DISTRIBUTED COMMUNICATION IN DISRUPTION TOLERANT NETWORK

by
Jingzhe Du

A thesis submitted to
the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
Computer Science

School of Computer Science

CARLETON UNIVERSITY

Ottawa, Ontario
April, 2012

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-89327-2

Our file Notre référence
ISBN: 978-0-494-89327-2

To Mom and Dad.

Abstract

With the widespread use of mobile wireless devices, a distributed network infrastructure without heavily relying on centralized servers becomes possible when nodes cooperate with each other. There are challenging issues that need to be addressed in this new direction, especially in Disruption (Delay) Tolerant Networks (DTN). We focus our work on three areas of problem study in DTN, including data storage, security and the use of directional antennae in improving DTN performance. In this thesis, we first describe a novel distributed storage protocol in DTN. We define local distributed location regions which are called cells to facilitate the data storage and lookup process. Our protocol resorts to storing data items in cells which have a hierarchical structure to reduce the storage space for mapping related information. A data item is mapped and stored in a node in the lowest level cell using Peer-to-Peer (P2P) techniques. We then describe a novel Distributed Key Establishment (DKE) protocol in Disruption (Delay) Tolerant Location Based Social Wireless Sensor and Actor Networks (DTLBS-WSAN). In DKE, we propose that sensor nodes use neighboring signatures to establish their keys. Pre-distributed keys are used by actor nodes to strengthen communication security. In DTLBS-WSANs, key (certificate) establishment, storage and lookup are performed in a distributed way. Multiple copies of a certificate can be stored at nodes to improve key security and counter network disruptions. After that, we address the neighbor discovery issue when directional antennae are available. We explore the neighbor discovery using only directional antennae first, propose deterministic and randomized algorithms for wireless networks. The deterministic algorithms use knowledge of the vertex coloring for efficient neighbor discovery while the randomized algorithms require knowledge only of an upper bound on the size of the network. Finally, we study the neighbor discovery issue using sensors having two antennae patterns and propose a cooperative approach to speed up neighbor discovery. Nodes use short range omnidirectional antennae to find nearby nodes and use long range directional antennae to find neighbors that can not be found otherwise. Neighboring nodes cooperate with each other to reduce delays and improve energy efficiency.

Acknowledgements

It is my great pleasure to have pursued my Ph.D. studies under the guidance of my supervisors, Dr. Evangelos Kranakis and Dr. Amiya Nayak. They instructed me through the hard course of this challenging process and all the past experiences have become treasure to me today. Their profound knowledge, insights and wisdom have always been, and will remain a source of inspiration to me. I express my sincere gratitude to them. I would also like to thank my thesis committee members, Dr. Paola Flocchini and Dr. Michel Barbeau for their guidance on my thesis research. Thanks also to other members of the thesis examination board, Dr. Yiqiang Zhao and Dr. Parimala Tulasiraman for their insightful questions and comments, all of which have helped shape this dissertation. I must also thank all my friends and family members who encouraged, supported me in many ways throughout the long study period. Finally, thanks is due to the Natural Sciences and Engineering Research Council of Canada (NSERC), and Ontario Ministry of Training, Colleges and Universities.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AOA Angle of Arrival

ARA Antenna Rotation Algorithms

CA Certificate Authority

CCSDS The Consultative Committee for Space Data System

CHT Cell-based Hash Table

CNSA China National Space Administration

CRA-DD Completely Random Algorithm

CRL Certificate Revocation List

DARPA Defense Advanced Research Projects Agency

DD Directional Transmission and Directional Reception

DHT Distributed Hash Table

DKE Distributed Key Establishment

DND Directional Neighbor Discovery

DO Directional Transmission and Omnidirectional Reception

DSR Dynamic Source Routing

DTLBSN Disruption (Delay) Tolerant Location Based Social Network

DTLBS-WSAN Disruption (Delay) Tolerant Location Based Social Wireless Sensor and Actor Networks

DTLSR Delay Tolerant Link State Routing

DTNRG Delay Tolerant Network Research Group

DTN Disruption (Delay) Tolerant Networks

D+O Directional and Omnidirectional Transmission/Reception

ESA European Space Agency

GHT Geographic Hash Table

GLR Geometric Localized Routing

GPG GNU Privacy Guard

GPRS General Packet Radio Service

GPS Global Positioning System

GPSR Greedy Perimeter Stateless Routing

GSM Global System for Mobile Communications

IETF Internet Engineering Task Force

IMEP Internet MANET Encapsulation Protocol

IPNSIG InterPlaNetary Internet Special Interest Group

IRTF Internet Research Task Force

LTP Licklider Transmission Protocol

MAC Medium Access Control

MANET Mobile Ad Hoc Network

MBR Model Based Routing

MEED Minimum Estimated Expected Delay

MF Message Ferrying

MHT Mobile Hash Table

MIMO Multiple Input and Multiple Output

NASA National Aeronautics and Space Administration

OD Omnidirectional Transmission and Directional Reception

OND Omnidirectional Neighbor Discovery

OO Omnidirectional Transmission and Omnidirectional Reception

P2P Peer-to-Peer

PGP Pretty Good Privacy

PKI Public Key Infrastructure

PRoPHET Probabilistic Routing Protocol using History of Encounters and Transitivity

RKA Russian Federal Space Agency

RSRMA Random Selection Rotation Mechanism Algorithm

SBA Scan Based Algorithm

TDOA Time Difference of Arrival

TOA Time of Arrival

UDG Unit Disk Graph

WSAN Wireless Sensor and Actor Networks

WSN Wireless Sensor Network

ZRP Zone Routing Protocol

Part I

Preface

Chapter 1

Introduction

Mobile Ad Hoc Network (MANET) is a collection of autonomous mobile nodes connected by wireless channels without any pre-existing network infrastructure. Typically, some of these mobile devices are part of the network only while they can communicate with the rest of the network. During network operations, there are situations where the communicating nodes may be disconnected (e.g., due to node mobilities). Disruption (Delay) Tolerant Network (DTN) is a specially designed network to address issues like this where intermittent communications and anomalies have the least adverse impact on the network.

We examine the performance improvements of nodes in DTN where these nodes cooperate with each other in a distributed way, in order to release the stringent requirement that centralized servers are needed. The first topic which we consider is how nodes can share information (data items) effectively in such a network setting. We subsequently propose solutions to address the distributed key establishment in the network. Since directional antennae can increase node communication range and improve network connectivity, the effective application of this antenna model is being explored in the thesis research.

1.1 Challenges and Motivation

Most network (including the Internet and MANET) protocols assume network connectivity. These protocols fail to work when network disruptions exist. Although existing communication technology has been trying to improve network connectivity, network disruptions (delays) happen from time to time. The centralized control mechanism further contributes to this situation where single server failure may have great impact on the network performance.

Nodes need to exchange information through network communications. Peer-to-Peer (P2P) has been proposed for information sharing on the Internet. However, existing P2P techniques rely on fast connections among distributed storage nodes and would not work properly when network disruptions (delays) exist. Node mobility can further increase the difficulty of P2P applications. It would be beneficial if nodes can still maintain acceptable data exchange services in DTN.

Security is the basic network function to prevent intentional or unintentional data item modification during information exchange. Network security is based on the proper key establishment. Nodes can either use pre-distributed symmetric keys or public key certificates for security functions such as data confidentiality, integrity and non-repudiation. It is well known that the traditional mechanisms are based on trusted third party nodes (or servers). The implementation of these strategies faces insurmountable difficulties in a dynamic Wireless Sensor Network (WSN) with disruptions and delays.

Directional antennae are known to reach further for the same amount of energy consumed. It can increase connectivity and reduce communication delay in DTN as a result. Although there are widespread uses of directional antennae in data communications, they are mainly applied in fixed settings. Less is known on how can these directional antennae be fitted in a random network. It is not a simple task for a node to find its neighbors using directional antennae and mobility can make this further challenging.

With nodes moving, it is difficult or even impossible for nodes to get the whole network topology information in most situations. Thus any algorithm which uses global topology information to achieve global computation should be restricted to distance k (for some small k) neighborhood of a node in such a situation. As in some DTN applications, a node can get its location information either by Global Positioning System (GPS) or localization algorithms. It can make decisions using the location information of its own and its distance k neighbors.

We propose solutions for both the data item exchange and key establishment mechanisms in DTN through nodes' cooperation in the distributed environment. We are aiming at providing secure network communications at the presence of network

disruptions and delays. In order to further improve the network performance, we propose neighbor discovery protocols when directional antennae are also available.

1.2 Problems of Interest

We focus on four problems of interest pertaining to the network performance improvements in DTN: 1) the distributed storage in disruption tolerant network; 2) the distributed key establishment in disruption tolerant location based social wireless sensor and actor network; 3) neighbor discovery in a sensor network with directional antennae; and 4) cooperative neighbor discovery using two antenna patterns.

1.2.1 Distributed Storage in DTN

With wireless devices being widely used in recent years, the necessity of enhancing the application solutions based on these mobile devices is increasing. However, distributed wireless mobile communications suffer from the lack of infrastructure disadvantages. Frequent network disruptions and delays are integral part of this wireless trend. Although protocols such as Peer-to-Peer information sharing on the Internet provide a variety of network solutions which meet customers' application needs, these applications have to address the adverse impact of network disruptions and long or variable delays before anything can be used in DTN.

One of the essential functions of the communication network is data sharing. Data have to be stored somewhere in the network before others can make use of them. Traditional data sharing is based on centralized servers while the emerging distributed P2P network without relying on server nodes can encourage nodes' participation. These existing solutions assume either direct or end to end communication paths. Although a few distributed storage solutions in MANET can be used in small densely connected wireless networks, they fail to work properly in a DTN setting. Distributed storage in DTN is the basis for further network application implementations which must be addressed.

1.2.2 Distributed Key Establishment in DTN

Communication needs confidentiality and integrity to protect parties involved and prevent intentional or unintentional modifications during data transmissions. Key establishment is the pre-condition for these security services. Over the years, communication nodes mainly make use of centralized server nodes for the proper key exchange among them. Servers can distribute symmetric or asymmetric keys for secure communications, which rely on fast communication links for the key establishment and verifications.

In a dynamic wireless network, the symmetric key approach is hard to implement because new nodes can join in a network even after the key deployment. Since asymmetric key approach needs to verify the authentication of the public keys on the fly, these verification procedures require fast links between the servers and the verifying nodes. In a distributed wireless network, these concerns have to be addressed and alleviated.

1.2.3 Neighbor Discovery using Directional Antennae

Without prior network topology information, neighbor discovery is the fundamental process which has to be applied before any other functions can be performed in DTN. A node periodically broadcasts its existence to others so that its neighbors can implement other protocols (e.g., routing protocol) based on this information.

Consider a network of n directional antennae in the plane. We consider the problem of efficient neighbor discovery in a network of sensors employing directional antennae. In this setting sensors send messages and listen for messages by directing their antennae towards a specific direction. The directional antennae can be rotated by the sensors as required so as to discover all neighbors in their vicinity. In order to reduce network communication delay through directional antennae, neighbor discovery in a (DD) model whereby sensors employ directional antennae for both transmission and reception is the first challenge that needs to be addressed in DTN.

1.2.4 Cooperative Neighbor Discovery with Two Antenna Patterns

Omnidirectional antennae which provide identical power density to all directions in an ideal situation have been extensively used in wireless communications. Omnidirectional antenna based neighbor discovery is the basis for modern wireless ad hoc communications. Due to wireless contentions, omnidirectional antennae are mainly used in situations where the communication ranges are limited.

Directional antenna, on the other hand, can reach nodes in faraway places with the same transmission power because the power is focused in a specific direction and thus is more efficient in DTN. When referring to neighbor discovery however, it is not as simple as omnidirectional antenna because it takes more rounds for neighboring nodes to face against each other and proper antenna orientation also plays an important role in the process. We further envisage the use of omnidirectional and directional antennae together so that the main task of directional antenna is to reach areas that otherwise cannot be reached through omnidirectional neighbor discovery. In the process, nodes can further reduce delay through cooperation. Researches until now have worked on neighbor discovery using only omnidirectional antenna (OO), neighbor discovery using only directional antenna (DD), or neighbor discovery using one antenna type for transmission and the other type for reception (OD and DO). However, it is not clear how the neighbor discovery model where both antenna neighbor discovery types are available ($D + O$) could work, which is an interesting direction to consider.

1.3 Summary of Contributions

In this thesis, we first describe a novel distributed storage protocol in Disruption (Delay) Tolerant Networks (DTN). Since DTNs cannot guarantee the connectivity of the network all the time, distributed data storage and lookup has to be performed in a store-and-forward way. In this work, we divide the network area into cells, where nodes have high probability of moving within these cells. Consequently we use the cells to facilitate the data storage and lookup process. Our protocol resorts to storing data items in the cells which can have a hierarchical structure to reduce the mapping

related information storage at nodes. A data item is mapped to the lowest level cell using Cell-based Hash Table (CHT) and stored in a node inside the lowest level cell using Peer-to-Peer (P2P) techniques. Multiple copies of a data item may be stored at nodes to speed up the data storage and lookup process. The cells are relatively stable regions and as a result, the data exchange overheads among nodes are reduced. Through experimentation using the NS-2 [1] simulator, we show that the proposed distributed storage protocol achieves higher successful data storage and lookup ratios with lower delays and limited data item exchange requirements than other protocols (e.g., MHT) in the literature.

We then describe a novel Distributed Key Establishment (DKE) protocol in Disruption Tolerant Location Based Social Wireless Sensor and Actor Networks (DTLBS-WSAN). We propose the use of neighboring signatures among sensor nodes to establish their keys. And we further strengthen the communication security through pre-distributed keys at the actor nodes. In DKE, guaranteed security can be achieved when actors are connected and cover the network area and nodes can get high security confidence even without actor nodes when the adversary (malicious node) density is small. In DKE, the distributed key (certificate) establishment, storage and lookup approaches are adopted. Multiple copies of a certificate can be stored at nodes to counter the adverse impact of network disruptions (delays) and to improve the key security.

Following that, we present our thoughts on neighbor discovery of the (DD) communication model whereby sensors employ directional antennae with identical transmission/reception beam widths. Our methodology is based on techniques for symmetry breaking in order to enable the sender/receiver communications. We provide both deterministic and randomized algorithms. The deterministic approaches introduce delays in the rotation of the antennae and exploit knowledge of the existence of a vertex coloring of the network, while the randomized approaches require knowledge only of an upper bound on the size of the network so as to accomplish neighbor discovery. In both instances we study the time complexities of the algorithms proposed when the sensor nodes are static.

Finally, we further explore the neighbor discovery issue when both directional and

omnidirectional antennae are available. Nodes exchange neighbor information with omnidirectional antennae, and neighbor information beyond the reach of omnidirectional antennae are collected using directional antennae. In our model, neighboring nodes cooperate with each other to speed up the neighbor discovery process. And three neighbor cooperation mechanisms are presented in the model. Through analysis, we show that the proposed protocol can reduce delays in the neighbor discovery process when the number of neighboring nodes increases following a Poisson distribution and contentions are taken into consideration. Through simulation, we present the improved delay performance and the energy efficiency of the proposed solution when it is compared with other neighbor discovery approaches in the literature.

In summary, our main contributions can be expressed in the following papers: a distributed storage protocol in disruption tolerant network [2], published in the proceedings of WISARN (2010); a distributed key establishment protocol in disruption tolerant location based social wireless sensor and actor network [3], published in the proceedings of CNSR (2011); a paper [4] on neighbor discovery in a sensor network with directional antennae, published in the proceedings of ALGOSENSORS (2011); a paper on cooperative neighbor discovery protocol using two antenna patterns [5], accepted in WWASN (2012). The routing protocols used in the simulations of [2] and [3] are based on the research work during my Master studies. The corresponding papers of that period are: a geometric routing protocol in disruption tolerant network [6, 7], published in the proceedings of WWASN (2009) and the journal IJPEDS (2010); a hop count based greedy face greedy routing protocol on localized geometric spanners [8], published in the proceedings of MSN (2009).

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 provides background materials for our research. Results related to our contributions during my Ph.D. studies are presented in Chapter 3. Chapter 4 provides the details of the proposed distributed storage protocol. Chapter 5 elaborates on our proposed distributed key establishment scheme in disruption tolerant location based social wireless sensor and actor networks. Chapter 6 analyzes the neighbor discovery process in a sensor

network with directional antennae. Following that, Chapter 7 gives details of our work on neighbor discovery using two antenna patterns. Chapter 8 concludes with possible future work.

Chapter 2

Background

We introduce the background material for the remainder of the thesis. We first describe the evolving history of DTN. Then an explanation of main research directions conducted by several major research groups in this field is presented, followed by a list of the DTN characteristics and some well known DTN applications. As a building block of DTN, routing protocols and their classifications are also examined.

2.1 History of Disruption (Delay) Tolerant Network

As the primary military research agency, Defense Advanced Research Projects Agency (DARPA) has been supporting the research in DTN because of possible communication disruptions commonly existed in the military wireless ad hoc networks, as well as in the interplanetary communications. The term "Disruption Tolerant Network" was widely used in their documentation. Kevin Fall [9] coined the term "Delay Tolerant Network" or DTN when he borrowed some ideas from Interplanetary research and applied them in terrestrial communications to enable services among heterogeneous types of networks. It is commonly assumed that the architectures and protocols designed for DTN can fit in both contexts because of their similarities. We adopt this assumption in the thesis research.

2.2 DTN Research Groups

There are several research groups (projects) working on the architectures and protocols of disruption (delay) tolerant network. Delay Tolerant Network Research Group (DTNRG), DARPA [10] DTN and The Consultative Committee for Space Data System (CCSDS) [11] (part of its members was previously known as InterPlaNetary Internet Special Interest Group (IPNSIG)) are three of them. The following

Figure 2.1 was used in [12] to depict the different but overlapping groups of people working on this topic. The core research is to extend the Internet architecture to cater to applications that may face significant delays or disruptions.

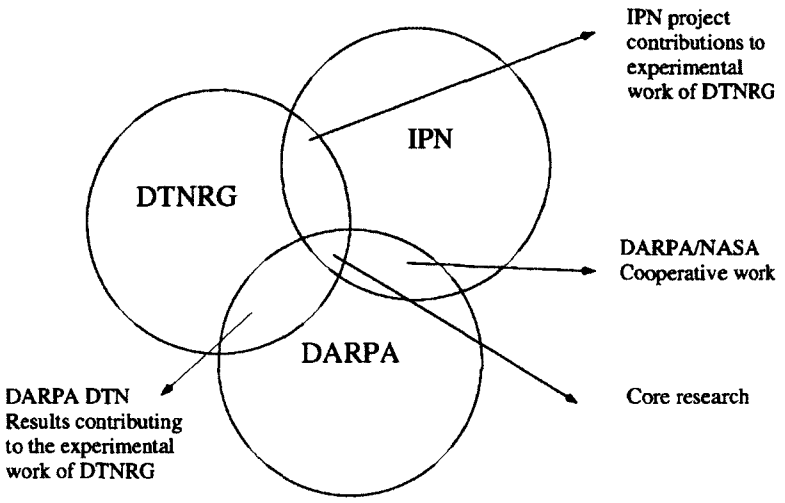


Figure 2.1: Main organizations working on DTN topic.

DTNRG (Delay Tolerant Network Research Group) [13] focuses their work on designing and implementing architectures and protocols for networks, where no continuous end-to-end connectivity can be assumed and which differ in the characteristics of the Internet. DTNRG is one of the main research groups in DTN society which publicly provide their research results. It is the most active research group in recent several years, which is also one of the chartered groups from Internet Research Task Force (IRTF) [14]. The so-called "bundle" protocol [15, 16] and Licklider Transmission Protocol (LTP) [17, 18] were proposed and developed by DTNRG, in order to make them standard DTN protocols in the near future.

2.2.1 Bundle Protocol

Bundle protocol is an overlay network that can run on top of nearly any combination of networks, including the Internet, space network, complex sensor network and other challenging networks. It aims to provide an approximation of end-to-end connectivity without being noticed by the end user. In bundle protocol, each transmitted data unit is called a "bundle" and contains all of the required control information as

well as the application data. It functions like an email by taking a store-and-forward approach. Figure 2.2 shows where the bundle layer is located inside the protocol stack, as given in [16, 19, 20].

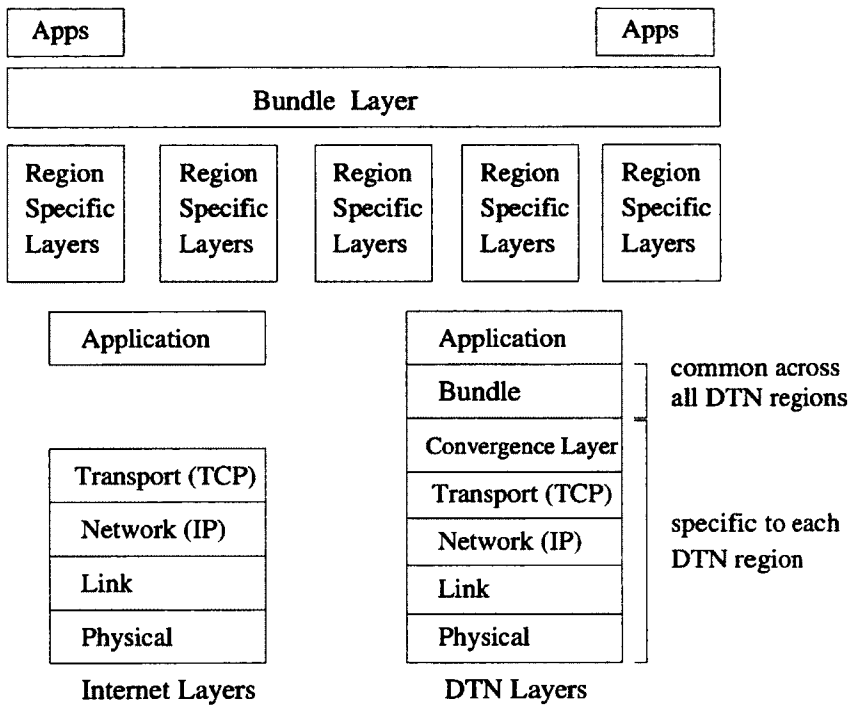


Figure 2.2: Bundle layer.

Bundle protocol uses overlay layer Endpoint Identifier (EID) to distinguish nodes. EIDs can be bound with the Internet addresses at a later stage. It guarantees the successful data transmission through custody transfer, either hop by hop or source to destination. Bundles may be fragmented and reassembled whenever necessary. Security services can also be integrated in the bundle protocol.

2.2.2 Licklider Transmission Protocol

The purpose of LTP is to provide reliability for data retransmission in a communication link characterized by long delays and/or frequent disruptions. LTP can serve as the underlying convergence layer over single hop communication for the bundle protocol. It categorizes data blocks into "red" and "green" blocks. "Red" blocks are those that require acknowledgements and retransmissions if data blocks

are lost while "green" ones are those where no retransmission is necessary to avoid long communication delays and under-utilization of the links.

2.3 Characteristics of Disruption (Delay) Tolerant Network

Traditional networking protocols from the Internet and MANET may not work properly in the disruption (delay) tolerant network because the network possesses at least one of the following special characteristics [19].

- Intermittent connection: end-to-end communication path between source and destination nodes does not guarantee to exist.
- Long or variable delay: long or variable propagation delays may exist due to long distance between nodes or other factors in the communication path.
- Asymmetric data rates: network may be heterogeneous and different parts of the network may use different topology or protocols and thus different data rates are possible.
- High error rates: communication path may be exposed to extreme environments which can lead to high error rates during data transmission.

Decentralized and distributed network can encourage the participation of nodes, however many existing protocols can fail in this setting without central control. To obtain quality communication service, measures should be taken to prevent the possible long delays and frequent network disruptions caused by the distributed communication nature.

2.4 DTN Applications

DTN solutions are necessary because there are many situations where traditional networking solutions do not work well. The use of DTN solution provides an effective way for the proper functioning of many applications which will not work otherwise. In this section, we list some primary DTN applications.

2.4.1 Inter-planetary Applications

Deep space communications [21] is one of the major DTN applications. Over the decades, National Aeronautics and Space Administration (NASA), European Space Agency (ESA), Russian Federal Space Agency (RKA), China National Space Administration (CNSA) and others have been planning a series of human or robotic missions to Mars, the Moon and elsewhere into the deep space. Compared with traditional network, communications in these missions face significant different sets of physical constraints. When human beings want to make further use of the outer space in the future, existing infrastructure based network communication systems would not be sufficient to provide a reliable service. Given the long delays involved, the network protocols in these missions should be designed to meet the special challenges and DTN technology should be explored for this purpose.

2.4.2 Tactical Military Communications

DARPA's DTN program is developing technologies that enable the military personnel in keeping proper communication even if network disruptions exist. Research [22] in this field shows that DTN approach outperforms traditional network solutions in a variety of network disruption scenarios. Even in the worst case situation, DTN approach can still reliably deliver data while traditional approach fails to work.

2.4.3 Lake Water Monitoring

It is a common practice to monitor the lake water quality before the utilization of lake resources can proceed. The lake water monitoring is necessary also because the water pollution in many lakes is becoming a serious problem. Compared with the use of Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS) or 3G mobile communications, the solution [23] of using sensor nodes with DTN communication capability, together with the data mules (e.g., data collection boats) is much more environmentally friendly and the total cost of the lake water monitoring application can be reduced.

2.4.4 Other Applications

The following is a list of some other (incomplete) DTN applications:

1. Internet connection in the developing countries (areas) [24]. In rural areas of some developing countries, the cost of traditional landline Internet connections is a burden to the villagers who want to use the Internet resources for better pricing of their products. The DTN solution, which combines wireless and asynchronous services, can meet the needs of the people in remote villages.

2. Underwater acoustic networking [25]. Acoustic communication among underwater sensors enhances the deep water resource exploration capability. The adverse environment conditions will cause high error rate, long delay and low data rate during communications which are typical characteristics of a disruption tolerant network. With around 71% of earth surface covered by sea water, more efforts should be placed in this field to enable effective DTN underwater application and facilitate the underwater utilization.

3. Wild life tracking [26, 27]. Tracking wild animals under study poses a new challenge because there are no known fixed routes of these wild animals. To collect as many useful data as possible, while at the same time save energy, memory and the cost of the wireless sensors, DTN approach is the viable solution to maintain a reliable wild life monitoring system.

2.5 DTN Routing Protocols

Clearly the routing protocols in DTN should take different approaches compared with the routing protocols in the Internet and MANET. In the survey of Zhang [28], DTN routing protocols are divided into deterministic and stochastic approaches, where deterministic approach means that routing paths can be calculated well in advance. Farrell *et al.* [29] supplemented the above classification with some schemes presented at a DTNRG meeting [30]. Since then, there have been a number of other approaches proposed in the literature. In [6, 7], we have proposed a geometric

routing protocol. We summarize the existing works and list them in the following subsections.

Oracle Scheme This scheme is called deterministic in [28]. This scheme works as if there exists an Oracle, with the full network topology information known in advance. During the routing process, routing paths could be properly calculated or arranged by using the available knowledge. One classic example is the routing arrangement in space networking, where spacecraft ephemeris, locations of the flight control centers for these space missions are carefully planned and known in advance. Thus communication routing paths can be scheduled by network operators. Knowledge of motion profiles approach [31] and Space and time routing [32] all belong to this scheme.

Model-Based Schemes In some situations, nodes in a DTN move in a predictable way. In other words, the statistical information for these nodes can be properly collected. Thus it is possible to build a model of the network according to statistical information and thereby construct a routing protocol with improved performance using the model. Model based routing [33] belongs to this scheme.

Epidemic Schemes To increase message delivery ratio, epidemic scheme can be used. This scheme can reduce delay at the cost of more data storage spaces and message transmissions. A node makes full use of all contacting nodes and forwards messages to every possible next hop although it is unsure which route will work (i.e., no effective differences in the routing can be identified among next hops) in this scheme. Epidemic [34] and Spray and wait [35] belong to this scheme.

Estimation Schemes In this scheme, nodes improve upon epidemic routing by estimating the probability of a successful forwarding according to the history or other available information. A node is chosen as the next hop when it has a higher chance of delivering a message to the destination than others. PRoPHET [36], Utility based routing [37], MEED [38], DTLSR [39] and contact duration-based probabilistic routing (PR_CD) [40] all belong to this scheme.

Erasure Coding Scheme Packets are encoded with redundancy information and divided into data blocks so that the packets could be transmitted in a more reliable manner. Packets can be recovered at the destination node if the number of received data blocks exceeds the recovery threshold even if some data blocks are lost. Erasure coding [41] belongs to this scheme.

Node Movement Control Scheme In this scheme, some mobile relay nodes are added to the network to increase the successful delivery rate. These nodes are called data mules or message ferries. The movement of these data mules are carefully controlled in order to improve the efficiency of this scheme. Message Ferrying [42] and Look-ahead routing [43] belong to this scheme.

Cluster Based Routing Scheme Mobile nodes can form clusters if they have similar mobility patterns. The nodes in the same cluster can then cooperate with each other in the routing process for communication overhead reduction and load balancing. [44] belongs to this scheme.

Geometric Routing Scheme Geometric location information is used to make routing decisions. Nodes construct geometric planar spanners and extract spanning trees from local Delaunay triangulation graphs in the direction from source to destination. Data packets are transmitted along the trees and with high probability they will be delivered with low delay. GLR [6] belongs to this scheme.

Chapter 3

Related Work

3.1 Introduction

Network nodes can communicate effectively with each other through cooperation. These nodes may work in a distributed manner and make their decisions independently. The effective information sharing among them is a necessary network function. In this chapter, we first present how nodes perform distributed data storage actions, both on the Internet and in the MANET. The application scenarios of the existing information sharing mechanisms are also presented.

Nodes in a wireless network need security provisioning during the communications because the transmitted data can be exposed to the receivers within range of the wireless signals. Key establishment is the first step in the security establishment. One of the widely used approach is the pre-distributed keys. Both symmetric and public key establishment solutions have been proposed in the literature. This chapter then focuses the attention on the related work of key distribution solutions in Wireless Sensor and Actor Networks (WSAN).

Compared with omnidirectional antenna, directional antenna can be used in the wireless network to improve transmitter/receiver gains. However, neighbor discovery is the necessary process before any data transmission advantages of using directional antennae can take place. We summarize existing neighbor discovery protocols using directional antennae first. Following that, related works on cooperative neighbor discovery using two antenna patterns are also described in details.

3.2 Distributed Storage Protocols

In this section, we first present the existing related peer-to-peer (P2P) data storage techniques on the Internet. After that we describe the distributed data storage

protocols in Mobile Ad Hoc Networks (MANET).

3.2.1 Peer-to-Peer

Peer-to-Peer (P2P) protocols can be classified into unstructured overlays or structured overlays. Unstructured P2P uses flooding or random walk for data item routing purposes and is hard to scale as a result. In structured P2P, one of the efficient routing algorithms is Chord [45], which uses a ring (circle) structure for routing purpose and has $O(\log n)$ routing table size and worst case routing distance in a network with n nodes.

In Chord [45], node keys are arranged in a circle and divide the circle to chords. Given an m bits key space, it will have keys ranging from 0 to $2^m - 1$. The mapping from addresses (data items) to keys uses hash functions (e.g., SHA-1 [46]). Each node has a successor and a predecessor. For example, in Figure 3.1, the successor of node v_1 is node v_2, and the predecessor of node v_{10} is node v_9 (value k is the key of a data item).

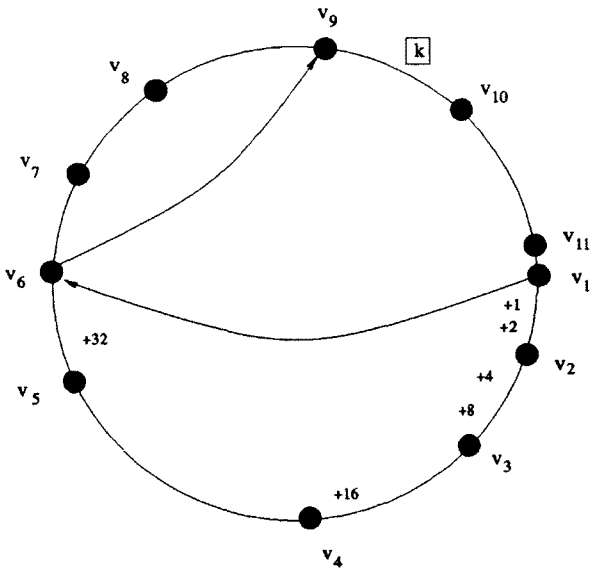


Figure 3.1: Chord routing.

