DESIGN AND ANALYSIS OF LOCATION CENTERED LARGE-SCALE OPPORTUNISTIC NETWORKS

by

SHANSHAN LU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2013

To my parents

For their endless love, support and encouragement

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my supervising professor Dr. Yonghe Liu, who provides me the opportunity to pursue my doctoral study and guides me throughout the study with great patience. Without his inspiration, encouragement and persistent help this dissertation would not have been possible.

I would like to extend my appreciation to my committee members Dr. Donggang Liu, Dr. Gergely Zaruba and Dr. David Kung for their interests, invaluable comments and suggestions in my research and for taking time to serve in my dissertation committee.

Last but not the least, I am extremely grateful to my parents and my husband for their understanding, support and sacrifice.

November 15, 2013

ABSTRACT

DESIGN AND ANALYSIS OF LOCATION CENTERED LARGE-SCALE OPPORTUNISTIC NETWORKS

SHANSHAN LU, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: Yonghe Liu

With the high penetration of mobile devices and their equipments of wireless interfaces in people's lives, the development of the scenario of Opportunistic Networks(OppNets) has become available. OppNets enable transmissions by exploiting temporary wireless links and connections opportunistically arising from mobility nature of nodes. The applications of OppNets have emerged in a wide range of areas.

OppNets have attracted a lot of attention from the wireless and mobile network research community, while little has been paid to the OppNets in large-scale scenario. The existing OppNets are not scaled well to large-scale area due to the high uncertainty embedded in transient contacts among people on the move. This dissertation introduces the large-scale opportunistic network that is centered and linked by populous locations in a physically large network field instead of transient contacts among mobile nodes. The relative stationary of mobile devices within these locations creates stabler and longer contact opportunities compared to those during movement and therefore provides much higher chances for successful message exchanges, potentially larger network capacity, and more flexibility for application design. A two-layer

architecture for the location centered large-scale opportunistic network is proposed in this dissertation. The base layer handles connection and enables device discovery and connection establishment within a location. The upper layer deals with routing which delivers messages to targeted locations and nodes.

For location targeted routing, two routing schemes, *LOcation based routing for OPportunistic networks* termed LOOP and *GEOcasting for OPPortunistic networks* termed Geoopp are proposed. Both adapt geographic greedy routing and fit it in the context of large-scale OppNets. The challenge lies in determining the future locations of nodes. LOOP and Geoopp differ in the determination. LOOP constructs a Bayes' predictive model to explore a node's mobility history and learn its mobility pattern, the regularity embedded in its movement. Instead of complicated data mining in LOOP, Geeopp offers a simple yet efficient way to learn the regularity in a novel approach. Geoopp introduces the concepts of inter-visiting time and contact availability per visiting to characterize a node's mobility. Chebyshev's inequality is employed to capture the regularity embedded in node mobility from visit and contact perspectives.

For node targeted routing, a *CAlendar based RouTing scheme* termed CART is proposed. While most existing routing schemes infer node mobility from context and social information, including LOOP and Geoopp which infer future locations from previous trajectory, CART studies a novel direction to obtain mobility without inference. CART explores the online public calendars for acquiring reliable node mobility and more importantly a global view of the evolving network topology. At the source node, CART can discover a path connecting source node and destination node, a node sequence from source to destination where pair of adjacent nodes share overlapped time and location. The message will be forwarded over the predetermined path.

For the connection layer, a device discovery approach and connection establishment scheme using ad-hoc Wi-Fi is proposed in this dissertation. It discovers devices within a larger range compared to commonly used Bluetooth and therefore enables the longer and stabler connections among mobile nodes inside a location. This approach is further enhanced with energy efficiency.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

Introduction

In recent years, personal mobile devices have become ubiquitous and an inseparable part of people's lives. The high penetration of the mobile devices and their equipments of wireless interfaces enable the availability and development of the scenario of Opportunistic Networks(OppNets). As a challenging scenario, OppNets represent the first attempt to close the gap between human and network behavior by utilizing a user-centric approach to networking and exploiting user nodes mobility as an opportunity, rather than a challenge, to improve data forwarding [1]. Basically, OppNets exploit humans mobility and their gregarious nature to enable a transmission only if two users carrying mobile devices are sufficiently close. In OppNets, two nodes wishing to communicate might never come into contact, it therefore routes messages between two nodes in store-carry-forward fashion where intermediate nodes delay the transmission of messages until new contact opportunities are available and messages are delivered to destination.

The applications of OppNets have emerged in a wide range of areas. OppNets bring connectivity in rural and developing regions where communication infrastructure is unavailable [2, 3]. In diaster recovery or emergency scenario where infrastructure has been damaged, OppNets can be adopted to allow mobile users to access diaster information. In well-connected urban area with availability of 3G coverage Wi-Fi access, OppNets are providing new types of applications and services. For example, delivering large amount of data through OppNets is cost-effective and meanwhile offloads the cellular infrastructure. OppNets support mobile social services, such as friend searching, management, content sharing, information seeking etc, as human relations can be sensed by the physical contacts among people. Mobile users sharing spatial locality in OppNet makes it perfect for applications with spatial significance, such as disseminating content within certain region. Some other applications include pervasive health care, intelligent transportation system, wildlife tracking etc.

OppNets have attracted a lot of attention from the wireless and mobile network research community. However, the existing works in literature commonly study OppNets deployed in relative small scenarios such as conferences, offices. This dissertation introduces location centered large-scale opportunistic network and proposes the network architecture. The motivations, challenges and contributions of this dissertation are discussed in this Chapter.

1.1 Motivations

The existing OppNets are not scaled well to large-scale area due to the high uncertainty embedded in transient contacts among people one the move. This dissertation introduces the large-scale opportunistic network that is centered and linked by populous locations in a physically large network field instead of transient contacts among mobile nodes. In real world, people participate activities in populous locations. They spend most of their times and interact in these locations such as offices, restaurant, stores etc. This social behavior results in the longer and stabler opportunistic contacts rather than short and transient ones normally considered among people while they are in motion. The opportunistic contacts enable data exchanges among users within the locations. People's movements from one location to another, for example from lunch places back to work, enable the exchanged data move among these locations. Therefore, the populous locations can serve as the backbone of the large-scale OppNets with large routing capacity as well as high bandwidth linking to other locations. One specific location can form an access network to the backbone and provide opportunities for delivering messages potentially over a long physical distance.

