2017

Analyzing discovery latency in mobile networks

Saud Alramzi

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

https://researchrepository.wvu.edu/etd/3995
ANALYZING DISCOVERY LATENCY IN MOBILE NETWORKS

Saud Alramzi

Problem Report submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of

Masters of Science
in
Electrical Engineering

Vinod Kulathumani, Ph.D, Chair
Xin Li, Ph.D
Yanfang Ye, Ph.D

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2017

Keywords: neighbor discovery, discovery latency, disco, manet

Copyright 2017 Saud Alramzi
ABSTRACT

Analyzing Discovery Latency in Mobile Networks

Saud Alramzi

Duty cycle is important in wireless sensor and mobile ad-hoc networks (MANETs) to extend their lifetime. Duty cycling in wireless networks lets the nodes wake up with infrequent fixed periods, based on pre-determined parameters. On the other hand, neighbor discovery is the process by which nodes learn about the neighbors. Nodes can use radio communications to discover other neighbors. Timely, neighbor discovery is crucial for routing. But, neighbor discovery becomes more challenging in energy-constrained, mobile environment with duty cycled nodes where a node may not know whether any neighbors are present, and what duty cycle those neighbors might operate at.

Disco is an asynchronous neighbor discovery and rendezvous protocol that allows two or more nodes to operate their radios at low duty cycles (1-2%). Disco obtains discovery and communication during infrequent, opportunistic encounters without requiring a global synchronization information. Disco nodes pick a pair of dissimilar prime number such that the sum of their reciprocal is equal to the desired duty cycle. A global counter increments with a fixed period. If one of the node’s prime numbers is divisible by the counter, the node will turn on its radio for one period. This protocol ensures that two nodes have some overlapping radio on-time within the boundary of the period, and discover each other despite of the independent set of duty cycle.

In this report, we seek to understand how Disco performs in mobile networks. We analyze the impact of different duty cycles, mobility speeds and network sizes on discovery latency in extreme mobile networks. We also scrutinize how fast Disco discovers that nodes have moved out of their neighborhood. We use NS3 to simulate Disco with different, duty cycles, mobility speeds, and network sizes.
Acknowledgements

First, I would want to thank my committee chair and advisor Dr. Vinod K. Kulathumani for constant support, providing me an opportunity to work with him and guiding me throughout my Master’s program. Also, I would like to extend my warm acknowledgments to both Dr. Xin Li and Dr. Yanfang Ye for taking their courses in Image Processing and Advanced Data Mining respectively, and for being part of my committee.

In addition, I would especially like to thank my family members and my lovely wife Mrs. Dalal Kamal, for her generous and unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation.

Furthermore, I would like to thank all my friends for their suggestions and helping me overcome all the tough times in Morgantown, especially Mr. Mrwan Ben Idris, and Ms. Sowmya Devaraja.

Last but not least, I would want to thank all of the faculty members in the Lane Department of Computer Science and Electrical Engineering for sharing expertise, and sincere and valuable guidance and encouragement extend to me.
# Table of Contents

Chapter 1  Introduction .................................................................................................................. 1
  1.1  Overview .................................................................................................................................. 1
  1.2  Summary of Contributions ........................................................................................................ 4
  1.3  Outline ...................................................................................................................................... 5

Chapter 2  Review of Prior work .................................................................................................. 6
  2.1  Birthday Protocol ..................................................................................................................... 6
  2.2  Quorum-based Protocol .......................................................................................................... 6
  2.3  U-Connect Protocol ................................................................................................................. 6
  2.4  Disco Protocol ......................................................................................................................... 6

Chapter 3  Design and Implementations ....................................................................................... 11
  3.1  Design ..................................................................................................................................... 11
  3.2  Experimental Setup ................................................................................................................. 13

Chapter 4  Results ........................................................................................................................ 18
  4.1  Discovery Latency .................................................................................................................... 18
    4.1.1  Different Duty Cycles ....................................................................................................... 18
    4.1.2  Different Mobility Speeds ............................................................................................... 18
  4.2  Failing Latency ....................................................................................................................... 20
    4.2.1  Different Duty Cycles ....................................................................................................... 20
    4.2.2  Different Mobility Speeds ............................................................................................... 23

