and obtain $P_{i,j}(t)$ from equation (4). The optimal p_t is the transmission probability that maximizes $P_{i,j}(t)$. In Figure 4, we plot the optimal value of p_t for different

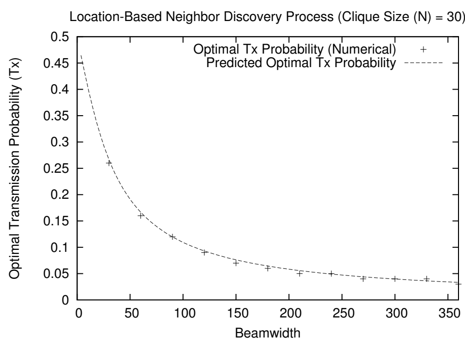


Figure 4: Optimal transmission probability for gossip-based algorithm

antenna beamwidths by solving equation (4) numerically. We observe from the figure that the optimal value of p_t obtained numerically matches almost exactly the optimal p_t we obtained for the direct discovery algorithm (see eq. (2), Section 3.1) which is given by the following equation:

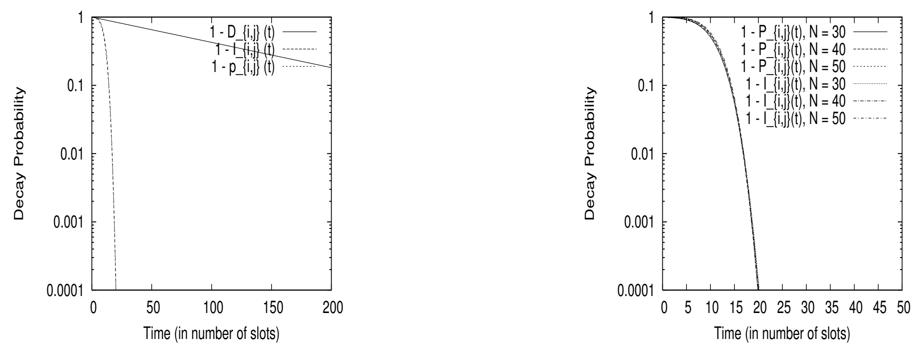
$$p_t = \frac{(2 + (k-1)\frac{\theta}{2\pi}) - \sqrt{(2 + (k-1)\frac{\theta}{2\pi})^2 - 4k\frac{\theta}{2\pi}}}{\frac{k\theta}{\pi}}$$

This is not surprising since, intuitively, the probability $P_{i,j}(t)$ for both algorithms is maximized when the probability of a successful transmission in a time slot, p_s, is maximized. Since p_s is the same for both the algorithms, the optimal p_t should also be the same.

In Figure 5(a), we plot the decay probability $1 - P_{i,j}(t)$, i.e., the probability that node i does not discover a given neighbor j within time t. This probability is computed numerically from (4) using the optimal value of p_t. For the graph shown in figure 5(a), we choose $k = 30$ and $\theta = 30^o$. The line labeled $1 - D_{i,j}(t)$ is the probability that node i does not discover node j within time t with the direct-discovery algorithm.

From Figure 5(a) we observe that the probability of not discovering a neighbor decays much faster for the gossip-based algorithm than for the direct-discovery algorithm. We also observe that the indirect discovery probability $I_{i,j}(t)$ dominates the direct discovery probability $D_{i,j}(t)$ since the lines $1 - I_{i,j}(t)$ and $1 - P_{i,j}(t)$ almost perfectly overlap each other. This graph suggests the potential benefit of indirect discovery in speeding up neighbor discovery. Given the promise of indirect discovery, we proceed to explore the following questions :

1. what is the fraction of neighbors discovered by a node in time t?



(a) Decay Probabilities of the Discovery Algorithms

(b) Effect of no. of Neighbors on Decay Probability

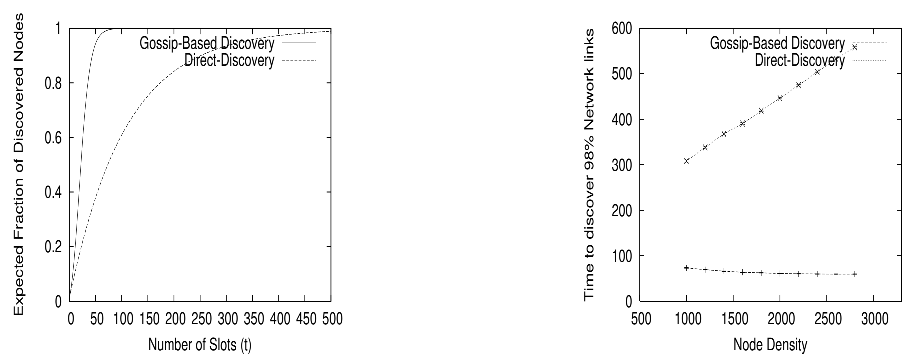
Figure 5: Decay Probabilities of Neighbor Discovery Algorithms

2. what is the time until a large fraction (say 98 %) of the entire topology is discovered?