Since the successor node may disappear from the network (because of failure or departure), each node can record a whole segment of the circle adjacent to it, i.e. the

c ($c \in N$) nodes following it, to improve the correctness of protocol operation. For the put (store a data item in the P2P network) or get (locate a data item) operation, routing is needed. In Chord, each node keeps a routing table called finger table and a node has at most m finger nodes in the table. The i^{th} entry in a node j's finger table is the first node with ID that succeeds ($\geq$) $j+2^{i-1}$. In Figure 3.1, the first item in node v_1's finger table is v_2 because v_2 is the first node that succeeds v_1+1, v_2 is also the second item of the routing table because it succeeds v_1+2. The third and fourth item in the routing table is v_3 because v_3 is the first node that succeeds v_1+4 and v_1+8. v_4 is the fifth item and v_6 is the sixth item in the routing table because it succeeds v_1+32 ($32=2^{6-1}$). When node v_1 wants to locate a key k, if node k is a finger node, then v_1 knows the address of the node which stores (should store) the key k. Otherwise, it first finds the closest finger node with ID smaller than k and asks this node to find the correct node. v_1 asks v_6 and v_6 also works the same way as v_1. It finds the closest finger node v_9 (e.g., when v_9 is the first node that succeeds v_6+16) in its finger table and asks v_9 to find the correct node. v_9 knows v_{10} is the owner of k and the locate operation returns its successor node v_{10}.

3.2.2 Data Storage in MANET

With only neighboring connections among nodes, the distributed data storage and look up in MANET is different from the DHT substrate on the Internet. There are several protocols [47, 48, 49, 50, 51, 52, 53] dealing with distributed data storage in MANET.

In [47], a Geographic Hash Table (GHT) was proposed to store data items in wireless sensor networks. GHT maps a data item to a location point and stores this data item at a node closest to this point. To improve the robustness of GHT, the approaches in [50, 51, 52] use the mapping of data items to equal sized rectangular areas or squares. However, these rectangular (or square) area schemes only fit for static networks and too much overhead can be introduced when nodes move around. In Figure 3.2, when nodes in rectangles A and B move back and forth in these two areas, the data items stored in these nodes will change owners frequently in the rectangular (square) area based schemes. Consequently the protocols cannot work

properly in this situation. In [50], the use of a few server nodes to store data items in a mapping region can also cause large maintenance overhead when servers frequently move across regions and server nodes can become bottlenecks when the number of data item look up processes increases.

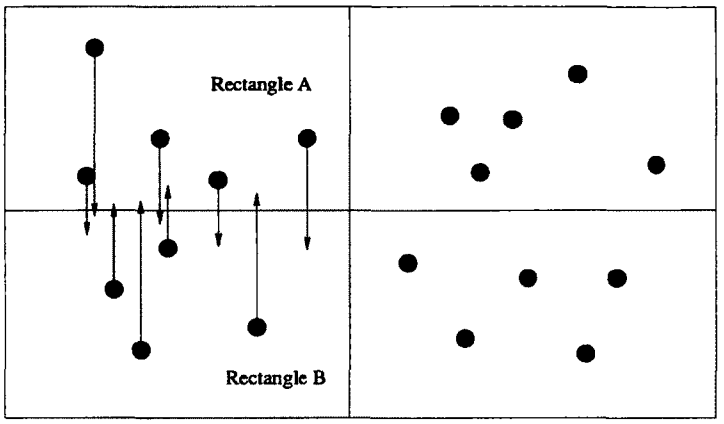


Figure 3.2: Rectangle area based distributed storage.

Similar to the mapping of data items to equal sized rectangular areas or squares, circles can also be used. The authors in [53] proposed the idea of a mobile structured peer to peer network, called Mobile Hash Table (MHT) to facilitate data storage and look up. In MHT, every node is assumed to know its moving trajectory and periodically broadcasts its trajectory information (with position, direction and speed) to all its neighbors. A data item d is mapped to a tuple (position p_d, direction, speed) and this data item is further stored at a node with the "closest matching" pattern. The storage node is the node which can keep the data item the longest time. Assume the data item d is stored at a node n_d and the node communication range is r. Then this node n_d should not be faraway than $\frac{r}{2}$ from p_d. When this node moves out of the circle with radius half its communication range ($\frac{r}{2}$) and with center a data item's matching position p_d, the data item has to be moved to another node. Node n_d chooses one of its neighboring node n_{next} (with position p_{next}) as the storage node, either by selecting the node which can keep the data item the longest time among all neighbors (i.e., $|p_{next} - p_d| < \frac{r}{2}$ is the longest where the direction and speed of n_{next} are taken into consideration), or by selecting a node which is closest to the mapped location if none of its neighbors' distance to the mapped location is within half the

node communication range. As shown in Figure 3.3, a data item with mapping location p_d has to be stored in, e.g., node a within $\frac{r}{2}$ (r is communication range) of p_d such that another node b can find the data item when it comes within $\frac{r}{2}$ of the mapped location. MHT outperforms GHT by storing a data item at a node related with a circle, rather than a node related with a location point.

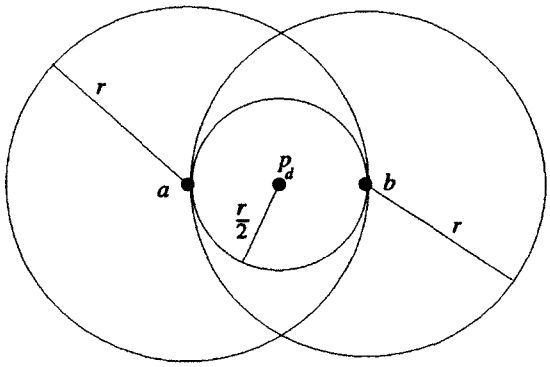


Figure 3.3: MHT data item storage.

The data lookup uses the same technique as shown in the data storage process. A node first maps the data item d to a specified position p_d and then routes its request towards p_d. Once the request message reaches a node in the circle with p_d as the center and $\frac{r}{2}$ as the radius, the request message is broadcasted to all nodes around. It should reach the node with the data item because the data item is supposed to be stored in one of the node that lies inside the circle. According to the mapping schemes in [53], the existing location based mechanisms would not work properly when the area of a region is very large, the number of nodes is small or when the nodes are not uniformly distributed (i.e., no node exists inside the circle around the mapped location). Existing geographic distributed storage protocols use GPSR [54] as the underlying routing protocol, which cannot work properly in a disruption (delay) tolerant network. In a network with long message delays and intermittent link connections, frequent data item exchange in these protocols can further degrade the operation performance.

In [49], the authors proposed Ekta, a DHT substrate in MANET, which is a combination of DSR [55] and Pastry [56]. However, it cannot scale well when the number of nodes increases.

3.3 Distributed Key Management Schemes

Key management schemes are necessary when security services are required in the distributed wireless networks. In symmetric key cryptography systems, the simplest solution is a single key scheme, in which all nodes in a network share a unique key for secure communications. However, once a node is captured, network security is broken. Another extreme scheme is to use pairwise pre-distributed keys. Each node has to store $n-1$ (n is the number of nodes) keys and a total number of $\frac{n\times(n-1)}{2}$ keys are needed for the whole network, which is impractical especially in a dynamic network with new nodes joining in. To solve the above difficulties in key distribution, while still providing reasonable security service level, Eschenauer and Gligor [57] proposed a pre-distributed symmetric key management scheme in Wireless Sensor Networks (WSN). Instead of keeping $n-1$ keys for every node, only a small subset of keys are randomly chosen from a large key pool and the keys stored at every node are significantly reduced. To strengthen the security against malicious nodes, a q-composite key scheme was proposed in [58]. Rather than using one single pre-distributed key as the link key in [57], q keys between two nodes are used to calculate a new link key.

Compared with the symmetric key approach, public key (asymmetric) cryptography [59] provides a security alternative. In Public Key Infrastructure (PKI) [60], a node needs to acquire its own certificate through a Certificate Authority (CA). When another node needs to communicate with this node, it will acquire and verify the certificate by going through the PKI chain until a trusted CA is found. In [61], a new key management protocol was proposed for WSAN, aiming at exploiting the resource abundant actor nodes and reducing the number of keys through a hierarchical approach, which was a combination of symmetric and public key system. In [62], another key management scheme in WSAN was proposed, in which a tree was established for key management with the sink as the root, sensors as leaves and actors connecting them. The reliance on central management makes these existing works vulnerable to attacks.

Since nodes in PKI need to check public key certificates of others, these look up procedures need to address the adverse impact of long or variable delays introduced

by DTN. In [2], a distributed storage mechanism called Cell-based Hash Table (CHT) was proposed to accelerate data item storage and look up, which can be used in facilitating the certificate storage and look up.

Different from PKI, self-generated public/private key pairs without the signature from CA can also be used in communication. GNU Privacy Guard (GPG) [63] and Pretty Good Privacy (PGP) [64] can be classified into this category. Existing algorithms (e.g., RSA [65] and ElGamal [66]) can be used in the key pair generation. In PGP, the author proposed the idea of "Web of Trust", in which a node can collect multiple signatures from multiple third party nodes which are called "trust introducers" for its self-generated public key certificate. A certificate is trusted once the verifier finds it is signed by a trusted "introducer" node. However, once a highly trusted node is broken (i.e., captured), the proper functioning of this scheme is lost. With less social trust relationship, it is difficult for a new node to obtain trust in the network. "Web of Trust" is based on human social networks.

Entity authentication is necessary in wireless communications, which is available through pre-distributed keys. Recently, there have been other approaches [67, 68] in identifying wireless nodes through radio frequency characteristics, which provides an additional layer for communication security through node hardware identity verification.

3.4 Neighbor Discovery with Directional Antennae

There are protocols using directional antennae in neighbor discovery processes. In [69], the authors proposed the gradual increase of directional communication range levels for neighbor discovery purposes. Nearby neighbors are discovered first and faraway neighbors will be discovered at later stages. Directional transmission and reception are used in this work. In [70], a direct discovery protocol and a gossip based neighbor discovery protocol using directional antennae in a static wireless network were proposed. During direct discovery process, a node discovers a neighbor node only when information is received from this neighbor, while nodes exchange their neighbors' location information to enable faster discovery in gossip based algorithm. The protocol tries to optimize the discovery probability in a randomized neighbor

discovery process using directional transmission and reception. In [71], a neighbor discovery protocol which considers node movements was proposed where directions with less possibility of discovering new nodes will be bypassed during neighbor scanning and neighbor discovery frequency is adjusted according to node mobility. It uses directional antenna for transmissions and omnidirectional antenna for receptions. In [72], two Scan Based Algorithms (SBA-D, SBA-R) and one Completely Random Algorithm (CRA-DD) were proposed, which use only directional antennae. In SBA-D, a node decides whether to scan or listen depending on node ID, while a node transmits at one direction or receives at the opposite direction with probability $\frac{1}{2}$ in SBA-R. SBA-D and SBA-R algorithms require perfectly synchronized antenna rotation direction, time and instantaneous antenna rotation to any direction, which are very strong assumptions. In CRA-DD, at each time slot, nodes decide whether to transmit/receive and which direction to transmit/receive completely randomly, which is the simplest algorithm one can imagine and it also requires instantaneous antenna rotation to any direction. In [73], an analytical model was proposed for synchronized 2D neighbor discovery protocols. The model is based on directional transmission and directional reception and a node transmits in one direction and receives in the opposite direction simultaneously.

3.5 Neighbor Discovery using Two Antennae Patterns

3.5.1 Existing Omnidirectional Neighbor Discovery Protocols

There are neighbor discovery protocols using omnidirectional antennae. Neighbor Discovery Protocol (NDP) in Zone Routing Protocol (ZRP [74]) is one of them. NDP is used for the discovery of one hop neighbors. A node repeatedly broadcasts its existence using a hello beacon, which includes the address of its own. Every node maintains a neighbor table to store the information of its neighboring nodes. Another neighbor discovery process can be found in the link/connection status sensing ($LCSS$) function inside the Internet Manet Encapsulation Protocol ($IMEP$ [75]). $IMEP$ uses Beacon packet to broadcast the existence of a node and uses Echo packets for its neighbors to acknowledge the reception of the Beacon packet.

Currently, there are also IETF draft protocols for 1 hop and 2 hop symmetric neighbor discovery ($NHDP$ [76]). $NHDP$ aims at providing connections among nodes with multiple addresses and interfaces. It also uses periodical Hello messages for neighbor discoveries.

3.5.2 Existing Directional Neighbor Discovery Protocols

Section 3.4 has presented some related directional neighbor discovery protocols in the literature. In Chapter 6 of the thesis, efficient neighbor discovery algorithms in a sensor network with one directional antenna for each node are proposed, which can be used in the proposed cooperative solution in Chapter 7.

3.5.3 Existing Cooperative Protocols

There are works of using cooperative diversity in improving wireless network performance. Antenna diversity can be classified into three categories: receive [77], transmit [78] or receive and transmit. In receive diversity, an array of multiple receivers is used and the receiver with the strongest received signal can be selected. In transmit diversity, redundant information over different antennae and symbol times is transmitted which provides diversity gain.

Transmit and receive diversity ($MIMO$) can be considered as spatial multiplexing. It transmits multiple uncoded symbols over different antennae and symbol times. In cooperative $MIMO$ [79], the source and destination nodes form clusters with their nearby nodes respectively in the network. A source distributes its m bits to m nodes in source cluster, which act as distributed transmitting antennae array and send m bits simultaneously to nodes in destination cluster acting as distributed receiving antennae array. The nodes in destination cluster send received signals to the destination for $MIMO$ processing to decode bits sent by source. Cooperative communications can increase coverage area and improve network connectivity [80, 81, 82, 83], with improved network capacity [84].

3.5.4 Existing Cluster Formation Algorithms

In [85], a survey on existing clustering algorithms was given. Some existing cluster formation algorithms require the weight $w(v)$ of a node v, which can be assigned based on node ID [86, 87], energy [88], proximity [89], mobility [90], node degree [87], communication cost or a combination of these [91, 92]. The $DMAC$ clustering algorithm ([93]) can be applied based on such node weights. In the algorithm, if a node has at least a clusterhead neighbor with bigger weight, it will join. Otherwise, it will be a clusterhead. Whenever a node failure occurs (e.g., due to mobility), a clusterhead will remove the corresponding failure node from its cluster and the nodes in a cluster will find new clusterhead if the failure node is the clusterhead.

Part II

Distributed Data Storage

Chapter 4

Distributed Storage in Disruption Tolerant Network

4.1 Introduction

Mobile Ad Hoc Networks (MANET) consist of autonomous mobile nodes connected by wireless channels without relying on pre-existing network infrastructure and the mobile devices are part of the network only while they can communicate with the others in the network. Existing ad hoc distributed data storage protocols usually assume that the network is dense and there is always a connected path from message (data item) source to destination. In situations where network partitions exist, these protocols drop the message if a path could not be found and thus perform insufficiently in terms of data item delivery. Disruption (Delay) Tolerant Networks (DTN) are proposed to address such issues in MANET where instantaneous source and destination node connections may not exist. There are increased DTN applications in recent years, including military communications [22], inter-planetary networks [21], wildlife tracking [26] and intermittent Internet connection in under-developed countries (areas) [24], to name a few.

Peer-to-Peer (P2P) is a decentralized way of networking in which network participants have equal responsibilities and capabilities. Distributed Hash Table (DHT) based P2P protocols [56, 45, 94] are well known for their efficiency in the storing and searching of data items. These protocols offer self-organizing and fault tolerant substrates for decentralized distributed applications. In DHT, an object is mapped to an ID through a one way hash function. DHT has been widely used in distributed data storage and lookup on the Internet through reliable and fast connections among nodes. In a large mobile wireless network with disruptions and delays, these existing protocols can fail to work.

Geographic data storage has been studied in MANET. Nodes could get their

location information either by Global Positioning System (GPS) or localization algorithms [95]. In existing geographic data storage schemes [47, 50, 53], a node makes data storage decisions according to the mapping of a data item to a specific location. During the storage process, a forwarding node sends this data item to the specific location according to the location information of its neighboring nodes and the data item is stored at the node which is closest to the mapped location. The lookup action is similar to this storage process. Since contemporaneous source to destination node connections may not exist in DTN, network disruptions have to be properly dealt with when geographic data storage is applied on the network. And since different DTN networks have different network characteristics (e.g., different movement patterns), a simple mapping approach (e.g., existing schemes) can introduce too much overhead and thus is not the best choice in dealing with different situations.

4.1.1 Contributions and Organization of the Chapter

In this chapter, we propose Cell-based Hash (mapping) Table (CHT), a novel distributed storage scheme in DTN. CHT maps a data item to a region, called cell. Cells are defined in such a way that node movements inside cells are far more frequent than the node movements crossing cell borders. CHT mapping works in a hierarchical manner. When a node wants to store a data item, it first needs to map this data item to the highest layer (level) cell where the data item should be placed. This data item is mapped either layer by layer down to the lowest level cell using only local information, or to a certain layer cell which is different from the cell of source node where the mapping stops. Then the data item is routed towards the available different cell by using DTN routing protocols, e.g., Geometric Localized Routing (GLR) [7] algorithm. Once it reaches a node in that cell, this node can then further map it down to a lower layer. A data item is mapped and stored at a node in the lowest level cell using DHT. Data item lookup request mapping works in the same way. Location diffusion is only performed in the lowest level cell or at most in several upper layers, no global location diffusion is needed to reduce the storage overhead.

We present the formal algorithm and compare it with MHT [53] (using DTN

routing protocol instead of GPSR [54]) and show that it is advantageous in communication overheads, delays and data item storage and lookup success ratios.

The rest of the chapter is organized as follows. Section 4.2 elaborates on our proposed solutions. Section 4.3 describes the details of experiments and analysis. Section 4.4 concludes with possible future work.

4.2 Distributed Storage Algorithm

Although the initial goal of the Internet is network robustness, problems have begun to emerge because of the increasing central management and the distributed communication nature (e.g., network congestion during information retrieval). Disruption tolerant networks, on the other hand, take this distributed nature into consideration at the very beginning. DTN considers node movements, distributed nature and possible unreliable connections.

Existing network (including the Internet) with DTN features can be integrated into a location based distributed storage mechanism, which is necessary for useful information storage and lookup. Although the existing Internet is IP based, authorities know which IP addresses (or address spaces) are mapped to where and individual organizations are responsible for their IP address location mapping.

We assume wireless connections are widely used in DTN. We propose the idea of using a distributed peer-to-peer (P2P) solution to counter network disruption (delay), without relying on centralized servers or super nodes. In DTN, there are long distance connections, as well as short distance connections. In reality, nodes in a network could have various connection interfaces, bandwidth or communication range (in other words, nodes are heterogeneous). Various fast connecting links should be used and considered besides wireless links to reduce communication delay in the network. Different wireless range links need to be considered also. As a special network scenario, nodes only have wireless channels and the antennae are omnidirectional.

Our proposed solutions use a mapping function $f(name) \rightarrow cell$ (we use the same symbol f in the following discussion which may take different parameters) to facilitate distributed data item storage and lookup in DTN. A data item first maps to a cell, then from a cell maps to a node in the cell. If a one to many mapping is necessary,

the mapping cells (or nodes) are better distributed evenly to counter network delay and disruptions. The mapping function $f(name) \rightarrow cell$ is also required to map a node name to a cell.

The following goals are kept in mind in designing the proposed solution: *high successful storage ratio*, *high successful lookup ratio* and *limited maintenance overhead*. Compared with the existing work (MHT [53]), the proposed Cell-based Hash (mapping) Table (CHT) achieves better successful storage and lookup ratios in DTN due to the adoption of cells. It also requires less maintenance overhead in a high mobile environment because frequent data item exchanges are avoided by storing a data item in a cell and allowing storage nodes keeping data items in a Peer-to-Peer distributed way as long as they move within the mapping cells. When multiple copies approach is adopted, the success ratio of CHT can be further improved with reduced data item lookup delay.

4.2.1 Delay-Tolerant Distributed Storage and Lookup

Network Partitioning

In the proposed solution, the network area is divided into cells (regions). Cells are not necessary to cover the whole area (gaps are allowed if it is impossible that there will be nodes in these gaps or nodes in these areas will not stay for long). Nodes can define their own cells with a variable size. The random sized area description can be stored in an approximate way (e.g., by scaling down a ratio) to save the storage space. Cells may be flat or with multiple layers. One example of the multiple layered cells is depicted in Figure 4.1. The layer of the three cells A, B and C inside a region is higher than the three sub-cells a, b and c inside cell A. There are five sub-cells inside c and a lowest layer cell α is inside sub-cell 3 of c. Nodes inside a cell will stay in it with high probability (e.g., $\geq 1 - \frac{1}{n}$ when the cell has n nodes). Region definition is a slow changing mechanism. Once the border of a region changes, this information is broadcasted to all other regions in the same layer, within the same upper region. A node does not need detailed global information to communicate with nodes in faraway places. In this way, local information (with very limited global information) is used to achieve global communication without relying on centralized nodes in the proposed

solution. A node outside a region can use the mapping function $f(name) \rightarrow region$ to find the region of interest and a node inside this region can use the mapping function $g(name) \rightarrow ID$ to locate the node which is responsible for storing the data item with corresponding name. As a special case, the network may be divided in such a way that cells can follow some specified shapes (e.g., hexagon, rectangle, etc.), where the required storage space for area description is greatly reduced.

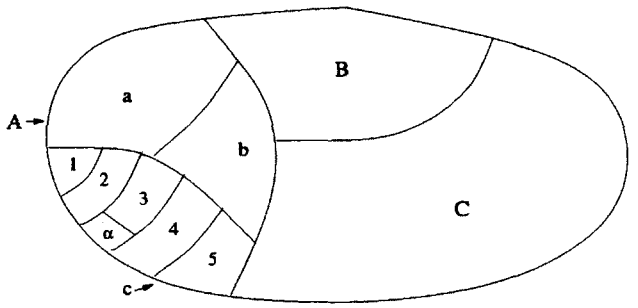


Figure 4.1: Location-based cells.

Different ways for network partitioning may exist. One of them is to make use of existing or predicted network characteristics. If there are high probabilities that nodes will move within certain areas, then the network partitions can be made based on this information, using either a flat (one layer) or a hierarchical structure, depending on the network size. In DTN, there are works on model based [33] and history based [36] routing protocols. Model and history statistical information can also be used in cell formation process. For example in Figure 4.2, we can partition the network region into four cells A, B, C, D based on the statistical information. When there are no nodes staying in E for extended length of time, we do not need to define it as a cell. And overall, the cells are not covering the whole network area. When nodes tend to move within their specific areas (regions) with high probability, a cell based data storage scheme is more robust with less message exchange overhead. Only when a node moves out of a cell, will message exchanges be necessary.

Layering

Layered regions are necessary when there are too many nodes distributed in a vast area. In the layered cell approach, the mapping function is $f(t,\ name,$

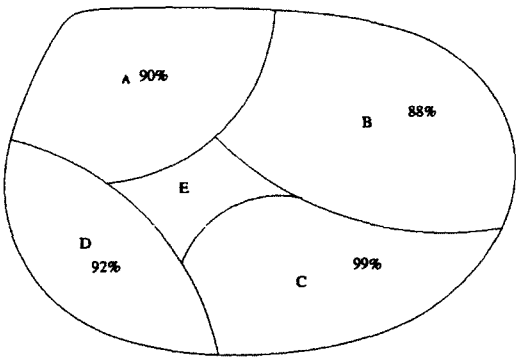


Figure 4.2: Model-based cells.

$parameter_1$, $parameter_2$, $\cdots$, $parameter_{n-1}$) $\rightarrow$ *lowest layer cell*, where the network has n layers with $layer_1$ the top layer, t is time, *name* is the data item (or node) name and $parameter_i$ is a layer specific parameter. The mapping function $f(t, name, parameter_i) \rightarrow layer_{i+1}$ means that by using layer specific parameter, a name could be mapped to a lower layer at time t. This mapping parameter may only need to be kept in $layer_i$, within the framework of the upper layer. In Figure 4.1 for example, a node in the lowest layer cell α can first map a data item to one of the three cells A, B or C inside the region. The mapping function can be $f = hash(data\ item\ name)\ mod\ 3$ (assume the whole region is a $layer_1$ cell) and 3 can be the layer specific parameter. If this data item maps to either cell B or C, nodes in those cells can further map it down. If it is mapped in cell A, a node in that cell then further maps the data item to a lower layer. When this data item is mapped into cell c, the node can use the mapping function $f = hash(data\ item\ name)\ mod\ 5$, where 5 is the corresponding layer parameter. This parameter needs only to be kept at nodes inside cell c. The boundary of a region in $layer_i$ is specified in a way that causes less nodes movement across region boundary, compared with nodes movement inside the corresponding region. In the proposed solution, the higher the layer, the less possible boundary changes are needed. It is apparent that most of the time lower layer local change would not affect faraway nodes in this way. Assume each layer cell has m sub-cells and the storage space for mapping related information (cell description) is the same, then a node only needs to store $m \times (n-1)$ items of mapping information when there is one $layer_1$ cell, rather than m^{n-1} items of mapping

related information. The number of cell descriptions without hierarchical approach is shown in Figure 4.3. Figure 4.4 shows another application example of the layered cell approach. In the figure, a city is the highest layer cell, districts are the middle layer cells and organizations are the lowest layer cells among the three layers.

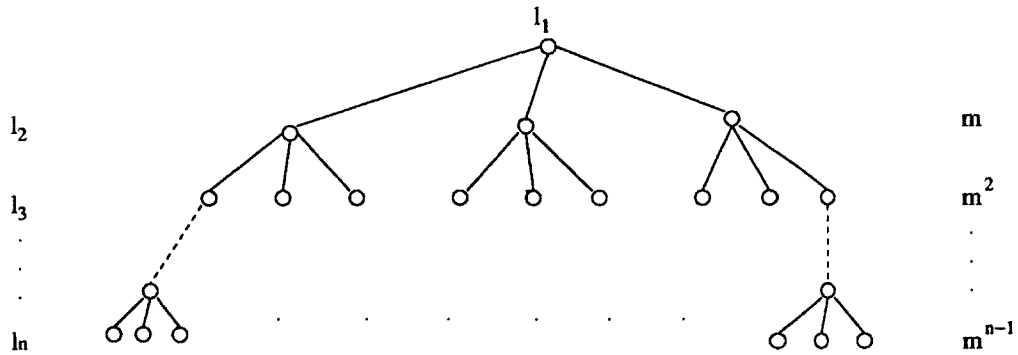


Figure 4.3: Cell descriptions at each layer.

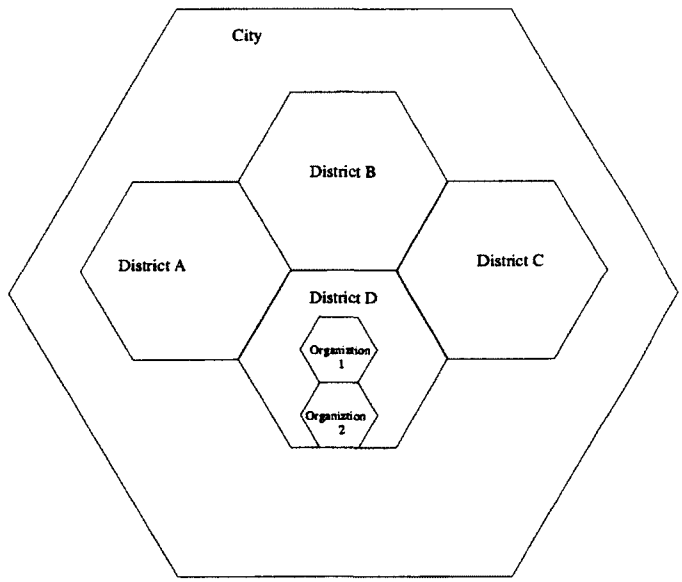


Figure 4.4: Layered cell application.

CHT with DHT

The proposed solution of distributed storage which uses P2P mechanism is called Cell-based Hash Table (CHT). In CHT, a cell is divided in a single layer if the area

and nodes number is small and multiple layers may be adopted if nodes number and its corresponding areas are large. In CHT, cells are used for data storage. Inside the lowest level cell, a data item is mapped to a node (or nodes if a one to many mapping is used) with closest matching ID(s) in the cell. Some existing DHT protocols [56, 45] can be used if nodes move within their cells with high probability, with the consideration that accurate data item to node mapping may not be complete because of the nature of DTN. A node with a specific data item will handover the item to another node in the cell when it moves out (crossing lowest cell border). If the probability that nodes move across cell borders is not low, then a simple balanced storage mechanism (e.g., only balance the number of data items stored at nodes without using DHT) may be a good choice.

The combination of CHT with DHT limits the use of DHT overlay to a small scale within the lowest layer cell. To further improve routing efficiency, we propose the use of Mchord (modified chord, which uses the ring structure of Chord) if the underlying DTN routing protocol is GLR. In our Mchord scheme, a data item is mapped to an ID and it is supposed to be stored at a node with ID closest to its own. If a node which stores the data item meets a neighboring node with a closer ID to the data item, it will give this data item to its neighbor. In Figure 4.5, when node 1 which has a data item with ID 26 meets another node 15, it will give this data item to node 15 because $|26 - 15| < |26 - 1|$ (i.e., the distance between 26 and 15 is less than the distance between 26 and 1 because nodes 1 to 26 are arranged in a circle). There are two modes in our approach, one is reactive and the other is proactive. In reactive mode, data item and lookup request are only sent to a neighboring node when the neighbor ID is closer to the stored data item ID (or request ID). While in proactive mode, a data item or request is always routed to a node with the closest ID according to the local knowledge of a node. Depending on the priority of the data item or request, the reactive or proactive mode can be selected.

Mapping

Two different rules exist for data items and nodes. A data item should stay at its mapping cell(s), while a node is allowed to move around. If nodes move fast and

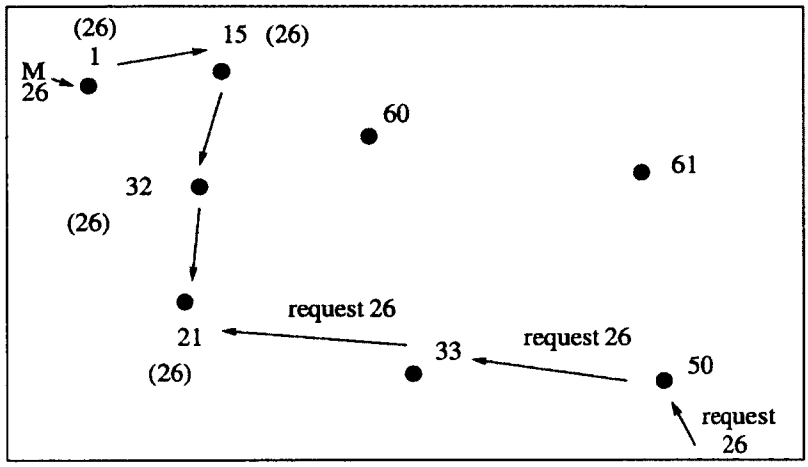


Figure 4.5: Mchord storage and lookup.

change their cells frequently, both CHT and DHT mapping should only be loosely coupled to reduce message exchange overhead. If nodes are relatively stable, mapping could be accurately coupled. For a node crossing multiple regions (large or small), it either can treat the mapping location as its home location and check regularly (when it requests a data item at that original location, as shown in Figure 4.6), or its ID can be mapped to more regions and thus the reply for a request should be sent to multiple mapping regions. The extreme situation is one cell, then it is the same as traditional P2P.

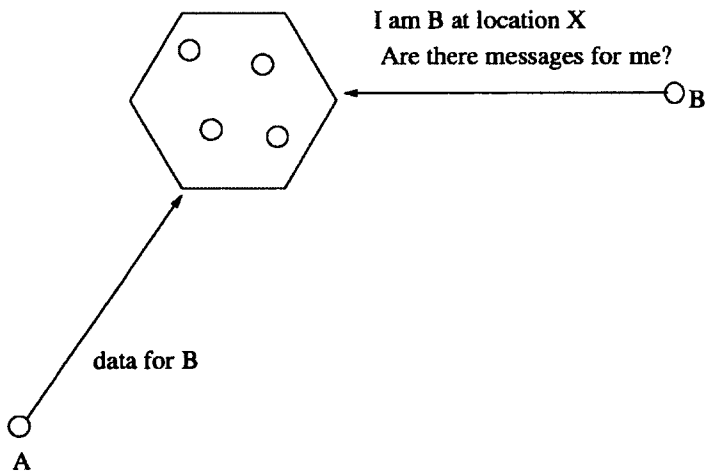


Figure 4.6: Data lookup.