Chapter 5  Future Work and Conclusion ..................................................................................... 28
  5.1  Future Work ............................................................................................................................ 18
  5.2  Conclusion ............................................................................................................................... 18

References ...................................................................................................................................... 29
List of Figures

Figure 4.1  The Discovery Latency for 50 nodes and the mobility of nodes is 2-4m/s .......... 3
Figure 4.2  The Discovery Latency for 50 nodes and the mobility of nodes is 3-7m/s .......... 7
Figure 4.3  The Discovery Latency for 50 nodes and the mobility of nodes is 7-15m/s .......... 8
Figure 4.4  The Discovery Latency for 50 nodes and the mobility of nodes is 15-25m/s ...... 15
Figure 4.5  The Discovery Latency for 100 nodes and the mobility of nodes is 2-4m/s ...... 18
Figure 4.6  The Discovery Latency for 100 nodes and the mobility of nodes is 3-7m/s ....... 19
Figure 4.7  The Discovery Latency for 100 nodes and the mobility of nodes is 7-15m/s ...... 23
Figure 4.8  The Discovery Latency for 100 nodes and the mobility of nodes is 15-25m/s .... 24
Figure 4.9  The Discovery Latency for 50 nodes and nodes’ duty cycle is 1% ................. 25
Figure 4.10 The Discovery Latency for 50 nodes and nodes’ duty cycle is 10%............... 26
Figure 4.11 The Discovery Latency for 50 nodes and nodes’ duty cycle is 20%............. 27
Figure 4.12 The Discovery Latency for 50 nodes and nodes’ duty cycle is 30%............. 28
Figure 4.13 The Discovery Latency for 50 nodes and nodes’ duty cycle is 50%............. 7
Figure 4.14 The Discovery Latency for 50 nodes and nodes’ duty cycle is 100%........... 8
Figure 4.15 The Discovery Latency for 100 nodes and nodes’ duty cycle is 1%............. 15
Figure 4.16 The Discovery Latency for 100 nodes and nodes’ duty cycle is 10%........... 18
Figure 4.17 The Discovery Latency for 100 nodes and nodes’ duty cycle is 20%.......... 19
Figure 4.18 The Discovery Latency for 100 nodes and nodes’ duty cycle is 30%......... 23
Figure 4.19 The Discovery Latency for 100 nodes and nodes’ duty cycle is 50%......... 24
Figure 4.20 The Discovery Latency for 100 nodes and nodes’ duty cycle is 100%...... 25
Figure 4.21 The Failing Latency for 50 nodes and the mobility of nodes is 2-4m/s ....... 26
Figure 4.22  The Failing Latency for 50 nodes and the mobility of nodes is 3-7m/s ............... 27
Figure 4.23  The Failing Latency for 50 nodes and the mobility of nodes is 7-15m/s ........... 28
Figure 4.24  The Failing Latency for 50 nodes and the mobility of nodes is 15-25m/s .......... 3
Figure 4.25  The Failing Latency for 100 nodes and the mobility of nodes is 2-4m/s .......... 7
Figure 4.26  The Failing Latency for 100 nodes and the mobility of nodes is 3-7m/s ........... 8
Figure 4.27  The Failing Latency for 100 nodes and the mobility of nodes is 7-15m/s .......... 15
Figure 4.28  The Failing Latency for 100 nodes and the mobility of nodes is 15-25m/s ....... 18
Figure 4.29  The Failing Latency for 50 nodes and nodes’ duty cycle is 1% ...................... 19
Figure 4.30  The Failing Latency for 50 nodes and nodes’ duty cycle is 10% .................... 23
Figure 4.31  The Failing Latency for 50 nodes and nodes’ duty cycle is 20% ..................... 24
Figure 4.32  The Failing Latency for 50 nodes and nodes’ duty cycle is 30% ..................... 25
Figure 4.33  The Failing Latency for 50 nodes and nodes’ duty cycle is 50% .................... 3
Figure 4.34  The Failing Latency for 50 nodes and nodes’ duty cycle is 100% .................... 7
List of Tables