The core of large-scale OppNets is the populous locations where mobile devices are relatively stationary. As a result, it makes the contact periods much longer and contact opportunity more predictable, as compared to those during movement. These provide much higher chances for successful message exchanges, potentially larger network capacity, and more flexibility for application design.

This dissertation studies and analyzes a two-layer location centered large-scale opportunistic network architecture. The base layer handles connection which is underlying technique and enables device discovery and connection establishment within a location. The upper layer deals with routing which delivers packets/messages to targeted locations and nodes in large-scale area. The key of the study is the exploitation of the stabler and longer contacts within locations.

The routing layer mainly discusses location targeted routing which aims to deliver information to all nodes within a geographic area rather than an arbitrary group of nodes. Location targeted routing can be highly desirable for a wide range of OppNet applications, such as location-targeted advertising for marketing, geographic notification for emergency situations, and geographically restricted service discovery.

While information and its discovery is readily available through services like Google, distributing information to targeted receivers is still a challenging issue. The difficulty arises in identifying and locating these targeted receivers. For example, coupon distribution problem [4, 5] is a typical example of such. In the coupon distribution problem, a store owner may choose to target a specific apartment complex or subdivision in order to reach potential customers. Another example is that a critical message such as tornado warning may need to be forwarded to the target area, without knowledge of specific destined nodes, to enable certain preventative actions. In these applications, targeted receivers are located in a particular geographical area and the problem transforms into distributing information to targeted locations. Opportunistic networking can facilitate this type of information forwarding to complement the service of normal wireless networks such as cellular or Wi-Fi, either due to its unavailability, high cost, or privacy constraints. In OppNets, messages can reach the destination location through the mobility of people who are carrying the message and be disseminated locally among the targeted receivers therein through physical contact opportunities. However supporting location targeted routing for OppNets is still an open problem.

1.2 Challenges

Unlike the traditional Internet architecture which assumes the availability of contemporaneous, reasonably low propagation delay, low packet loss rate, end-to-end paths between any two nodes [6], for OppNets, everything in the architecture, protocol and paradigm needs to be reconsidered. Among these research issues, forwarding has attracted the most attention in the research community. In OppNets, no assumption is made on the existence of a complete path between any two nodes wishing to communicate. Two mobile nodes might never be connected at the same time. Nevertheless, opportunistic networking techniques allow such nodes to exchange messages between them by operating messages in store-carry-forward fashion. Usually, this comes at the price of additional delay in messages delivery, since messages are often buffered in the network waiting for a path towards the destination to be available. So far, the main focus of research on OppNets has been on routing and forwarding issues. Finding routes towards the desired destination in such disconnected environments is regarded as the most compelling issue due to the dynamic underlying topology, the scarcity of network resources and the lack of global information.

When dealing with routing and forwarding issue, some concerns such as privacy and cost are of high importance in designing routing and forwarding schemes. We take storage for example, even though memory is cheap nowadays, messages in OppNets are usually far larger than them in conventional networks, since whole files can be accommodated in a single message. For an average message size of 1MB, with 1000 messages generated in the whole network, Epidemic routing scheme should reserve nearly 1GB just for routing purpose.

The proposed large-scale OppNets impose new challenges in solving the routing and forwarding issues. The large-scale OppNets focus on the longer and stabler connections among mobile nodes within populous locations to enable message exchange. The exploitation of these longer and stabler contact opportunities for the purposes of facilitating message delivery is the key challenge. Beside, the high level heterogeneity of mobile nodes in a large-scale OppNets leads to more complex and diverse node mobility, while exploring node mobility plays an essential role in routing algorithm design as the communications among nodes in OppNets are enabled by their movements.

Another challenge arises from the wireless communication in the settings of node mobility, discovery of encounters among mobile nodes. The base of OppNets is the short-range wireless communication of mobile devices carried by their users in order to make users aware of each other. It helps introduce people to each other who are in close proximity. Therefore, device discovery through short-range wireless communication interface is an underlying technique to enable the message exchange in OppNets. Each mobile node in OppNets needs to provide the functionality of discovering mobile devices in neighborhood. This also introduces the concern of energy efficiency, as the wireless interfaces need to be turned on for a rather long time for capturing contact opportunities. The commonly used Bluetooth approach has relatively short communication range and its enabled connections can be frequently broken during people's movement. To enable the longer and stabler connections among mobile nodes inside a location, a device discovery approach with longer communication range is required in large-scale OppNets.

1.3 Contributions of this Dissertation

A two-layer location centered large-scale opportunistic network architecture is proposed in this dissertation. Specifically, a device discovery and connection establishment approach using ad-hoc Wi-Fi is introduced in base layer. Routing schemes are presented in upper layer for solving the location/node targeted routing issues in the large-scale networks. Part of the works and results in this dissertation have been published at conferences [7, 8, 9]. The contributions are highlighted in this section.

1.3.1 Location/Node Targeted Routing

This dissertation proposes three routing schemes for location targeted and node targeted routing in large-scale OppNets. For location targeted routing, we present two routing schemes, *LOcation based routing for OPportunistic networks* termed LOOP and *GEOcasting for OPPortunistic networks* termed Geoopp. Both adapt geographic greedy routing and fit it in the context of OppNets. Geographic routing is naturally the choice for routing messages toward destined locations. Traditional location based routing assumes that each node knows its 1-hop neighbor and the relay decision is made based on their relative positions. A node sends message to its 1-hop neighbor that makes most progress towards the destination. However the assumption is not satisfied in sparse OppNets. A message needs to be carried for a while before the node meets available neighbor, as a result, messages are forwarded in store-carry-forward style. In the adapted geographic greedy routing, it is assumed that each node knows its future positions. The relay decision is made based on its current and future positions. A message will be forwarded to a node, if it can carry the message and make the most progress on closeness toward destination. The challenge lies in determining the future locations of nodes. LOOP and Geoopp differ in the determination.

In reality, human movements often exhibit a high degree of repetition including regular visits to certain places and regular contacts during daily activities [10]. Song *et al.* measures uncertainty in human trajectories and their results reveal that human moves are highly predictable. It is feasible to predict a node's future locations based on its mobility history. This enables high fidelity prediction of a mobile node's future locations based on its mobility trace. Predicted future locations can then be used to determine routing strategies for message forwarding toward the destination location.