Storage with Multiple Copies Option

Definition 1. *(**Cell Center**) We define the traffic hub (the point where most cell communications pass through) in a cell as the cell center. Given the coordinates of a cell center, a node knows its relative direction with respect to the cell center. A cell center is indeed a communication center.*

When a data item (or request) message forwarding is necessary, this message is first sent to a node in the cell with the shortest distance to the destination cell center. In Figure 4.7, when node a_1 in a $layer_i$ cell with center c_1 has a data item for another $layer_i$ cell with center c_2, it needs to route the data item towards c_2. However, once the data item enters into the $layer_i$ cell with center c_2 and stores at node a_2, a_2 will decide the $layer_{i+1}$ cell with center c' where this data item should be placed. Similarly, a node a_3 in $layer_{i+1}$ cell with center c' will decide the $layer_{i+2}$ cell, where the data item will finally be stored at a node d.

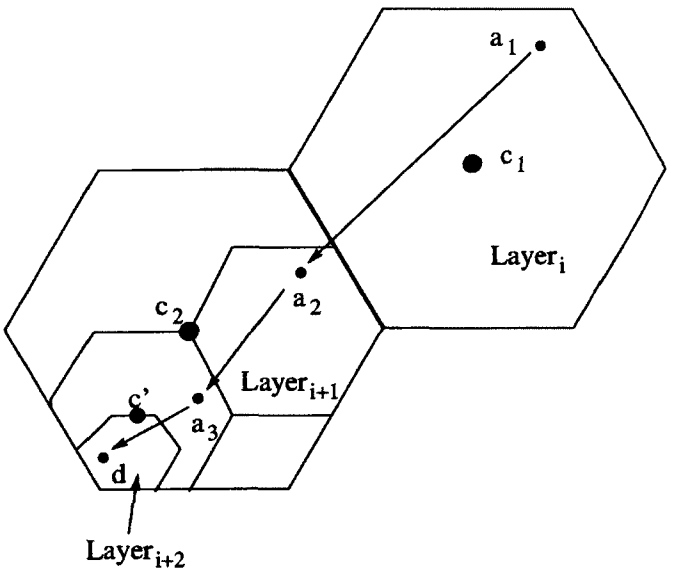


Figure 4.7: Multiple layer storage.

In CHT with Mchord, a data item can always be kept by a matching node so long as this node moves within the mapping cell. When a node stores a data item whose mapping ID is the same as its ID, there is no need to switch this data item with other nodes unless it leaves a cell. Even if a node stores a data item whose mapping

ID does not equal to its ID, it would not give this data item to others unless another node with closer ID to the data item is found. When two nodes have the same distances (|node ID - data item mapping ID|), the node with smaller ID is chosen as the store node. This process converges to the node with the closest matching ID to the data item. In MHT however, a data item needs to be frequently exchanged among nodes because of the nature of mobility and its mapping mechanism. Due to the above reason, the maintenance overhead of CHT is less than that of MHT.

Multiple copies approach can be used in the proposed DTN P2P storage. In traditional P2P network, some protocols have proposed multiple copies approach. DKS(N, K, f) (f is a replication factor) [96] and Tapestry [94] are two of them. The need of multiple copy approach is further necessary because of the characteristics of the DTN.

Data Lookup

When a node performs data lookup, it first maps the data item to a cell identifier (certain layer), and then use any possible connections to route its request towards this cell according to the lookup mapping. The node in that certain layer further maps the data item lookup into its sub-layer and forwards the lookup. If mapping is loosely coupled, data item lookup should be performed in the following order: first the mapping node in the cell, then any other node in the cell which has the data item and finally, nodes in surrounding cells should be checked. For nodes in a cell, if they perform P2P storage, their view of network may be incomplete. So counter measures in lookup are necessary even if accurate mapping is adopted. A step by step option similar to the loosely coupled situation should also be used.

In DTN, data item lookup uses the closest node to the cell to store request upon partition. It will be kept in a node that does not have a closer neighbor to the cell temporarily. Alternative ways (e.g., face routing) can also be used to send the lookup request to the cell. In a certain region, it is possible that some nodes may move out temporarily and other nodes have to store all their data items. When the moving out nodes return, the nodes with the data items have to distribute data back to them. Another approach for nodes is to keep data items while away if they will come back

for sure, so as to save some data exchange cost. If this approach is adopted, the lookup request should be able to tolerate the incurred delays.

To accelerate data lookup, cache is used. Nodes use cache to temporarily store their newly forwarded data items. If a new request matches one of the data items in the cache, no further lookup is needed. However, the cache size cannot be large. There are tradeoffs between cache size and the number of stored messages.

Location Diffusion

We assume nodes know their location and time. For every node, its description could be a tuple (location, time, movement pattern). When nodes meet each other, they will exchange (location, time) pairs. If there are multiple layer cells, all nodes in a lowest level cell report and store each other's location together with time stamp through location diffusion. In case there are two location reports concerning one node, the newer report prevails. This location information provides a node with an overall picture of who is in the lowest layer cell and can be used in the data storage and lookup process.

We show the difference of Chord and Mchord which makes use of location diffusion in Figure 4.8. In Chord, when node 1 wants to find the data item with mapping ID 22, it asks node 17 in its finger table first. Node 17 asks node 20 and then from node 20 finds node 23 as the storage node. Node 20 needs to notify node 1 of this information and node 1 also needs to route request message to node 23. Since instant connection may be impossible in DTN, these several rounds looking up can introduce long routing delays. In Mchord, a node simply finds the closest ID in its location table and routes the request to the node. If this location table is accurate, it surely saves routing time. Since Mchord can make use of DTN routing location information and can be easily integrated with the routing protocol (i.e., GLR [7]), the number of message exchange is reduced.

4.2.2 Analysis of Multiple Copies Approach

Multiple copies approach is useful in accelerating data item storage and lookup. In order to optimize the algorithms and protocols, we calculate the effects of the

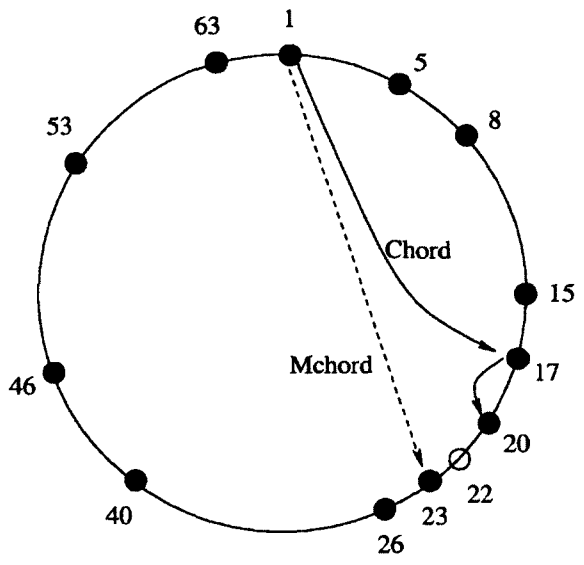


Figure 4.8: Chord and Mchord lookup.

multiple copies approach. Assume every step uses an equal sized time interval T, a node is in the same state (alive or die in a certain cell) at every step and the probability $p = Pr[a\ node\ is\ alive\ over\ time\ interval\ T]$. We store a data item in s nodes and calculate the probability $Pr[item\ still\ exists\ in\ a\ node\ after\ k\ steps]$. If the probability that a node is alive is p, then the probability that a node either dies or moves out (of a cell) is $1-p$. If s copies are stored at nodes, the probability that an item exists in at least a node after k steps is $p_e = 1-[(1+p+p^2+...+p^{k-1})(1-p)]^s = 1-(1-p^k)^s$. We use OCTAVE [97] to plot this probability with varying k, s.

We plot figures with 0 to 20 steps and 0 to 20 duplicate copies. The probability that a node is alive is 0.8 (Figure 4.9) and 0.9 (Figure 4.10) respectively. It is clear that multiple copies approach can improve the chances that a data item stays in a cell.

The figures with 10 and 30 duplicate copies are also plotted with various probabilities that nodes are alive, ranging from 0.1 to 1. The probability that a data item stays in a cell also improves when the probability that a node is alive increases. The results are shown in Figure 4.11 and Figure 4.12.

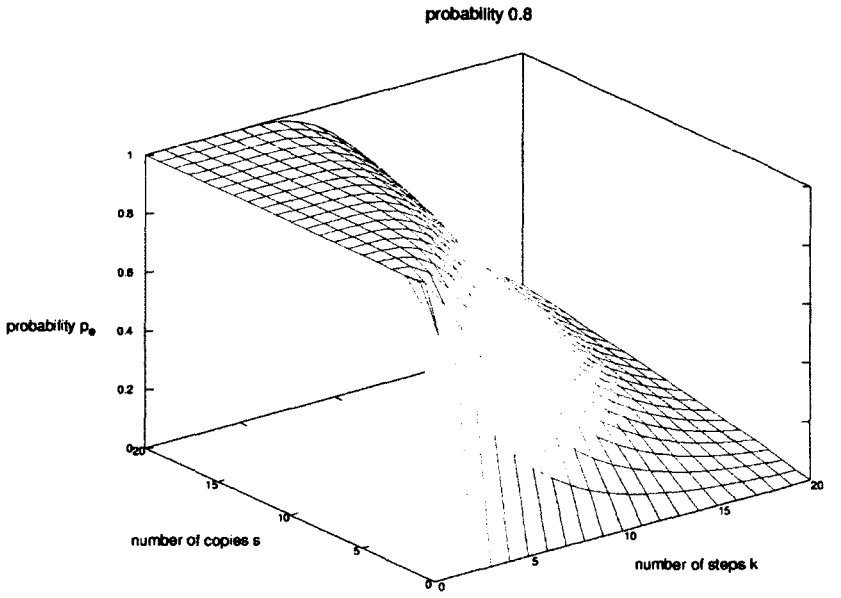


Figure 4.9: Multiple copies with 0.8 alive probability.

4.2.3 Operational Procedures

The proposed solution can be used in DTN applications, which include data item publication and storage, specific data item lookup and data item browsing. A node can publish its own data item and store it in DTN using CHT. If a node knows the name of the data item, it can map it to the specific cell (region) according to the working procedure of CHT. It is still possible that a node wants to browse available names of data items and find its own item of interest. In this case, we introduce a special type of name, called item inventory, which stores records concerning available data items. This name is reserved and can be mapped to a cell, just like other data item mappings. However, we emphasize that this inventory may not be complete in DTN. A record in the data item inventory includes the name and the data item description. Data item inventories can be classified into different levels. If a data item is mainly used locally, it needs only to be recorded in the local data item inventory.

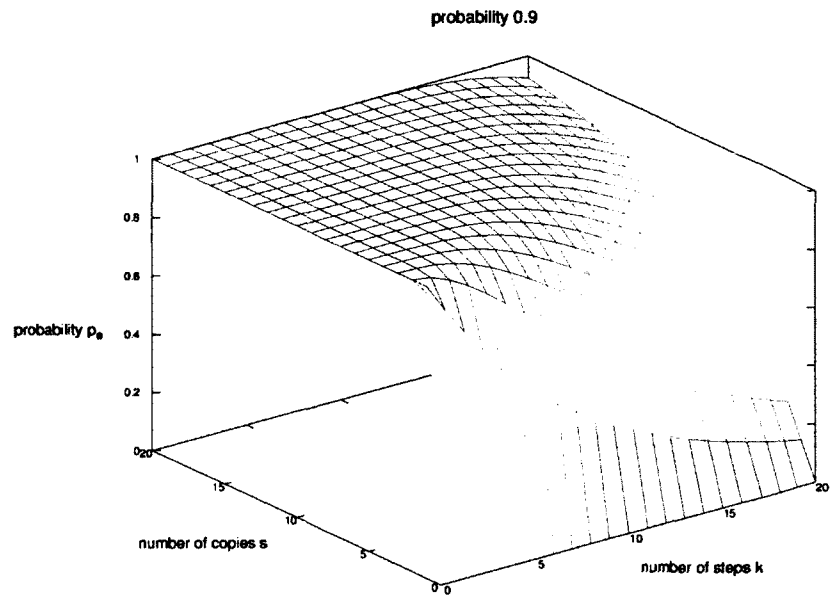


Figure 4.10: Multiple copies with 0.9 alive probability.

Nodes can also check global data item inventory to find data items that interest them. Due to the distributed nature of the protocol, records of the data items are mainly stored at the inventories in their locality.

In a distributed environment, data item synchronization can be used to prevent data item lost. A data item source node can periodically check the existence of the data item, in case a data item cannot be found within reasonable time delay, it can propagate its data item to the storage site.

4.3 Experimental Evaluation

In order to evaluate our CHT distributed storage strategy, we perform simulations to compare CHT with MHT in DTN. During the experiments, we pay great attention to the key attributes, including maintenance overheads, data item storage latency and success ratios of the distributed storage and lookup.

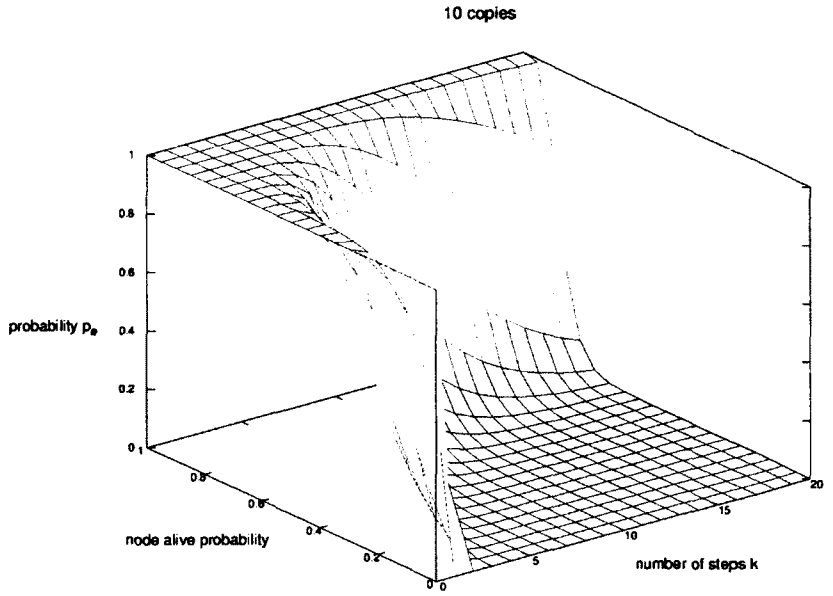


Figure 4.11: Probability figure with 10 copies.

4.3.1 Simulation Environment

The CHT is implemented using the NS-2 [1] simulator. This simulation environment includes full simulation of the IEEE 802.11 physical and MAC layers, which makes the simulation better reflect the real world. A *Random Waypoint Model* [98] is chosen as the motion pattern. For the propagation model, we have chosen *Two Ray Ground* which considers both the direct path and a ground reflection path. The simulation parameters are shown in Table 4.1. The simulation time is 1000 seconds, through which we can clearly identify the performance differences between the CHT and MHT.

Through simulation, we show that data item maintenance overhead is significantly reduced if a data item is mapped to a cell instead of mapping to a circle around a location point. The delay performance and the success ratios for storing and searching a data item in a cell in CHT are also better than those in MHT.

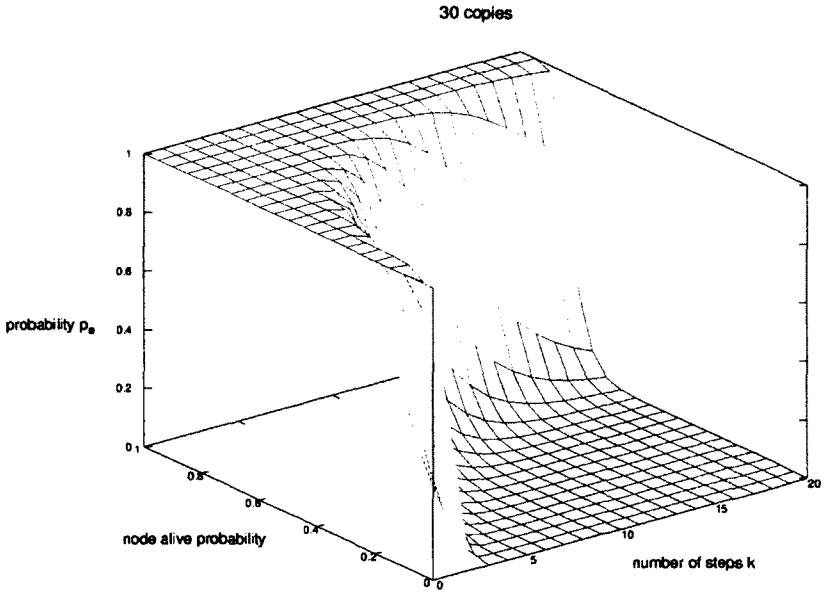


Figure 4.12: Probability figure with 30 copies.

For the simulation results, all points in the figures, as well as numbers in the tables are obtained as an average of 10 different runs with 10 different network topologies and movement patterns. The confidence intervals (t-distribution) for the numbers are calculated at 95% confidence level.

4.3.2 Single Cell Data Storage and Maintenance

We implement CHT with Mchord proactive mode on top of GLR routing protocol (GLR single copy approach is used for accurate maintenance overhead calculation). Although initially we want to compare CHT with original MHT which works on top of GPSR routing protocol, simulation results show that the original MHT can only achieve 6%±2% successful data item storage ratio in DTN. As a result, MHT on top of GLR is implemented for comparison with our proposed solution. We store 20 data items at each scenario in both CHT and MHT. Our simulation results show that

Table 4.1: Parameters of the simulations.

Parameter	Value
Number of mobile nodes	50
Mobility	0-20m/s (default), 0-50m/s
Transmission range	100m
Data rate	1 Mbps
Propagation model	*Two Ray Ground*
Simulation time	1000 seconds
Link layer queue length	150
Topology size	1500m×300m
Pause time	0 seconds
Packet payload size	1000 bytes
Antenna model	Omnidirectional

CHT experiences significant less data item handovers compared with that of MHT (one handover means from source node to the mapping node or from one qualified store node to another qualified store node), as shown in Figure 4.13. Table 4.2 shows the average data item storage delay (the latency when a data item first reaches a qualified storage node). It is clear that storage delay in CHT is also less than that of MHT. In the simulation, all data items have been properly stored at their storage sites in CHT while the success ratio in MHT is only 91.5%±4.85%.

Table 4.2: Data item storage delay.

Protocol	Delay (seconds)
CHT	15.07 ± 2.14
MHT	81.92 ± 14.65

4.3.3 Single Cell Lookup Success Ratio

The lookup success ratios of CHT single cell and MHT at different maximum nodes moving speeds are also evaluated. Lookup actions are performed after 20 data items are stored in the mapping nodes. CHT uses accurate lookup mapping in the

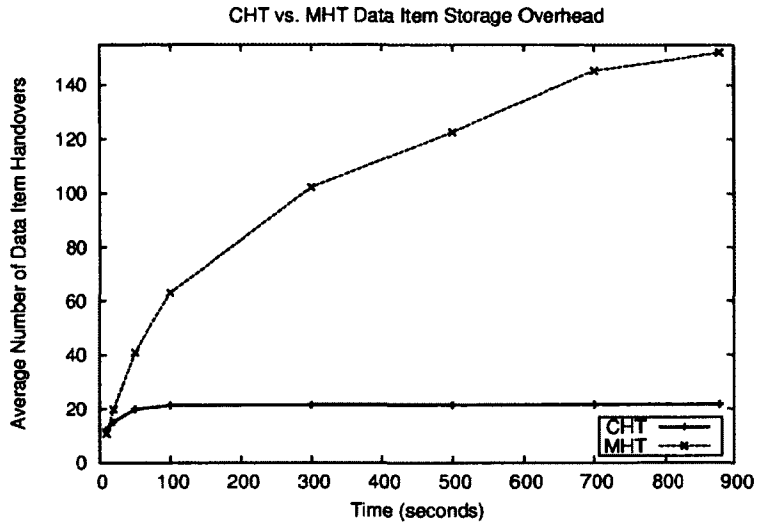


Figure 4.13: Single cell storage maintenance overhead.

simulation. In MHT, a lookup is unsuccessful if a request message for a data item enters into the circle around the mapped location while the data item cannot be found there. Figure 4.14 clearly shows the superiority of CHT over MHT in DTN. The MHT lookup success ratios are low and different nodes moving speeds do not have significant impacts on the results.

4.3.4 Multiple Cell Data Storage and Maintenance

Furthermore, we evaluate the multiple cell storage in CHT. Since MHT maps data items to locations, it does not consider multiple cells and works in the same way throughout the simulation. We divide the topology area into 9 cells with 500m×100m cell size and 5 nodes in each cell. So 90% nodes move within cells and 10% move globally. We assume rectangle centers as cell centers.

The simulation results clearly show that our multiple cell mapping scheme also significantly reduces the data item exchange overhead. Similar to the single cell situation, the longer the time, the more savings can be observed when compared with MHT. Since there are nodes which move globally and data items should be kept at their mapping cells, data item handovers exist over time in CHT, as shown in Figure 4.15.

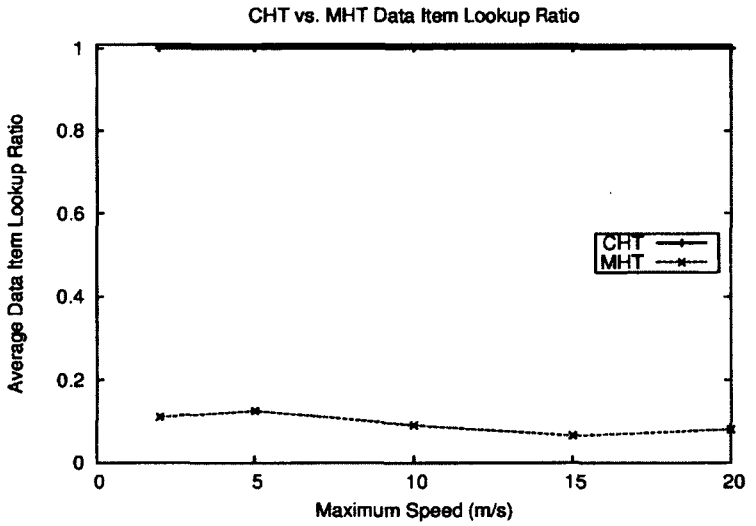


Figure 4.14: Single cell lookup ratio.

4.3.5 Multiple Cell Lookup Ratio

The lookup success ratios of CHT multiple cell and MHT are also evaluated. Figure 4.16 also clearly shows the lookup advantages of CHT over MHT. Since there are nodes moving around, the CHT lookup success ratios in multiple cell scenarios are lower than the success ratios in the single cell scenarios.

4.4 Conclusions

We have proposed a novel distributed data storage mechanism in DTN, called Cell-based Hash Table (CHT). CHT uses cells (flat or hierarchical) to divide regions, and cells are divided in such a way that nodes inside a cell have high probability of moving within. A data item is mapped to a lowest level cell by using local information, and it is further stored at a node according to a modified chord mechanism (or DHT). Due to the use of CHT, data storage maintenance overhead is greatly reduced in DTN. Compared with existing MHT storage mechanism, we have shown in the simulation that our scheme has less storage maintenance overhead with higher success ratio and less delay. As future work, we plan to further study our CHT protocol, exploring the efficient distributed cell formation process in DTN.

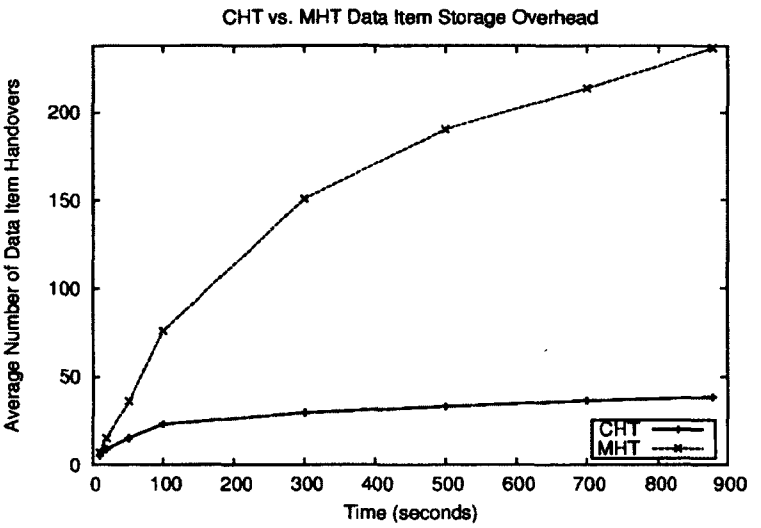


Figure 4.15: Multiple cell storage maintenance overhead.

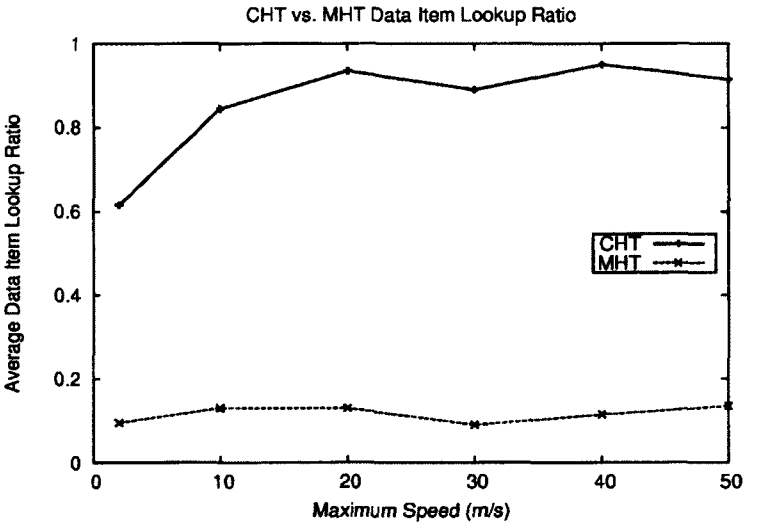


Figure 4.16: Multiple cell lookup ratio.

Part III

Distributed Security Establishment

Chapter 5

Distributed Key Establishment in DTLBS-WSAN

5.1 Introduction

Wireless Sensor and Actor Network (WSAN) improve the robustness of Wireless Sensor Network (WSN) through the use of actor nodes. In the network, wireless sensor nodes are used to gather and relay neighboring environment information to the responsible receiver nodes, while actors are usually resource independent nodes which work according to the input of sensor nodes to take appropriate actions. WSAN works in a distributed way to perform tasks such as environment monitoring, home automation and battlefield surveillance [99].

In a dynamic WSAN, communication delays and disruptions have to be taken into consideration for the proper functioning of the network. Disruption (Delay) Tolerant Networks (DTN) have been widely studied recently, which are also called opportunistic networks or challenged networks. One of the purposes of DTN is to address issues in wireless networks where instantaneous source and destination node connections may not exist.

Location information can be used in WSAN. Wireless nodes could get their location information either by global positioning system (GPS) or localization algorithms [95]. In case there are malicious nodes, the accuracy of location information can be improved through secure localization algorithms [100, 101]. In DTN, there are location based routing protocols (e.g., GLR algorithm [6]), which can fit in a disruption tolerant WSAN.

Wireless communications among nodes may express their social connections. Existing human social networks [102] have a tendency of building relationships and can facilitate network applications [103]. However, these social networks are based on node similarities (e.g., similar origin, interest, etc.) and are not local. They would not work properly when the similarities cannot be found in a WSAN.

WSAN needs security provisioning in a hostile environment. However, existing key management protocols [57, 58, 104] mainly work in a network that is densely connected and focus on the use of symmetric keys. Existing protocol [61] that uses symmetric and public keys highly relies on centralized mechanism and thus is hard to scale when some nodes are captured.

5.1.1 Contributions and Organization of the Chapter

In this chapter, we propose a novel distributed key establishment (DKE) scheme in disruption (delay) tolerant location based social wireless sensor and actor networks (DTLBS-WSAN). DKE is based on neighbor cooperation, with a node trusting its neighboring nodes with high probability. The working procedure of distributed public key certificate establishment scheme without Certificate Authority (CA) is presented. To further improve security, we can equip actor nodes with pre-distributed symmetric and public/private keys. We show that security is guaranteed when actors are connected and cover the sensor deployment area (*Powerful Model*) and high security confidence level can be achieved in the distributed system when the density of malicious nodes is small, even without central management (*None Actor Coverage Model*). We also show that security assurance can be improved when actors increase their transmission powers so that they are connected and cover the entire network area while a sensor node has shorter transmission range than that of actors (*Semi Powerful Model*). We use a mechanism called "safety margin" to counter the malicious certificate attacks. We propose the use of location based social networks (cells) to facilitate and strengthen the security of distributed certificate storage and lookup in wireless sensor and actor networks with disruptions and delays.

The rest of the chapter is organized as follows. Section 5.2 elaborates on our proposed solutions. We theoretically analyze the security strength of the proposed key management scheme in Section 5.3. Section 5.4 describes the details of experiments and analysis. Section 5.5 concludes with possible future work.

5.2 Distributed Key Establishment Algorithm

In this section, we present the basic features of the proposed key management scheme and its security properties, deferring the detailed analysis of the scheme's security strength against the malicious attacks to the next section.

5.2.1 Definitions in Secure DTLBS-WSAN

Definition 2. *(***Disruption tolerant location-based social network***) Disruption (Delay) Tolerant Location Based Social Network (DTLBSN) emphasizes the importance of nodes to stay at (around) a specified location during some period of time, which have a tendency of being static during dynamic unrelated movements, but can cooperate with each other. When nodes are composed of wireless sensors and actors, it is called DTLBS-WSAN.*

With wireless devices being widely used in recent years, local (and possibly distant) wireless communication without relying on centralized server nodes become possible. These devices may possess diverse sensing capabilities and can cooperate with each other in their local communication range. The use of wireless peer-to-peer emails in an organization is one such example. In [105], the author listed the security threats to this emerging way of communication. The attack to the single network key approach (one key is used by all mobile devices) is likely and the risk is critical. Due to this reason, proper key management has to be proposed. Current network graph partitioning mainly focuses on dense connected static graph. DTLBSNs are focusing on a community of nodes which may even be divided, sparse or evolving. Traditional social networks are about nodes (people) having similarities (familiarities) while DTLBSNs are about nodes with diversities (for some reason, they happen to be in close neighborhood, which may or may not be due to similarities). In a location-based social network, nodes mainly cooperate and communicate with others within their social network.

Definition 3. *(***Semi-security***) Communications between nodes with pre-distributed keys are considered secure, while the communication between nodes with self-generated*

public/private keys without the signatures of trusted nodes is considered as semi-secure.

Semi-secure communication provides a platform for security establishment without relying on centralized nodes or servers. A node generates its public key certificate, asks for signatures from multiple neighboring nodes and stores it in a distributed manner.

5.2.2 Notation

We describe the key establishment protocol using the formal notations described in [106]. With A, B denote specific nodes, K_{ab} denotes the shared symmetric key between nodes A, B. K_a, K_b denote the public keys of A, B and K_a^{-1}, K_b^{-1} their corresponding private keys. A message M encrypted with K_{ab} takes the form $\{M\}_{K_{ab}}$, while $\{M\}_{K_a^{-1}}$ means that M is signed by node A.

5.2.3 Adversary Model

We assume the adversary (or a malicious node) can overhear, intercept and manipulate any messages passing through it. However, we assume malicious nodes are randomly deployed with low density. We assume nodes are not allowed to own multiple IDs and malicious nodes with multiple IDs will be detected. This can be achieved through hardware fingerprinting [67] or location cross checking techniques [107]. We differentiate between cooperative malicious nodes and independent malicious nodes (malicious nodes which act independently). The following definition is used in the proposed solution.

Definition 4. *(k* **cooperative malicious nodes***) k ($k \geq 2$) cooperative malicious nodes means that exactly k malicious nodes cooperate with each other and share the information which they possess, including their own public/private keys as well as intercepted information through covert channels.*

5.2.4 Key Pre-distribution and Distributed Key Establishment

The use of pre-distributed keys in existing WSAN key establishment protocols assumes the need of sensor to sensor security. In practice, not every sensor node needs to communicate with every other sensor node. In a location-based social network, most network communications exist among wireless sensors in their locality, as well as between wireless sensors and their corresponding actors. Since nodes mainly communicate with other location based social relations (nodes) that they are unable to know in advance, we propose the distributed approach that they establish public/private keys through neighboring cooperation.