Since it is difficult to derive answers to these questions analytically, we resort to simulation to answer them.

4.3 Simulation Results

In Figure 6(a) we plot the fraction of neighbors discovered by a node as a function of time for the gossip-based algorithm as well as the direct-discovery algorithm. The simulation scenario consists of 2000 nodes uniformly distributed in a square with area $9 \times 10^6 m^2$. Nodes have a transmission range of 200m and have an antenna beamwidth of 30 degrees. For the simulation results shown here (and in the rest of the paper), each node only has information about node density based on which it calculates the expected number of its neighbors. The transmission probability p_t is then calculated based on this estimate. The results are averaged over 20 runs each corresponding to a different node placement. The same node placements are used for both the neighbor discovery algorithms. We do not show the confidence intervals, since the confidence interval widths were observed to be very small. (The 95% confidence interval widths for simulations of both the algorithms were within 2.3% of the corresponding mean values).



(a) Fraction of Discovered Neighbors

(b) Time to Discover 98% of Graph

Figure 6: Comparison of Gossip-based and Direct discovery algorithms

The simulation results clearly indicate that nodes discover their neighbors much faster using the gossip-based algorithm than using the direct-discovery algorithm. To quantify the difference between the two algorithms, we observe that at the end of 50 time slots, the expected fraction of neighbors discovered by a node using direct-discovery algorithm is 0.37 while the fraction is 0.94 using the gossip-based algorithm. In other words, in 50 time slots a node discovers 2.5 times more neighbors with the gossip-based algorithm than with the direct-discovery algorithm.

We observed in Figure 6(a) that each node quickly discovers a large fraction of its neighbors using the gossip-based algorithm. A related, but system-wide, metric of interest is the time until all nodes in the network *collectively* discover a certain fraction of the entire underlying graph. More formally, let $G = (V, E)$ represent the actual underlying graph where V is the set of nodes and E is the set of directed edges representing pairs of nodes that are neighbors of each other. Formally, $E = \{(u, v)|u, v \in V, d(u, v) \leq r\}$, where $d(u, v)$ represents the euclidean distance between nodes *u* and *v*, and *r* is the node transmission range. Let $G(t) = (V, E(t))$ represent the graph discovered by time t. The set $E(t)$ is the union $\cup_{v \in V} E_v(t)$, where $E_v(t)$ represents the set of edges discovered by node v by time t. We then ask the question: what is the time, T_f, until $\frac{|E(t)|}{|E|} \geq f$?

In Figure 6(b), we plot $T_{0.98}$ as a function of node density. We also plot the 95% confidence intervals on the graph. However, the confidence intervals are very small and hence, not noticeable. We observe that $T_{0.98}$ increases with node density for the direct-discovery algorithm. This is not surprising since the direct-discovery algorithm allows a node to discover at most one node per time slot and with the increase in node density the number of neighbors of a node also increases. Interestingly, $T_{0.98}$ for the gossip-based algorithm is insensitive to an increase in node density. In fact, $T_{0.98}$ initially decreases with an increase in node density. This is because, as density increases, the probability of a nodes discovering its neighbors indirectly increases and offsets the decrease in the probability of direct discovery. In Figure 5(b), we numerically evaluate $1 - I_{i,j}(t)$ and $1 - P_{i,j}(t)$ for different node densities using the analysis in Section 4.2. The probability of a successful transmission p_s decreases with increasing node density and so does the probability of direct discovery $D_{i,j}(t)$. Although the indirect discovery probability $I_{i,j}(t)$ also depends on p_s, an increase in node density means that a node can discover its neighbors indirectly from more nodes. This more than offsets the decrease in p_s resulting in an overall increase in $I_{i,j}(t)$. In Figure 5(b), we observe that $I_{i,j}(t)$ and $P_{i,j}(t)$ almost overlap with each other, despite an increase in the number of neighbors. This indicates that $P_{i,j}(t)$ is insensitive to an increase in node density. This explains the insensitivity in the time to discover a certain fraction of neighbors with respect to node density as observed in our simulations and validates the analysis in Section 4.2.

4.4 Practical Considerations

We next consider several practical issues associated with our gossip-based neighbor discovery algorithm.

1. Fraction of nodes with location information: So far, we assume that each node knows its location information. In practice, the gossip-based algorithm can still be used without modification even if only a fraction of nodes know their location information. We simulate the gossip-based neighbor discovery algorithm by varying the fraction of nodes in the network with location information (f). In Figure 7, we plot the expected fraction of neighbors discovered by a node against time for different values of f. We observe that the performance degrades gracefully and approaches the performance of the direct-discovery algorithm as the fraction of nodes with location information becomes smaller. The ability of the gossip-based algorithm to operate without change even when only a fraction of nodes have location information demonstrates its flexibility.