Table 3.1 Examples for a pair of prime numbers for 1%, 10%, and 20% duty cycle ............... 2
Table 3.2 Three different cases when two nodes have a 1% duty cycle............................ 10
Table 3.3 The experiment parameters............................................................................... 10
Table 4.1 Percentages of undiscovered neighbors’ nodes for Figure 4.19 network.......... 16
Chapter 1

Introduction

1.1 Overview

Duty cycle is important for wireless sensor and mobile ad-hoc networks (MANETs). The devices in the MANET are termed as nodes. Node’s duty cycle determines its active time. A 60% duty cycle means the node is active for 60% of the time but off for 40% of the time. By varying the duty cycle ratio, we can let the node wake up with infrequent fixed periods.

Neighbor discovery in mobile networks is the process in which nodes learn about their neighbors. Neighbor discovery is divided into two parts, the first part is when nodes discover their neighbors. The second part is when nodes discover their neighbors are no longer in communication range. Neighbor discovery has proven challenging because of changing topology and no information regarding the wake-up schedules of the neighbor nodes. To minimize energy consumption, the nodes usually apply duty-cycling to their radios, which means they repeatedly switch on their radios and sleep. A successful discovery can take place if and only if one radio listens on the channel at the same point in time at which another one sends a packet. As the clocks of these devices are completely unsynchronized, the point in time when two devices meet for the first time is random.

Minimizing discovery latency times is vital for the success of routing protocols in mobile networks. This includes discovering new neighbors and discovering that the existing neighbors have failed because of node mobility. The latency to discover new neighbors is discovery latency, while the latency to discover that nodes have moved out of communication range is failing latency. Both latencies have to be small to ensure scalable routing, as explained by the repair time scaling wall.
The repair time scaling wall [8] occurs because as network size increases, the average duration that a path remains connected decreases, while the average duration to repair a path remains constant. When the average path connectivity interval falls below the path repair interval, the scaling wall is reached. It has been shown that the path repair interval is roughly equal to the link failure estimation delay in the system. Thus, faster link discovery is critical for extending the scalability of routing protocols for MANETs.

Asynchronous networks refer to nodes that have independent duty cycles, and have no information about other nodes in the network. When desired duty cycle is assigned to the nodes, it is very difficult to achieve neighbor discovery. Operating a radio at low duty cycle and ensuring the discovery to be fast, reliable, and predictable over a range of communication is a challenge. Asynchronous duty cycles are advantageous as they reduce collisions. The various neighbor discovery protocols are Birthday protocol, Quorum-based protocol, disco and so on. The results of Birthday protocol and Quorum-based protocol have problems with asynchronous neighbor discovery. [2,6]

Disco is an asynchronous neighbor discovery protocol that allows two or more nodes to operate their radios at low duty cycles, and yet still discover and communicate with one another during infrequent periods without requiring any prior synchronization information [4]. In disco, nodes pick a pair of prime number such that the sum of their reciprocals is equal to the desired duty cycle. Each node increments a local counter with a globally-fixed period. If a node’s local counter value is divisible by one of its prime numbers, the node will turn on its radio for one period.

In [4], the cumulative distribution of discovery latencies for disco is compared with Birthday and Quorum protocols operating at 5% duty cycle. It’s proven that Disco achieves better neighbor discovery than these protocols. Disco protocol is a practical solution to the asynchronous neighbor discovery, and rendezvous problem that works by scheduling radio wake times at multiples of prime numbers, which ensures deterministic pairwise discovery and mimic the discovery latencies. The relationship between slot length, beacon rate, discovery latency,
discovery rate, and duty cycle has been analyzed. The disco protocol has been validated for limited network sizes and duty cycles.

*The objective of this work is to validate the disco protocol in mobile networks, for large network sizes under varied duty cycles.*

### 1.2 Summary of Contributions

In this work, in-depth analysis has been done on the concept of Disco protocol. Since Disco protocol was only validated on limited duty cycles and small networks sizes, the results were based on:

- We implemented and verified that Disco ensures discovery in mobile networks.
- We analyzed both discovery and failing latencies in mobile networks using Disco protocol using network simulator, ns3 for network sizes 50, and 100 nodes under different node speeds, duty cycles, and densities.
- We compare both discovery and failing latencies in networks with fixed mobility speed versus various duty cycles.
- We compare both discovery and failing latencies in networks with fixed duty cycle versus various mobility speeds.