We are aiming at a key agreement mechanism with different levels of security (e.g., guaranteed security, semi-security with high confidence, etc.), with possible guaranteed security expansion and negative trust (malicious) node key deletion.

Key Pre-distribution

Key pre-distribution can be used to increase network communication security. Before deployment, actors are loaded with symmetric keys, as well as public key certificates. Symmetric keys are used for data confidentiality and private keys are used for digital signatures. We assume these nodes are equipped with tamper resistant devices so that the keys are destroyed once they are captured. Actor nodes are also called trusted nodes. We assume actors have abundant resources for computational needs and they are trusted in their location based social networks (cells).

Distributed Key Establishment

We define the distributed key establishment process as follows:

I Setup Phase:

1. When a sensor node, say node A wants to setup its certificate, it generates public/private key pairs K_a, K_a^{-1} and its public key certificate $Cert_a$ (Figure 5.1, the Multiple Issuer and Extended Certificate Signature fields are left blank at this step).

2. It inquires its neighboring nodes to gather the information of which nodes are willing to sign its certificate.

3. The neighboring nodes which are willing to sign it respond to node A with their IDs. An actor node will reply to node A when it receives the request.

4. Node A picks up s nodes (node N_i has $ID_i, i = 1, 2, ...s$) out of all the replies and sends its signature requests, together with its public key certificate (Figure 5.1, without contents in the Extended Certificate Signature field at this step). Any replying actor node will be included in the s nodes.

5. These nodes sign the certificate using their self generated (pre-distributed for actors) private keys ($K_{N_i}^{-1}$) and return back their signatures $\{Cert_a\}_{K_{N_i}^{-1}}$ to A.

6. When A receives all replies, it attaches the signatures to its certificate (Figure 5.1).

7. Node A stores its certificate in a distributed manner (i.e., using CHT [2]) and multiple copies of the certificate can be stored simultaneously.

8. All nodes follow the same procedure and certificates of all nodes are established and stored properly.

9. At the same time, some pre-distributed keys are stored at trusted nodes (actors) and communications between nodes with pre-distributed keys are considered secure.

II Verification Phase:

1. Node B acquires the certificate $Cert_a$ for A, either from A or from distributed storage sites.

2. It chooses c out of s signatures and verifies them, depending on its confidence requirement.

3. If a signature is from a trusted node, then $Cert_a$ is immediately valid by this node checking.

4. A node can also check its neighbors to see if there is a trusted neighbor node which has a shared symmetric key with one of the signing node and verifies it accordingly.

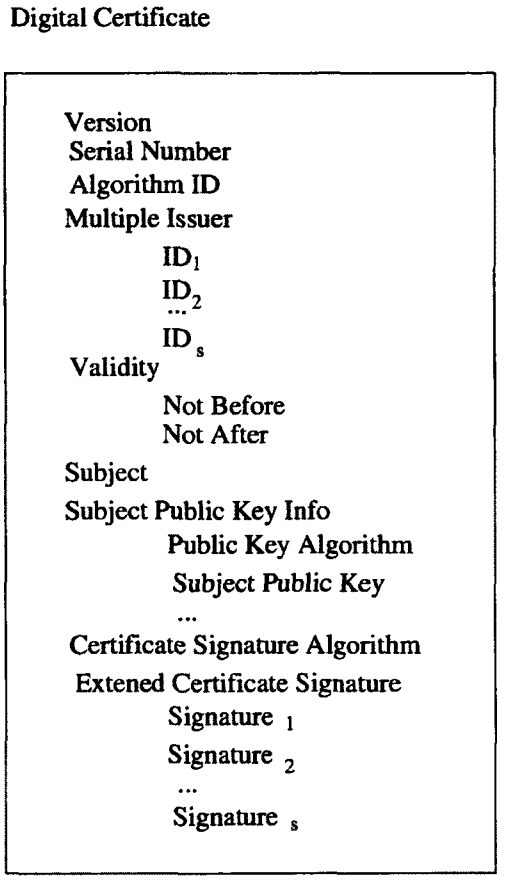


Figure 5.1: Distributed certificate.

Properties of Distributed Key Establishment

The distributed key establishment protocol has the following properties:

Theorem 1. *The difficulty that an independent malicious node can successfully make changes to a certificate of another node without being discovered (through the verification with signing nodes) is as hard as attacking the key generation algorithm.*

Proof. Since the number of signing nodes of a certificate is at least two, any node which wants to forge a certificate should possess at least two pairs of public/private keys. Although a node can generate any number of public/private key pairs as it wants, the key pairs are not verifiable because the node ID and public/private keys at a node should be unique. The malicious node then needs to guess the private keys of other signing nodes in order to successfully make changes to the certificate, which is as difficult as attacking the key generation algorithm.

Similarly, the following result can be obtained.

Corollary 2. *When there are s ($s > k$) signatures, the difficulty that k cooperative malicious nodes can successfully make changes to a certificate of another node without being discovered (through the verification with signing nodes) is as hard as attacking the key generation algorithm.*

Theorem 3. *k cooperative malicious nodes can successfully make changes to a certificate of another node when there are s ($s \leq k$) signing nodes, however undeniable evidence exists once the malicious activity is being discovered.*

Proof. To effectively change the content of a certificate without being discovered during verification phase, the malicious nodes have to use their valid private keys and pretend to be the original signing nodes. When k cooperative malicious nodes modify the content of a certificate, their IDs and signatures (which are generated using their private keys) are attached to the certificate. Once their activity is discovered, another node can prove this through their IDs and signatures.

Session Key Establishment

After the establishment of public key certificates, communicating nodes can exchange session keys encrypted through the public keys via a challenge/response protocol. Since the number of clock cycles needed by the processor to compute security function in symmetric key cryptography is much smaller than the number of clock cycles in asymmetric cryptography, the combination of symmetric session keys with public keys improves the encryption/decryption speed and reduces energy consumption, as shown in [108].

5.2.5 Distributed Certificate and Certificate Revocation List Storage

It has been shown in [2] that Cell-based Hash Table (CHT) can be used to facilitate distributed data storage and lookup in DTN. We use CHT to store the distributed certificates in the proposed solution when there is high probability that sensor and actor nodes will stay at their deployment location based social network regions (cells). Multiple copies of a certificate can be stored in different mapping regions to strengthen certificate security and speed up the certificate lookup process in DTLBS-WSAN.

As with the distributed certificate storage, Certificate Revocation List (CRL) for malicious or captured sensor nodes should also be stored in a distributed manner.

5.2.6 Re-Keying and Key Revocation

When a certificate is going to expire, the owner needs to update its certificate accordingly. This node generates new public/private key pairs and the corresponding certificate. It then asks its neighbors to sign the certificate according to the procedures described in the key setup phase. It will also sign it using its previous private key so that a distributed storage node can replace the old certificate with the new one simply by verifying its signature.

When a node has been captured, its certificate has to be revoked. The trusted node (i.e., actor) in its location based social network can issue signed certificate revocation message to the distributed CRL storage sites so that the storage nodes can verify the authentication of this message and list this revoked certificate in the CRL following verification. Distributed CRL addresses the issues when nodes are compromised. It is stored at the same mapping site where the corresponding certificates are stored. Once a certificate revocation message is received, the revoked certificate can be deleted.

5.3 Analysis of Distributed Key Establishment Security Strategies

The security strengths of the proposed key establishment scheme are different when node transmission power differs. In *Powerful Model*, where actors are connected

and cover the network area and sensors possess same transmission range as that of actors, guaranteed security can be achieved. In *Semi Powerful Model*, actors increase their transmission power so that they are connected and cover the network while the sensor communication range may be far shorter than that of the actors. A sensor node can get improved security assurance in this model. When actors cannot cover the network (*None Actor Coverage Model*), we show that several security mechanisms can be applied to ensure key is secure with high confidence. We present our analysis in the sequel.

5.3.1 Guaranteed Security in Powerful Model

Theorem 4. *If there are m actors distributed in a unit disk square according to Poisson process with communication range*

$$r \geq \sqrt{\frac{\log m + \log\log m + c(m)}{m\pi}} \tag{5.1}$$

with $c(m) \to \infty$, when $m \to \infty$ and sensors have the same communication range with actors, then a sensor node will have at least one actor node as its certificate signing node and its certificate can be verified by any other nodes, with high probability.

Proof. It is shown in [109] that with probability 1 a network with m nodes with communication range $r \geq \sqrt{\frac{\log m + c(m)}{m\pi}}$ with $c(m) \to \infty$, when $m \to \infty$ is connected and the network is covered when the communication range satisfies inequality (5.1) (according to Theorem 3.11 in [110]. In order to save energy, the growth magnitude of $c(m)$ can be selected as small as possible so long as $c(m) \to \infty$ when $m \to \infty$ is satisfied). When (5.1) holds, each sensor node will have at least one actor node in its distance one neighborhood and all the actors are connected. According to DKE, every sensor node will have at least an actor node as its signing node. Any certificate can be verified through this connected actor network. Since a source node can get a genuine key for any destination node, secure communication is guaranteed in this model.

5.3.2 Security Assurance in Semi Powerful Model

Theorem 5. *If there are m actors distributed in a unit disk square according to Poisson process with communication range satisfying inequality (5.1), while sensors have much smaller communication range than the communication range of actors, then a sensor node can get security assurance once its certificate is signed by an actor node through routing path in its locality, with high probability.*

Proof. As shown in Theorem 4, the network is connected and covered when node communication range satisfies inequality (5.1). Actor nodes can broadcast their existence together with their public keys to all sensors in their coverage areas. So a sensor node will be either in distance one neighborhood of at least one actor node where it can communicate with the actor directly, or not. In the latter case, it encrypts its certificate using the actor's public key and tries to send its certificate to the actor node through local routing. The routing path should be local because the sensor node is close to the actor (i.e., within the actor's communication range). When the actor node signs its certificate and returns back to it (through direct transmission), the sensor node can verify the signed certificate using the already received actor public key. If there are any discrepancies during the routing process, the sensor node can re-initiate the routing process until it successfully gets its certificate signed. A sensor node can get security assurance once it receives the proper confirmation from the actor. Its certificate can be verified by others through the connected actor network.

5.3.3 High Confidence in None Actor Coverage Model

Security Confidence with Key Pre-distribution

We assume nodes with pre-distributed keys are trusted. Assume the probability that a node is trusted is p_t. Once a copy of the signature from a trusted node is checked, a node will be confident that this certificate is the original which belongs to the owner. The probability that at least a signature belongs to the trusted node is (at least) $1-(1-p_t)^s$ (s is the number of signatures), in which case the checking node is fully confident with the certificate. (assume full trust to the actor nodes). We use Octave [97] to plot this probability with varying p_t, s, as shown in Figure 5.2.

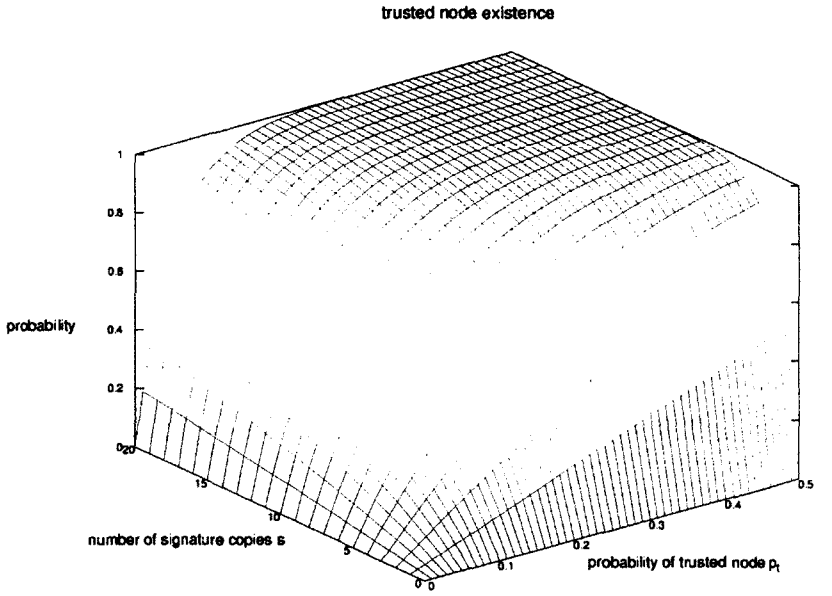


Figure 5.2: Trusted node probability.

Security under k-Cooperative Malicious Attack

There are different possible attacks towards the proposed public key scheme. Malicious nodes can attack the certificate in the certificate setup phase, in distributed storage process or during certificate lookup process. The attack in the certificate setup phase is possible when all the signatures are selected from the malicious neighbors. Assume nodes are randomly deployed and malicious nodes are generated independently and randomly with probability p, it is difficult (the probability is at most p^s, with s the number of signatures) for malicious nodes to be selected as signing nodes during setup phase so that they can make changes to the certificate later. When k ($k \geq s$) malicious nodes cooperate with each other and possess each other's private keys, they can modify the content of an established certificate during distributed storage or lookup process. By doing so, all the original Issuer IDs and

authentic signatures in the certificate are replaced by the s malicious IDs and their bogus signatures. There are several ways to counter this attack. One is by checking the IDs to see if there are possibilities that all those signing nodes happen to be in the neighborhood of the certificate owner during the certificate setup phase. The other one is through multiple copies storage and lookup approach. With multiple copies approach, if there are discrepancies, a checking node will be alerted. Also non-malicious nodes should cooperate with each other to counter this attack. We further analyze the security strengths of the proposed scheme in the following.

Theorem 6. *If there are i nodes on a routing path in a network with n nodes, the probability that there is at least a malicious node inside the i nodes is less than $1-(\frac{n-k-i+1}{n-i+1})^i$. When $i=\frac{n+1}{\alpha k+1}$, this value is at most $1-\frac{1}{\sqrt[e]{e}}$.*

Proof. Assume the probability that there is at least a malicious node on routing path is p. Then $p = 1-\frac{n-k}{n}\times\frac{n-k-1}{n-1}\times\cdots\times\frac{n-k-i+1}{n-i+1} \leq 1-(\frac{n-k-i+1}{n-i+1})^i$, where $1-(\frac{n-k-i+1}{n-i+1})^i = 1-(1-\frac{1}{\frac{n-i+1}{k}})^i \approx 1-\frac{1}{\sqrt[e]{e}}$, when $i=\frac{n+1}{\alpha k+1}$ and $n\to\infty$.

Theorem 7. *When multiple copies (u copies, $u\in N$ and $u\geq 2$) approach is adopted, if the routing paths for these copies are disjoint, then the probability that all the paths have malicious nodes is less than $(1-\frac{1}{\sqrt[e]{e}})^u$, when the routing path length is at most $\frac{n+1}{\alpha k+1}$.*

Proof. Assume the probability that there is at least a malicious node on routing path is p_j for routing path j, with $j=1,2,\cdots,u$. We start by evaluate p_1. From Theorem 6, it is clear that $p_1 \leq 1-\frac{1}{\sqrt[e]{e}}$, when $i=\frac{n+1}{\alpha k+1}$ and $n\to\infty$.
Assume m_1 nodes in routing path 1 and at least one malicious node in the m_1 nodes. Then $p_2 \leq 1-\frac{(n-m_1)-(k-1)}{n-m_1}\times\frac{(n-m_1)-(k-1)-1}{n-m_1-1}\times\cdots\times\frac{(n-m_1)-(k-1)-i+1}{n-m_1-i+1} < 1-\frac{n-k}{n}\times\frac{n-k-1}{n-1}\times\cdots\times\frac{n-k-i+1}{n-i+1} \leq 1-(\frac{n-k-i+1}{n-i+1})^i = 1-\frac{1}{\sqrt[e]{e}}$, when $i=\frac{n+1}{\alpha k+1}$ and $n\to\infty$.
Similarly, we can show that $p_j < 1-\frac{1}{\sqrt[e]{e}}$ for $j>2$. And we conclude that the probability that all the paths have malicious nodes is less than $(1-\frac{1}{\sqrt[e]{e}})^u$ $(u\geq 2)$. It is clear that with multiple copies approach, the security of certificate storage and lookup process is greatly improved.

Distance k Safety Margin

We can further use distance k (for some small k) safety margin to counter malicious certificate attacks. In forwarding a certificate (storage or lookup), a non-malicious node will forward it to the next hop node and at the same time, broadcast it to distance k neighbors. Only next hop receiver needs to do the same thing. A node will compare two certificates if they are generated by the same node ID (owner). In Figure 5.3 (Δ means malicious and circles without labels means unreachable), when node B forwards a copy of a certificate to node M (malicious), it will also forward this certificate to other nodes in its distance k neighborhood (except node A which has the certificate already). This certificate can reach node C through nodes D, E, F when $k = 4$. When M tries to modify the certificate, discrepancies will appear at node C.

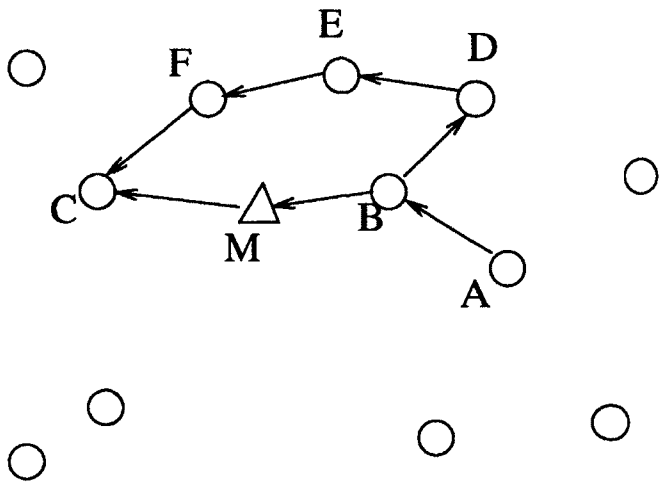


Figure 5.3: Distance k safety margin.

5.4 Experimental Evaluation

5.4.1 Simulation Environment in NS2

We evaluate the performance of the proposed solution using the NS-2 [1] simulator. A random waypoint model is chosen as the motion pattern. The simulation parameters are shown in Table 5.1.

Table 5.1: Parameters of the simulations.

Parameter	Value
Number of mobile nodes	50/200
Mobility	0-20m/s(uniform distribution)
Transmission range	100m
Propagation model	*Two Ray Ground*
Simulation time	300/600 seconds
Topology size	1500m×300m/1500m×600m
Pause time	0/600 seconds
Antenna model	Omnidirectional

Through simulation, we show that a node in a low malicious density cell should not select the nodes from high malicious density cells as certificate signing nodes and *Powerful Model* is an ideal security provisioning for a cell when malicious node probability in the cell is high. Our experimentation results on "safety margin" approach show that it can increase certificate security during storage and lookup process. A node has a high probability of receiving an original certificate from nodes on the routing path ahead of a malicious node even if it receives a bogus certificate from the malicious node, when the node communication range is not too small. The increase in the node communication range can improve the effectiveness of the "safety margin" approach. For the simulation results, all points in the figures, as well as numbers in the tables are obtained as an average of 10 different runs with 10 different network topologies and movement patterns. The confidence intervals (t-distribution) for the numbers are calculated at 95% confidence level.

5.4.2 Effect of Multiple Location-based Regions

We evaluate the effect of multiple location regions with varying malicious probabilities. The network topology area is divided into 9 (1-9) cells with 500m×200m size each and malicious probabilities are 0.2 (5 cells) and 0.8 (4 cells), with different probability cells separate each other. As shown in Table 5.2, the malicious neighbors probabilities for non-malicious nodes in 0.8 cells are consistent with the deployment probability in general. However, the low probability (0.2) cells are greatly affected by their neighbor cells. It is clear that if most of the nodes of a cell are malicious,

a neighbor cell node is better not to choose them as signing nodes. With more malicious nodes in a high probability (0.8) cell, *Powerful Model* is a good choice for a few non-malicious nodes in that cell.

Table 5.2: Effect of the neighbor cells.

Parameter	Column 1	Column 2	Column 3
Row 1	0.351±0.09	0.799±0.06	0.282±0.06
Row 2	0.709±0.11	0.341±0.05	0.724±0.07
Row 3	0.336±0.11	0.797±0.07	0.303±0.06

5.4.3 Distance k Safety Margin

We evaluate the distance k safety margin using GLR [6] routing protocol single copy approach. In the simulation, we set $k = 2$. If a node is on the routing path and receives a data packet, we evaluate whether it can receive the data packet through distance k safety margin approach. Figure 5.4 shows the result. It is clear that the probability that a routing node is also in distance k safety margin increases when node communication range increases.

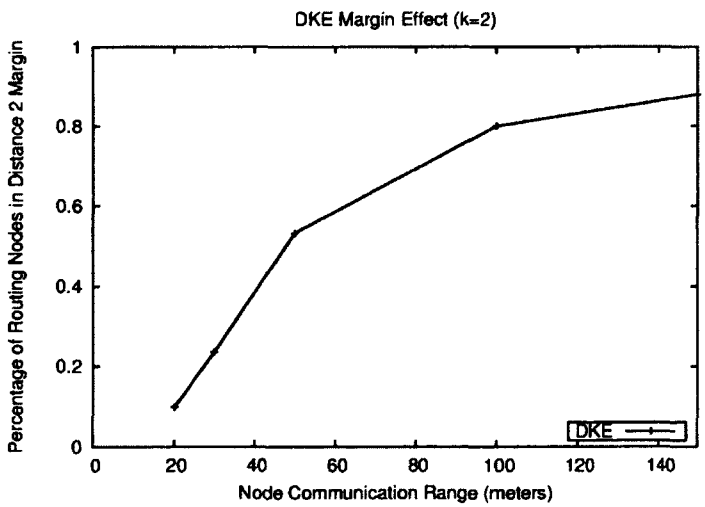


Figure 5.4: Distance two safety margin effect.

The impact of various maximum nodes' moving speeds to the distance k safety margin effects when the node communication range is 100 meters is also evaluated

and the results are shown in Figure 5.5. Different nodes' moving speeds affect the percentage of nodes which are also in distance k safety margin slightly and higher mobility ($\geq$ 10 m/s) outperforms lower mobility ($\leq$ 5 m/s).

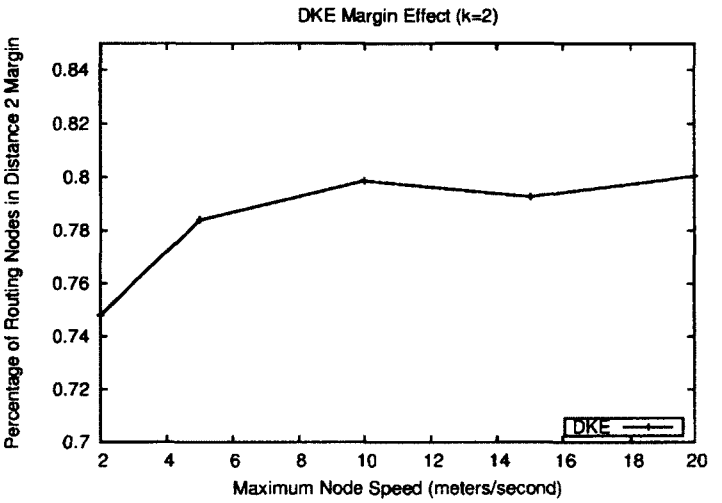


Figure 5.5: The effect under different speeds.

5.5 Summary

We have proposed a novel distributed key establishment mechanism, called DKE. DKE uses a combination of key pre-distribution and neighbor key establishment to set up key pairs at nodes in DTLBS-WSAN. Public key certificates and certificate revocation list of nodes are stored in a distributed way to improve the security and counter network disruptions. We have proved that guaranteed security can be achieved when actor nodes are powerful so that they are connected and cover the entire network area. We propose the use of "safety margin" approach to thwart malicious certificate attacks. Through simulation, we have shown the effectiveness of distance k safety margin approach in improving the certificate security.

As future work, we plan to further study security models in DTLBS-WSAN, exploring theoretically distributed trust establishment in a hostile environment.

Part IV

Neighbor Discovery with Directional Antennae

Chapter 6

Neighbor Discovery in a Sensor Network with Directional Antennae

6.1 Introduction

Directional antennae are known to reduce energy consumption because they can reach further for the same amount of energy consumed. Thus the application of directional antennae in DTN has the potential of improve network performance. Neighbor discovery using directional antennae is the first issue that need to be addressed in this new DTN model.

6.1.1 Motivation

Unlike sensors with omnidirectional antennae sensors with directional antennae take longer to discover their neighbors if contentions are neglected. This is due to the fact that although sensors may be within transmission range the sender (respectively, receiver) sensor may not necessarily be located within the given sector determined by the beaming antenna of the transmitting sensor. This raises the question of what algorithms to employ so as to attain efficient communication (e.g., routing, broadcasting, etc.) using only directional antennae. This approach can be particularly beneficial in delay tolerant sensor networks, for example, whereby sensors may be able to take advantage of opportunistic appearances of sensors due to mobility and other factors.

For a given radius $r > 0$, assume that a given sensor, say S, can reach all other sensors within the disc having centre S and radius r. For our study, it will suffice to consider the following directional antenna model. We assume that either 1) the sensors are standing on a swivel and can rotate in any desired direction or 2) the sensors' coverage area can be divided into non-overlapping sectors that can

be activated by an antenna switch so as to reach other sensors within a particular region. It is clear that in the former mode of operation the rotation of the antenna is continuous around the circle while in the latter the circular sectors are in discrete predefined sectors around the circle. We will not elaborate further in this chapter the differences and similarities between these two modes of operation for directional antennae.

6.1.2 Preliminaries and Notation

In this subsection we present the Unit Disk Graph (UDG) model and discuss several related antenna models that are related to our study.

Unit Disk Graph (UDG)

To simplify the research in wireless communications, Unit Disk Graph (UDG) model was introduced in the literature. UDG is defined as follows:

- It is a graph $G(V, E)$ with V the set of vertices and E the set of edges.
- V: Vertices are the sensor nodes.
- E: Edges between vertices represent connectivity, i.e., two vertices u, v are connected if and only if their distance is at most 1 unit.

Communication Models with Directional Antennae

Several communication models are possible for a pair of sensors with omnidirectional and directional antennae. Consider the pair (XY), where the first parameter X indicates the capability of the sender sensor and the second parameter Y the capability of the receiver sensor. To be more precise, X, Y may take either of the values O, D, where O means omnidirectional and D directional antenna. Thus, the (XY) *communication model* for a pair of communicating sensors means that the sender uses antenna of type X and the receiver of type Y. We also assume a *duplex* communication model whereby sensors can send and receive messages at the same time ignoring collisions. It is clear from the previous discussion that

- in the (OO) model two sensors can communicate if they are within transmission range of each other,

- in the (DO) (respectively, (OD)) model, the sender (respectively, receiver) must turn its antenna so as to reach its neighbor, and

- in the (DD) model both sender and receiver must direct their antennae towards each other at the same time.

More specifically, in all four models the sensors must be within range of each other so as to communicate. However, in the (DO) and (OD) models the sensor with the directional antenna must also turn its antenna toward the other sensor, while in the (DD) model both sensors' antennae must face against each other. Therefore it follows that (DD) is the slowest and (OO) is the fastest in the neighbor discovery process among the four communication models when contentions in the communications are not taken into consideration.

More general model is also possible whereby a sensor's transmission beam width is not necessarily the same with its reception beam width. To simplify notation and terminology, in this chapter we will limit ourselves to the (DD) communication model with identical transmission/reception beam widths. Our results generalize without much difficulty to this more general model.

The *neighbor discovery* process usually entails the exchange of identities (e.g., MAC addresses) between two adjacent nodes. It will not be necessary to go into the details of such an exchange and for our purposes it will be sufficient to assume that this is a one step process whereby one sensor sends its identity and the other acknowledges by sending back its own. Throughout this chapter we will assume that the sensors have distinct identities but their corresponding locations (i.e., (x, y)-coordinates) in the plane are not known to each other.

Antenna Models

The transmission area of an omnidirectional antenna is modelled by a circular disk in the plane while the transmission area of a directional antenna is modelled by

a circular sector in the disk. We assume that sensors have the capability to rotate their directional antenna and change sectors so as to establish communication.

Consider a set of n sensors in the plane. Each sensor u is equipped with a directional antenna having beam width ϕ_u. Further we will assume that $\phi_u = \frac{2\pi}{k_u}$, for some integer k_u.[1] In particular, if $k_u = 1$ then we have an omnidirectional antenna at u. The sensors are synchronous and can rotate their antennae counter-clockwise (see Figure 6.1). Assume that the Unit Disk Graph (UDG) formed by

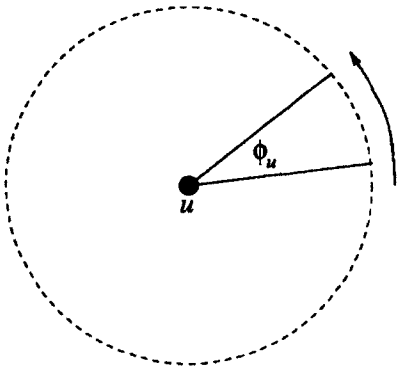


Figure 6.1: An antenna at u rotating counter-clockwise.

the sensor set V is connected and c-colorable, i.e., there is a coloring of its vertices $\chi : V \to \{0, 1, \ldots, c-1\}$ such that if sensors u, v are adjacent in the UDG then u and v have different colors, i.e., $\chi(u) \neq \chi(v)$. Observe any "integer based" identity scheme, e.g., the n sensors are numbered $0, 1, 2, \ldots, n-1$, that provides different numbering to different sensors satisfies this property (albeit it is not efficient).

6.1.3 Contributions and Organization of the Chapter

In this chapter, we propose novel neighbor discovery algorithms in a (DD) communication model whereby sensors employ directional antennae with identical transmission/reception beam widths and each sensor has only one directional antenna. Our methodology is based on symmetry breaking techniques to enable sender/receiver communication. We provide 1) deterministic algorithms that exploit knowledge of

[1]It turns out that this assumption is not required for the subsequent results; we use it because it makes the proofs simpler.

a vertex coloring of the network and introduce delay in the rotation of the antennae according to the coloring, and 2) randomized algorithms without requiring extra knowledge. Only an upper bound on the size of the network is enough to accomplish neighbor discovery, with high probability. In both instances we study tradeoffs on the efficiency of the algorithms proposed when nodes are static. In addition, we simulate all our algorithms and examine their performance using the NS-2 simulator environment.

The rest of the chapter is organized as follows. Deterministic algorithms on neighbor discovery are presented in Section 6.2. As an alternative scenario, Section 6.3 gives out the randomized algorithm and its analysis. Section 6.4 describes our simulation results and we conclude with possible future directions in Section 6.5.

6.2 Deterministic Algorithms for Neighbor Discovery

In this section we give algorithms for neighbor discovery in the (DD) communication model and analyze their complexity. First we give a simple lower bound that indicates the complexity of the neighbor discovery problem.

In all the results below as measure of complexity for neighbor discovery we will use the time required for sensors to discover each other and we will ignore collisions during simultaneous transmissions. For two sensors, this is the number of steps until the first successful send/receive exchange. For a sensor network, this is the minimum for any algorithm taken over the maximum time required for any two adjacent sensors in the network to communicate.

6.2.1 Lower Bound

In a setting whereby two adjacent sensors know each other's location all they need to do is turn their antennae towards each other in the specified locations. Therefore the observation below is useful when sensors do not know each other's location.

Theorem 8. *Consider two sensors u, v within communication range of each other and respective antenna beam widths $\frac{2\pi}{k_u}$ and $\frac{2\pi}{k_v}$, respectively. If the sensors do not know each other's location then any algorithm for solving the neighbor discovery*

problem in the (DD) communication model requires at least $\Omega(k_u k_v)$ time steps.

Proof. For a successful communication to occur each sensor must be within the beam of the other sensor's antenna at the same time. Since the sensors do not know each other's location they must attempt transmissions in all their respective sectors. This completes the proof of Theorem 8. □

6.2.2 Antenna Rotation Algorithms

Given these preliminary definitions we consider the following class of Antenna Rotation Algorithms (ARA). For each sensor u, let d_u be an integer delay parameter and k_u be defined so that $\phi_u = \frac{2\pi}{k_u}$. Given u, d_u, k_u the sensor executes the following algorithm.

Antenna Rotation Algorithm: $ARA(d_u, k_u)$

1. **Start** at a given orientation;
2. **Repeat for ever**
   (a) **For** $i := 0$ **to** $d_u - 1$ **do**
   /*For d_u steps stay in chosen sector*/
      i. **send** message to neighbor(s);
      ii. **listen for** messages from neighbor(s) (if any);
   (b) **Rotate** antenna beam one sector counter-clockwise;
   /*rotate by an angle equal to ϕ_u*/

Remarks and Observations on the *ARA* Algorithm.