LOOP formulates the mobility pattern mining as a multi-label classification problem and constructs a Bayes' predictive model to explore a node's mobility history and learn its mobility pattern. This mobility pattern will then be used to predict the node's future mobility. Based on the prediction, the ability of the node to deliver a message to the destination is quantified through defined closeness improvement metric. The metric will be the determining factor for choosing proper relaying nodes in proposed strategy. Predicting future locations by extracting regular mobility pattern exhibits some essential advantages: LOOP preserves nodes' privacy and is easily scaled with network size (amount of mobile nodes). More importantly we can achieve total distributed operation as each node can choose its individual routing strategy and parameters without affecting others.

LOOP mines the previous trajectory to extract the node mobility pattern. Instead of the complicated data mining operation, Geeopp offers a simple yet efficient way to learn the regularity. Geoopp characterizes a node's mobility by inter-visiting time and contact availability per visiting to capture the regular visits and contacts in a specific region. Chebyshev's inequality is employed to compute the probabilities that a node visiting a region and having contact inside. Similar with LOOP, Geoopp only uses a node's own visiting and contact information, and does not require any extra routing information to be delivered across the network. Therefore, privacy is preserved and it is highly scalable as network size increases.

While most existing routing schemes infer node mobility from context and social information, including LOOP and Geoopp which infer future mobility from previous trajectory, a scheme that obtains mobility without inference is desirable. For node targeted routing, we propose a *CAlendar based RouTing scheme* termed CART which studies utilizing online public information to obtain node's mobility to aid the routing. CART explores the online public calendars for acquiring reliable node mobility and more importantly accessing a global view of the evolving network topology. At the source node, CART can discover a path connecting source node and destination node, a node sequence from source to destination where pair of adjacent nodes share overlapped time and location. The message will be forwarded over the predetermined path.

1.3.2 Device Discovery

A device discovery approach and connection establishment scheme using adhoc Wi-Fi is proposed in this dissertation. It provides much longer communication range compared to commonly used Bluetooth and therefore enables the stabler and longer connections among mobile devices within a location. This scheme employs beacon stuffing method for a device to announce its existence in its neighborhood by broadcasting beacons stuffed with useful information in the field of SSID to remote devices. This method can allow a device to discover multiple devices and their running applications without performing connection establishment attempt while limiting the communication for device discovery only on MAC layer. A score-based scanning schedule for device discovery operation is developed to enhance energy efficiency. Once the neighbor is discovered, a connection establishment scheme is presented to handle the connection between the discovered pair of devices.

1.4 Organization of Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 reviews related works in the fields of mobility models, OppNets routing schemes and device discovery approaches. The schemes LOOP, Geoopp for location targeted routing and CART for node targeted routing are presented in Chapter 3, 4 and 5 respectively. We then propose the device discovery and connection establishment scheme using Ad-Hoc Wi-Fi in Chapter 6. The conclusion are drawn in Chapter 7 with a discussion of future works that can be further explored.

CHAPTER 2

Related Works

This chapter introduces related works to the research problems to be addressed by this dissertation. The road map of this chapter is as follows. Section 2.1 presents a survey on mobility models for understanding human mobility. Section 2.2 investigates the existing techniques for solving node targeted routing issue in OppNets. Section 2.3 discusses the research works regarding the location targeted routing issue for OppNets. Section 2.4 reviews the approaches for device discovery in OppNets.

2.1 Understanding Human Mobility

In OppNets, message delivery is achieved through opportunistic contacts among nodes in the course of human mobility. The most challenging problem in OppNets is finding the route between two disconnected mobile nodes. Obviously, uncovering habits in human mobility can facilitate the design and analysis of routing schemes. In this section, we review the existing research works on real trace analysis and mobility models for the purpose of learning human mobility characteristics and patterns. Understanding human mobility not only helps in routing scheme design but also enables us to choose appropriate mobility models in simulation studies when the mobility characteristics of real world environment are known. We start by discussing the statistical properties and patterns regarding human mobility drawn from the real trace analysis. We then review the existing mobility models, learn the mobility characteristics they attempt to emulate and fit them into a taxonomy.

2.1.1 Learn Mobility From Real Trace Analysis

We first review the existing research works on real trace analysis. Statistical properties and patterns regarding human mobility drawn from the real trace data help understand the nature of mobility. [11] lists about 34 sets of traces publicly available, and there are several initiatives: CRAWDAD, UNC/FORTH and MobiLib that provide repositories for real data traces.

Chaintreau *et al.* [12] analyze the distribution of inter-contact time (the time gap separating two contacts between the same pair of devices) on extensive sets of public traces. They observe that the distribution of the inter-contact time may be well approximated by a power law over the range (10 minutes, 1 day). In further, [13] discovers that the distribution of inter-contact time follows a power law in most cases, and the distribution decays exponentially after half a day.

Rhee *et al.* [14] reveal that the mobility patterns in outdoor settings closely resemble those of Levy walks: Their flight distributions and pause time distributions closely follow (truncated) power-law distributions. A flight is defined as a straight line trip from one location to another that a node makes without a directional change or pause. The pause time refers to the duration that a node stays at the same location.

Several observations regarding the distribution of movement velocity and direction angle are reported in [15]: Speed and acceleration follow a Normal distribution; The direction angle variation follows a Lognormal distribution. Velocity and direction angle change have a temporal dependence unlike the random mobility models.

Song *et al.* [16] address another important questions: To what degree is human behavior predictable? This paper explores the limits of predictability in human dynamics by studying the mobility patterns of mobile phone users. By measuring the entropy of each individuals trajectory, they find a 93% potential predictability in user mobility across the whole user base.

Researchers have been intensely analyzing and modeling the trace of campus scenario, as most of the publicly available traces are collected in the campus environments. There are some general mobility patterns from the analysis of the campus trace data: users tend to visit popular locations; a user's movement is dependent on its community; a user's movement will vary over time [11].

2.1.2 Mobility Models Taxonomy

Mobility models where mobile users and their movement paths are usually created probabilistically aim to capture various characteristics and patterns of realistic mobility. We review the existing mobility models and learn the characteristics. We then classify them into four categories, from random, spatial, temporal and social dimension, based on the mobility characteristics they attempt to capture. The taxonomy of mobility models is shown in Fig. 2.1.