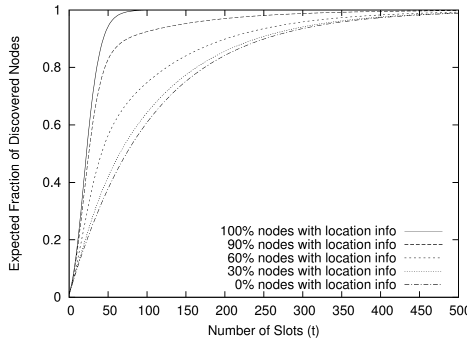


Figure 7: Performance of gossip-based algorithm when only a fraction of nodes have location information

2. Message Size: In the gossip-based algorithm, a node's message consists of not only the identities of its discovered neighbors but also their co-ordinates. Thus, message length grows as more and more nodes are discovered. This may not be a serious concern if the node density is not large, but for very dense networks the message size can be reduced by compressing the neighbor information. Clever encoding of the location information should be possible since nodes that are geographically close to each other will have very similar location information.

3. Physical Obstacles: While the gossip-based algorithm works well in a free-space environment, the presence of physical obstacles can cause nodes to incorrectly infer another node as its neighbor. In other words, even though two nodes may be geographically close to each other, they may still not be able to communicate with each other. In such an environment, the location-discovery phase must be followed by a "pruning" phase. In this phase, each node solicits a response from each "potential" neighbor that it has discovered indirectly by sending out probe messages exactly in the direction of its potential neighbor. This is possible since the node already knows the location information of its potential neighbors. The absence of a response after sufficient number of retries causes a removal of that node from the neighbor list. While this "pruning" slows down the discovery process, the algorithm still discovers neighbors more quickly than the direct-discovery algorithm. This is because each node only probes potential neighbors that are discovered indirectly.

4.5 Algorithm Enhancements

In the gossip-based algorithm described earlier, nodes only gossip about their discovered neighbors. However, the gossip-based algorithm also allows a node to gossip about other discovered nodes which are not its neighbors. By including the identities of other discovered nodes (which are not its neighbors) in its gossip message, a node can potentially help its neighbors to discover their neighbors faster. However, this extra information comes at the cost of an increased message length. One possible solution is to only gossip about a fraction of such non-neighbors at a time. A more detailed study of the various tradeoffs and analysis of discovery probability using this enhanced algorithm is an interesting direction for future research.

5 Asynchronous Discovery Algorithms

So far, we have assumed the existence of a slotted, synchronous system for both the direct-discovery and gossip-based algorithms, and obtained the transmission probability, p_t, that maximizes the probability of discovering a neighbor. We next outline how our synchronous algorithms can operate asynchronously.

5.1 Direct-Discovery Algorithm

We discuss the asynchronous version of direct-discovery algorithm and its optimal design. As in the previous section, nodes have a fixed beamwidth θ. We analyze the case in which nodes have a directional transmitter and omni-directional receiver. The extension to other antenna models is straightforward.

The asynchronous direct-discovery algorithm operates as follows. Each node listens for a random time interval. Upon the expiry of this time interval, the node transmits in a random direction and then returns to listen mode. All transmissions are of fixed duration, τ.

5.2 Analysis

Consider a random node, say i, with k-1 neighbors, $j = 1, 2, \ldots, k-1$. Let $N_i(t) = (N_{i,1}(t), N_{i,2}(t), \ldots, N_{i,k-1}(t))$ be a k-tuple, where each $N_{i,j}(t)$ represents the number of transmissions successfully received by node i from its neighbor j in time t.

For simplicity, we assume that the listen intervals of each node are exponentially distributed with rate λ. We further assume that the transmission duration is very small, i.e., $\tau \approx 0$. This assumption means that the inter-transmission times of a node are exponentially distributed with rate λ. Given these assumptions, the inter-transmission time from a given node j to its neighboring node i is also exponentially distributed with rate $\lambda' = \frac{\theta}{2\pi}\lambda$, since $\frac{\theta}{2\pi}$ is the probability that node j's transmission covers node i. Since node i has $k-1$ such neighbors, the time between two successive transmissions to node i is exponentially distributed with rate, $\lambda'' = (k-1)\lambda'$. Or, in terms of λ, $\lambda'' = (k-1)\frac{\theta}{2\pi}\lambda$.

We are interested in the event that node i does not discover its neighbor j in time t, i.e., $N_{i,j}(t) = 0$. We condition this event on the number of transmissions from node j to node i by time t, which we represent by the random variable $X_{i,j}(t)$.