There are several issues concerning interpretations of the execution of the rotation algorithm which are worth discussing.

- In Step 1 the initial antenna orientation is selected. There are many consistent ways to define this but for simplicity in this chapter it is taken to be the bisector of the angle which defines the antenna beam. Also, if the sensors are equipped with a compass then we may assume that they all start with identical

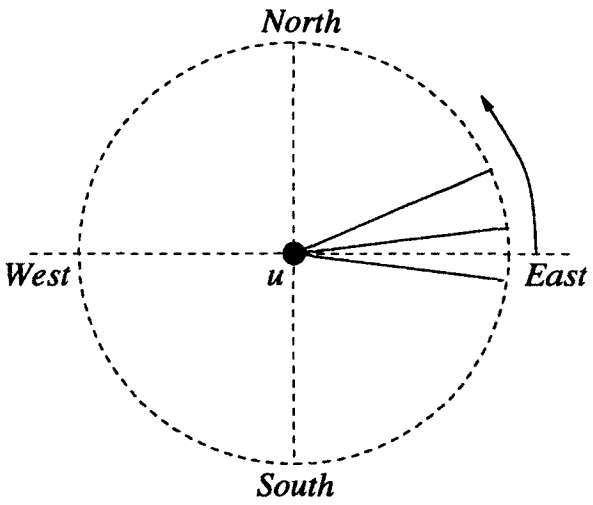


Figure 6.2: An antenna at u with sectors counted counter-clockwise.

orientations, say East (see Figure 6.2). Otherwise, the initial orientation may be chosen in an arbitrary manner. It turns out that our analysis is valid in this more general setting.

- The main neighbor discovery algorithm is executed in Step 2. We are interested in measuring the number of steps until all (available) neighbors are discovered. For the duplex communication model being considered here, it is clear that two sensors u, v will be able to discover each other if (see Figure 6.3)

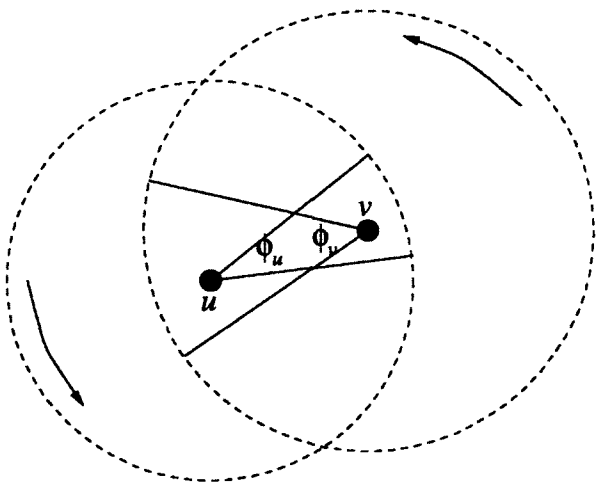


Figure 6.3: Neighbor discovery for sensors u, v.

1. each sensor is within each other's range, and
2. the corresponding antennae of the two sensors are oriented so that each sensor is within the other sensor's beam at the same time.

These are the basic requirements we employ in order to prove the correctness and running time of our algorithm.

- In Step 2a, the algorithm imposes a *rotation delay*, i.e., for d_u (equal to the delay imposed) steps the sensor sends messages and also listens for messages from neighbors. The delay imposed in Step 2a is required so as to break symmetry and ensure that neighboring sensors' antennae are within each other's beam range and will eventually communicate using the (DD) communication model. There are several possibilities here. The sensor may elect to send/receive messages 1) at each step during the delay interval $[0, d_u - 1]$, 2) select a time within the delay interval $[0, d_u - 1]$ at random. In our analysis we will assume the former.

- Step 2b involves rotation of the antenna by ϕ_u which is also equal to the beam width of the antenna. This ensures that after each rotation a new region (located counter-clockwise from the old region) is covered. Several possibilities exist, for example 1) allow overlap between the new and old antenna beaming location, 2) select the new antenna beaming location at a sector chosen at random among the k_u possible sectors in the disk.[2]

6.2.3 Complexity of Deterministic Antenna Orientation Algorithm

Now we consider the complexity of the various antenna orientation algorithms. Assume the sensor network is synchronous. Recall our basic assumption that there is a coloring $\chi : V \to \{0, 1, \ldots, c-1\}$ of the vertices of the sensor network using c colors. Table 6.1 summarizes the results of this section.

[2]The point of these assumptions is to consider collision models. In this chapter we assume that the sensors send/receive messages at each step during the delay interval. Further, if we were to analyze a collision model we would have to assume that the corresponding intervals of adjacent nodes are disjoint.

Antenna at u	Knowledge	Running Time	Theorems
$2\pi/k$	Identical	$O(k^{c-1})$	Theorem 9
$2\pi/k$	Identical	$O(k(c\ln c)^3)$	Theorem 10

Table 6.1: List of theorems and running times of deterministic algorithms.

The simplest possible delay model is for a sensor to wait "sufficient amount of time"' so as to send to (receive from) the desired node.

However, there are choices of delay under which sensors with directional antennae will never be able to communicate as illustrated in Figure 6.4.

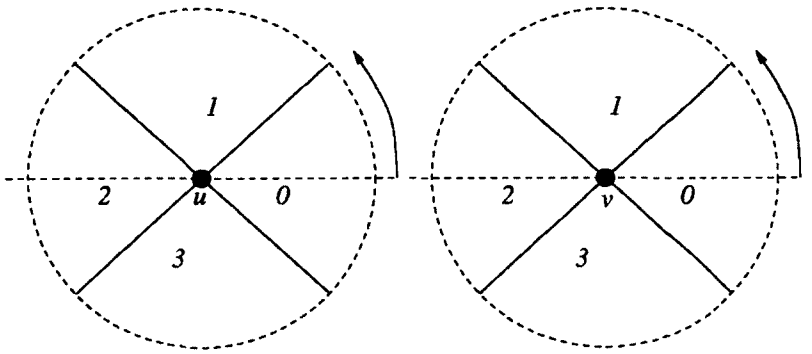


Figure 6.4: Neighbor discovery for sensors u, v is not possible.

Example 1. *Assume the antenna beam width is* $\frac{2\pi}{4} = \frac{\pi}{2}$ *and the four sectors are labelled* $0, 1, 2, 3$. *Both sensors depicted in Figure 6.4 start beaming East. Sensor* u *employs delay* $d_u = 2$ *and sensor* v *delay* $d_v = 1$. *Sensors can communicate only if* u*'s antenna faces East and* v*'s antenna faces West at the same time. Observe that sensor* u *faces East only at time* $t = 0, 1, 8, 9, 16, 17, \ldots$ *while sensor* v *faces West only when* $t = 2, 6, 10, \ldots$. *Therefore* u, v *can never communicate.*

The previous example indicates that sensor delays must be chosen judiciously so as to enable communication. The first theorem considers the simplest model whereby a sensor delays the rotation of its antenna sufficient time so as to allow all its neighbors' antennae to perform a complete rotation.

Theorem 9. *Consider a set of sensors in the plane with identical antenna beam widths equal to* $\phi = \frac{2\pi}{k}$ $(k \geq 1)$. *For each sensor* u *let the delay be defined by*

$d_u := k^{\chi(u)}$. *If each sensor u executes algorithm $ARA(d_u, k)$ then every sensor in the network will discover all its neighbors in at most k^{c-1} time steps.*

Proof. Consider two adjacent sensors u, v. Clearly, $\chi(u) \neq \chi(v)$ since they must have different colors. By assumption, $d_u = k^{\chi(u)}$ and $d_v = k^{\chi(v)}$. Without loss of generality assume that $\chi(u) < \chi(v)$. Observe that for each chosen sector the sensor v beams its antenna in this sector for $k^{\chi(v)}$ steps. But $k^{\chi(v)} = k^{\chi(v)-\chi(u)}k^{\chi(u)}$ and hence $k^{\chi(v)}$ is a multiple of $k^{\chi(u)}$. In particular, while sensor v waits in a given sector the other sensor u will execute $k^{\chi(v)-\chi(u)}$ rotations around the circle before returning to its original sector. It follows that sensors u, v will discover each other within the specified number of steps. This completes the proof of Theorem 9. □

The running time of the algorithm depends on the coloring being used in Theorem 9. If no knowledge on the network is available then any integer identity scheme will work, however this will typically be of size $\Omega(n)$ thus giving an exponential running time $k^{\Omega(n)}$. If the sensor network is bipartite (e.g., tree) then it is easy to see that $c = 2$ is sufficient. For random UDGs with range at the connectivity threshold the number of colors required is $c = \Theta(\log n)$ in which case the running time of the algorithm is about $k^{\log n} = n^{\log_2 k}$, which is polynomial in n with exponent $\log_2 k$.

Instead of an exponential increase where c is the quantity growing in Theorem 9, we can further reduce the running time. Indeed, this is the case as shown by the next theorem.

Theorem 10. *Consider a set of sensors in the plane with identical antenna beam widths equal to $\phi = \frac{2\pi}{k}$. Assume the sensor network is synchronous. Suppose that the delays d_u at the nodes are chosen so that*

1. $\gcd(k, d_u) = 1$, *and* $d_u > k$, *for all* u, *and*
2. *if* u, v *are adjacent then* $\gcd(d_u, d_v) = 1$.

If each sensor u executes algorithm $ARA(d_u, k)$ then every sensor in the network will discover all its neighbors in at most $O\left(k(\max_u d_u)^3\right)$ time steps. In addition, the delays d_u can be chosen so that every sensor in the network will discover all its neighbors in at most $O(k(c \ln c)^3)$ time steps. In particular, this is at most $O\left((c \ln c)^3\right)$ time steps provided that $k \in O(1)$.

Proof. Without loss of generality, in the proofs below we assume that the sensors can determine a fixed starting antenna sector facing East, say (see Figure 6.2). Proofs carry over to the more general case and the necessary modifications are omitted. Consider two adjacent sensors u, v. Without loss of generality assume that

1. sensor u is to the left of sensor v, and

2. that both antennae orientations are initially set to *East*, say.

First we consider the case when the line segment connecting u to v is horizontal. Observe that u, v can communicate when v's antenna is facing *West* which is sector $\lfloor \frac{k}{2} \rfloor$. Since $\gcd(d_u, d_v) = 1$, by Euclid's algorithm there exist integers $0 < a_u < d_v, 0 < a_v < d_u$ such that

$$a_u d_u = a_v d_v + 1. \tag{6.1}$$

Lets look at sensor u first. Recall that because of the delay constrains of the algorithm, the sensor stays in the same sector for d_u steps before it rotates its antenna. After $d_u k$ steps sensor u will be in its starting position and, clearly, the same applies for any time duration that is a multiple of $d_u k$. Thus sensor u is in its initial position (facing East) at time $j a_u d_u k$, for any $j > 0$. If we multiply both sides of Equation $a_u d_u = a_v d_v + 1$ by jk we have that

$$j a_u d_u k \;=\; j a_v d_v k + jk$$

It follows that at time $t = j a_u d_u k$ the sensor at u is facing *East*. If there is a j such that $jk = \lfloor \frac{k}{2} \rfloor d_v + r$ for $0 \leq r < d_v$, then sensor v is facing West and therefore the sensors u, v can discover each other. Starting from $j = 1$, with $k \leq \lfloor \frac{k}{2} \rfloor d_v$, we can find a j such that,

$$jk \leq \lfloor \frac{k}{2} \rfloor d_v < jk + k \tag{6.2}$$

which means that $jk + k = \lfloor \frac{k}{2} \rfloor d_v + r$, with $r \leq k < d_v$. A simple modification of the proof will prove the result when the two sensors are not necessarily on a horizontal line.

The number of rotations required is $ja_u d_u k$, where j satisfies Inequality (6.2). Since $ja_u d_u k \leq k(\max_u d_u)^3$ it follows that $k(\max_u d_u)^3$ is an upper bound on the time required by all pairs of sensors to discover each other.

If $k \in O(1)$ (this is a reasonable assumption since in practice k is a constant) then we can satisfy the conditions of Theorem 10 by choosing the d_us to be prime numbers. Since the number of colors is c, we will need c prime numbers (one for each color class of vertices of the graph). Hence by the prime number theorem the largest prime needed in order to define the delays $\{d_u : u \in V\}$ will be in the order of the c-th prime number, which is in $O(c \ln c)$. Therefore every sensor in the network will discover all its neighbors in at most $O((c \ln c)^3)$ time steps. This completes the proof of Theorem 10. □

Theorem 10 can be improved further with only slight modifications in the proof even in the case where $\frac{2\pi}{\phi}$ is not necessarily an integer. To this end define $k := \lfloor \frac{2\pi}{\phi} \rfloor$. We can modify algorithm $ARA(d_u, k)$ to a new algorithm $ARA'(d_u, \phi)$ as follows: we still have k sectors and we can modify Step 2b in algorithm $ARA(d_u, k)$ so that the antenna at u rotates along the corresponding sectors $0, 1, \ldots, k-1$ (thus there is overlap between the new and the old sector). It is easy to prove the following generalization of Theorem 10.

Theorem 11. *Consider a set of sensors in the plane such that the antenna beam width of sensor u is equal to ϕ. Define $k := \lfloor \frac{2\pi}{\phi} \rfloor$ Assume the sensor network is synchronous. Suppose that the delays d_u at the nodes are chosen so that*

1. *$gcd(d_u, k) = 1$ and $d_u > k$, for all u, and*

2. *if u, v are adjacent then $\gcd(d_u, d_v) = 1$.*

If each sensor u executes algorithm $ARA'(d_u, \phi)$ then every sensor in the network will discover all its neighbors in at most $k(\max_u d_u)^3$ time steps. In addition, the delays d_u can be chosen so that every sensor in the network will discover all its neighbors in at most $O(k(c \ln c)^3)$ time steps. In particular, this is at most $O((c \ln c)^3)$ time steps provided that $k \in O(1)$.

Proof. With some simple modifications, this is identical to the proof of Theorem 10. Details are left to the reader. □

Observe that for a random UDG at the connectivity threshold we have that $c = \Theta(\ln n)$ and therefore the running time of the algorithms in Theorems 10 and 11 will be $O((\ln n \ln \ln n)^3)$.

6.3 Randomized Neighbor Discovery Algorithms

In this section we consider several randomized algorithms. The main advantage of the algorithms in Theorems 12 and 13 is that no a priori knowledge of coloring or of any proper identity scheme is required; just an upper bound n on the size of the network. Moreover, the algorithm in Theorem 14 requires only a bound on the antennae beam widths. Table 6.2 summarizes the results of this section.

Antenna at u	Knowledge	Running Time	Theorems
$2\pi/k$	Identical	$kn^{O(1)}$	Theorem 12
$2\pi/k$	Identical	$O(k^2 \log n)$	Theorem 13
$2\pi/k_u$	$\max_u k_u \leq k$	$O(k^4 \log n)$	Theorem 14

Table 6.2: List of theorems and running times of randomized algorithms.

6.3.1 Deterministic Algorithm with Selection of Random Delay

In this algorithm each sensor u selects a random prime number as delay d_u (in a range $k..R$ to be specified) and runs the deterministic algorithm $ARA(d_u, k)$.

Randomized Antenna Rotation Algorithm: $RARA(k_u; R)$

1. **Select** $d_u \rightarrow RANDOMPRIME(k_u..R)$;
2. **Execute** $ARA(d_u, k_u)$;

Theorem 12. *Consider a set of sensors in the plane such that the antenna beam width of sensor u is equal to $\phi = \frac{2\pi}{k}$. Assume the sensor network is synchronous. If each sensor u executes algorithm $RARA(k; R)$, where $R = n^{O(1)}$ and n is an upper*

bound on the number of sensors, then every sensor in the network will discover all its neighbors in at most $kn^{O(1)}$ expected time steps, with high probability.

Proof. For every node u, let $N(u)$ denote the neighborhood of u and $\deg(u)$ the degree of u. Further, let $D = \max_u \deg(u)$ denote the maximum degree of a node of the sensor network. By the prime number theorem, the number of primes $\leq R$ and $> k$ is approximately equal to $\frac{R}{\ln R} - \frac{k}{\ln k}$ and therefore the probability that the primes chosen by two adjacent nodes, say u and v, are different is $1 - \frac{1}{\frac{R}{\ln R} - \frac{k}{\ln k}}$.

Let E_u be the event that the prime chosen at u is different from all the primes chosen by its neighbors. It is easily seen that

$$\begin{aligned}
\Pr[E_u] &= 1 - \Pr[\neg E_u] \\
&= 1 - \Pr\left[\exists v \in N(u)(d_u = d_v)\right] \\
&\geq 1 - \sum_{v \in N(u)} \Pr\left[d_u = d_v\right] \\
&\approx 1 - \deg(u)\frac{1}{\frac{R}{\ln R} - \frac{k}{\ln k}} \\
&\geq 1 - D\frac{1}{\frac{R}{\ln R} - \frac{k}{\ln k}}.
\end{aligned}$$

Similarly, we can prove that

$$\begin{aligned}
\Pr\left[\bigcap_u E_u\right] &= 1 - \Pr\left[\bigcup_u \neg E_u\right] \\
&\geq 1 - \sum_u \Pr[\neg E_u] \\
&\geq 1 - nD\frac{1}{\frac{R}{\ln R} - \frac{k}{\ln k}} \\
&\geq 1 - \frac{1}{n}.
\end{aligned}$$

By choosing R in $n^{O(1)}$ and recalling that $D \leq n$ we see that all the primes chosen by all the nodes in the network are pairwise distinct, with high probability. The claim concerning the expected number of time steps follows immediately from the analysis of the antenna rotation algorithm in Theorem 10. This completes the proof of Theorem 12. □

6.3.2 Algorithm with Random Selection of Rotation Mechanism

In the algorithms below we assume that the antenna beam width of u is equal to $\frac{2\pi}{k}$. In the main algorithm a sensor chooses "rotation mechanism" between two given rotation mechanisms independently at random. In the first mechanism, the antenna cycles k rounds with no sector delay, while in the second the antenna cycles only one round but with delay k per sector. The two rotation mechanisms can be described formally as follows.

Rotate with no Sector Delay: $Mech_0(u, k)$
/*Cycle k rounds with no sector delay*/

1. **For** $j = 1$ **to** k **do**;
   (a) **For** $i = 0$ **to** $k - 1$ **do**
      i. send message to neighbor(s) in sector i;
      ii. listen for messages from neighbor(s) (if any) in sector i;
      iii. Rotate antenna one sector;

Rotate with Delay k per Sector: $Mech_1(u, k)$
/*Cycle one round with delay k per sector*/

1. **For** $i = 0$ **to** $k - 1$ **do**;
   (a) **For** $j = 1$ **to** k **do**
      i. send message to neighbor(s) in sector i;
      ii. listen for messages from neighbor(s) (if any) in sector i;
   (b) **Rotate** antenna one sector;

Random Selection Rotation Mechanism Algorithm: $RSRMA(u; k)$
/*Choose rotation mechanism at random*/

1. **Select** $bit \rightarrow RANDOM(\{0, 1\})$;
2. (a) **If** $bit = 0$ **then**
   Execute $Mech_0(u, k)$;

(b) **If** $bit = 1$ **then**

Execute $Mech_1(u, k)$;

Thus algorithm $RSRMA(u, k)$ selects the rotation mechanism at random. We can prove the following theorem.

Theorem 13. *Consider a set of n sensors in the plane with identical antenna beam width equal to $\phi = \frac{2\pi}{k}$. Assume the sensor network is synchronous. If each sensor u executes algorithm $RSRMA(u; k)$ for $O(\log n)$ times then every sensor in the network will discover all its neighbors in at most $O(k^2 \log n)$ expected time steps, with high probability.*

Proof. The proof of correctness is not difficult. The sensor flips a coin. If the outcome is head ($bit = 0$, step 2a) then it rotates the antenna k rounds around the circle; in each round it rotates the antenna with no delay and sends messages and listens for messages. However, if the outcome is tail ($bit = 1$, step 2b) then it rotates the antenna once around the circle; in each sector it sends messages and listens for messages k times and then rotates the antenna one sector. Now consider two sensors u, v within range of each other and assume, without loss of generality, that u is to the left of v (The same proof will work regardless of the direction of the line segment uv connecting u to v). Both sensors start beaming *East*. We know that a necessary and sufficient condition to establish communication is for u's antenna to beam *East* and v's antenna to beam *West* at the same time. If both sensors coin-flips give the same bit then the sensors will select the same rotation mechanism and their antennae will not face "against" each other. However, if their coin-flips give different bits then it is clear that their corresponding antennae will face *East* and *West*, respectively, at the same time.

Let $m = 3 \log n$ and suppose that all sensors run algorithm $RSRMA(u; k)$ for m times. The only case that two adjacent sensors u, v cannot communicate in m steps is that the coin flips yield identical outcomes m times. In particular we have two random binary strings of length m each one drawn from u and another from v. The probability that the strings are identical is equal to $2^{-m} = n^{-3}$ since $m = 3 \log n$.

Finally, we can prove the main result of the theorem. Let $E_{u,v}$ denote the event that sensors u, v can communicate (at some time). Consequently, from the discussion

above we conclude that

$$\Pr[\neg E_{u,v}] \leq n^{-3}, \text{ for any pair } u, v \text{ of sensors.} \tag{6.3}$$

Therefore we obtain that the probability that any two adjacent sensors communicate is at least

$$\begin{aligned}
\Pr[\forall u, v E_{u,v}] &= 1 - \Pr[\neg(\forall u, v E_{u,v})] \\
&= 1 - \Pr[\exists u, v \neg E_{u,v}] \\
&= 1 - \Pr\left[\bigcup_{u,v} \neg E_{u,v}\right] \\
&\geq 1 - \sum_{u,v} \Pr[\neg E_{u,v}] \\
&\geq 1 - n^2 \frac{1}{n^3} \\
&= 1 - \frac{1}{n}.
\end{aligned}$$

This proves our assertion and completes the proof of Theorem 13. □

6.3.3 Algorithm if Bound on Antenna Beam Widths is Known

We now indicate how to extend Theorem 13 to the case of sensors with arbitrary antenna beam widths. First of all, we modify the rotation mechanisms by introducing the delay as a parameter.

Rotate with no Sector Delay: $Mech'_0(u, k_u, d)$
/*Cycle d rounds with no sector delay*/

1. **For** $j = 1$ **to** d **do**;

   (a) **For** $i = 0$ **to** $k_u - 1$ **do**

      i. send message to neighbor(s) in sector i;
      ii. listen for messages from neighbor(s) (if any) in sector i;
      iii. Rotate antenna one sector;

Rotate with Delay d per Sector: $Mech_1'(u, k_u, d)$
/*Cycle one round with delay d per sector*/

1. **For** $i = 0$ **to** $k_u - 1$ **do**;

   (a) **For** $j = 1$ **to** d **do**

      i. send message to neighbor(s) in sector i;

      ii. listen for messages from neighbor(s) (if any) in sector i;

   (b) **Rotate** antenna one sector;

Following the proof of Theorem 13, observe that if two adjacent sensors u, v execute the following algorithm for $m = 3 \log n$ times then they will discover each other with high probability.

Random Selection Rotation Mechanism Algorithm (u): $RSRMA'(u; k_u, k_v)$
/*Choose rotation mechanism at random*/

1. **Select** $bit \rightarrow RANDOM(\{0, 1\})$;
2. (a) **If** $bit = 0$ **then**
   Execute $Mech_0'(u, k_u, k_v)$;

   (b) **If** $bit = 1$ **then**
   Execute $Mech_1'(u, k_u, k_v)$;

and

Random Selection Rotation Mechanism Algorithm (v): $RSRMA'(v; k_v, k_u)$
/*Choose rotation mechanism at random*/

1. **Select** $bit \rightarrow RANDOM(\{0, 1\})$;
2. (a) **If** $bit = 0$ **then**
   Execute $Mech_0'(v, k_v, k_u)$;

   (b) **If** $bit = 1$ **then**
   Execute $Mech_1'(v, k_v, k_u)$;

This idea is for each sensor to use the neighbor sensor's antenna beam width to determine an appropriate delay. However, this will not work because sensor u (respectively, v) does not necessarily know the beam width of v's (respectively, u's) antenna. However, this difficulty is easy to resolve if an upper bound, say k, on $\max\{k_u, k_v\}$ is known by both u and v. Namely, sensor u executes algorithm $RSRMA'(u; k'_u, k'_v)$ and sensor v executes algorithm $RSRMA'(u; k'_u, k'_v)$, for all pairs (k'_u, k'_v) such $k'_u, k'_v \leq k$. To maintain synchronicity all k^2 pairs of algorithms are executed in the same lexicographic order by all pairs of sensors each algorithm for $m = 3 \log n$ times. Clearly, the running time of the algorithm is $O(k^4 \log n)$ with high probability.

Putting these ideas together and repeating the proof of Theorem 13 it is easy to prove the following theorem.

Theorem 14. *Consider a set of n sensors in the plane such that sensor u has antenna beam width equal to $\phi_u = \frac{2\pi}{k_u}$. Assume the sensor network is synchronous and that an upper bound k is known to all sensors so that $\max_u k_u \leq k$. If each sensor u executes algorithm $RSRMA'(u; a, b)$, for each pair (a, b), with $a, b \leq k$, for $O(\log n)$ times then every sensor in the network will discover all its neighbors in at most $O(k^4 \log n)$ expected time steps, with high probability.* □

6.4 Simulations

We perform simulations to compare the performance of different proposed algorithms and evaluate the effects of different antenna beam widths in the neighbor discovery process. During the experiments, we pay great attention to the key attributes, including the delay and energy saving efficiency of using directional antennae.

6.4.1 Simulation Environment

The proposed solutions are implemented using the NS-2 [1] simulator. This simulation environment includes full simulation of the IEEE 802.11 physical and MAC layers, which makes the simulation better reflect the real world. We migrate directional antenna code from IIT TENS project ([111], based on NS-2 version 2.19b) to NS-2 version NS-2.33 and add the ideal directional antenna type with various beam

widths. The simulation parameters are shown in Table 6.3.

Table 6.3: Parameters of the simulations.

Parameter	Value
Number of mobile nodes	50
Beam Width Angle	20°, 30°, 40°, 45°, 50°, 60°, 90°, 120°, 360°
Transmission range	250m
Data rate	1 Mbps
Propagation model	*Two Ray Ground*
Simulation time	200 - 20000 seconds
Link layer queue length	50
Topology size	1000m×1000m
Antenna model	directional

Simulation results show that the proposed neighbor discovery algorithms using directional antenna can significantly reduce energy consumption when they are compared with the omnidirectional antenna. Through simulation, we show that the random selection of rotation mechanisms can achieve desirable neighbor discovery delays.

For the simulation results, all points in the figures, as well as numbers in the tables are obtained as an average of 10 different runs with 10 different network topologies. The confidence intervals (t-distribution) for the numbers are calculated at 95% confidence level.

6.4.2 Deterministic Neighbor Discovery

We first perform simulations on algorithm ARA, with sectors k varying from 3 to 12. Graph coloring algorithm [112] is used to get the color $\chi(u)$ for a node u. We set antenna rotation delay as $d_u = k^{\chi(u)}$ units (two seconds per unit). In the simulation, nodes with wider beam width can find neighbors faster than the nodes with narrower beam width, as shown in Figure 6.5.

As a special case of the ARA algorithm, we select prime numbers for the coloring and set rotation delay d_u as the prime numbers (the special case is denoted as $ARAR$ in the simulation). Simulation results show that $ARAR$ is faster than the ARA algorithm with delay $d_u = k^{\chi(u)}$ (Figure 6.5) in neighbor discovery process. Figure 6.6

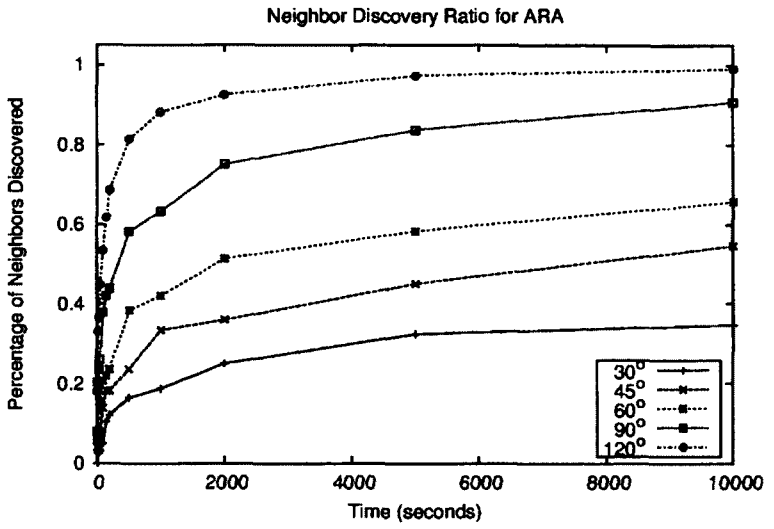


Figure 6.5: Neighbor discovery results for *ARA* algorithm with varying beam width.

is the *ARAR* comparison when nodes take different beam widths.

Random Selection Rotation Mechanism Algorithm

Without global knowledge, randomized neighbor discovery algorithms (*RSRMA* and *RSRMA'*) can be used in the network. In the simulation, *RSRMA* is the best among all the algorithms (better than the deterministic neighbor discovery algorithms), even without global information. Similar to the deterministic algorithms, nodes with wider beam width antennae outperform those with narrower beam width antennae. Figures 6.7 and 6.8 illustrate the effects.

6.4.3 Delay Comparison with Different Beam Width

We also plot a figure to show the delay performance comparison of different antenna rotation algorithms. Figure 6.9 shows the situation when node beam widths are randomly chosen (from $20°, 30°, 40°, 45°, 50°$ and $60°$), and clearly $50°$ does not divide $360°$.

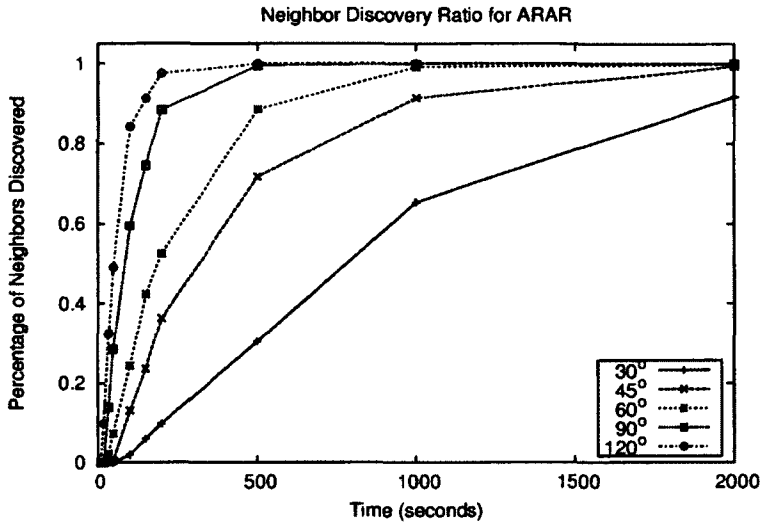


Figure 6.6: Neighbor discovery results for ARA algorithm with prime number coloring.

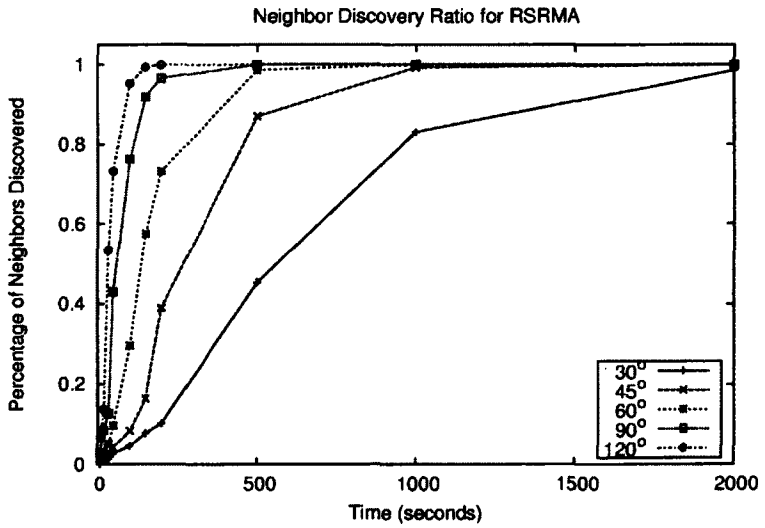


Figure 6.7: Neighbor discovery for $RSRMA$ algorithm with varying beam width.