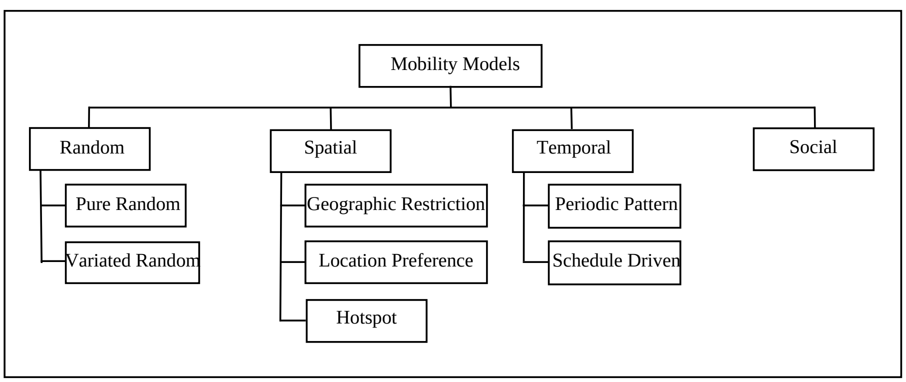

Figure 2.1. Mobility Models Taxonomy.

Random mobility models [17] [18] and their variations [19] are widely used where nodes are assumed to be deployed in a rectangular sized unobstructed simulation area. Nodes can move to anywhere in the given simulation playground according to the specification of the respective mobility model. The movement patterns described in random mobility models are not the reflection of people's movement in real lives. However, as they are simple, analytical tractable, they are still intensively employed in contemporary research today.

In the spatial class, the mobility patterns are mainly about geographic restriction, location preference and hotspot. For mobility models [20][21] with geographic restriction, the movement of a node is restricted within some areas defined for node movement. In particular, the movements of vehicles are bounded to the freeways or local streets in the urban area, and the pedestrians may be blocked by the buildings and other obstacles. Location preference refers to the human mobility characteristic of the re-appearance at the set of preferred locations, as in real lives, people tend to visit an array of locations more frequently than others, such as home, work site, school, grocery store etc, named preferred locations. Mobility models that emulate location preference are presented in [22][23][24][25]. It is observed that people are always attracted to populous places, so called hotspot, for example, the student center, gym in a campus, the shopping mall in a city, etc. Hotspot is defined as a part in the movement area, where the nodes spend a proportionally larger fraction of time. Some papers attempt to incorporate this type of movement characteristic into their mobility models [26][27].

In the temporal class, mobility models show periodic pattern and can be schedule driven. Periodic pattern represents the periodical return to the same locations, as regularity is embedded in human mobility in daily movement. For example, a typical working day in urban area for a mobile node includes a trip from home to work in the morning, the stay at working space during daytime and a trip towards home after work. The route is repetitive for the mobile node at working days. Periodic pattern is usually combined with location preference in mobility models. The following mobility models display the periodic pattern [22][24][25]. Schedule driven models [28][29] reproduce realistic temporal patterns of movements by explicitly including repeating daily activities in the human schedules.

Recently mobility models [30][31][32][33][34][35] exploring social relationships of humans begin to attract attentions in the field. It is believed that people's movements are strongly influenced by social relationships of humans. For example, we do not meet people pure randomly, instead we meet some people such as family members, friends and colleagues, more frequently and regularly. Exploring social relationships to facilitate the development of mobility models is the most active activity in the current research. The social relationships are often described in the form of social graph which help construct mobility models.

2.1.3 Populous Mobility Models

After introducing the mobility models taxonomy, we give the details of some of the popular algorithms in this area. Random Walk, also referred as Brownian Motion, was originally designed to simulate the unpredictable movement of particles in physics. In Random Walk model, a mobile node moves from its current location to a new location by randomly choosing a direction and speed in which to travel. It is thus suitable to simulations where the movement patterns of mobile nodes are completely unpredictable. Random Waypoint Model can be seen as an extension of Random Walk in the addition of pause time [36]. A mobile node travels towards the next destination in the simulation area at a selected speed. Upon arrival, the node takes a pause time before starting the movement again. Random Waypoint Model is the most populous mobility model employed in performance evaluation and a foundation for building complicated mobility models.

ORBIT Model presented in [22] is the first work to emulate location preference. It is based on the observation that the movement of a mobile user exhibits a partially repetitive "orbital" pattern involving a set of "hubs". The set of hubs are the preferred locations for the mobile user. Each node selects a subset of these spots and moves between them based on a predefined customizable behavior. For example, on a typical weekday, a node leaves home for school in the morning, visits the gymnasium in the evening, and returns home at night. Similarly, the student may stay in his home town for a few weeks and visit friends and family in other cities over some weekends, forming yet another higher level nation-wide orbit.

Small World In Motion (SWIM) Model proposed in [23] is based on a simple intuition on human mobility: People go more often to places not very far from their homes and where they can meet a lot of other people. It claims that it is unlikely (though not impossible) that we go to a place that is far from home, or that is not so popular, or interesting. Not only that, usually there are just a few places where a person spends a long period of time, such as home and work office or school. The model assigns to each agent a so called home, which is a randomly and uniformly chosen point on the plane. The agent then selects a destination for next moves depending on the weight of each site, which grows with the popularity of the place and decreases with the distance from the home. The home and frequently visited locations are the set of preferred locations for the mobile node.

A Time Variant Community (TVC) Model presented in [24] attempts to capture characteristics observed from empirical wireless WLAN traces: people tend to spend most of time at a handful of frequently visited locations and recurrence daily schedule. TVC divides a simulation cycle into two types of periods: Normal movement periods (NMP) of fixed lengths T_n and concentration movement periods (CMP) of fixed lengths T_c. The later is created to capture the periodical re-appearance behavior. A high probability for a node to visit its community during the CMP is assigned, so it re-appears at its community with high probability. These two types of time periods occur alternatively.

Note that ORBIT [22] and TVC [24] also display the property of periodic pattern. A mobile node in ORBIT follows a orbital pattern of visiting to a set of preferred locations. It is thus achieved that the node visits a preferred location periodically. For TVC Model, a mobile node reaches its preferred locations during concentration movement period of a simulation cycle while the cycle is repeated continually. Hence the regular return to the preferred locations is realized in the movement of the mobile node.