### 1.3 Outline

The rest of our report will be organized as follows. In chapter 2, we discuss in detail regarding the neighbor discovery and neighbor discovery protocols, in particular about disco protocol. We present the design aspects of the disco protocol in chapter 3. In chapter 4 we have briefed the implementation details for mobile networks. We have analyzed neighbor discovery protocol for mobile networks by presenting the simulation results obtained by ns-3 simulations in chapter 5. In chapter 6 and 7 we present the future work and conclusion.
Chapter 2

Background

Neighbor discovery has been a challenge in both sensor and wireless networks. In these networks, neighbor discovery and energy efficiency are significant. So, neighbor discovery is a challenge in mobile networks. In this section, we will discuss some of the existing neighbor discovery protocols and their consequences.

The two neighbor discovery types are synchronous and asynchronous. In synchronous neighbor discovery, nodes in the network know the wake up patterns of their neighbors. On the other hand, in asynchronous neighbor discovery, nodes do not have any information on neighbors’ wake-up schedules [7]. In our work, we will explore more on asynchronous neighbor discovery protocols.

2.1 Birthday Protocol

“Birthday” protocol [2] proposed by McGlynn and Borbash is a probabilistic protocol. Birthday protocol uses random independent transmissions to discover adjacent nodes. In this protocol, the nodes choose to stay in a state (transmit, listen or energy saving) for a time slot. Thus, the node which wants to share discovery message should be in transmit state and node which is intended to receive the discovery message should be listen state. Only when the states overlap in this pattern, the neighbors will be successfully discovered. Energy saving state is an idle state. Although it is an efficient saving-energy protocol for static ad hoc networks, it does not provide predictable rendezvous times and exhibits a long tail for discovery. This protocol is not suitable for network with changing topology.
2.2 Quorum-based Protocol

Quorum-based protocol [6] proposed by Tseng et al. is a power-saving neighbor discovery protocol for multi-hop ad hoc network. In this protocol, a common duty cycle is used by all the nodes in the network. This ensures that nodes know the wake-up time of the node to communicate with. The concept of the protocol is to let nodes send beacons $O(1/n)$ of the beacon intervals. The sequence of beacon interval is divided into sets of continuous $n^2$ where $n$ a global parameter indicating the duty cycle.

In each set on intervals, the $n^2$ intervals are arranged as a 2-dimentional $n \times n$ array in a row-major manner. On the $n \times n$ array, a node arbitrarily picks one column and one row of entries to transmit and receive, respectively, a total of $2n - 1$ intervals in each set of $n^2$ intervals. Since $n$ is a global constant parameter, all nodes have the same duty cycle, which limits the flexibility of the Quorum protocol. Thus, Quorum-based protocol lacks asynchronous duty cycles in the network.

2.3 U-Connect Protocol

U-Connect protocol [1], proposed by Kandhalu, Lakshmanan, and Rajkumar is a neighbor discovery protocol that achieves neighbor discovery at minimal with saving energy costs, in a symmetric/asymmetric asynchronous duty-cycled neighbor discovery. The design is as follows, each node picks a prime number $p$ and stays active for 1 every $p$ slots, and nodes will discover one another. They also exclude the case if prime is 2.

2.4 Disco Protocol

Disco is an asynchronous neighbor discovery and rendezvous protocol that allows nodes discover each other at low duty cycle [4]. Disco ensures deterministic pairwise discovery without requiring global coordination of duty cycles.

This approach is adopted from Sun Zi’s two-millenia old Chinese Remainder Theorem and ensures discovery in bounded time, even if nodes independently choose their prime numbers. The Chinese Remainder Theorem requires co-primes to guarantee a solution to the simultaneous
congruencies [5]. The protocol is as stated, each node selects a pair of prime numbers, such that the desired duty cycle will be the sum of their reciprocals. Each node will stay active if the counter/timer is divisible by one of its primes.