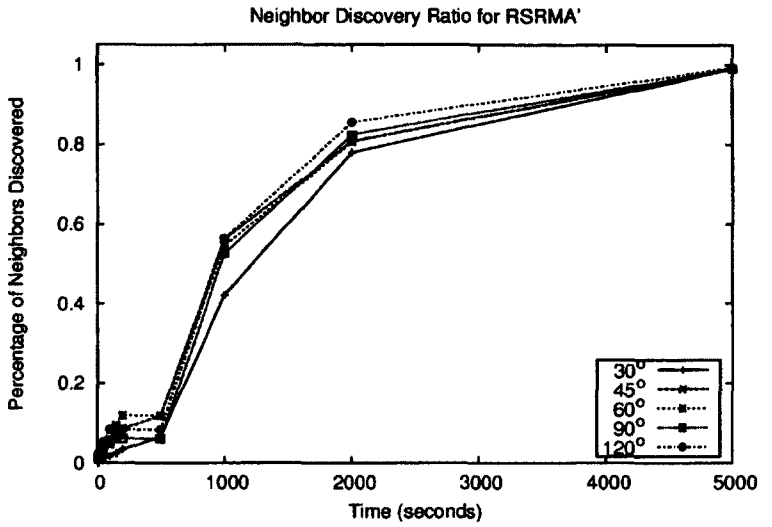


Figure 6.8: Neighbor discovery results for $RSRMA'$ algorithm with varying beam width.

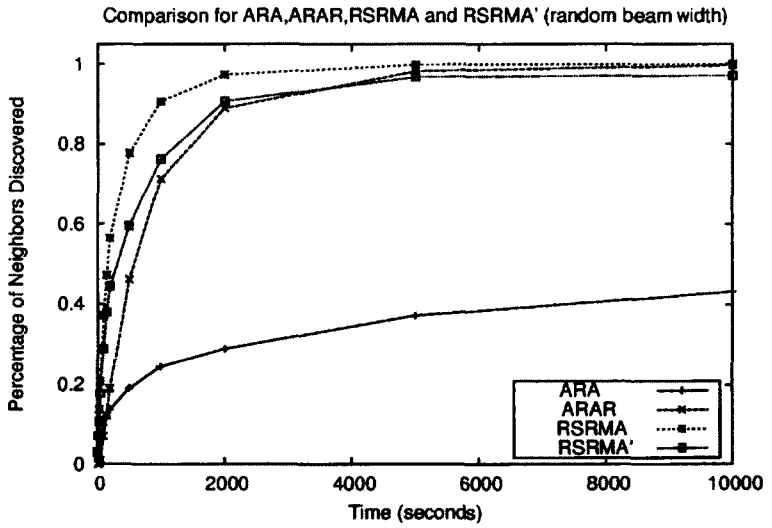


Figure 6.9: Algorithm comparison with beam width randomly chosen.

6.4.4 *RSRMA* Delay Comparison with Omnidirectional Antenna

When algorithm *RSRMA* is used, we compare its performance of using directional antennae with that of omnidirectional antennae (360°). Figure 6.10 shows that *RSRMA* algorithm can achieve desirable neighbor discovery delays when antenna beam width is 120°. Although directional antenna is not as fast as its omnidirectional counterpart, it consumes significant less energy, which we will show in section 6.4.5.

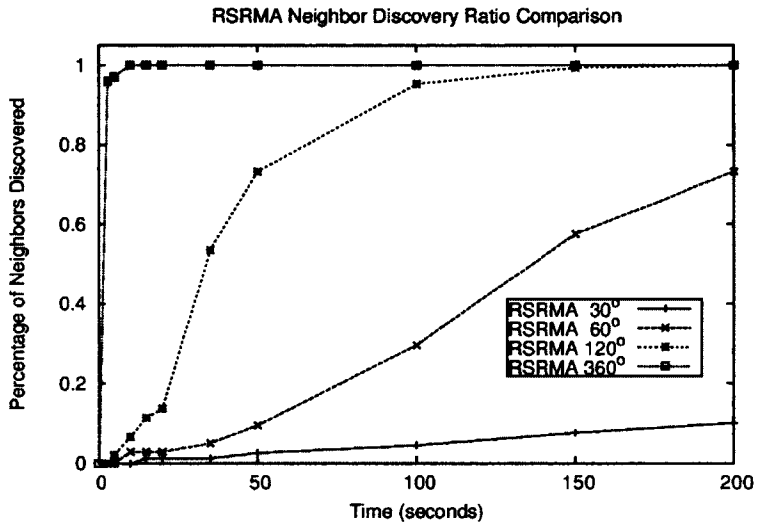


Figure 6.10: Delay comparison with omnidirectional antenna in neighbor discovery.

6.4.5 Energy Comparison

We compare the energy consumption (only transmission energy is collected) of different antenna rotation algorithm with different beam widths. The transmission and reception gains (compared with omnidirectional antenna) for a directional antenna with beam width α (in radians) are set as $\frac{2\pi}{\alpha}$ (two dimensional antenna gain). Table 6.4 summarizes the results. The transmission interval is one second and the total energy consumption is collected at 20000 seconds. For the same antenna type, different algorithms do not have much impact on the results. However, the energy consumption of all directional antennae are much smaller than that of the omnidirectional antenna (2π).

Antenna at u	$ARAR$	$RSRMA$	$RSRMA'$
$\pi/6$	1.2923 $\pm$0.0005	1.2879 $\pm$0.0002	1.3056$\pm$ 0.0013
$\pi/4$	2.9307 $\pm$0.0024	2.9106$\pm$ 0.0015	2.9378$\pm$ 0.0031
$\pi/3$	5.2638$\pm$ 0.0066	5.2082$\pm$ 0.0049	5.2356$\pm$ 0.0049
$\pi/2$	12.1621$\pm$ 0.033	11.9635$\pm$ 0.0329	12.4557$\pm$0.0681
$2\pi/3$	22.3643 $\pm$ 0.0872	21.7582$\pm$ 0.0823	21.9526$\pm$ 0.0883
2π	281.087$\pm$3.3943	281.087$\pm$3.3943	281.087$\pm$3.3943

Table 6.4: List of energy consumptions for the algorithms.

6.5 Conclusion and Open Problems

An interesting class of problems arises in considering the efficiency of broadcasting in the single channel UDG model, i.e., 1) there is a single send/receive channel and multiple transmissions on the same node produce packet collisions, and 2) a link between two sensors u, v exists if and only if $d(u, v) \leq 1$. In general, broadcasting with omnidirectional antennae requires scheduling of transmissions (typically using *group testing* techniques) so as to avoid collisions. Clearly, if broadcasting time with omnidirectional antennae without collisions is B then the result of Theorem 10 indicates that broadcasting in the directional antennae model can be accomplished in time $O(B(c \ln c)^3)$, where c is the number of colors of a vertex coloring of the sensor network. The main question arising is whether we can improve on this time bound when using directional antennae.

Chapter 7

Cooperative Neighbor Discovery using Two Antennae Patterns

7.1 Introduction

Omnidirectional antennae are widely used in wireless network communications where nodes are randomly deployed. A node transmits data in all directions with the same transmitting power. Although wireless nodes can make use of omnidirectional antennae in the communication process, it is not efficient in terms of energy savings when two nodes communicate with each other because most transmitting energy is wasted. Directional antennae [113], on the other hand, reduce energy consumption and consequently can increase transmission distance. They can also reduce wireless contentions. With the increasing connection opportunities, communication delays can be reduced when directional antennae are applied in the network.

A node can get its location information either through *GPS* or Time of Arrival (TOA), Time Difference of Arrival (TDOA) or Angle of Arrival (AOA) techniques ([114], p.167). Some existing localization algorithms [115] can be applied in the location calculation when the number of anchor nodes is limited.

With the application of directional antennae, routing delays in communication can be reduced because less routing hops are needed. In a dynamic network however, nodes have to collect neighboring nodes' information through a neighbor discovery process before the advantages of directional antennae can be put into reality.

Since the purpose of neighbor discovery is to inform the nearby nodes of a node's existence, broadcast is apparently more efficient. However, long range omnidirectional antennae will introduce too many contentions. In this chapter, we propose the use of omnidirectional antennae for discovering nearby neighbors first. Our protocol then uses long range directional antennae to discover neighboring nodes which can

not be discovered through the omnidirectional antenna neighbor discovery. We propose neighbor cooperation mechanisms to speed up the neighbor discovery processes.

7.1.1 Contributions and Organization of the Chapter

In this chapter, we propose a novel neighbor discovery scheme in wireless networks in the presence of two antennae patterns. Nodes use omnidirectional antennae with small communication range to broadcast their existence and use long range directional antennae to find nodes that can not be reached in distance k (omnidirectional antennae) neighborhood. In the proposed solution, nodes cooperate with each other to speed up the neighbor discovery process. Through analysis, we compare the delay performance of existing neighbor discovery protocols when contentions in the neighbor discovery process are taken into consideration and identify the critical points in the delay performance. And the analysis clearly shows that neighbor discovery delay is reduced through the cooperative scheme. Through simulation, we also show the desirable delay and energy performance of the proposed solution.

The rest of the chapter is organized as follows. Section 7.2 elaborates on our proposed solutions. We theoretically analyze the proposed scheme in Section 7.3. Section 7.4 describes the details of experiments and analysis. Section 7.5 concludes with possible future work.

7.2 Cooperative Neighbor Discovery Strategy Using Two Antenna Patterns

In this section, we propose the use of local omnidirectional (omnidirectional distance k neighborhood, for some small k) and global directional antenna patterns in neighbor discovery. We assume a node can get its location information either through GPS or localization algorithms, which are reasonable assumptions with existing technologies.

7.2.1 Overview of Neighbor Discovery using Directional and Omnidirectional Antennae

Unlike data transmissions which mainly focus on specific users, the goal of neighbor discovery is to find all neighbors as quickly as possible. Although an omnidirectional antenna is ideal for this broadcasting based neighbor discovery purpose, a lot of contentions can occur when the number of neighboring nodes increases. Compared with omnidirectional antennae, directional antennae incur significant less contentions and can communicate with more nodes with the same power when a node rotates its antenna to cover a total of 2π degrees.

With communication range r, an omnidirectional antenna will consume power proportional to πr^2 while an ideal directional antenna with beam width α radians will consume power proportional to $\frac{\alpha}{2}r^2$ ([116]). Figure 7.1 shows the comparison of directional and omnidirectional antennae. When transmitting power is the same, the narrower the beam width α of the directional antenna, the longer the directional antenna radius r_d. If the omnidirectional antenna radius is r_o, then $r_d = r_o \times \sqrt{\frac{2\pi}{\alpha}}$, when both transmission powers are the same. In practice, long range communications usually are performed using directional antennae (e.g., satellite communications).

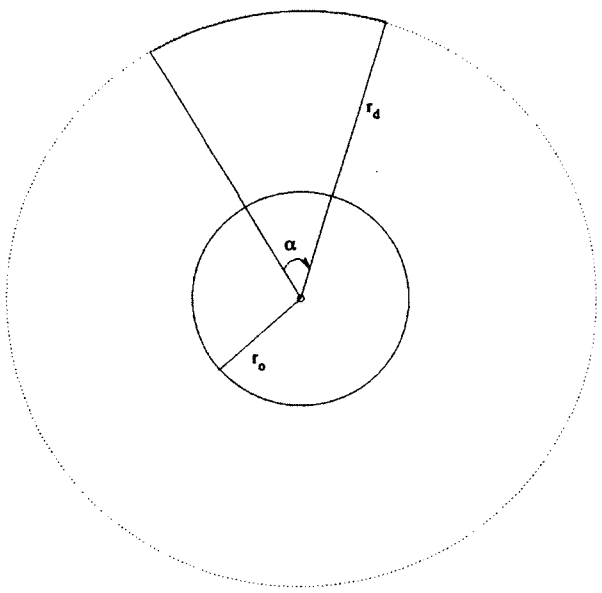


Figure 7.1: Omnidirectional and directional antennae.

We consider the situation when directional and omnidirectional antennae are both available. We denote DD directional transmission and reception and OO omnidirectional transmission and reception. Neighbor discovery consists of cycles of Omnidirectional Neighbor Discovery (OND) process and Directional Neighbor Discovery (DND) process. Nodes cooperate with each other to speed up the propagation of neighbor information in both processes.

7.2.2 Neighbor Discovery Using Omnidirectional Antenna

The proposed solution combines two processes within one neighbor discovery cycle. First, $ONDP$ is applied in omnidirectional antenna distance k neighbors of a node. Nodes exchange information of their own, as well as the discovered omnidirectional distance k neighbors.

In the omnidirectional neighbor discovery process, a node broadcasts its existence, together with its coordinates to its neighbors. A node keeps a list of neighboring nodes. It first discovers its distance one neighbors. And in the next cycle, each node will broadcast its own location information, together with its distance one neighbor information. Distance two neighbor information will be exchanged in the following cycle. If distance i ($i < k$) information has been found, distance $i + 1$ information will be exchanged in the next cycle. This process repeats until all distance k neighbor information has been exchanged. A node updates its distance k neighbor list whenever it receives new neighbor information. The format of the beacon packet for a node b is shown in the following:[1]

[1]Here x, y, z denote the x coordinate, y coordinate and z coordinate of a node with the corresponding ID, where the subscript of the values for a node with ID_{ij} means that the node is in distance i ($1 \leq i \leq k$) omnidirectional neighborhood and it is the j-th ($1 \leq j \leq m_i$) discovered distance i neighbor.

$$\begin{cases} \text{Own information: } (ID_b, x_b, y_b, z_b), \\ \text{Omnidirectional antennae distance } 1..k \text{ neighbor information} \\ \text{distance 1 } (ID_{11}, x_{11}, y_{11}, z_{11})(ID_{12}, x_{12}, y_{12}, z_{12}) \cdots (ID_{1m_1}, x_{1m_1}, y_{1m_1}, z_{1m_1}) \\ \text{distance 2 } (ID_{21}, x_{21}, y_{21}, z_{21})(ID_{22}, x_{22}, y_{22}, z_{22}) \cdots (ID_{2m_2}, x_{2m_2}, y_{2m_2}, z_{2m_2}) \\ \cdots \\ \text{distance } k \ (ID_{k1}, x_{k1}, y_{k1}, z_{k1})(ID_{k2}, x_{k2}, y_{k2}, z_{k2}) \cdots (ID_{km_k}, x_{km_k}, y_{km_k}, z_{km_k}) \end{cases} \tag{7.1}$$

7.2.3 Neighbor Discovery Using Directional Antenna

Although neighbor discovery process using omnidirectional antennae is simpler than the process of using directional antennae because it does not need to scan its neighborhood area one by one, it may not be able to discover all the neighbors because of its communication range limitations. In Figure 7.2 for example, nodes b, c can not communicate with node a without using directional antennae.

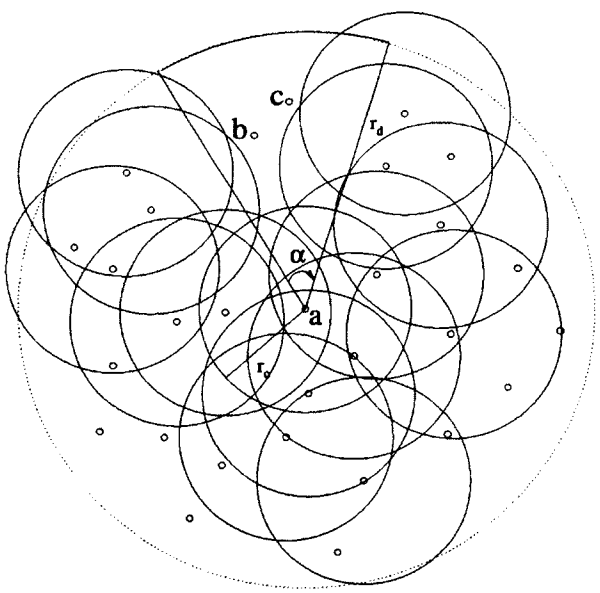


Figure 7.2: Directional neighbor discovery.

After the omnidirectional antenna neighbor discovery process, directional antennae are used. Existing neighbor discovery algorithms (e.g., $RSRMA$ algorithm in [4]) can be used in the proposed solution. A node within the directional antenna coverage area which has already received the omnidirectional neighbor discovery information may not reply to the directional neighbor discovery inquiry to reduce contentions.

7.2.4 Cooperative Neighbor Discovery

When directional neighbor discovery is applied, it is possible that some nodes within the directional antenna coverage area may not receive a beacon message because their antennae rotate in directions that are unavailable to receive or because there exist other reasons (e.g., due to contentions), while some nearby nodes can properly receive it. These nearby nodes can cooperate with the nodes which fail to receive beacon signals. They will exchange the received directional information through omnidirectional antennae. In Figure 7.3, when node b receives location information from node a, it will broadcast this information to all its neighbors. Through the cooperative operation, neighbor discovery delay can be reduced.

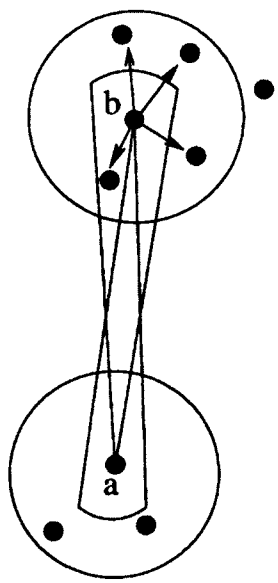


Figure 7.3: Cooperative neighbor discovery.

In the cooperation process, directional antennae can also be used in the propagation of omnidirectional neighbor information, i.e., a node will transmit its omnidirectional neighbor information in its beacon packet. In $ONDA$, a node transmits its distance k omnidirectional neighbor information, together with its directional neighbor information to all its omnidirectional neighbors through omnidirectional antenna. In $DNDA$, a node only transmits its distance k omnidirectional neighbor information to all its directional neighbors through directional antenna. After the reception of the neighbor information, a node can decide independently which nodes are its neighbors (both omnidirectional and directional). The format for both directional and omnidirectional beacon packets is

$$\begin{cases} \text{Own information: } (ID_b, x_b, y_b, z_b), \\ \text{Omnidirectional antennae distance } 1..k \text{ neighbor information} \\ \text{distance 1 } (ID_{11}, x_{11}, y_{11}, z_{11})(ID_{12}, x_{12}, y_{12}, z_{12}) \cdots (ID_{1m_1}, x_{1m_1}, y_{1m_1}, z_{1m_1}) \\ \text{distance 2 } (ID_{21}, x_{21}, y_{21}, z_{21})(ID_{22}, x_{22}, y_{22}, z_{22}) \cdots (ID_{2m_2}, x_{2m_2}, y_{2m_2}, z_{2m_2}) \\ \cdots \\ \text{distance } k \ (ID_{k1}, x_{k1}, y_{k1}, z_{k1})(ID_{k2}, x_{k2}, y_{k2}, z_{k2}) \cdots (ID_{km_k}, x_{km_k}, y_{km_k}, z_{km_k}) \\ \text{Directional antennae neighbor information} \\ \text{(optional and default none for directional beacon)} \\ (ID_1, x_1, y_1, z_1)(ID_2, x_2, y_2, z_2) \cdots (ID_m, x_m, y_m, z_m) \end{cases} \tag{7.2}$$

7.2.5 Cooperation Mechanisms

Consider a set of nodes in the plane with directional communication range r_d. We use the directional antenna rotation algorithm $RSRMA$ [4] for the discovery of neighbors. We propose cooperation mechanisms in the following.

Non-clustered random cooperation

In this model, a node randomly chooses its directional antenna direction and runs Random Selection Rotation Mechanism Algorithm (RSRMA) independently. Nodes cooperate with each other in their distance k omnidirectional neighborhood during neighbor discovery cycle for information on discovered directional neighbors.

We show the improvement of the random cooperation approach in distance one omnidirectional neighborhood using Figure 7.4. In the figure, node a and its omnidirectional distance one neighbors are within the directional antenna coverage areas of the nodes around b, and vice versa. Once node a and node b discover each other, they can exchange neighbor information for all the nodes around a and b. Without cooperation, each node within distance one neighborhood of a has to discover each node within distance one neighborhood of b independently. Assume this probability is p_r. Clearly, the expected time for node a to discover node b independently is $\frac{1}{p_r}$. Also assume nodes have degree d (omnidirectional), contentions and time in omnidirectional communication is negligible. Then the probability that at least a node from distance one neighborhood around a (including a) discover at least a node from distance one neighborhood around b (including b) is $1-(1-p_r)^{(d+1)^2}$. With cooperation, we can take the inverse to calculate the expected time for a to discover b ($\frac{1}{1-(1-p_r)^{(d+1)^2}}$), which is much faster, especially when p_r is small and d is large.

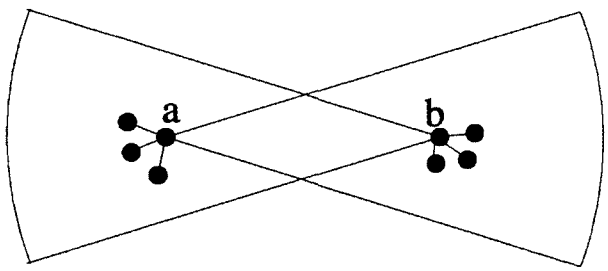


Figure 7.4: Random cooperation.

Clustered minimum delay coverage

Clustered solution can be adopted only when nodes can quickly discover all omnidirectional distance k nearby neighboring nodes. In this model, nodes establish

clusters first. We define that the weight of a node v by $w(v) = (deg(v), ID(v))$, which is first distinguished by node degree $deg(v)$ (omnidirectional), the higher degree the higher weight and then by node ID, the bigger node ID the higher weight. Existing cluster formation algorithms ([93]) can be used based on the node weights and the node with the largest weight is the clusterhead. Nodes in a cluster follow the same antennae rotation mechanism. After cluster formation, nodes will point their antennae so that when they rotate, they will cover all the coverage area within minimum rotation time. Clustered solution requires at most distance two information exchange within the cluster if the clusterhead can reach the other nodes in its cluster directly [93].

Since a clusterhead knows how many nodes are in its cluster, it can assign nodes in its cluster with different antenna orientation directions. For example, in Figure 7.5, the clusterhead can decide the three nodes (including itself) in its cluster point their antennae in sectors 0 (clusterhead), 4 and 8 when there are 12 antenna rotation sectors. The sector distance between two consecutive directional antenna directions is 4 sectors. The assignment is chosen in such a way that the differences of the sector distances are less or equal to one. When nodes exchange distance k ($k \geq 3$) neighbor information, each node can calculate independently the cluster information and therefore there is no need for extra cluster formation messages. Nodes can also decide their antenna orientation directions independently. A clusterhead will indicate its antenna direction (e.g., in sector zero if the sectors are numbered from 0 to $k_s - 1$) and the antenna rotation mechanism of $RSRMA$ in its omnidirectional beacon message. All other nodes inside its cluster will follow the same antenna rotation mechanism. Their antenna orientation directions are chosen independently by nodes according to the following rules. Starting from the antenna orientation sector of the clusterhead, if there are $k_s - 1$ other nodes inside the cluster, then these nodes will select sectors consecutively, starting from sector 1 to sector $k_s - 1$ according to the decreasing order of their weights. If the number of nodes in the cluster is $c < k_s$. We set $d = \lfloor \frac{k_s}{c} \rfloor$. Then there exist $i, j \in N$ with $i + j = c$ such that $id + j(d+1) = k_s$. Nodes select their antennae directions according to the decreasing order of their weights. Starting from the clusterhead, the first i nodes will select

their antennae directions separating d sectors apart, and the remaining j nodes will select their antennae directions separating $d+1$ sectors apart.

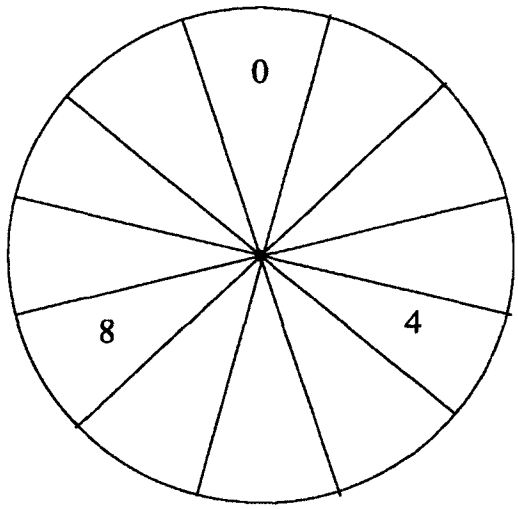


Figure 7.5: Antenna direction.

Selective Directional Scan

Selective directional scan suits the situation when omnidirectional antennae are used in distance k neighborhood of nodes and the maximum range of distance k neighbor exceeds the directional antenna communication range. In this model, directional antennae are only used to scan the area that is not covered by omnidirectional distance k neighbor discovery process.

After omnidirectional neighbor discovery, a node can calculate the directional sectors that are not covered by omnidirectional neighborhood, i.e., there are possible neighbors in those sectors that are yet to be discovered. In Figure 7.6 for example, assume after OND node a finds that all neighboring nodes except those located in the three shaded sectors have been discovered. In DND, node a then runs the modified $RSRMA$ algorithm which only scans these three sectors and for each sector the delay time is the same as the original $RSRMA$ algorithm. The neighboring node b follows the same process. They can discover each other with high probability (refer [4] for the proof). Since the number of scanned sectors is less than the original k,

neighbor discovery delay is reduced.

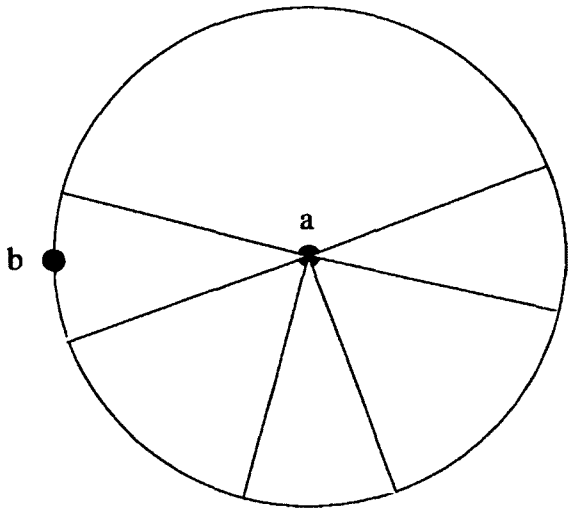


Figure 7.6: Selective directional scan.

7.3 Delay and Energy Analysis

In the analysis, we quantitatively compare the cooperative two antennae approach with the pure omnidirectional (OO) and directional (DD) approaches, as well as omnidirectional transmission and directional reception (OD) and directional transmission and omnidirectional reception (DO) approaches in the literature. Assume a successful neighbor discovery happens when no collision occurs. Through analysis, we show that our approach is faster than other approaches with high probability when the number of neighboring nodes increases following a Poisson distribution. Table 7.1 summarizes the results of this section.

7.3.1 Omnidirectional Transmission and Reception

We present the omnidirectional antenna delay and energy analysis, where delay is defined as the time slots needed to discover neighbors. Assuming that nodes are distributed uniformly and independently following the Poisson process, we calculate the delay for omnidirectional antenna when there are n neighboring nodes in the coverage area of a node. If a node broadcasts in a time slot with probability p, then

Protocol	Running Time	Transmitter Gain	Receiver Gain
OO	$T_o = \frac{1}{p(1-p)^{n-1}}$	1	1
DD	$T_d = \frac{3k^2 n \log n}{(n-1)p(1-p)^{\frac{n}{k^2}-1}}$	$\frac{2\pi}{\phi}$	$\frac{2\pi}{\phi}$
OD	$T_{od} = \frac{1}{p(1-p)^{\frac{n}{k}-1}}k$	1	$\frac{2\pi}{\phi}$
DO	$T_{do} = \frac{1}{p(1-p)^{\frac{n}{k}-1}}k$	$\frac{2\pi}{\phi}$	1
$D+O$	$T_{d+o} = \frac{T_d \times p_r}{1-(1-p_r)^{(d+1)^2}}$ (random cooperation)	$\frac{2\pi}{\phi}(D)$; $1(O)$	$\frac{2\pi}{\phi}(D)$; $1(O)$

Table 7.1: List of running times to discover a neighboring node and transmitter/receiver gains.

the expected time for a node to discover another node is $T_o = \frac{1}{p(1-p)^{n-1}}$. The energy consumption is proportional to πr^2 for transmission, and we set the receiver gain as 1.

7.3.2 Directional Transmission and Reception

We show directional antenna delay and energy analysis in the following. For directional antenna, if there are k sectors, then the expected time for a node to discover another node is $T_d = \frac{1}{p(1-p)^{\frac{n}{k^2}-1}} \times \frac{3k^2 \log n}{1-\frac{1}{n}} = \frac{3k^2 n \log n}{(n-1)p(1-p)^{\frac{n}{k^2}-1}}$ when $RSRMA$ algorithm ([4]) is applied in directional neighbor discovery. Clearly, when n increases, $T_o \geq T_d$ will occur (i.e., when $\frac{3k^2 n \log n}{n-1}((1-p)^{(1-\frac{1}{k^2})})^n \leq 1$). The energy consumption is proportional to $\frac{\phi}{2}r^2$ for transmission and the receiver gain is $\frac{2\pi}{\phi}$.

7.3.3 Omnidirectional Transmission with Directional Reception

Since there has been some work on omnidirectional transmission and directional reception, its delay and energy analysis is presented here. The expected time for a node to discover another node is $T_{od} = \frac{1}{p(1-p)^{\frac{n}{k}-1}}k$ when the directional antennae scan the consecutive sectors one by one with no extra sector delay. Clearly, when n increases, $T_{od} \geq T_d$ will happen (i.e., when $\frac{3kn \log n}{n-1}((1-p)^{(\frac{1}{k}-\frac{1}{k^2})})^n \leq 1$). The energy consumption is proportional to πr^2 for transmission and the receiver gain is $\frac{2\pi}{\phi}$.

7.3.4 Directional Transmission with Omnidirectional Reception

This part discusses directional antenna transmission and omnidirectional reception delay and energy analysis. The expected time for a node to discover another node is $T_{do} = \frac{1}{p(1-p)^{\frac{n}{k}-1}}k$ when the directional antennae scan the consecutive sectors one by one with no extra sector delay. Clearly, when n increases, $T_{do} \geq T_d$ will happen (i.e., when $\frac{3kn\log n}{n-1}((1-p)^{(\frac{1}{k}-\frac{1}{k^2})})^n \leq 1$). The energy consumption is proportional to $\frac{\phi}{2}r^2$ for transmission and the receiver gain is 1. It is clear that DO has the same delay performance with OD in the analysis.

7.3.5 Cooperative Two Antennae Patterns

The proposed solution $(D+O)$ uses cooperative omnidirectional and directional antennae neighbor discovery approaches. The energy consumption is proportional to $\frac{\phi}{2}r^2 + \pi r_o^2$ for transmission (where r, r_o are the radius of directional and omnidirectional antennae respectively), and the receiver gain is $\frac{2\pi}{\phi}$ for directional antenna and 1 for omnidirectional antenna. The expected time T_{d+o} for a node to discover a neighboring node depends on the cooperation mechanism. It is clear that $T_{d+o} \leq T_d$. In the random cooperation mechanism, we have shown that $T_{d+o} = \frac{T_d \times p_r}{1-(1-p_r)^{(d+1)^2}}$ using only omnidirectional distance one cooperation, which is much faster than T_d.

7.4 Experimental Evaluation

In order to evaluate our delay and energy saving strategies using the cooperative two antenna patterns, we perform relevant simulations. During the experiments, we pay great attention to the key attributes, including delay performance and energy efficiency of the protocols.

7.4.1 Simulation Environment

The proposed solution is implemented using the NS-2 [1] simulator. This simulation environment includes full simulation of the wireless communication physical layer, which makes it better reflect the real world. A random waypoint model is chosen as the motion pattern. For the propagation model, we have chosen *Two*

Ray Ground which considers both the direct path and a ground reflection path. We have migrated directional antenna code from IIT TENS project ([111], based on NS2 version 2.19b) to NS-2.33 and added the ideal directional antenna. The simulation parameters are shown in Table 7.2.[2] The beacon interval is one second.

Table 7.2: Parameters of the simulations.