Sheep and Maverick model [26] features the formation of hot spots at random times and places that they refer to as filling, their disaggregation that they refer to as scattering, and dynamics between both. The Sheep models user clumping, filling process, in some predefined zones. The basic idea is that users tend to move towards places where there are already more people, creating a self-enhanced clumping. The sheep model has a major drawback, which prevents it to fit reality: as soon as it has reached an absorbing state, it does not evolve any more. A hotspot is suddenly disaggregated before reforming in other zones. Mavericks brought to model the scattering process coexist with sheep. Mavericks choose their target zone uniformly, including their current zone, independently from other mobiles. At each transition, an individual decides to behave as a maverick with probability α, and as a sheep with probability $1 - \alpha$.

The Working Day Movement Model (WDM) [29] consists of three different major activities that the nodes can be participating. They are being at home, working and some evening activities with friends. These activities are very common and capture a large portion of a working day for the majority of people. It is developed by combining various sub models together: home activity sub model, office sub model, evening sub model and transportation sub model. The Working Day Movement Model is more flexible than agenda driven model but still suffers from the complexity for analytical tractability issue.

Social relationships among people can be described as a social graph, where nodes represent individuals and weighted edges count for the strengths of the social connection between them. A community based mobility model (CMM) present in [30] is the most influential model in this class. The model allows collections of hosts, called community, to be grouped together in a way that is based on social relationships among the individuals. The key idea is that nodes in the same community tend to share the same interest. A node will travel towards a location if other nodes that it has strong social relationships with is moving towards the destination.

2.2 Routing in Opportunistic Networks

A large set of research results on routing exists for OppNets in the literature. We classify them into two categories, deterministic, and non-deterministic which is further divided into three subcategories, context-oblivious, context-aware and social relationship exploited. The classification is shown in Fig. 2.2. In deterministic routing, the complete path of a message is determined at the source node. The route is therefore determined once and does not change as the message traverses the network. However, it is on the basis of the complete knowledge of the topological evolution of the network. In non-deterministic routing the next-hop of a message is determined at each hop along its forwarding path. It allows the message carrier to utilize local information to make the relay decision.

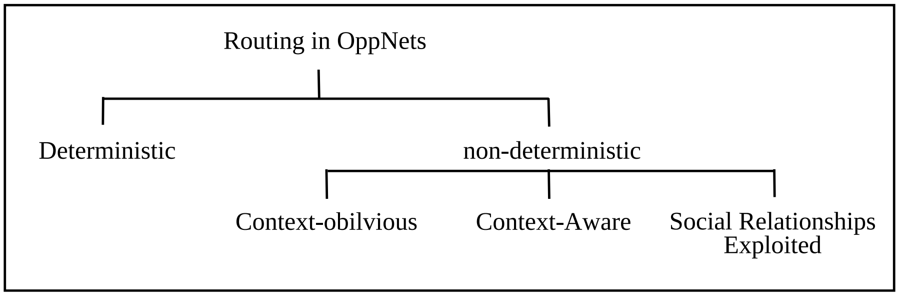

Figure 2.2. Routing in OppNets.

2.2.1 Deterministic Routing

If each node in the network has exact knowledge of the network topology, a complete path of a message can be determined at the source node and the message can be routed over predetermined paths deterministically. Deterministic routing achieves optimal performance at the minimum price. The challenge is the acquiring of the complete topological knowledge. The authors of [37] presents a Linear Programming formulation that uses the knowledge oracles to determine the optimal routing for minimizing average delay in the network.

2.2.2 Context-Oblivious Routing

Context-oblivious protocols do not exploit any contextual information regarding the status or behavior of the devices, users, and environment. In epidemic scheme [38], messages are forwarded to any encountered mobile nodes. The message carrier who is willing to forward a stored message to its destination attempts to capture all contact opportunities to spread the message and hope one of the copies will arrive at its destination as soon as possible. Each potential relay for forwarding a message is treated equally without identifying its degree of delivery ability. Therefore, the highest successful delivery ratio and lowest delay can be achieved at the extremely large cost of resource consumption in terms of traffic, buffer and energy and so on. However, in most cases, the mobile nodes in a network cannot afford the luxury cost. For example, mobile nodes only dedicate limited resource to help the message forwarding and the contact duration could be too short to exchange all messages. The delivery performance is hence lowered with the limited resources available.

In order to achieve a certain degree of delivery performance while consuming reasonable resources, a group of controlled flooding scheme emerges. An example is Spray and Wait [39] where a fixed number of copies of the message are sprayed into the network, hoping at least one of the copies will be forwarded to the destination through opportunistic contacts. However, the problem of Spray and Wait scheme is that it is difficult to determine the fixed number of copies sprayed into the network. Although it could be measured in a studied network, the determination of this parameter in a random opportunistic network remains a challenge.

A Spray and Focus scheme is proposed in [40] to improve Spray and Wait in certain scenarios. When the mobility of each node is restricted to a small local area, then it may not be possible to deliver one of the copies to the destination. The spray phase of Spray and Focus is the same as Spray and Wait. A fixed number of copies of the message are sprayed into the network. In the second phase (focus phase) rather than waiting for the destination to be encountered, each relay can forward its copy to a potentially more appropriate relay, using a carefully designed utility-based scheme.

2.2.3 Context-Aware Routing

Context-aware routing learns and exploits context around. Context information can cover various ranges, such as workplace, home address, profession, and email address, the mobility pattern of nodes and so on. All information that can help in routing messages is context information. Compared to the context-oblivious routing where all potential relays are treated equally, the context-aware routing attempts to identify and forward the message to relays that have high probability to deliver the message to its destination based on context information. The resources cost can be reduced as a result by not spreading the message in the whole network.

Prophet [41] uses past encounters to predict the probability of future encounters. Each mobile node maintains its contact history with all the other mobiles nodes it has encountered before. When two mobile nodes meet with each other, both of them calculate their contact frequencies with the message destination and the node that has the higher frequency will be selected to forward this message. It is assumed in the prophet that a mobile node which had frequent encounters with another node in the history will have high probability to meet this node again in the future. Similarly, Encounter-Based Routing presented in [42] uses an encounter-based metric for optimization of message passing that maximizes message delivery ratio while minimizing overhead both in terms of extra traffic injected into the network and control overhead, as well as minimizing latency as a second order metric. However, the large contact history makes it not suitable in a large-scale opportunistic network.

HiBop [43] a history based routing protocol for OppNets uses personal profile as context information. It is based on an assumption that a mobile node that has similar personal profile with the message destination tends to have high probability to meet with the destination node. For example, a user who lives in the same community with the message destination can help forward the message, as they will probably run into each other and hence have the contact opportunity for message delivery. However the exploitation of private information will incur the privacy issue.