In this protocol, symmetric (all nodes use same duty cycle) and asymmetric (nodes use different duty cycle) prime pairs were considered. Results have shown that asymmetric prime pairs dramatically reduce the discovery latency, by 30-50%. Disco protocol achieves discovery faster than the other neighbor discovery protocols for asynchronous duty cycles, allows nodes to independently select their own duty cycle. Hence, the protocol offers a provable upper bound on discovery latency, and performs better than expected in practice.
Chapter 3

Design and Experimental set up

Disco algorithm randomly chooses a pair of prime number for nodes such that the sum of their reciprocals is equal to the desired duty cycle. In this section, we will illustrate in-depth about the Disco algorithm.

3.1 Design

Choosing Primes

It is more efficient to choose primes such that, the reciprocal of one of the prime is close to the duty cycle, while the other prime number is relatively large. It is ensured that sum of those reciprocals will be close to the desired duty cycle. This is because when one of the reciprocal node’s prime number is close to the desired duty cycle, its wake-up schedule will be more efficient. For instance, consider a network of 1% duty cycle (as in table 3.1), node can choose prime pairs of (139, 479), (131, 457), and so on. The first pair’s duty cycle will get close to a 0.93%, which is close to 1%. As a matter of fact, a node cannot choose a same pair of primes (i.e. a node chooses 211 and 211). Table 3.1 demonstrates the calculation of nodes prime number and its desired duty cycle calculation.

Table 3.1: Examples of a pair of prime numbers for 1%, 10% and 20% duty cycle.

<table>
<thead>
<tr>
<th>Node pair of primes</th>
<th>Desired duty cycle</th>
<th>Sum of the reciprocal</th>
<th>Node’s duty cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>(139,479)</td>
<td>1%</td>
<td>$\frac{1}{139} + \frac{1}{479} = \frac{618}{66581}$</td>
<td>0.928%</td>
</tr>
<tr>
<td>(23,37)</td>
<td>10%</td>
<td>$\frac{1}{23} + \frac{1}{37} = \frac{60}{851} = 0.07$</td>
<td>7%</td>
</tr>
<tr>
<td>(7,47)</td>
<td>20%</td>
<td>$\frac{1}{7} + \frac{1}{47} = \frac{54}{329} = 0.164$</td>
<td>16.4%</td>
</tr>
</tbody>
</table>
When a counter is divisible by one of the node’s prime numbers, the node will turn on its radio and beacons for a fixed period termed as slot duration. Furthermore, if two or more nodes turn on their radios during the same period, then they can exchange beacons and discover each other. As indicated in Table 3.1, the nodes duty cycles cannot be specified with fine granularity, especially with large duty cycles. Terminologies used in our scheme is as follows.

**Definition 1:** Discovery
It refers to the process by which nodes learn about their current one-hop neighbors for the first time.

**Definition 2:** Discovery Latency (DL)
Discovery latency will be the time taken by the node to first discover once they become neighbors. It refers to the delay between the time two nodes are in communication range \(T_n\), to the time when they first discover each other \(T_{b}^{T_n}\).

\[
\text{Discovery Latency, } DL = T_{b}^{T_n} - T_n
\]

where, \(T_{b}^{T_n}\) indicates when two nodes’ radios are overlapping while they are in neighborhood.

**Definition 3:** Failing Latency (FL)
Failing latency is observed only amongst the nodes that have been discovered already. It refers to the delay between the time two nodes are not in communication range \(T_f\), to the time when the two nodes have already learned about each other’s wake-up patterns \(T_{b}^{T_f}\).

\[
\text{Failing Latency, } FL = T_{b}^{T_f} - T_f
\]

where, \(T_{b}^{T_f}\) indicates that when two nodes should be active in the same period but they are not neighbors. Table 2 shows the equations of both discovery and failing latencies.
Choosing Primes

The prime numbers chosen will control the wake-up time of the nodes radio. Choosing pair primes is quite challenging. Let’s consider an instance on how we consider the primes. For simplicity consider 2 nodes. Let the desired duty cycle for the nodes be 1%. The possible prime pairs for the 2 nodes are as indicated in Table 3.2.