Let $p_{i,j}$ represent the probability that node i successfully receives a transmission from a given neighbor j. Therefore, the probability that node i successfully receives m transmissions from a node j given n transmissions from node j in time t, follows a binomial distribution with parameters n and $p_{i,j}$.

$$P(N_{i,j}(t) = m | X_{i,j}(t) = n) = \binom{n}{m} {p_{i,j}}^m (1 - p_{i,j})^{n-m}$$

$$P(N_{i,j}(t) = 0 | X_{i,j}(t) = n) = (1 - p_{i,j})^n$$

Removing the conditioning on the number of transmissions $X_{i,j}(t)$,

$$P(N_{i,j}(t)=0)=\sum_{j=0}^{t/\tau}(1-p_{i,j})^j.\frac{e^{-\lambda' t}(\lambda' t)^j}{j!} \tag{8}$$

The upper limit of t/τ in the summation in (8) is based on the fact that a node cannot simultaneously schedule more than one transmission to a given node. In the limit as transmission time approaches zero i.e., $\tau \to 0$, the fraction t/τ in (8) approaches ∞. Substituting in (8) yields:

$$P(N_{i,j}(t)=0)=\sum_{j=0}^{\infty}(1-p_{i,j})^j\frac{e^{-\lambda' t}(\lambda' t)^j}{j!} \tag{9}$$

Simplifying (9) and expressing λ' in terms of λ yields:

$$P(N_{i,j}(t)=0)=e^{-\frac{\theta}{2\pi}\lambda p_{i,j}t} \tag{10}$$

We next determine $p_{i,j}$, the probability that node i successfully receives a transmission from node j. This probability is simply the probability that node i receives no other transmission in a time period of 2τ. More precisely, a transmission from node j to node i that starts at time instant t is successful only if no other node transmits to node i during the time interval $[t-\tau, t+\tau]$.

Recall from our earlier discussion, that inter-reception times at node i are exponentially distributed with rate $\lambda''=(k-1)\frac{\theta}{2\pi}\lambda$. In addition, node i performs its own transmissions with rate λ. Therefore,

$$\begin{aligned} p_{i,j} &= e^{-2\tau(\lambda''+\lambda)} \\ &= e^{-2\tau\lambda((k-1)\frac{\theta}{2\pi}+1)} \end{aligned}$$

Rewriting (10) in terms of λ yields

$$P(N_{i,j}(t)=0)=e^{-\frac{\theta}{2\pi}f(\lambda)t}$$

where,

$$f(\lambda)=\lambda e^{-2\tau\lambda((k-1)\frac{\theta}{2\pi}+1)}$$

Returning to the neighbor discovery problem, we seek a value of λ that minimizes the probability, $P(N_{i,j}(t)=0)$. This corresponds to finding the optimal rate λ that maximizes $f(\lambda)$. Standard calculations yields:

$$\lambda=\frac{1}{2\tau((k-1)\frac{\theta}{2\pi}+1)} \tag{11}$$

This result for optimal value of λ is reminiscent of the expression we obtained earlier for the optimal transmission probability p_t for a synchronous system in that the optimal transmission rate λ of a node is inversely proportional to the number of its neighbors k. An interesting question is how the performance of asynchronous discovery algorithm compares to its synchronous counterpart. In other words, we wish to find out how much synchronization helps improve the performance of neighbor discovery algorithm.

5.3 Comparison of Asynchronous and Synchronous Direct-Discovery Algorithm

We now compare the times required by the two direct-discovery algorithms. Our metric for comparison is the time t until the probability that a node i discovers a given neighbor j exceeds p, i.e., the minimum time t such that $P_{i,j}(t) \geq p$. We will assume that node i has k-1 neighbors and beamwidth θ.

For the synchronous direct-discovery algorithm,

$$P_{i,j}(t) = 1 - (1 - p_{i,j})^t$$

where $p_{i,j}$ denotes the probability of successful transmission from node j to node i in a time slot. Assuming the optimal value of $p_t = \frac{2\pi}{\theta k}$, we obtain

$$p_{i,j} = \frac{1}{k}(1 - \frac{1}{k})^{k-2}(1 - \frac{2\pi}{\theta k})$$

The minimum time, t_s, until $P_{i,j}(t_s)$ exceeds p is given by:

$$t_s = \frac{\log(1-p)}{\log(1 - \frac{1}{k}(1 - \frac{1}{k})^{k-2}(1 - \frac{2\pi}{\theta k}))} \tag{12}$$

Note that time t_s in (12) is measured in number of slots. For small x, $\log(1 + x) \approx x$. Hence,

$$t_s = \frac{\log(1-p)}{-\frac{1}{k}(1 - \frac{1}{k})^{k-2}(1 - \frac{2\pi}{\theta k})} \tag{13}$$

For the asynchronous direct-discovery algorithm, we obtain the minimum time, t_a, until $P_{i,j}(t_a)$ exceeds p from (10) and (11) as:

$$t_a = -2e(k - 1 + \frac{2\pi}{\theta}) \log(1 - p)$$

For large k, we can approximate the above as,

$$t_a = -2ek \log(1 - p) \tag{14}$$

The expression for t_a is in number of slots assuming each slot has duration τ.