MaxProp [44] prioritizes both the schedule of packets transmitted to other peers and the schedule of packets to be dropped. These priorities are based on the path likelihoods to peers according to historical data and also on several complementary mechanisms, including acknowledgments, a head-start for new packets, and lists of previous intermediaries.

Some research works exploit the mobility patterns of nodes in network when making relay decision. MobySpace [45] forms a high-dimensional Euclidean space based on nodes' mobility patterns. Each dimension represents the probability for a node to be found in a particular location. In particular, for each of the nodes in this network, there is a well defined probability of finding that node at each of N locations. This set of probabilities is a node's mobility pattern, and is described by a point, its MobyPoint, in an N dimensional Euclidean space, the MobySpace. A message is forwarded to nodes having mobility patterns that are successively closer to the mobility pattern of the destination. In simple words, MobySpace routing prefers to give custody of a message to a node that has mobility habits similar to those of the messages destination. The Euclidean distance is used to measure this similarity. Thus, the distance d_{ij} between the points i and j is $\sqrt{\sum_{k-1}^{n}(x_{ik} - x_{jk})^2}$.

In CAR [46], a relay decision is made based on node mobility and past colocation with message destination. It assumes that a highly mobile node is a good carrier as it meets many other nodes and the past colocation indicates that the node will meet the destination again in the future. The context information used in CAR are change degree of connectivity and the future host colocation. Change degree of connectivity is defined as the number of connections and disconnections that a node experienced over the last T seconds. A high value $U_{cdc_n}(t)$ means that node n recently changed a large number of its neighbors. For node n's colocation $U_{col_{h,i}}(t)$, a value of 1 means that n has been colocated with i at time t. These values are fed into Kalman filter predictors, which yield the predictions $\hat{U}_{cdc_h}$ and $\hat{U}_{col_{h,i}}$ of these utilities at time $t+T$. The final single utility is produced from multi-criteria decision theory and represents the how good a node n is for delivering messages to i.

Authors of [47] describe a technique for trajectory prediction and contact estimation for mobile nodes that uses a Time Homogeneous Semi Markov model. GeoDTN [48] models human mobility as a probability distribution around known anchor points, which makes the future movement of a node predictable. Then a neighbor score is calculated between a node and the destination node to measure the probability of them being at a common location. A node that has higher neighbor score with the destination is considered as a better relay. Kalantari *et al.* [49] introduce a model for capturing mobility inspired by thermodynamics, in particular, by the way heat is exchanged between objects.

All of the context-aware forwarding schemes utilize context information to forward message in OppNets. In order to collect context information, these schemes generally require learning neighbors' behaviors and incur privacy issues [50] and high overhead.

2.2.4 Social Relationships Exploited Routing

Recently certain schemes attempt to discover the social relationships among users in OppNets and utilize them for making relay decisions.

A labeling strategy discussed in [51] assigns each mobile node in the network with a label telling others about its affiliation or group, just like the name badge in the conference. Messages are forwarded to destinations or to next-hop nodes belonging to the same group, having the same label, as the destination. It is based on a social observation that people from the same affiliation tend to meet more often than people outside the affiliation and hence can be good forwarders to relay messages to the other members in the same affiliation with the same label.

SimBet [52] captures hidden social network structure by node centrality, betweenness and social similarity to the destination node. Messages are firstly relayed toward nodes with higher centrality to increase the possibility of meeting more nodes and finding the potential routes to the final destination. Similarly, BubbleRap [53] focuses on two specific aspects of society: community and centrality. The human society is divided into communities. Locally within a community or globally in the whole network, the people who have high centrality are more popular, and interact with more people than others. People can be ranked to measure its centrality. A message carrier first bubbles the message up the hierarchical ranking tree using the global ranking, until it reaches a node which is in the same community as the destination node. Then the local ranking system is used instead of the global ranking, and the message continues to bubble up through the local ranking tree until the destination is reached or the message expires. The concept of centrality is widely used in social aware forwarding schemes. PeopleRank [54] inspired by the PageRank algorithm used in Google's search engine to measure the relative importance of a Web page within a set of pages identifies the most popular nodes (in a social context) to forward the message to, given that popular nodes are more likely to meet other nodes in the networks.

FairRoute [55], a routing algorithm for delay tolerant networks is inspired by the social processes of perceived interaction strength, where messages are preferably forwarded to users that have a stronger social relation with the target of the message; and assortativity, that limits the exchange of messages to those users with similar "social status".

Social-aware routing needs to uncover the complicated social network structure, which requires either a prior knowledge or challenging data mining operation. These schemes also tend to create unbalanced load for different nodes [55].

2.3 Location Targeted Routing

Delivering information toward a geographical area, rather than a particular node, also named Geocasting is indeed widely studied in the wireless domain. A survey is presented in [56] for geocasting protocols which classifies them into three categories: flooding, directed flooding and no flooding. However, all these routing schemes implicitly assume that the underlying wireless network is fully connected and there exists a complete end-to-end path between any pair of nodes wishing to communicate with each other. This assumption has largely prevented their applications to OppNets. While numerous routing algorithms are proposed for OppNets, supporting geocasting service for OppNets is still an open problem [57].

In the literature, several works are presented for geocasting in delay tolerant networks. Burns *et al.* [58] present MV algorithm for forwarding bundles from a mobile source to a stationary destination in delay tolerant networks. It is based on observed meetings between peers and visits of peers to geographic locations. In order to receive the routing information, mobile nodes have to exchange their visiting and meeting history. It is a violation of privacy if it is implemented in OppNets where mobile devices are carried by human. Another similar routing algorithm is proposed in [59] to address the geocasting problem in delay tolerant networks where expected visiting rate to the destined region is introduced for the intermediate node to make geocast routing decisions. An intermediate node sends a copy to its encounter, only if the encountering node has a higher expected visiting rate to the destined region than itself. Piorkowski *et al.* [60] present a geocast service for mobile partitioned networks called GeoMobCast. It is based on the discovery of mobility maps designed to capture the collective mobility pattern in an area such as a city.