Table 3.2: Three different cases when two nodes have a 1% duty cycle.

<table>
<thead>
<tr>
<th>Case</th>
<th>Node 1 primes</th>
<th>Node 2 primes</th>
<th>1st discovery (slot number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(211,223)</td>
<td>(199,229)</td>
<td>41989th slot</td>
</tr>
<tr>
<td>2</td>
<td>(131,383)</td>
<td>(139,457)</td>
<td>18209th slot</td>
</tr>
<tr>
<td>3</td>
<td>(131,479)</td>
<td>(131,487)</td>
<td>131st slot</td>
</tr>
</tbody>
</table>

In case 1, the prime-pairs for node-1 is (211, 223) and for node-2 it is (199, 229). Further calculations on the overlapping wake up times for these nodes indicate that the 1st discovery in ideal scenario happens at time slot number 41989. In case 2, the prime-pairs chosen for node 1 and node 2 are (131, 383) and (139,457) respectively. The 1st discovery can happen most likely if no collision happens at slot number 18209. Furthermore, in case 3, the prime-pairs chosen are (131, 479) and (131, 487) for node1 and node2. This indicates that the 1st discovery could happen at slot 131. Also, if the 1st neighbor discovery fails in case 1 and case 2, for the next discovery the wait time would be longer where as in case 3, the neighbor discovery happens more often.

Choosing a reciprocal of a prime number that is close to duty cycle will ensure discovery. That is, low discovery times are possible if one of the reciprocal of prime numbers is very close to the desired duty cycle, while the other prime number is a larger number. Hence, it is more important to randomize the choice of prime numbers to reduce the chance that two nodes have the same pair of prime number when the desired duty cycle is the same. Since in our network setup we have density of 10 nodes, it’s important we should choose at least 5 primes that are close to the reciprocal of the desired duty cycle, and assign them randomly to the nodes. This technique will ensure more discoveries more than the first two previous instances.
3.2 Experimental setup

In this section, we provide the implementation details for neighbor discovery in mobile networks. We have simulated and analyzed the results using a network simulator, NS-3 on network sizes of 50 nodes and 100 nodes [3].

Parameters

Table 3.3 indicates the parameters considered in our experimental setup. Neighbor check is how often the nodes check for this already discovered neighbors. This pairwise neighbor check is done for every 10ms. Slot duration indicates the time slot the node is going to be active once their radio is turned on. The slot duration is 20ms.

Table 3.3: The experiment parameters.

<table>
<thead>
<tr>
<th>Type of parameters</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor Check</td>
<td>10ms</td>
</tr>
<tr>
<td>Slot Duration</td>
<td>20ms</td>
</tr>
<tr>
<td>Density (neighbor region capacity)</td>
<td>10 nodes</td>
</tr>
<tr>
<td>Mobility Speed (m/s)</td>
<td>2-4 3-7 7-15 15-25</td>
</tr>
<tr>
<td>Duty cycle (%)</td>
<td>1, 10, 20, 30, 50, 100</td>
</tr>
<tr>
<td>Network Size</td>
<td>50, 100 nodes</td>
</tr>
</tbody>
</table>

Density of the network indicates the number of neighbors in its one-hop communication range. We have considered it to be up to 10 nodes. Mobility speed indicates the speed with which the nodes move in the network. It is varied as indicated in the table 3.3. Duty cycle indicates the active time of the node. It is varied between 1% and 100%. 100% duty cycle indicates the nodes are awake for entire duration and 1% indicates its active time is minimal and is sleeping most of the time. The network size of this experiment setup will be either 50 or 100 nodes.
Chapter 4

Results

In this chapter, we analyze the results of disco protocol on 50, and 100 network sizes. We have evaluated the discovery latency for varying duty cycles, mobility speeds and network sizes in a network. Also, we have studied failing latency for varying duty cycles, mobility speeds and network sizes. The analysis is based on the simulation results obtained for 10 simulations.

4.1 Discovery Latency

Discovery Latency is a function of the nodes’ duty cycles, as well as the neighbor check time.