Parameter	Value
Number of mobile nodes	25/50 (default)/75/100/125/150
Beam Width Angle	20°, 30°, 45°, 60°, 90°, 120° (default)
Mobility	none
Transmission range (D)	100/150/200/250 (default)/300/350/500m
Transmission range (O)	100/150/200/250(default)/300/350/500m
Data rate	1 Mbps
Propagation model	*Two Ray Ground*
Simulation time	1000 seconds
Link layer queue length	50
Topology size	1000m×1000m
Antenna model	directional/omnidirectional

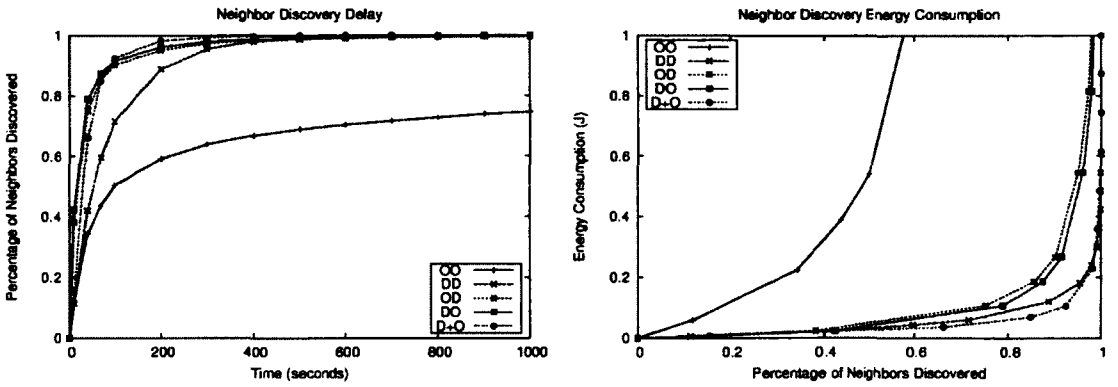
Simulation results in the proposed neighbor discovery delay tolerance and energy saving performance are collected. In the simulation, we assume nodes have no knowledge of the number of neighbors around. A node decides whether to transmit or receive in a time slot randomly, i.e., the node tosses a coin and decides to transmit when it is head and to receive when it is tail. The simulation implements the random cooperation in omnidirectional distance one neighborhood. Through simulation, we have shown the improved neighbor discovery effectiveness of the cooperative approach through the comparison with other protocols.

For the simulation results, all points in the figures are obtained as an average of 20 different runs with 20 different network topologies and movement patterns.

7.4.2 Delay and Energy Comparisons

The delay and energy performances of OO, DD, OD and DO with 250 meters communication range in both antenna types are tested (the number of nodes is

[2]The 250m default omnidirectional communication range is used only in OO, OD, DO, while the omnidirectional communication range in $D + O$ is 100m.



(a) Delay comparison (b) Energy comparison

Figure 7.7: Comparison of delay and energy in 250m communication range and 120 degree, 50 nodes.

50 and the directional antenna angles are 120 degree), which are also compared with the proposed cooperative solution with 250m directional communication range (omnidirectional communication range in $D + O$ is 100m which is used throughout the simulation.[3]). The data on the percentage of neighbors discovered is collected. Figure 7.7a shows the results. Energy consumption is also collected when certain percentages of the neighbors have been discovered, which is shown in Figure 7.7b.

It is clear that the delay performance of the proposed solution is better than other approaches in discovering all neighbors. The energy consumption of the proposed cooperative approach is also better than the energy consumptions of OO, OD, DO and DD when the same percentage of neighbors are discovered.

7.4.3 Factors Affecting Delays

Different communication ranges, total number of nodes or directional antennae angles can change the delay performance of the protocols. The proposed solution performs the best when node communication range increases or when there are more nodes in the network. Otherwise it degrades its performance gracefully and is very close to the best solution.

[3]The meaning of communication range in the following refers to the parameter of all antennae in the simulated protocols except omnidirectional communication range in $D + O$.

Effects of Communication Ranges

With the same node deployment, node communication range plays an important role in the neighbor discovery process because the increase in the range means the increase in the average number of neighboring nodes, which can lead to a contention increase. The communication range effects on the neighbor discovery process are shown in Figures 7.8a (100 meters), 7.8b (150 meters), 7.8c (200 meters), 7.8d (300 meters), and 7.8e (350 meters).

Simulation results show that when the node communication ranges are 100 meters and 150 meters, all approaches have very similar performance. However, when the node communication ranges are 300 meters and 350 meters, the proposed solution significantly outperforms other approaches.

Effects of the Number of Nodes

The effects of various node densities are also evaluated. We compare the situations when there are 25, 75, 100, 125 and 150 nodes at 250m communication range and 120 degree. The results are shown in Figures 7.9a (25 nodes), 7.9b (75 nodes), 7.9c (100 nodes), 7.9d (125 nodes), and 7.9e (150 nodes). When the node densities become larger, there are contention increases. However, the increase in the proposed solution is much smaller than the increases of other approaches. When the number of nodes increases over 100, the proposed approach is the best in neighbor discovery delay performance among all the protocols.

Effects of Directional Angles

The directional angle of nodes affects the delay performance because varying angle changes the number of neighboring nodes when directional antennae are used. The simulation results are shown in Figures 7.10a (30 degree), 7.10b (45 degree), 7.10c (60 degree), and 7.10d (90 degree).

Since the smaller angle increases the number of sectors, OD and DO outperform the proposed solution when there are 50 nodes in 250 meter communication range. In all the scenarios, with the help of omnidirectional neighbors, the proposed solution

is better than the *DD*. In all the delay comparisons, *OD* and *DO* have very similar simulation results. When the number of nodes and node communication range increase, the proposed solution improves performances quickly, even when the directional angle is 20 degree. The results are shown in Figures 7.11a (50 nodes and 250 meters), 7.11b (150 nodes and 350 meters), and 7.11c (150 nodes and 500 meters). In Figure 7.11c, the proposed solution is the best when the neighbor discovery time exceeds 600 seconds.

7.4.4 Factors Affecting Energy Consumption Efficiencies

We also check the energy efficiencies of protocols when certain percentage of neighbors are discovered. The energy is computed according to the number of bits transmitted. Although the proposed cooperative solution transmits more bits, it is more energy efficient than other protocols in most situations. The results of node communication range, total number of nodes and directional antennae angle impacts on energy efficiency are collected, as shown in the following.

Effects of Communication Ranges

Same as the delay performance, node communication range also plays an important role in the energy efficiency of the neighbor discovery process. The results of the node communication range effects on the energy efficiency are shown in Figures 7.12a (100 meters), 7.12b (150 meters), 7.12c (200 meters), 7.12d (300 meters), and 7.12e (350 meters).

Simulation results show that the proposed solution has the best energy efficiency performance when node communication ranges are 300 meters and 350 meters.

Effects of Number of Nodes

The effects of different node densities on energy efficiencies are also evaluated. The results are shown in Figures 7.13a (25 nodes), 7.13b (75 nodes), 7.13c (100 nodes), 7.13d (125 nodes), and 7.13e (150 nodes). It is clear that the proposed solution is the most energy efficient protocol when different number of nodes are available (except in the 25 nodes scenario).

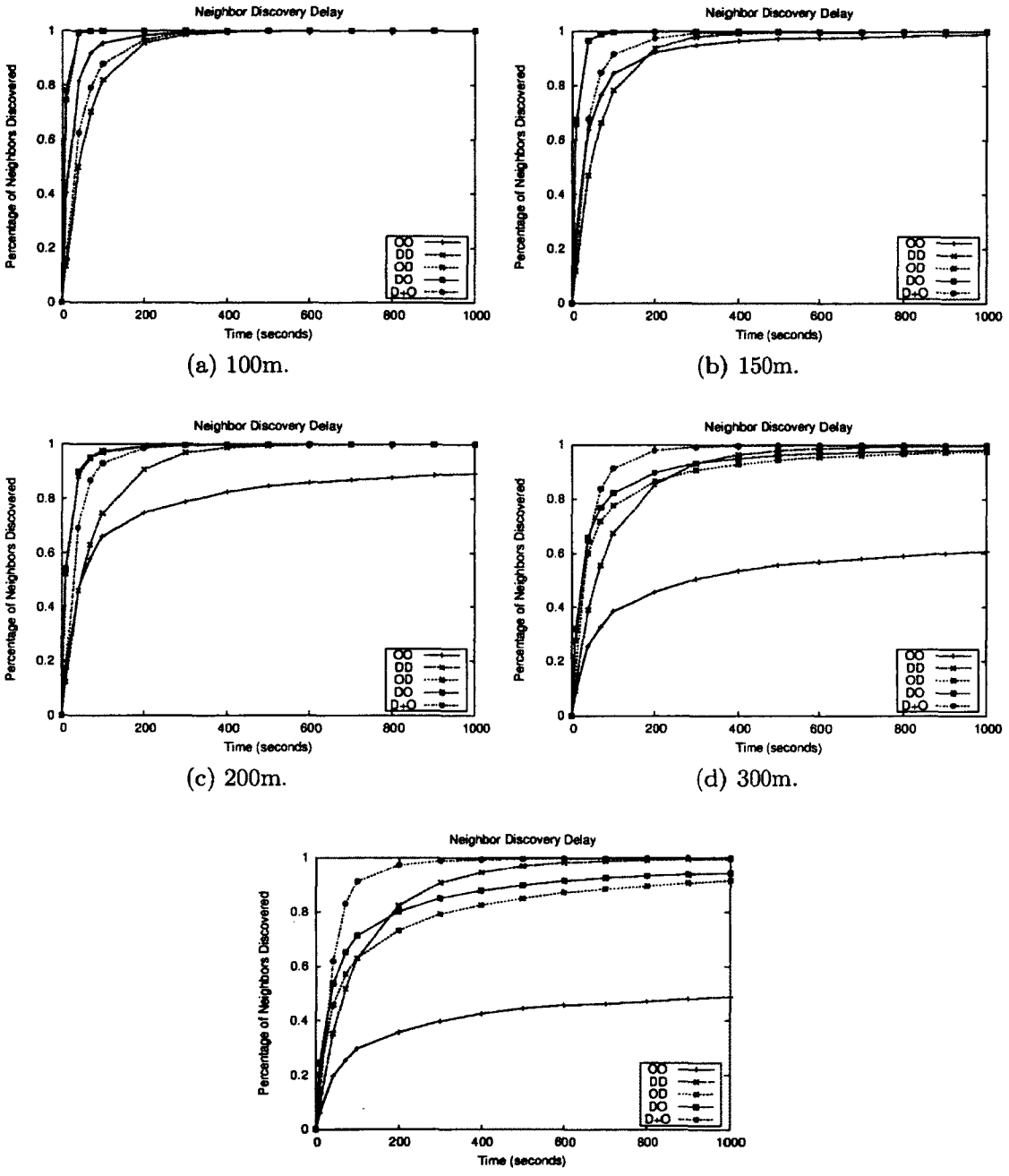
Effects of Directional Angles

The effects of directional angles on the energy efficiencies are similar to the effects on the delay performance. The results are shown in Figures 7.14a (30 degree), 7.14b (45 degree), 7.14c (60 degree), and 7.14d (90 degree).

Although the energy efficiency of the proposed solution is not the best when both directional angle and the communication range are small, the performance differences are not significant. When the number of nodes and node communication range increase, the proposed solution outperforms *OO*, *OD* and *DO*, as shown in Figure 7.15c (the directional antenna angle is 20 degree and there are 150 nodes with communication range 500 meters). The effects of varying number of nodes and different communication range settings when the directional antenna angle is 20 degree are shown in Figure 7.15a (50 nodes and 250 meters) and 7.15b (150 nodes and 350 meters).

7.5 Conclusions

We have proposed a novel cooperative neighbor discovery mechanism in wireless networks. In the proposed solution, nodes use small range omnidirectional antennae to discover nearby neighbors and use long range directional antenna to find nodes that can not be reached by omnidirectional antennae. Nodes cooperate with each other to speed up the neighbor discovery process. Through analysis, we have shown that the proposed solution has advantages over other neighbor discovery mechanisms when each node has more neighbors around. Simulation results show that the proposed solution has increased energy efficiency with reduced delays when node communication range or (and) the number of nodes increase. As future work, we plan to further study directional communication establishment in a wireless network with disruptions and delays, exploring experimentally the integrated directional antenna mechanisms into an applicable solution.



(a) 100m.

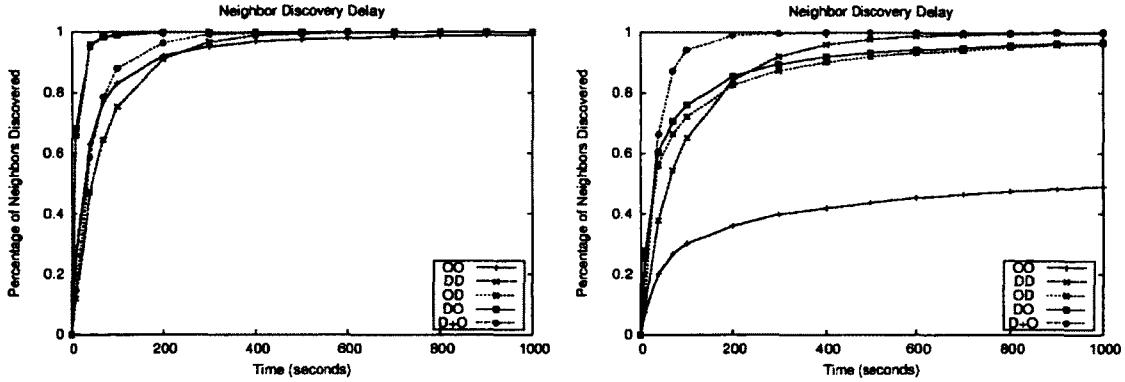
(b) 150m.

(c) 200m.

(d) 300m.

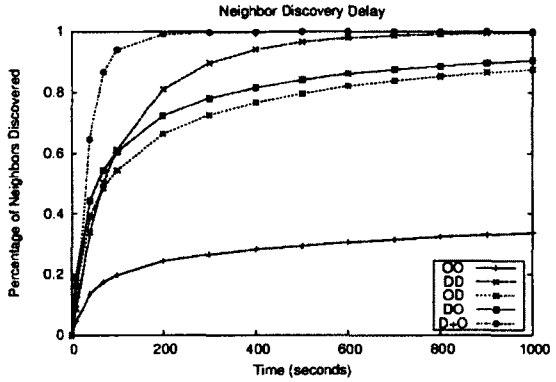
(e) 350m.

Figure 7.8: Comparison of percentage of neighbors discovered at various communication ranges, 50 nodes and 120 degree.

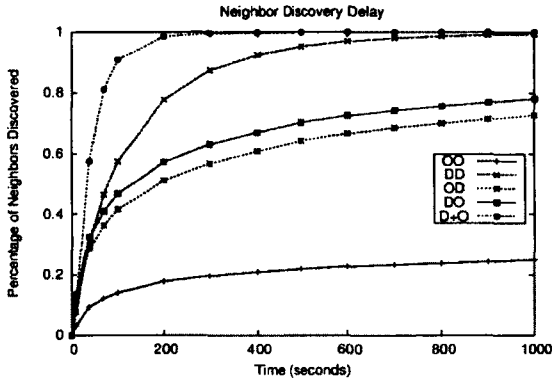


(a) 25 nodes.

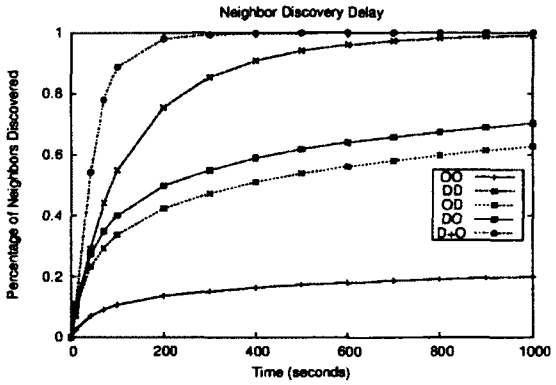
(b) 75 nodes.



(c) 100 nodes.

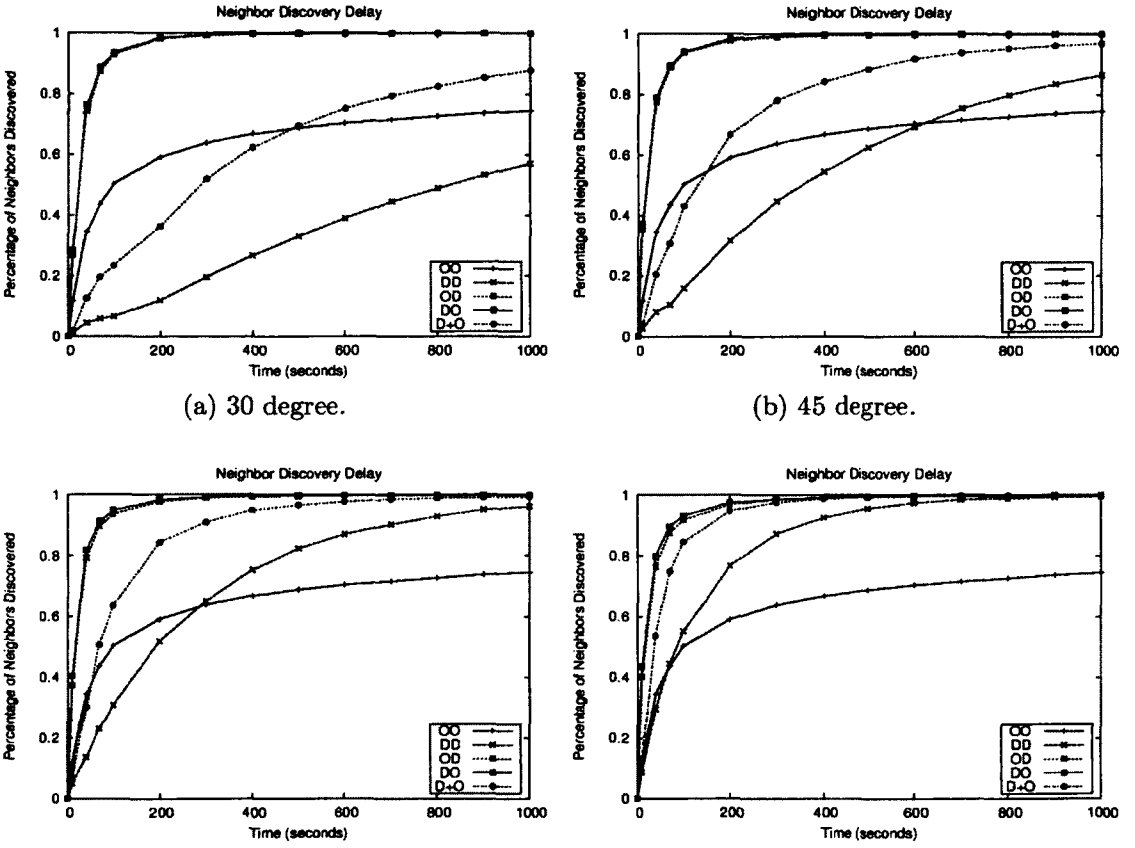


(d) 125 nodes.



(e) 150 nodes.

Figure 7.9: Comparison of percentage of neighbors discovered at various number of nodes, 250m communication range and 120 degree.



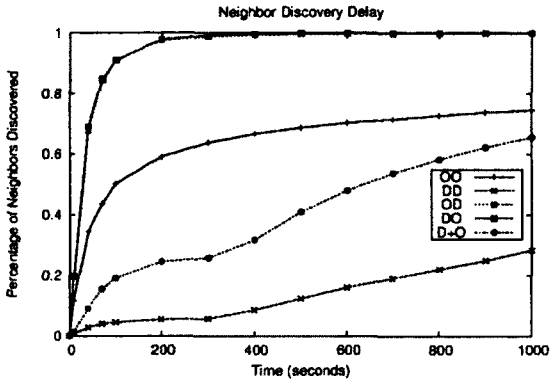
(a) 30 degree.

(b) 45 degree.

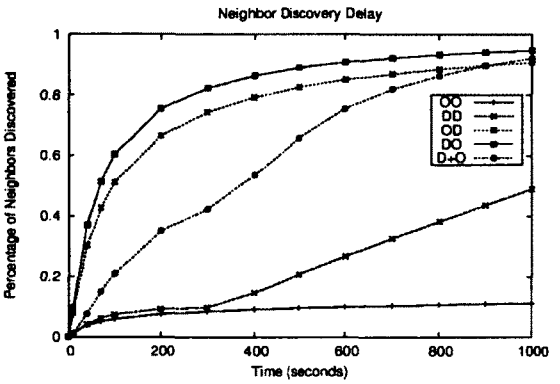
(c) 60 degree.

(d) 90 degree.

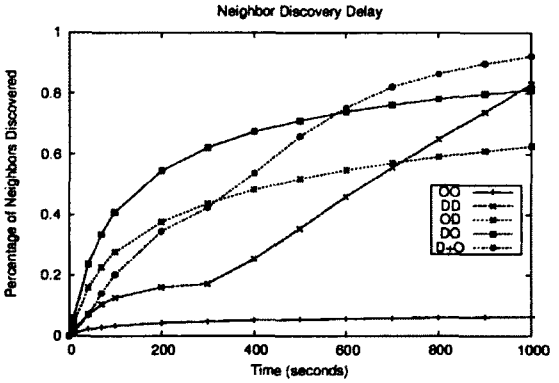
Figure 7.10: Comparison of percentage of neighbors discovered at various angles, 50 nodes and 250m communication range.



(a) 20°, 50 nodes and 250m.

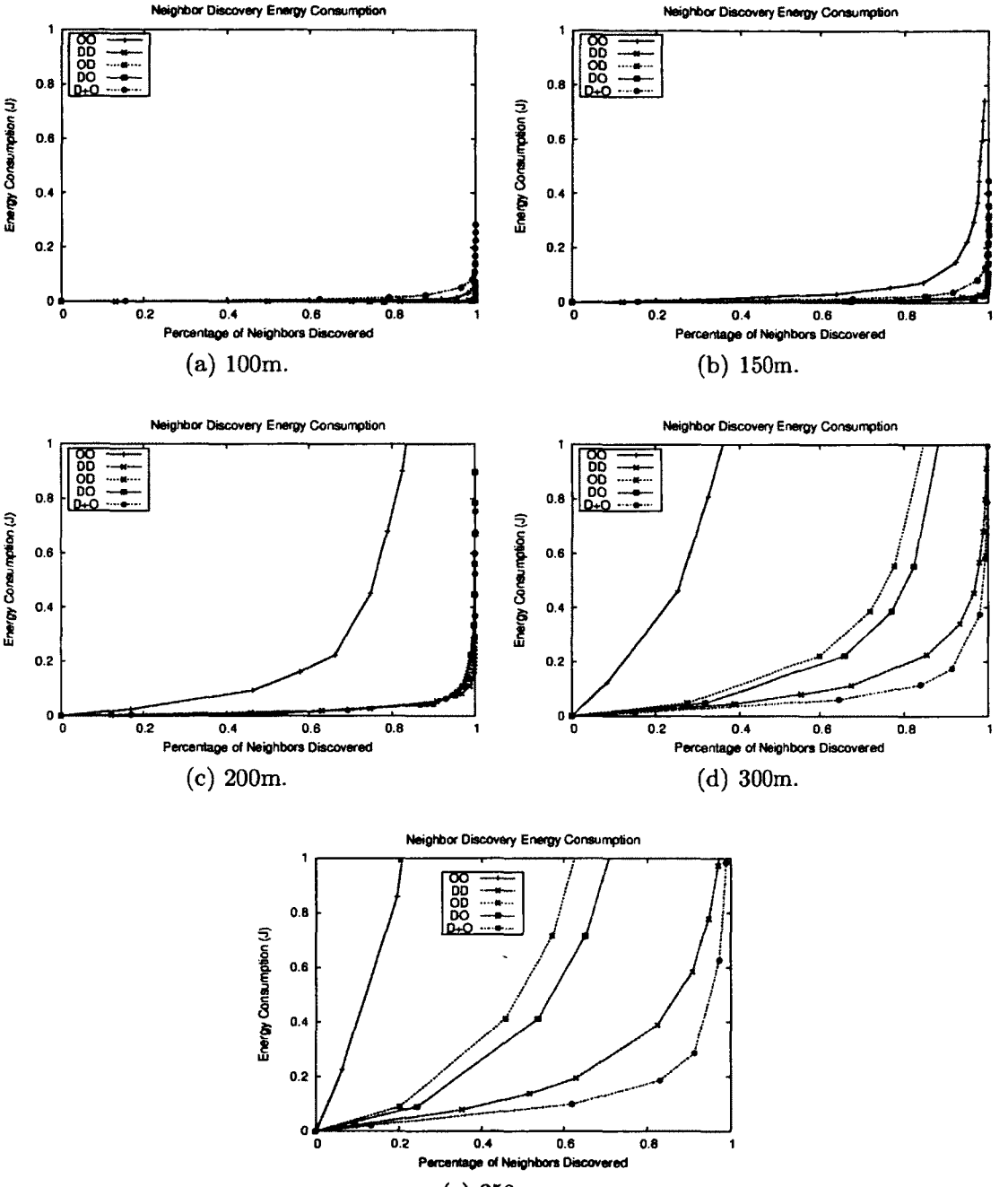


(b) 20°, 150 nodes and 350m.



(c) 20°, 150 nodes and 500m.

Figure 7.11: Delay comparison at 20 degree.



(a) 100m.

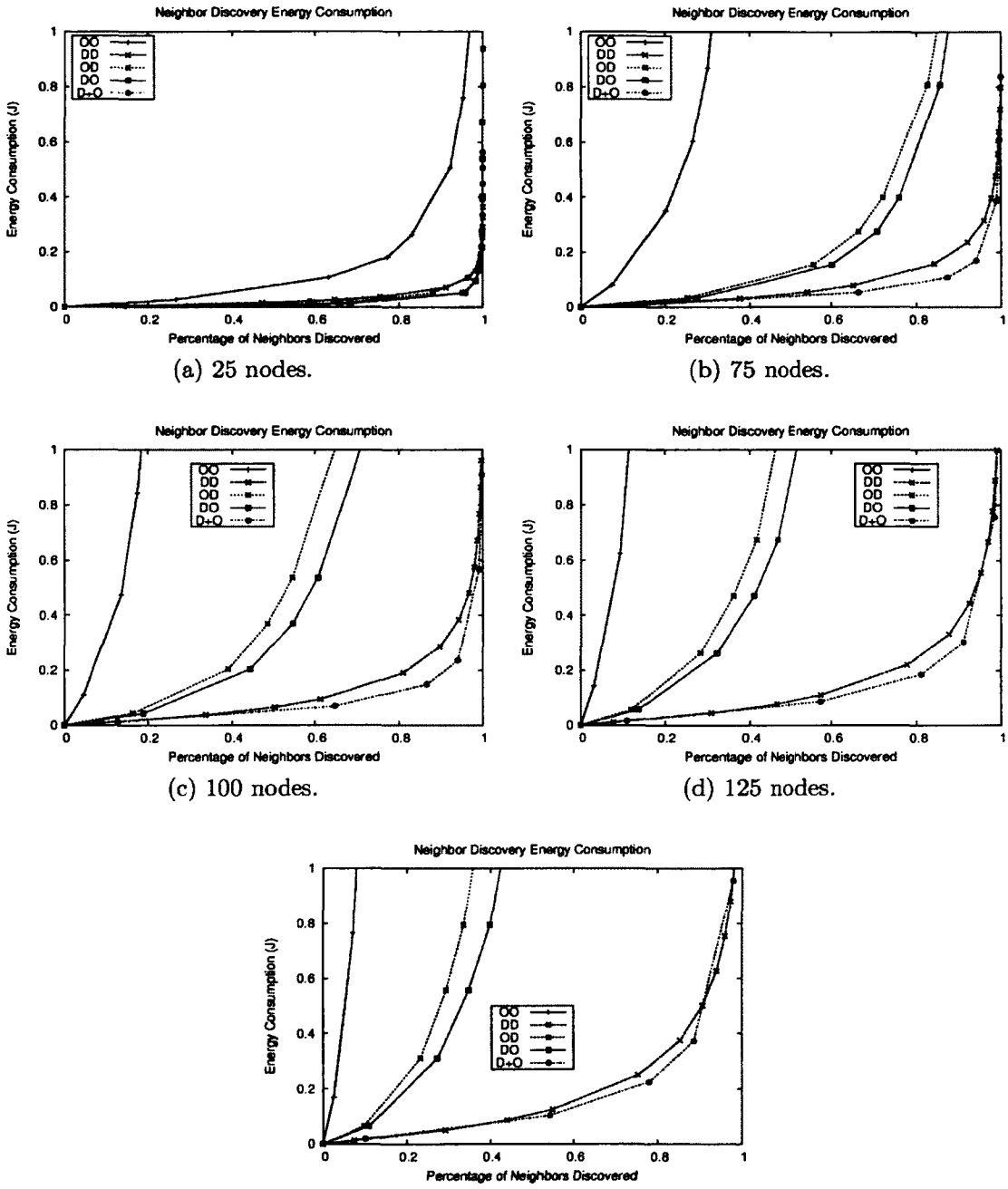
(b) 150m.

(c) 200m.

(d) 300m.

(e) 350m.

Figure 7.12: Energy comparison at various communication ranges, 50 nodes and 120 degree.



(a) 25 nodes.

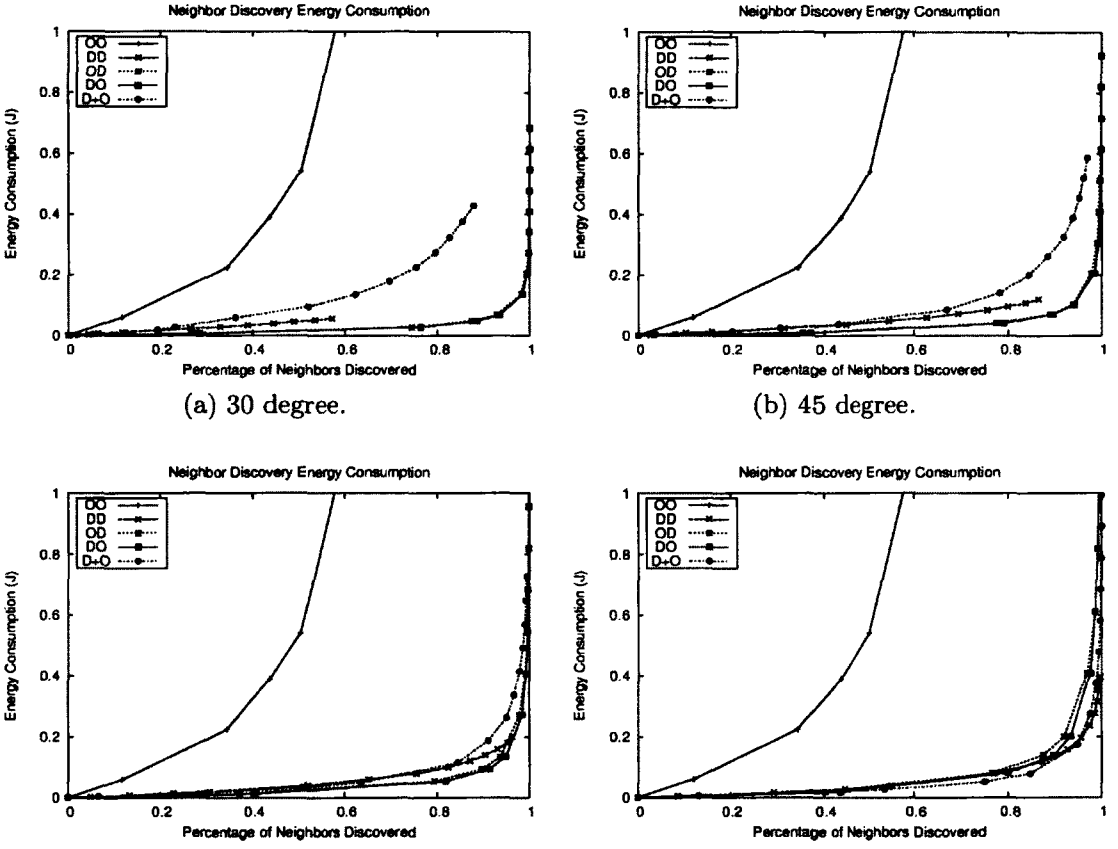
(b) 75 nodes.

(c) 100 nodes.

(d) 125 nodes.

(e) 150 nodes.

Figure 7.13: Energy comparison at various number of nodes, 250m communication range and 120 degree.