The ratio R of (14) to (13) yields

$$R = 2e(1 - \frac{1}{k})^{k-2}(1 - \frac{2\pi}{\theta k})$$

For large k, $(1 - \frac{1}{k})^{k-2} \rightarrow e^{-1}$ and $(1 - \frac{2\pi}{\theta k}) \rightarrow 1$. Hence,

$$R \approx 2$$

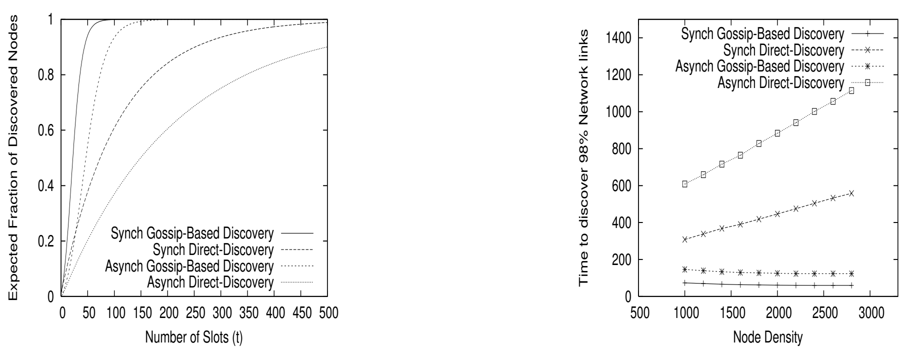
Thus, our analysis suggests that for dense networks, the asynchronous algorithm requires an amount of time to discover a neighbor that is approximately twice the time required by the synchronous algorithm. This factor of two slowdown in asynchronous discovery algorithms is observed in our simulations as well and will be discussed in more detail in Section 5.5.

5.4 Asynchronous Gossip-Based Algorithm

The operation of asynchronous gossip-based algorithm is similar to that of the direct-discovery algorithm, except for the additional information contained in the messages. In Section 4, we found that the optimal transmission probability p_t for the synchronous gossip-based algorithm is the same as that for the synchronous direct-discovery algorithm. For both algorithms, the probability of a successful transmission $p_{i,j}$ is the same and the discovery probability is maximized when $p_{i,j}$ is maximized. Since the probability of a successful transmission remains the same even for asynchronous versions of the algorithms, the optimal transmission rate, λ, for the gossip-based algorithm is the same as that for the direct-discovery algorithm.

5.5 Simulation Results

In Figure 8, we compare the asynchronous direct and gossip-based discovery algorithms with their synchronous counterparts. The simulation setting is exactly the same as considered in Section 4, viz. each node with a beamwidth of 30^o and transmission range of 200m. For the results in Figure 8(a), we simulate 2000 nodes in a square with area $9 \times 10^6 m^2$. In Figure 8(a), we plot the expected fraction of neighbors discovered by a node against time for



(a) Expected Fraction of Discovered Neighbors

(b) Time to Discover 98% of Graph

Figure 8: Comparison of Synchronous and Asynchronous Discovery Algorithms

the various discovery algorithms. Not surprisingly, the gossip-based algorithms outperform the direct-discovery algorithms and the synchronous discovery algorithms outperform their asynchronous counterparts.

In Figure 8(b), we plot the time, $T_{0.98}$, to discover 98% of the graph against node density. We also plot the 95% confidence intervals which are too small to be noticeable. We observe that for the asynchronous gossip-based algorithm, $T_{0.98}$, is insensitive to node density and in fact decreases initially with increasing density. This behavior is exactly same as that observed for the synchronous gossip-based algorithm. $T_{0.98}$ for the direct-discovery algorithms, however, increases with node density. Another interesting observation is that, for a given node density, $T_{0.98}$ for the asynchronous discovery algorithms is approximately twice the corresponding $T_{0.98}$ for their synchronous counterparts. This slowdown by a factor of two in the simulations is exactly as predicted in our analysis in Section 5.3. In fact, in a classical paper by Roberts [12], it was shown that the throughput achieved using slotted ALOHA is twice that of pure ALOHA. An explanation for this result as pointed out in [12] is that the "vulnerable" period of a transmission is only one time slot (since nodes transmit only at slot boundaries) in slotted ALOHA, while the "vulnerable" period in pure ALOHA is twice the length of a packet or two time slots. Because of the similarity of the ALOHA protocols to our discovery algorithms, we see that the values of T_k for asynchronous algorithms are approximately twice that for their synchronous counterparts.

5.6 Algorithm Enhancements

We propose an enhancement to the asynchronous discovery algorithm in which each node senses if it is currently receiving a transmission, before it transmits. Details of this algorithm and its analysis is available in A. Our analysis shows that, by using sensing, there is an increase in probability of discovering a neighbor within a given amount of time. However, this increase is only small.

6 Antenna Beamwidth Selection

So far, our analysis was based on determining the optimal transmission probability p_t, given that nodes have a fixed beamwidth, θ. However, the process of neighbor discovery depends not only on p_t but also on node beamwidths. A large beamwidth covers a large transmission area and is therefore more likely to cause greater interference compared to a smaller beamwidth. However, a large beamwidth is advantageous since it potentially allows many nodes to simultaneously receive a transmission. Thus, there is a trade-off between the speed of discovery of neighbors and probability of collision. The goal, then, is to find the node beamwidth that maximizes the expected number of neighbors discovered by a node in a given amount of time. Our goal is motivated by the fact that information about a large number of neighbors is essential for almost all routing protocols in order to construct optimal routes in the network.