Geocasting for vehicular networks characterized by very high node mobility but constrained by roads has also been studied in the literature. Lee *et al.* [61] present a topology-assisted geographic-opportunistic routing (TO-GO) that incorporates topology assisted geographic routing with opportunistic forwarding for vehicular networks. MoVe scheme [62] leverages the knowledge of relative velocities of a vehicle and its neighboring nodes to predict the closest distance that they are predicted to get to the destination. In general, a vehicle is considered to be a better relay if it is moving toward the destination. Leontiadis *et al.* [63] propose GeOpps, a geographical delay tolerant routing algorithm that exploits information from the vehicles' navigation system to route messages to a specific location. The future positions of vehicles can be derived from the navigation system embedded in vehicles. GeoDTN+Nav [64] is a hybrid geographic routing solution enhancing the standard greedy and recovery modes exploiting the vehicular mobility and onboard vehicular navigation systems to efficiently deliver packets for vehicular networks. However, The different mobility features of vehicles from human make it difficult to apply these schemes to OppNets formed by mobile devices carried by people.

2.4 Device Discovery in Opportunistic Networks

Mobile devices equipped with short communication interfaces are becoming increasingly popular, which provide the foundation for wide deployment of OppNets. Bluetooth and Wi-Fi are two main short communication techniques underlying OppNets.

Bluetooth is a universal short-range radio link operating in the unlicensed 2.45 GHz ISM band. The physical channel is divided into timeslots of 625 microseconds. Bluetooth communicates strictly in a point-to-point fashion. Devices organize themselves in piconets of one master and multiple slaves. The master coordinates communication in the piconet and routes messages between slaves at the physical link layer. The slaves only communicate with each other on the logical link layer. A scatternet is a group of piconets that are connected by Bluetooth devices that are a member of multiple piconets [65].

Several papers have studied Bluetooth device discovery [65, 66, 67]. A Simple Neighbor Discovery (SND) procedure is proposed in [68]. The scheme is shown to be configurable in the trade-off between discovery time and overhead. The authors found three weaknesses of Bluetooth in current neighbor discovery procedure. First, inquiry takes a lot of time and therefore it requires too much overhead if used regularly. Second, it is very inefficient to transmit data simultaneously with the inquiry. Third, the inquiry assumes asymmetric roles, which is not well suited to an ad hoc network of peer nodes. The Simple Neighbour Discovery (SND) procedure is based on beacon packets, which are sent by a node regularly with the appropriate information to establish a connection. Other nodes can scan for these beacon packets to discover the neighbor. SND assumes symmetric roles, it is faster than the inquiry and it enables more bandwidth for data transmission during the discovery period.

Two adaptive algorithms are proposed in [69] in order to save energy when the device is unlikely to encounter a neighbor. The authors adaptively choose parameter settings depending on a mobility context to decrease the expected power consumption of Bluetooth-enabled devices. Both adaptive schemes for selecting the discovery mode are based on locally available information.

However, Bluetooth is not suitable in the location centered large-scale OppNet to support stable and long connectivity. 802.11 Wi-Fi has a much longer communication range and faster device discovery speed than Bluetooth. A novel Inter- BSS DLS (iDLS) protocol is proposed in [70, 70] to facilitate a more general approach that allows a device to discover neighbors in a different BSS and set up direct communication between any two stations regardless of the BSS membership and restrictions. However, this approach relies on APs to relay message for device discovery.

CHAPTER 3

LOOP: A Location Based Routing Scheme

This chapter introduces a location based routing scheme for large-scale OppNets termed *LOOP* that aims at forwarding messages to a *destination location/area*[1], instead of forwarding to specific nodes.

LOOP exploits the regularity embedded in human movements. We formulate the mobility pattern mining as a multi-label classification problem and construct a Bayes predictive model to explore a nodes mobility history and learn its own mobility pattern. We can then predict a mobile nodes future locations based on its mobility pattern with high confidence. Closeness improvement metric is defined to measure the improvement in terms of distance a node can carry a message to its destination on the basis of mobility pattern. The metric will be the determining factor for choosing proper relaying nodes in the proposed strategy.

We first give an overview of the scheme in Section 3.1. We then present the scheme in Section 3.2 and enhance it by exploring user calendars in Section 3.3. We perform analytical study in Section 3.4 and extensive simulation study in Section 3.5. The results show that LOOP can achieve significant performance gains over well known existing routing algorithms in opportunistic networks.

3.1 Overview

As a special type of delay tolerant networks, OppNet largely relies on human relays. Message routing is achieved over store-carry-forward paradigm by exploiting

[1]For simplification, we will use only location to denote the destination area from now on.

temporary wireless links and connections opportunistically arising from mobility nature of nodes. Our routing scheme, termed *LOOP* for *LOcation based routing for large-scale OPportunistic networks*, exploits the regularity embedded in human moving pattern. Human movements often exhibit a high degree of repetition including regular visits to certain places and regular contacts during daily activities [10]. This enables high fidelity prediction of a mobile node's future locations based on its mobility trace. Predicted future movement can then be used to determine routing strategies for message forwarding toward the destination location.

Correspondingly, our design focuses on two challenges therein: i) How to effectively mine the movement history to reliably predict a mobile device's future locations; and ii) how to select nodes for message relay toward destination location given the predicted locations. We formulate the mobility pattern mining as a multi-label classification problem and construct a Bayes' predictive model to explore a node's mobility history and learn its mobility pattern. This mobility pattern will then be used to predict the node's future mobility. Based on the prediction, the ability of the node to deliver a message to the destination is quantified through defined closeness improvement metric. The metric will be the determining factor for choosing proper relaying nodes in proposed strategy. Note that LOOP is a multi-copy forwarding scheme where a message can be sprayed to multiple relays.

We further enhance LOOP by exploiting readily available user calendars. Accurate mobility information regarding a person can be discovered from one's associated calendars. To the best of our knowledge, calendars are rarely considered for message routing in OppNets. We present *LOOP assisted with user Calendar* termed *LOOPC* to incorporate users' calendar information in addition to discovered mobility patterns when we evaluate a node's delivery probability toward destination. In the literature, calendars have rarely been considered for message routing in opportunistic networks. In calendar based routing scheme, we explore the online public calendar to gain people's mobility knowledge and therefore facilitate the message forwarding in opportunistic networks. LOOPC exploits private calendars kept by users on their mobile devices to facilitate the routing process. Note that using private calendars is not invasion of privacy of users as the calendar content is restricted within mobile device itself.