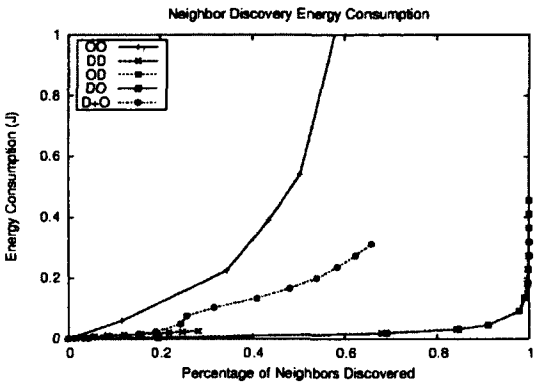
(a) 30 degree.

(b) 45 degree.

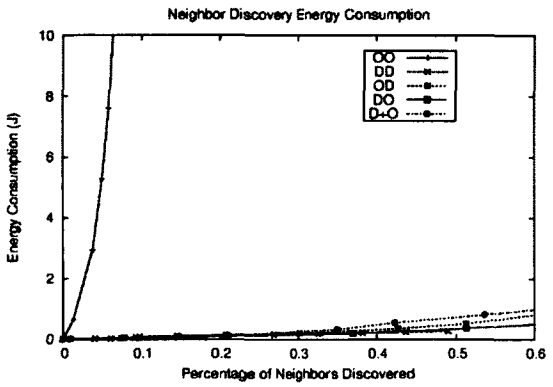
(c) 60 degree.

(d) 90 degree.

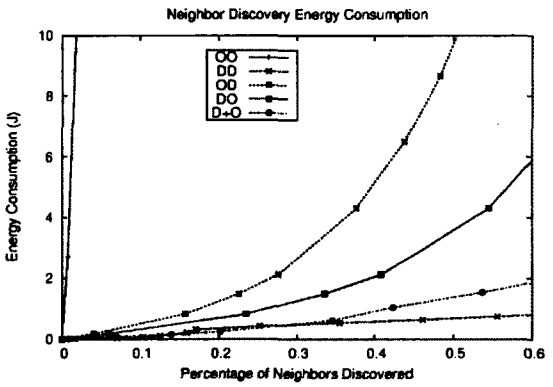
Figure 7.14: Energy comparison at various directional angles, 50 nodes and 250m communication range.



(a) 20°, 50 nodes and 250m.



(b) 20°, 150 nodes and 350m.



(c) 20°, 150 nodes and 500m.

Figure 7.15: Energy comparison at 20 degree.

Part V

Conclusion

Chapter 8

Conclusions and Future Work

We set out to improve network performance in a disruption (delay) tolerant network (DTN). Through our research, we have proposed a distributed storage protocol in DTN and identified a distributed key establishment solution to counter the network disruptions and delays. Since directional antennae can establish long communication links with less energy consumption when it is compared with omnidirectional antennae, we have also worked on the application of directional antennae in DTN. Because neighbor discovery is the first issue that needs to be addressed, we have subsequently proposed two neighbor discovery protocols in the presence of directional antennae.

The following sections summarize our work and point out possible future directions that interest us.

8.1 Contributions

8.1.1 Distributed Storage in Disruption Tolerant Network

In our work on distributed data storage, we first examined the existing works on the distributed storage on the Internet using P2P, as well as the distributed storage in MANET. We have pointed out the weakness of applying these solutions in DTN. Different from existing location based data storage protocols which use equal sized geometric shapes (e.g., rectangles, circles), we have proposed CHT through the use of Cell-based approach. In CHT, the network area is divided into cells and these cells can have different shapes. Nodes in a cell have high probability to move within their cells. Our protocol then stores data items in the cells which can have a hierarchical structure to reduce the storage of mapping related information. A data item first maps to the lowest level cell using CHT and then from a cell maps to a node in the cell using P2P. Our solution is indeed a combination of CHT and DHT, which limits

the use of DHT overlay to the lowest level cell. And nodes only needs to exchange their location information with other nodes inside their lowest level cell.

Through simulation, we have shown significant data item storage performance improvement when CHT is compared with the existing MHT protocol in the literature. Our protocol can quickly store data items in the mapping nodes with less maintenance overhead than MHT, in both single cell and multiple cell scenarios. Since data lookup is also part of the protocol, lookup success ratios have also been evaluated and CHT outperforms MHT in all scenarios when nodes have different maximum moving speeds.

8.1.2 Distributed Key Establishment in DTLBS-WSAN

We have proposed a Distributed Key Establishment (DKE) protocol in wireless sensor and actor network and focused on the disruption (delay) tolerant location based social network (DTLBSN) in which nodes cooperate with each other in the network communications during dynamic unrelated movements. In DKE, a node first generates its public and private key pairs and then uses neighboring signatures to establish its public key certificate. The node stores its certificate in a distributed manner using CHT and multiple copies of the certificate can be stored simultaneously. We have further improved the network security with pre-distributed symmetric and public/private keys at the actor nodes. We have categorized three models and presented our analysis on their security strength. In the *powerful model*, the actors are connected and cover the network area and sensors have the same transmission range as that of the actors. We have shown that guaranteed security can be achieved in this model. In *semi-powerful model*, actors have the same properties as those in powerful model but sensors have shorter transmission range than that of actors, a node can get security assurance once its certificate is signed by an actor node. In *none actor coverage model*, we have shown that high security confidence can be achieved. We have defined k cooperative malicious nodes and focused on the security strength against them in the analysis.

We have proposed the use of distance k safety margin approach to counter malicious certificate attacks. In the approach, a certificate is forwarded to the next hop

node and at the same time broadcasted to distance k neighbors of a node. Only next hop receiver needs to do the same thing. Through simulation, we have proved the effectiveness of this approach.

8.1.3 Neighbor Discovery in a Wireless Sensor Network with Directional Antennae

In situations when both transmission and reception use directional antennae, we have shown that there are instances that two nodes may not be able to discover each other, and as a result, nodes have to choose their rotation delays judiciously to enable communication. We have proposed deterministic and randomized neighbor discovery schemes using only directional antennae and analyzed the time complexity for nodes to discover each other. In deterministic Antennae Rotation Algorithm (ARA), nodes have knowledge of their vertex coloring. In ARA, if a node u sets its antenna rotation delay as $k^{\chi(u)}$, where k is the number of sectors and $\chi(u)$ is the color of the node u, we have proven that the time complexity is $O(k^{c-1})$ when there are c colors. If the delay of a node u is set as d_u, we have shown that the delays d_u can be chosen so that $gcd(d_u, k) = 1$ and $gcd(d_u, d_v) = 1$ when the node v is adjacent to node u. And it has been proved that the time complexity of $O(k(c \ln c)^3)$ can be achieved. We have also presented three randomized neighbor discovery algorithms and their analysis. The first algorithm uses random prime delay within range $k..n^{O(1)}$ and nodes can discover their neighbors in time $kn^{O(1)}$ with high probability. The second randomized algorithm is $RSRMA$, in which a node tosses a coin; it cycles k rounds with no sector delay if the coin is head and cycles one round with delay k per sector if the coin is tail. We have proved that it takes $O(k^2 \log n)$ expected time steps for neighboring nodes to discover each other, with high probability. The last proposed randomized antennae rotation algorithm is $RSRMA'$ which fits for the situation that the number of sectors of nodes u (k_u) and of node v (k_v) are not equal. And we have identified the time complexity of $RSRMA'$ as $O(k^4 \log n)$, where k is an upper bound on the number of sectors with $k_u \leq k$ and $k_v \leq k$.

We have evaluated the delay performance and energy consumption of the proposed algorithms. We have provided the results that the proposed solution is energy

efficient in neighbor discovery when it is compared with omnidirectional antennae. Simulation results have shown that $RSRMA$ performs the best among all the proposed algorithms.

8.1.4 Cooperative Neighbor Discovery using Two Antennae Patterns

We have shown that neighbor discovery delay performance and energy efficiency can be further improved when both directional and omnidirectional antennae patterns are available at the nodes. We have proposed the cooperative neighbor discovery approach in the presence of two antennae patterns. In a neighbor discovery cycle, a node performs neighbor discovery using short range omnidirectional antenna first and then uses long range directional antenna to discover neighbors that can not be reached otherwise. A node transmits its omnidirectional distance k neighbor information, as well as its directional neighbor information in its omnidirectional beacon packets while omnidirectional distance k neighbor information is contained in its directional beacon packets. We have proposed three cooperation mechanisms. In the random cooperation mechanism, a node points its directional antenna into a random sector initially and rotates its antenna independently according to the $RSRMA$ algorithm and cooperates with its omnidirectional neighbors in distance k neighborhood. In clustered minimum delay coverage, nodes form clusters first and nodes in a cluster try to rotate their directional antennae in different directions if there are $c < k$ nodes when there are k sectors. The goal is to minimize the differences of the sector distances between any two consecutive sectors occupied by these directional antennae. Nodes in the same cluster follow the same antennae rotation mechanism in $RSRMA$. This mechanism suits for the situation where nodes can communicate with its omnidirectional neighbors quickly and reliably. In selective directional scan cooperation mechanism, after a node collects its distance k neighbor information, it only uses its directional antenna to scan the areas that are not covered by omnidirectional distance k neighbors.

We have analyzed the delay and energy performances of existing neighbor discovery protocols. Theoretical results have shown that DD improves its performance

when nodes are distributed uniformly and independently and the number of neighbors of a node increases following a Poisson process in a contention based neighbor discovery model. DD eventually outperforms OO, OD and DO with the increase of neighboring nodes. We have also shown that the proposed solution can further improve the neighbor discovery performance.

Through simulation, we have presented the comparisons of different neighbor discovery schemes. The results have shown that the proposed $D+O$ scheme improves the delay performance with desirable energy efficiency when node communication range and the number of neighboring nodes increase. It ultimately becomes the best for a large number of neighbors at the nodes.

8.2 Future Work

8.2.1 Distributed Storage Protocol in DTN

There are still open problems in the distributed storage process in a network with disruptions and delays. In CHT, we have used statistical data in the cell partition. Further work on the dynamic cell formation process is necessary when we also need a distributed cell formation process as the basis for the distributed cell maintenance process. The cell formation process should work autonomously. Nodes will try to formulate their cells through network evolving process, by gradually accumulating the network statistical information, from local neighbors to faraway nodes.

8.2.2 Distributed Security Establishment in DTLBSN

We plan to further study the distributed security establishment in DTLBSN. Trust is an important aspect of network security because it is related with the network actions which a node will take. A trusted node can access certain information from others and the contents (e.g., executable code) which it provides can be regarded as reliable. In the Distributed Trust Establishment (DTE), we plan to first define rules and the trust value of nodes will be calculated independently based on these pre-defined rules. Nodes in a network may not be familiar with each other, but they cooperate with each following these rules.

8.2.3 Efficient Communication using Directional Antennae

We have proposed two neighbor discovery protocols at the presence of directional antennae. Further application of these protocols on network communications is an interesting future research direction. One important question is how can nodes reduce routing (communication) delays in DTN with the directional antennae. Since with the same transmission power, a directional antenna can reach faraway places compared with omnidirectional antenna, energy efficiency should also be considered, together with the delay performance.

Bibliography

[1] The Network Simulator, NS-2, `http://www.isi.edu/nsnam/ns/`, accessed February 1, 2012.

[2] J. Du, E. Kranakis, and A. Nayak. "Distributed Storage in Disruption Tolerant Network." In "Proceedings of 1st International Workshop on Wireless Sensor, Actuator and Robotic Networks (WiSARN), (WoWMoM Workshops 2010)," pages 1–6. Montreal, Canada (2010).

[3] J. Du, E. Kranakis, and A. Nayak. "Distributed Key Establishment in Disruption Tolerant Location Based Social Wireless Sensor and Actor Network." In "Proceedings of 8th Annual Conference on Communication Networks and Services Research CNSR 2011," pages 109–116. IEEE (2011).

[4] J. Du, E. Kranakis, O. Morales, and S. Rajsbaum. "Neighbor Discovery in a Sensor Network with Directional Antennae." *7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS)* pages 57–71 (2011).

[5] J. Du, E. Kranakis, and A. Nayak. "Cooperative Neighbor Discovery Protocol in a Wireless Network using Two Antenna Patterns." *9th Workshop on Wireless Ad hoc and Sensor Networks (WWASN2012), to appear* .

[6] J. Du, E. Kranakis, and A. Nayak. "A Geometric Routing Protocol in Disruption Tolerant Network." In "Proceedings of 6th Workshop on Wireless Ad hoc and Sensor Networks (WWASN2009), June 22, 2009, (ICDCS Workshops 2009, June 22-26)," pages 109–116. Montreal, Canada.

[7] J. Du, E. Kranakis, and A. Nayak. "A geometric routing protocol in disruption tolerant network." *International Journal of Parallel, Emergent and Distributed Systems* **25**(6), 489–508 (2010).

[8] J. Du, E. Kranakis, and A. Nayak. "A hop count based greedy face greedy routing protocol on localized geometric spanners." In "2009 Fifth International Conference on Mobile Ad-hoc and Sensor Networks," pages 231–236. IEEE (2009).

[9] K. Fall webpage, `http://kfall.net/ucbpage/`, accessed February 10, 2012.

[10] Defence Advanced Research Projects Agency (DARPA), accessed February 3, 2012, `http://www.darpa.mil`.

[11] The Consultative Committee for Space Data System (CCSDS), accessed February 3, 2012, `http://www.ccsds.org`.

[12] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald. "When TCP Breaks: Delay-and Disruption-Tolerant Networking." *IEEE INTERNET COMPUTING* pages 72–78 (2006).

[13] DTNRG Website, `http://www.dtnrg.org`, accessed February 1, 2012.

[14] Internet Research Task Force (IRTF), `http://irtf.org`, accessed February 3, 2012.

[15] RFC5050, `http://tools.ietf.org/html/rfc5050`, accessed February 1, 2012.

[16] RFC4838, `http://tools.ietf.org/html/rfc4838`, accessed March 6, 2011.

[17] RFC5325, `http://tools.ietf.org/html/rfc5325`, accessed February 7, 2012.

[18] RFC5326, `http://tools.ietf.org/html/rfc5326`, accessed February 7, 2012.

[19] F. Warthman. "Delay-Tolerant Networks (DTNs): A Tutorial." *Interplanetary Internet Special Interest Group, May 2003. Accessed February 1, 2012, at* ***http://www.ipnsig.org/reports/DTN_Tutorial11.pdf*** .

[20] Helsinki University of Technology DTN Comnet, accessed February 10, 2012, at `http://www.netlab.tkk.fi/~jo/dtn/`.

[21] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. "Delay-tolerant networking: an approach to interplanetary Internet." *Communications Magazine, IEEE* **41**(6), 128–136 (2003).

[22] R. Krishnan, P. Basu, J. Mikkelson, C. Small, R. Ramanathan, D. Brown, J. Burgess, A. Caro, M. Condell, N. Goffee, *et al.* "The spindle disruption-tolerant networking system." In "2007 Military Communications Conference (MILCOM 2007)," pages 1–7. IEEE (2008).

[23] SeNDT Project, `http://down.dsg.cs.tcd.ie/sendt/`, accessed February 1, 2012.

[24] A. Pentland, R. Fletcher, and A. Hasson. "DakNet: Rethinking Connectivity in Developing Nations." *IEEE Computer* **37**(1), 78–83 (2004).

[25] Z. Guo, G. Colombo, B. Wang, J. Cui, D. Maggiorini, and G. Rossi. "Adaptive routing in underwater delay/disruption tolerant sensor networks." In "Fifth Annual Conference on Wireless on Demand Network Systems and Services, 2008. WONS 2008.", pages 31–39. IEEE (2008).

[26] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet." *ACM SIGPLAN Notices* **37**(10), 96–107 (2002).

[27] ZebraNet Website, `http://www.princeton.edu/~mrm/zebranet.html`, accessed February 3, 2012.

[28] Z. Zhang. "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges." *Communications Surveys & Tutorials, IEEE* **8**(1), 24–37 (2006).

[29] S. Farrell and V. Cahill. *Delay and Disruption Tolerant Networking.* Artech House, Inc. Norwood, MA, USA (2006).

[30] IETF-65 DTNRG group meeting proceedings, Dallas, TX, USA, 2006, `http://www.ietf.org/proceedings/06mar/DTNRG.html`, accessed February 1, 2012.

[31] R. Handorean, C. Gill, and G. Roman. "Accommodating transient connectivity in ad hoc and mobile settings." *Pervasive Computing* pages 305–322 (2004).

[32] S. Merugu, M. Ammar, and E. Zegura. "Routing in Space and Time in Networks with Predictable Mobility." *College of Computing Technical Reports, Georgia Institute of Technology, 2004. Accessed February 1, 2012, at* ***http://smartech.gatech.edu/handle/1853/6492*** pages 1–13.

[33] C. Becker and G. Schiele. "New mechanisms for routing in ad hoc networks through world models." *Proceedings of the 4th CaberNet Plenary Workshop, Pisa, Italy* pages 1–4 (2001).

[34] A. Vahdat and D. Becker. "Epidemic routing for partially connected ad hoc networks." *Duke University, 2000. Accessed February 1, 2012, at* ***http://www.cs.duke.edu/~vahdat/ps/epidemic.pdf*** .

[35] T. Spyropoulos, K. Psounis, and C. Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." In "Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking," pages 252–259. ACM (2005).

[36] A. Lindgren, A. Doria, and O. Schelen. "Probabilistic routing in intermittently connected networks." *ACM SIGMOBILE Mobile Computing and Communications Review* **7**(3), 19–20 (2003).

[37] T. Spyropoulos, K. Psounis, and C. Raghavendra. "Single-copy routing in intermittently connected mobile networks." *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks. IEEE SECON 2004.* pages 235–244 (2004).

[38] E. Jones, L. Li, J. Schmidtke, and P. Ward. "Practical Routing in Delay-Tolerant Networks." *IEEE Transactions on Mobile Computing* pages 943–959 (2007).

[39] M. Demmer and K. Fall. "DTLSR: delay tolerant routing for developing regions." *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions* pages 1–6 (2007).

[40] C. Lin, W. Chang, L. Chen, and C. Chou. "Performance Study of Routing Schemes in Delay Tolerant Networks." In "22nd International Conference on Advanced Information Networking and Applications Workshops," pages 1702–1707. IEEE (2008).

[41] S. Jain, M. Demmer, R. Patra, and K. Fall. "Using redundancy to cope with failures in a delay tolerant network." *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications* pages 109–120 (2005).

[42] W. Zhao, M. Ammar, and E. Zegura. "A message ferrying approach for data delivery in sparse mobile ad hoc networks." *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing* pages 187–198 (2004).

[43] Y. Xian, C. Huang, and J. Cobb. "Look-ahead routing and message scheduling in delay-tolerant networks." *Computer Communications* pages 2184–2194 (2011).

[44] H. Dang and H. Wu. "Clustering and cluster-based routing protocol for delay-tolerant mobile networks." *IEEE Transactions on Wireless Communications* **9**(6), 1874–1881 (2010).

[45] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. "Chord: a scalable peer-to-peer lookup protocol for internet applications." *IEEE/ACM Transactions on Networking (TON)* **11**(1), 17–32 (2003).

[46] N. Standard. "Federal information processing standards publication 180-1." *US Department of Commerce, National Institute of Standards and Technology* **131** (1995).

[47] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. "Data-centric storage in sensornets with GHT, a geographic hash table." *Mobile Networks and Applications* **8**(4), 427–442 (2003).

[48] M. Li, W. Lee, and A. Sivasubramaniam. "Efficient peer-to-peer information sharing over mobile ad hoc networks." In "Second Workshop on Emerging Applications for Wireless and Mobile Access (MobEA II), in conjunction with the World Wide Web Conference (WWW)," pages 1–6. Citeseer (2004).

[49] H. Pucha, S. Das, and Y. Hu. "Ekta: An efficient DHT substrate for distributed applications in mobile ad hoc networks." *Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)* pages 163–173 (2004).

[50] K. Seada and A. Helmy. "Rendezvous regions: a scalable architecture for service location and data-centric storage in large-scale wireless networks." In "Proceedings of the 18th International Symposium on Parallel and Distributed Processing," pages 218–225. IEEE. ISBN 0769521320 (2004).

[51] R. Tanushetty, L. Ngoh, and P. Keng. "An efficient resiliency scheme for data centric storage in wireless sensor networks." In "2004 IEEE 60th Vehicular Technology Conference (VTC2004-Fall).", volume 4, pages 2936–2940. IEEE. ISBN 0780385217. ISSN 1090-3038 (2005).

[52] F. Araujo, L. Rodrigues, J. Kaiser, C. Liu, and C. Mitidieri. "CHR: a distributed hash table for wireless ad hoc networks." In "25th IEEE International Conference on Distributed Computing Systems Workshops," pages 407–413. Citeseer (2005).

[53] O. Landsiedel, S. Gotz, and K. Wehrle. "Towards scalable mobility in distributed hash tables." In "Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing," pages 203–209. IEEE Computer Society (2006).

[54] B. Karp and H. Kung. "GPSR: greedy perimeter stateless routing for wireless networks." *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* pages 243–254 (2000).

[55] D. Johnson, D. Maltz, J. Broch, *et al.* "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks." *Ad Hoc Networking* **5**, 139–172 (2001).

[56] A. Rowstron and P. Druschel. "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems." In "Middleware 2001," pages 329–350. Springer (2001).

[57] L. Eschenauer and V. Gligor. "A key-management scheme for distributed sensor networks." In "Proceedings of the 9th ACM Conference on Computer and Communications Security," pages 41–47. ACM (2002).

[58] H. Chan, A. Perrig, and D. Song. "Random key predistribution schemes for sensor networks." In "Proceedings of 24th IEEE Symposium on Security and Privacy," pages 197–213. ISBN 0769519407. ISSN 1081-6011 (2003).

[59] W. Diffie and M. Hellman. "New directions in cryptography." *IEEE Transactions on Information Theory* **22**(6), 644–654 (1976).

[60] ITU-T Recommendation X.509 | ISO/IEC 9594-8: "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", 2001.

[61] Y. Lee and S. Lee. "A New Efficient Key Management Protocol for Wireless Sensor and Actor Networks." *International Journal of Computer Science* **6**(2), 15–22 (2009).

[62] Z. Dai, Z. Li, B. Wang, and Q. Tang. "RTKPS: A Key Pre-distribution Scheme Based on Rooted-Tree in Wireless Sensor and Actor Network." *Advances in Neural Networks–ISNN 2009* pages 890–898 (2009).

[63] GNU Privacy Guard, `http://www.gnupg.org`, accessed February 2, 2012.

[64] P. Zimmermann. *The official PGP user's guide.* MIT Press (216 pp. May 1995).

[65] R. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* **21**(2), 120–126 (1978).

[66] T. ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms." In "Advances in Cryptology," pages 10–18. Springer (1985).

[67] M. Barbeau, J. Hall, and E. Kranakis. "Detection of rogue devices in bluetooth networks using radio frequency fingerprinting." In "Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks," pages 4–6 (2006).

[68] D. A. Knox and T. Kunz. "AGC-based RF Fingerprints in Wireless Sensor Networks for Authentication." In "Proceedings of 1st International Workshop on Wireless Sensor, Actuator and Robotic Networks (WiSARN), (WoWMoM Workshops 2010)," pages 1–6. Montreal, Canada (2010).

[69] G. Pei, M. Albuquerque, J. Kim, D. Nast, and P. Norris. "A neighbor discovery protocol for directional antenna networks." In "IEEE Military Communications Conference, 2005. MILCOM 2005.", pages 487–492. IEEE. ISBN 0780393937 (2006).

[70] S. Vasudevan, J. Kurose, and D. Towsley. "On neighbor discovery in wireless networks with directional antennas." In "INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE," volume 4, pages 2502–2512. IEEE. ISBN 0780389689. ISSN 0743-166X (2005).

[71] X. An and R. Hekmat. "Self-adaptive neighbor discovery in ad hoc networks with directional antennas." In "Mobile and Wireless Communications Summit, 2007. 16th IST," pages 1–5. IEEE. ISBN 1424416620 (2007).

[72] Z. Zhang and B. Li. "Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: algorithms and comparisons." *IEEE Transactions on Wireless Communications* **7**(5), 1540–1549. ISSN 1536-1276 (2008).

[73] J. Park, S. Cho, M. Sanadidi, and M. Gerla. "An analytical framework for neighbor discovery strategies in ad hoc networks with sectorized antennas." *Communications Letters, IEEE* **13**(11), 832–834. ISSN 1089-7798 (2009).

[74] Z. Haas and M. Pearlman. "ZRP: a hybrid framework for routing in ad hoc networks." In "Ad Hoc Networking (ed. Perkins CE)," pages 221–253. Addison-Wesley, Chapter 7. ISBN 0201309769 (2001).

[75] M. Corson, S. Papademetriou, P. Papadopoulos, V. Park, and A. Qayyum. "An internet MANET encapsulation protocol (IMEP) specification." Technical report, Internet-Draft, draft-ietf-manet-imep-spec-01.txt (1999).

[76] T. Clausen, C. Dearlove, and J. Dean. "ID: MANET Neighborhood Discovery Protocol (NHDP)." *Work In Progress, http://tools.ietf.org/id/draft-ietfmanet-nhdp* .

[77] W. Jakes. *Microwave Mobile Communications.* Wiley & Sons (1975).

[78] S. Alamouti. "A simple transmit diversity technique for wireless communications." *IEEE Journal on Selected Areas in Communications* **16**(8), 1451–1458 (1998).

[79] C. Wang, X. Hong, X. Ge, X. Cheng, G. Zhang, and J. Thompson. "Cooperative mimo channel models: a survey." *Communications Magazine, IEEE* **48**(2), 80–87 (2010).

[80] Y. Tu and G. Pottie. "Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through AWGN channels." In "IEEE Vehicular Technology Conference," volume 1, pages 130–134. ISSN 0740-0551 (2002).

[81] A. Scaglione and Y. Hong. "Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances." *IEEE Transactions on Signal Processing* **51**(8), 2082–2092. ISSN 1053-587X (2003).

[82] S. Song, D. Goeckel, and D. Towsley. "Collaboration improves the connectivity of wireless networks." In "Proceedings of 25th IEEE International Conference on Computer Communications. INFOCOM 2006.", pages 1–11. IEEE. ISBN 1424402212. ISSN 0743-166X (2007).

[83] L. Wang, B. Liu, D. Goeckel, D. Towsley, and C. Westphal. "Connectivity in cooperative wireless ad hoc networks." In "Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing," pages 121–130. ACM (2008).

[84] A. Ozgur, O. Lévêque, and D. Tse. "Hierarchical cooperation achieves linear capacity scaling in ad hoc networks." In "INFOCOM 2007. 26th IEEE International Conference on Computer Communications.", pages 382–390. IEEE. ISBN 1424410479. ISSN 0743-166X (2007).

[85] A. Abbasi and M. Younis. "A survey on clustering algorithms for wireless sensor networks." *Computer Communications* **30**(14-15), 2826–2841 (2007).

[86] A. Ephremides, J. Wieselthier, and D. Baker. "A design concept for reliable mobile radio networks with frequency hopping signaling." *Proceedings of the IEEE* **75**(1), 56–73 (1987).

[87] C. Lin and M. Gerla. "Adaptive clustering for mobile wireless networks." *IEEE Journal on Selected Areas in Communications* **15**(7), 1265–1275 (1997).

[88] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. "An application-specific protocol architecture for wireless microsensor networks." *IEEE Transactions on Wireless Communications* **1**(4), 660–670 (2002).

[89] M. Demirbas, A. Arora, and V. Mittal. "Floc: A fast local clustering service for wireless sensor networks." In "Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS/DSN 2004)," pages 1–6 (2004).

[90] A. McDonald and T. Znati. "A mobility-based framework for adaptive clustering in wireless ad hoc networks." *IEEE Journal on Selected Areas in Communications* **17**(8), 1466–1487 (1999).

[91] O. Younis and S. Fahmy. "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks." *IEEE Transactions on Mobile Computing* **3**(4), 366–379 (2004).

[92] S. Bandyopadhyay and E. Coyle. "An energy efficient hierarchical clustering algorithm for wireless sensor networks." In "22nd Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003.", volume 3, pages 1713–1723. IEEE (2003).

[93] S. Basagni. "Distributed clustering for ad hoc networks." In "Proceedings of Fourth International Symposium on Parallel Architectures, Algorithms, and Networks, 1999.(I-SPAN'99)," pages 310–315. IEEE (1999).

[94] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz. "Tapestry: A resilient global-scale overlay for service deployment." *IEEE Journal on Selected Areas in Communications* **22**(1), 41–53 (2004).

[95] A. Savvides, C. Han, and M. Strivastava. "Dynamic fine-grained localization in Ad-Hoc networks of sensors." *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* pages 166–179 (2001).

[96] L. Alima, S. El-Ansary, P. Brand, and S. Haridi. "DKS (N, k, f): A Family of Low Communication, Scalable and Fault-Tolerant Infrastructures for P2P Applications." In "Proceedings of the 3rd International Symposium on Cluster Computing and the Grid," pages 344–350. IEEE Computer Society (2003).

[97] GNU OCTAVE, `http://www.gnu.org/software/octave/`, accessed February 1, 2012.

[98] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. "A performance comparison of multi-hop wireless ad hoc network routing protocols." In "Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking," pages 85–97. ACM (1998).

[99] I. Akyildiz and I. Kasimoglu. "Wireless sensor and actor networks: research challenges." *Ad Hoc Networks* **2**(4), 351–367 (2004).

[100] J. Alfaro, M. Barbeau, and E. Kranakis. "Secure Localization of Nodes in Wireless Sensor Networks with Limited Number of Truth Tellers." In "Proceedings of the 2009 Seventh Annual Communication Networks and Services Research Conference-Volume 00," pages 86–93. IEEE Computer Society Washington, DC, USA (2009).

[101] W. Shi, M. Barbeau, and J. Corriveau. "Cross Verification-based Detection of the Evil Ring Attack in Wireless Sensor Networks." In "Proceedings of

1st International Workshop on Wireless Sensor, Actuator and Robotic Networks (WiSARN), (WoWMoM Workshops 2010), 2010," pages 1–6. Montreal, Canada.

[102] R. Hill and R. Dunbar. "Social network size in humans." *Human Nature* **14**(1), 53–72 (2003).

[103] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and F. Ferhatosmanoglu. "Crowd-Sourced Sensing and Collaboration Using Twitter." In "Proceedings of 11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)," pages 1–9. Montreal, Canada (2010).

[104] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili. "A pairwise key predistribution scheme for wireless sensor networks." *ACM Transactions on Information and System Security (TISSEC)* **8**(2), 228–258 (2005).

[105] M. Barbeau. "Assessment of the True Risks to the Protection of Confidential Information in the Wireless Home and Office Environment." In "Proceedings of 11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)," pages 1–6. Montreal, Canada (2010).

[106] M. Burrows, M. Abadi, and R. Needham. "A logic of authentication." *ACM Transactions on Computer Systems (TOCS)* **8**(1), 18–36 (1990).

[107] W. Shi, M. Barbeau, and J. Corriveau. "Detection of the Evil ring attack in wireless sensor networks using cross verification." In "2010 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)," pages 1–6. IEEE (2010).

[108] D. Carman, P. Kruus, and B. Matt. "Constraints and approaches for distributed sensor network security (final)." *DARPA Project report,(Cryptographic Technologies Group, Trusted Information System, NAI Labs)* (2000).

[109] P. Gupta and P. Kumar. "Critical power for asymptotic connectivity." In "Proceedings of the 37th IEEE Conference on Decision and Control," volume 1, pages 1106–1110. ISBN 0780343948 (1998).

[110] P. Hall. *Introduction to the Theory of Coverage Processes.* Wiley New York. ISBN 0471857025 (1988).

[111] Raman, B., Chebrolu, K.: The Enhanced Network Simulator (TENS), `http://www.cse.iitk.ac.in/users/braman/tens/`, accessed February 1, 2012.

[112] Culberson, J., `http://webdocs.cs.ualberta.ca/~joe/Coloring`, accessed February 1, 2012.

[113] H. YAGI. "Beam transmission of ultra short waves." *Proceedings of the IEEE* **85**(11), 1864–1874. ISSN 0018-9219 (1997).

[114] M. Barbeau and E. Kranakis. *Principles of Ad Hoc Networking.* John Wiley & Sons Ltd. (2007).

[115] M. Barbeau, E. Kranakis, D. Krizanc, and P. Morin. "Improving distance based geographic location techniques in sensor networks." *Ad-Hoc, Mobile, and Wireless Networks* pages 197–210 (2004).

[116] E. Kranakis, D. Krizanc, and E. Williams. "Directional versus omnidirectional antennas for energy consumption and k-connectivity of networks of sensors." *Principles of Distributed Systems* pages 357–368 (2005).