For a given transmission power, the transmission range of a node depends on its beamwidth. Let d_o be the range of a transmission when a node transmits omni-directionally. By using a directional transmitter, a node can focus all its power in the direction of transmission. With a beamwidth of θ, a node can radiate $\frac{2\pi}{\theta}$ times more power in direction of transmission as compared to an omni-directional antenna, thus extending its transmission range. This quantity of $\frac{2\pi}{\theta}$ is called the **gain** of the directional antenna. The transmission range, $d(\theta)$, then is the distance at which received signal strength is the same as the signal strength at distance d_o when using an omni-directional antenna. If α is the pathloss exponent, then we obtain the following relationship between d_o, d and θ.

$$\frac{1}{d_o^\alpha} = \frac{2\pi}{\theta}\frac{1}{d(\theta)^\alpha} \tag{15}$$

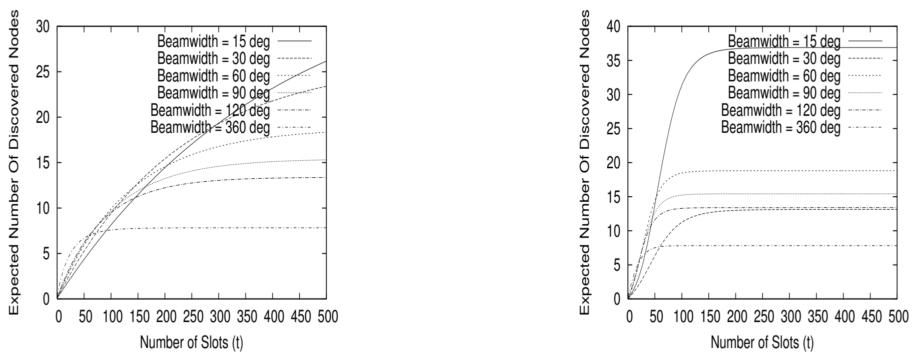
Rewriting equation (15):

$$d(\theta) = d_o(\frac{2\pi}{\theta})^{\frac{1}{\alpha}} \tag{16}$$

All nodes at distance $d(\theta)$ or less from the node are its *potential neighbors*. It is easy to see from equation (16) $d(\theta)$ increases as node beamwidth decreases and, hence, the number of its potential neighbors.

For a given antenna beamwidth and neighbor discovery algorithm, the number of discovered neighbors increases with time. The number of discovered neighbors also depends on the number of potential neighbors of the node which is a function of the beamwidth and node density. We thus formulate the optimal beamwidth selection problem as follows. Given that nodes are uniformly distributed with density λ, what is the optimal choice of beamwidth that maximizes the expected number of discovered neighbors by time t?

In order to address this question, we simulate our synchronous neighbor discovery algorithms for different antenna beamwidths. The results of our simulation are shown in Figure 9. The simulation involves 2000 nodes placed in a square with area $9 \times 10^6 m^2$. We used the simple model given by (16) to determine the transmission range of nodes for different choices of beamwidths. For our simulations, we choose an omni-directional transmission radius $d_o = 107m$ and the path loss exponent $\alpha = 4$. Depending on the choice of beamwidth, nodes choose the transmission probability p_t as provided by our analyses in Sections 3 and 4. We observe from Figure 9 that for a given neighbor discovery



(a) Direct-Discovery Algorithm

(b) Gossip-Based Algorithm

Figure 9: Effect of beamwidth on neighbor discovery

algorithm, the choice of beamwidth depends on the time t that is allocated to the neighbor discovery process. For both neighbor discovery algorithms, larger values of t yield larger expected numbers of discovered neighbors using narrower beamwidths. However, if t is small, using a large beamwidth results in a greater number of neighbors being discovered as seen in Figure 9(a). As t increases, no new neighbors are discovered and hence narrower beamwidths are more preferable. This is indeed the case even for the gossip-based algorithm too. However, the advantage of using a larger beamwidth seems insignificant in the case of the gossip-based algorithm, as seen from Figure 9(b), and suggests that nodes can use the narrowest possible beamwidth even for small t.

If nodes are equipped with antennas that can dynamically adjust their beamwidths, an interesting question is whether there is any benefit in varying the node beamwidth during discovery. For instance, from Figure 9(a) we observe that, for the direct-discovery algorithm, nodes can initially transmit with a beamwidth of 2π until $t = 25$. After $t = 25$, no additional neighbors are discovered if nodes transmit omnidirectionally and so nodes can reduce their beamwidth to the next largest beamwidth and continue the discovery process. An interesting problem is to determine in what step sizes should nodes reduce their beamwidth and after what time intervals should nodes switch to smaller beamwidths. We do not consider this problem in this paper, and is a topic for future work. Figure 9(b), however, suggests little benefit in varying beamwidths for the gossip-based algorithm.