4.1.1 Different Duty Cycles

In this case we are analyzing the discovery latency by varying the duty cycle keeping the mobility constant. Figure 4.1 – Figure 4.4 are results for network size 50 nodes and Figure 4.5 – Figure 4.8 are results for network size 100 nodes. We see two patterns in our results.

1. Discovery latency increases as the duty cycle varies from 1% to 10%
   The prime number pair chosen does not result in duty cycle close to the desired prime number. This is seen when the duty cycle is 10%.

2. Discovery latency decreases as the duty cycle varies from 10% to 100%
Figure 4.1: The discovery latency for 50 nodes network and the mobility of nodes is 2-4 m/s.

Figure 4.2: The discovery latency for 50 nodes network and the mobility of nodes is 3-7 m/s.
Figure 4.3: The discovery latency for 50 nodes network and the mobility of nodes is 7-15 m/s.

Figure 4.4: The discovery latency for 50 nodes network and the mobility of nodes is 15-25 m/s.
Figure 4.5: The discovery latency for 100 nodes network and the mobility of nodes is 2-4 m/s.

Figure 4.6: The discovery latency for 100 nodes network and the mobility of nodes is 3-7 m/s.
Figure 4.7: The discovery latency for 100 nodes network and the mobility of nodes is 7-15 m/s.

Figure 4.8: The discovery latency for 100 nodes network and the mobility of nodes is 15-25 m/s.
4.1.2 Different Mobility Speeds

In this case we are analyzing the discovery latency by varying the mobility speeds of the node keeping the duty cycle constant. Figure 4.9 – Figure 4.14 are results for network size 50 nodes and Figure 4.15 – Figure 4.20 are results for network size 100 nodes. We observe that the discovery latency decreases as the mobility of the nodes increases. We see this pattern because when the nodes are moving with slower speed they remain neighbors for longer time than when they move with faster speed.

In order to ensure this, we have evaluated the percentage of nodes that have been neighbors but not discovered. This Table 4.1 shows that as the mobility increases the percentage of undiscovered neighbors’ increases.

\[
\text{Percentage of undiscovered neighbors, } P_n = \frac{N_{n_{br}}}{N_{n_{br}} + N_d} * 100
\]

where, \( N_{n_{br}} \) - total number of nodes that have been neighbors but never discovered
\( N_d \) - total number of nodes that have been discovered

<table>
<thead>
<tr>
<th>( P_n ) (2-4m/s)</th>
<th>8.568%</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_n ) (3-7m/s)</td>
<td>10.496%</td>
</tr>
<tr>
<td>( P_n ) (7-15m/s)</td>
<td>12.790%</td>
</tr>
<tr>
<td>( P_n ) (15-25m/s)</td>
<td>14.247%</td>
</tr>
</tbody>
</table>

Table 4.1: Percentages of undiscovered neighbors’ nodes for Figure 4.9 network.
Figure 4.9: The discovery latency for a 50 nodes network and nodes’ duty cycle is 1%.

Figure 4.10: The discovery latency for a 50 nodes network and nodes’ duty cycle is 10%.
Figure 4.11: The discovery latency for a 50 nodes network and nodes’ duty cycle is 20%.

Figure 4.12: The discovery latency for a 50 nodes network and nodes’ duty cycle is 30%.
Figure 4.13: The discovery latency for a 50 nodes network and nodes’ duty cycle is 50%.

Figure 4.14: The discovery latency for a 50 nodes network and nodes’ duty cycle is 100%.
Figure 4.15: The discovery latency for a 100 nodes network and nodes' duty cycle is 1%.

Figure 4.16: The discovery latency for a 100 nodes network and nodes' duty cycle is 10%.
Figure 4.17: The discovery latency for a 100 nodes network and nodes' duty cycle is 20%.

Figure 4.18: The discovery latency for a 100 nodes network and nodes’ duty cycle is 30%.
Figure 4.19: The discovery latency for a 100 nodes network and nodes’ duty cycle is 50%.

Figure 4.20: The discovery latency for a 100 nodes network and nodes’ duty cycle is 100%.
4.2 Failing Latency

Failing latency is observed only amongst the nodes that have been discovered already and they have moved out of the communication range when they had to exchange information.