7 Conclusions and Future Work

In this paper, we considered the problem of neighbor discovery in wireless networks with directional antennas. We proposed two classes of probabilistic neighbor discovery algorithms, viz. *Direct-Discovery Algorithms* in which nodes discover their neighbors only when they hear transmissions from their neighbors and *Gossip-Based Algorithms* in which nodes gossip about location information about their neighbors. We first considered the operation of these algorithms in a slotted, synchronous system and find the transmission probability that maximizes the probability of discovering their neighbors. Simulations of the algorithms demonstrated that the *Gossip-Based Algorithms* are insensitive to increase in node density i.e., the time required to discover a given fraction of neighbors remains unaffected with the increase in node density. The gossip-based algorithm also has an interesting property that it operates without any modification even if only a fraction of nodes have location information and its performance degrades gracefully to that of direct-discovery algorithm when none of the nodes have location information. We also described how the synchronous algorithms can be modified to operate asynchronously and analytically derive its optimal algorithm parameters. Finally, we discussed how choice of antenna beamwidths affects the expected number of neighbors discovered by the neighbor discovery algorithms.

There are a number of future directions from this work. Analytical derivation of the time to discover a given fraction of the entire topology is an interesting problem. Deriving analytical bounds for the decay probability for the gossip-based algorithm is another future goal, as it will help determine how well the gossip-based algorithm performs in comparison to the direct-discovery algorithm as a function of the various algorithm parameters. When nodes can dynamically adjust their beamwidths, designing beamwidth varying algorithms to maximize the number of neighbors discovered is an interesting open question. Some applications require nodes to discover only sufficient number of neighbors to achieve k-connectivity. Designing beamwidth varying algorithms to rapidly achieve k-connectivity is another interesting direction for future work.

References

[1] L. Bao, J.J. Garcia-Luna-Aceves. Transmission Scheduling in Ad Hoc Networks with Directional Antennas. In *Proceedings of ACM MOBICOM 2002*, Atlanta, GA, 2002.

[2] A. Baptista, T. Leen, Y. Zhang, A. Chawla, D. Maier, W. Feng, W. Feng, J. Walpole, C. Silva, J. Freire. Environmental Observation and Forecasting Systems: Vision, Challenges and Successes of a Prototype. *Encyclopedia of Physical Science and Technology*, Academic Press, Third Edition, Vol. 5., pp 565-581.

[3] R. Choudhury, X. Yang, R. Ramanathan, N. Vaidya. Using Directional Antennas for Medium Access Control in Ad Hoc Networks. In *Proceedings of ACM MOBICOM*, Atlanta, GA, 2002.

[4] R. Choudhury, N. Vaidya. Impact of Directional Antennas on Ad Hoc Routing. In *Proceedings of Personal and Wireless Communications*, Venice, 2003.

[5] A. Keshavarzian, E. Uysal-Biyikoglu, F. Herrman and A. Manjeshwar. Energy-efficient link assessment in Wireless Sensor Networks *Proc. of IEEE INFOCOM*, Hong Kong, March 2004.

[6] L. Li, J. Halpern, P. Bahl, Y. Wang and R. Wattenhofer. Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multihop Networks In *Proceedings of ACM Principles of Distributed Computing*, Newport, RI, 2001.

[7] A. Mainwaing, R. Szewczyk, D. Culler, J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[8] Self-Organizing Neighborhood Wireless Mesh Network. http://research.microsoft.com/ bahl/MS_Projects/projects.html

[9] M. McGlynn and S. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks In *Proceedings of ACM MOBIHOC*, Long Beach, CA, 2001.

[10] R. Ramanathan. On the performance of beamforming antennas. In *Proceedings of ACM MOBIHOC*, Long Beach, CA, 2001.

[11] R. Ramanathan and R. Hain. Topology Control of Multihop Radio Networks using Transmit Power Adjustment. In *Proceedings of IEEE INFOCOM*. Tel Aviv, Israel, March 2000.

[12] L. Roberts. ALOHA Packet System With and Without Slots and Capture. *Computer Communications Review*, April 1972, pp. 28-42.

[13] L. Schwiebert, S. Gupta, J. Weinmann. "Research Challenges in Wireless Networks of Biomedical Sensors". In *Proceedings of ACM MOBICOM*, 2001, pp. 151-165.

[14] S. Vasudevan, J. Kurose and D. Towsley. On Neighbor Discovery in Wireless Networks With Directional Antennas. UMass Comp. Sci. Technical Report, July 2004, TR# 04-53.

[15] R. Wattenhofer, L. Li, P. Bahl and Y. Wang. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[16] F. Zhao, J. Shin and J. Reich. "Information-Driven Dynamic Sensor Collaboration for Tracking Applications". In *IEEE Signal Processing Magazine*, March 2002.

APPENDIX

A Asynchronous Neighbor Discovery Algorithm With Sensing

The analysis of the asynchronous algorithm described in Section 5 assumes that nodes transmit as soon as their listen period expires, even if they are receiving a transmission from their neighbors. By avoiding transmission during an ongoing reception, nodes can increase the probability of successfully receiving a transmission. We, therefore, modify the asynchronous algorithm described in Section 5 as follows. Each node listens for an exponentially distributed interval with rate λ. At the end of the interval, if the node is receiving a transmission it listens again for an exponentially distributed time interval with rate λ. However, if there is no ongoing transmission, then the node transmits in a random direction with a fixed beamwidth θ. After transmission, the node listens again.

Our goal is to determine an optimal λ that minimizes $P(N_{i,j}(t) = 0)$, the probability that node *i* hears no transmission from its neighboring node *j* in time *t*.

It can be seen that the inter-transmission times of a node for the modified algorithm is an exponential random variable with rate $\lambda^{'} = \lambda p_{idle}$, where p_{idle} is the probability that the node is not receiving a transmission at a given time instant. We can use the analysis in Section 5 to obtain the optimal λ'. However, the probability of a successful transmission from node *j* to node *i* $p_{i,j}$ for the modified algorithm is different since node *i* senses an ongoing reception before transmitting its own packet. The probability $p_{i,j}$ is given by:

$$p_{i,j} = p_{i,j} = e^{-2\tau\lambda'(k-1)\frac{\theta}{2\pi}}$$

Using the analysis in Section 5, the optimal λ' is:

$$\lambda^{'} = \frac{1}{2\tau(k-1)\frac{\theta}{2\pi}} \tag{17}$$

However, our goal is to obtain the optimal λ and not $\lambda^{'}$, therefore

$$\lambda = \frac{1}{2\tau(k-1)\frac{\theta}{2\pi} p_{idle}} \tag{18}$$

From (18), we need to determine p_{idle} in order to obtain λ. Recall that p_{idle} represents the probability that a given node (designated as node *m* in our discussion) is not receiving any transmission at any given time instant.

Consider the time line of node *m* shown in Figure 10.

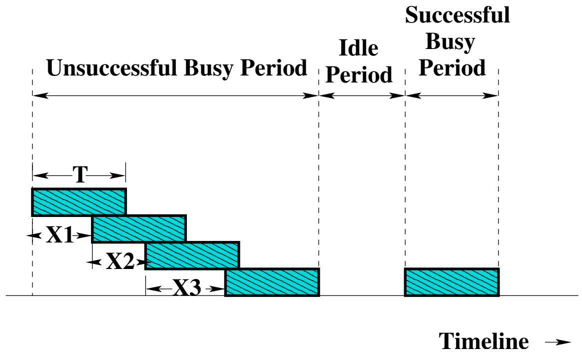


Figure 10: Time Line of a Node

Let *B* represent the mean duration of the busy period and *I* represent the mean duration of the idle period at node *m*. Then p_{idle} is given by:

$$p_{idle} = \frac{I}{B+I} \tag{19}$$

The inter-arrival times at node *m* are exponentially distributed with rate, $\lambda^{''} = (k-1)\frac{\theta}{2\pi}\lambda^{'}$.

Substituting from equation (17) for $\lambda^{'}$, yield

$$\lambda^{''} = \frac{\pi}{\theta\tau} \tag{20}$$

Now the mean idle period of node *m* is given by :

$$I = \frac{1}{\lambda''} = \frac{\theta\tau}{\pi} \tag{21}$$

The busy period of node *m* is given by :

$$B = X_1 + X_2 + \ldots + X_R + \tau$$

where X_i's are iid exponential random variables with rate λ'' while *R* is geometrically distributed with mean $\frac{1}{p}$ where $p = e^{-\lambda''\tau}$. In words, p represents the probability of no transmission arriving for the duration of an ongoing transmission, τ.

The sum $X_1 + X_2 + \ldots + X_R$ is exponentially distributed with rate $\lambda'' p$. Hence, the mean busy period of node *m* is:

$$B = \frac{1}{\lambda'' p} + \tau = \tau(1 + \frac{\theta}{p\pi}) \tag{22}$$

Replacing B and I in (19) with expressions in (22) and (21) yields

$$p_{idle} = \frac{e^{-\frac{\pi}{\theta}}}{1 + e^{-\frac{\pi}{\theta}}(1 + \frac{\pi}{\theta})} \tag{23}$$

Substituting in equation (18) for p_{idle}, yields the expression for λ.

Substituting the optimal vLue of λ into the expression for $P(N_{i,j}(t) = 0)$ yields:

$$P(N_{i,j}(t) = 0) = \exp\{-\frac{\theta}{2\pi} \frac{1}{2e\tau(k-1)\frac{\theta}{2\pi}} t\} \tag{24}$$

For the asynchronous algorithm described in Section 5,

$$P(N_{i,j}(t) = 0) = \exp\{-\frac{\theta}{2\pi} \frac{1}{2e\tau((k-1)\frac{\theta}{2\pi} + 1)} t\} \tag{25}$$

It is easy to see that $P(N_{i,j}(t) = 0)$ is smaller in (24) than in (25). In other words, node *i* has greater probability of discovering *j* within a given time using the modified algorithm . However, it can be seen from the expressions that the decrease in $P(N_{i,j}(t) = 0)$ with the reception sensing is very small.