4.2.1 Different Duty Cycles

In this case we are analyzing the failing latency by varying the duty cycles of the node keeping the mobility constant. Figure 4.21 – Figure 4.24 are results for network size 50 nodes and Figure 4.25 – Figure 4.29 are results for network size 100 nodes. We see two patterns in our results.

1. Failing latency slightly increases as the duty cycle varies from 1% to 10%
   The prime number pair chosen does not result in duty cycle close to the desired prime number. This is seen when the duty cycle is 10%.
2. Failing latency decreases as the duty cycle varies from 10% to 100% significantly
Figure 4.21: The failing latency for a 50 nodes network and the mobility of nodes is 2-4 m/s.

Figure 4.22: The failing latency for a 50 nodes network and the mobility of nodes is 3-7 m/s.
Figure 4.23: The failing latency for a 50 nodes network and the mobility of nodes is 7-15m/s.

Figure 4.24: The failing latency for a 50 nodes network and the mobility of nodes is 15-25m/s.
Figure 4.25: The failing latency for a 100 nodes network and the mobility of nodes is 2-4 m/s.

Figure 4.26: The failing latency for a 100 nodes network and the mobility of nodes is 3-7 m/s.
Figure 4.27: The failing latency for a 100 nodes network and the mobility of nodes is 7-15 m/s.

Figure 4.28: The failing latency for a 100 nodes network and the mobility of nodes is 15-25 m/s.
4.2.2 Different Mobility Speeds

In this case we are analyzing the failing latency by varying the mobility speeds of the node keeping the duty cycle constant. Figure 4.29 – Figure 4.34 are results for network size 50 nodes. We observe that the failing latency is not affected by changing the mobility speed.

![Figure 4.29: The failing latency for a 50 nodes network and nodes’ duty cycle is 1%.

![Figure 4.30: The failing latency for a 50 nodes network and nodes’ duty cycle is 10%.]
Figure 1.31: The failing latency for a 50 nodes network and nodes’ duty cycle is 20%.

Figure 4.32: The failing latency for a 50 nodes network and nodes’ duty cycle is 30%.
Figure 4.33: The failing latency for a 50 nodes network and nodes’ duty cycle is 50%.

Figure 4.34: The failing latency for a 50 nodes network and nodes’ duty cycle is 100%.
Chapter 5

Future Work and Conclusion

5.1 Future Work

In this work, we analyzed both discovery and failing latency in mobile networks for network sizes 50 and 100 nodes. In future, we would like to analyze the results for discovery and failing latency on larger networks (>1000 nodes). Bian et al. [9] presented a comprehensive survey about most advanced neighbor discovery protocols in mobile sensing applications. The authors adopted two-level classification. At the higher level, neighbor discovery can be categorized into direct and indirect approaches.

Direct neighbor discovery approach means when a node discovers another node, the two nodes exchange their wake up schedules only. While in indirect neighbor discovery approach, the technique uses a direct neighbor discovery approach as building blocks and exploit the collaboration of direct discovered neighbors to discover new neighbors indirectly. In other words, when two nodes discover each other, they will exchange what their list of nodes which have been discovered. An extend approach to my implementations is to enable indirect neighbor discovery approach.

5.2 Conclusion

After studying Disco; an asynchronous neighbor discovery and rendezvous protocol, which allows nodes to operate at low duty cycle, yet they discover each other regardless of any prior synchronization information. The Disco’s propose has not been yet tested on large networks. It motivates me to analyze this protocol on large mobile networks (50 nodes/100 nodes) using a network simulator (NS-3).
Neighbor discovery has been a fundamental problem in the sensors networks, wireless networks, particularly, in mobile wireless networks. This work verifies that Disco protocol can handle large networks. Analyzing both discovery and failing latencies on those networks will give the network designer a clear picture based on their needs.

Results have been evaluated that Disco can achieve enormous number of discoveries over a short amount of time (100 sec), while discovery latencies on a certain duty cycle can vary with different mobility speeds; still they were very satisfying.
References