Our scheme has several advantages when applying into the scenario of OppNets. First, LOOP preserves privacy and is highly scalable as network size by solely relying on node's own movement information for routing and avoid routing data exchange among nodes. Second, our scheme achieves total distributed control. Each node calculates its own delivery probability and decides if it is a proper relay toward the destination. Each node, based on its own state, can choose its individual forwarding strategy without involving network wide changes.

3.2 The Scheme of LOOP

In this section, we detail LOOP, a location based routing scheme for large-scale opportunistic networks. Firstly, we present a Bayes's predictive model to learn a node's mobility patterns and predict its future movement based on its mobility trace. Secondly, a mobile node's ability to deliver a message is evaluated by defined closeness improvement metric. Finally, we present strategy to make relay selections.

3.2.1 Movement Prediction

In this section, we describe Bayes' predictive model for movement prediction in LOOP. We assume that each message forwarded in the network contains information headers including $<$ Message ID $>$, $<$ destind location $>$ and $< TTL >$. The message ID can be generated with creation time and message title. The goal here is to find proper relays to route the message from source location to destination location. In order to achieve this, predicting a node's future locations is necessary as this will determine if a node is a proper relay for that message.

We assume that human movement exhibits high degree of repetition, introducing regular mobility pattern. A predictive model is constructed to recognize these mobility patterns. We define regular mobility pattern as a set of tuples in terms of <day type, time slot, locations>. Here a day of different day types, such as weekday and weekend day, is divided into a number of time slots. We refer one tuple as a mobility pattern, denoting that the node regularly visits certain locations at a specific day type and time slot. We can then predict that the node will visit these locations at such day type and time slot in the future. In this chapter, we will use regular mobility pattern and predicted movement interchangeably.

In order to discover the mobility pattern, mobility trace of the node shall be collected. Given the mobility trace, we then formulate the mobility pattern extraction as a multi-label classification problem and propose BR-NB (Binary Relevance Naive Bayes) and CC-NB methods (Classifier Chain Naive Bayes) to learn and generate the mobility pattern.

3.2.1.1 Collecting Mobility Trace

In this chapter, mobility trace is represented as a series of mobility records. $R = \{r_1, r_2, r_3, ..., r_i, ...\}$ is the collection of mobility records. Each tuple $r_i \in R$ which refers to one mobility record in the trace is in the format $r_i =< d_i, t_i, l_i >$, which indicates that the node visited location l_i at time slot t_i on day d_i. We remark that location information can be easily obtained in smart mobile devices, either by Global Positioning System (GPS) equipped in the device or through web service such as Geolocation API provided by Google. Acquired location can be recorded in address or as a pair of longitude and latitude coordinates.

Mobility trace collection is triggered when two conditions are satisfied simultaneously. 1. Current date, time slot or location differs from its counterpart in latest mobility record; 2. Opportunistic contact is currently available. The second condition is necessary as it takes availability of contact opportunity (next relay), into consideration. If both are satisfied, we generate a mobility record using current date, time slot and location and insert it into the mobility trace. A mobile node might return back to the same location multiple times on the same date and within the same time slot and this leads to repeated records. Only the last is kept in trace as the visiting times under the same set of date and time slot is not of our concern.

3.2.1.2 Extracting Regular Mobility Patterns

Extracting node's mobility pattern from mobility trace is modeled as a multi-label classification problem which associates an instance with a subset of a finite set of labels. Let $X = \{F_1, ...F_m\}$ denote instance with a vector of features. The set of labels is $\mathcal{L} = \{l_1, l_2, ..., l_n\}$. The goal of the multi-label classification problem is to learn a classifier $h(x)$ from the training set which predicts a subset of labels $SL \subseteq L$ for each unseen instance x.

The multi-label classifier is specified for extracting regular mobility pattern from mobility trace as following. The features are day types and time slots $F_1 = D = \{d_1, d_2, ...\}$ and $F_2 = T = \{t_1, t_2, ...\}$. A label $l \in \mathcal{L}$ is a location present in the mobility trace, and the label set $\mathcal{L}$ contains all the unique locations present in the mobility trace. The task is to learn a classifier h to associate a subset of labels $SL \subseteq L$ for a given instance $x = \{d, t\}$. In other words, the classifier h predicts the locations the node can reach for a given tuple of day type and time slot. An association of the instance and the label subset (x, SL) serves as a mobility pattern.

The label subset SL can also be represented as $\hat{y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_n]$ where $\hat{y}_i = 1$ if ith label in label set L is associated to x; otherwise $\hat{y}_i = 0$. So the task transforms to learn a classifier $h(x)$ to generate $\hat{y}$ for a given instance x. The new task is illustrated in eq. (3.1).

$$
\begin{aligned}
h(x) &: x \rightarrow \hat{y} \\
\hat{y}_i &= \begin{cases} 1 & l_i \in SL \\ 0 & l_i \notin SL \end{cases}
\end{aligned}
\tag{3.1}
$$

In this chapter, we propose BR-NB method which applies the naive Bayes classifier to construct the binary classifier and employs binary relevance method to solve the multi-label classification problem. One of the most common ways to solve the multi-label classification problem is binary relevance method (BR) which learns a binary classifier $h_i(x)$ for each label and then outputs $\hat{y}_i$. The binary relevance method is shown in eq. (3.2). The set of binary classifiers $\{h_1(x), h_2(x), ..., h_n(x)\}$ generates the final classification result $\hat{y}$.

$$
\begin{aligned}
\hat{y} = [\hat{y_1}, \hat{y_2}, ..., \hat{y_n}] = [h_1(x), h_2(x), ..., h_n(x)] \\
\hat{y}_i \in \{0, 1\}
\end{aligned}
\tag{3.2}
$$

The naive bayes classifier can be applied to construct the binary classifier. The constructed binary classifier is shown in eq. (3.3). In the training data set, $P(y_i = 1)$ is calculated as the probability of instances that are labeled with l_i. $P(d/t|y_i = 1)$ is the probability of instances that have day type d or time slot t and under the condition that they are labeled with l_i. On the other hand, $y_i = 0$ indicates that the instance is not labeled with l_i.

$$\begin{aligned} \hat{y_i} &= h_i(x) \\ &= argmax_{y_i \in 0,1} P(y_i)P(d|y_i)P(t|y_i) \end{aligned} \tag{3.3}$$

When the classification result $\hat{y}$ for a given instance $x = \{d, t\}$ is acquired through BR-NB, we then transform $\hat{y}$ back to label subset SL. For all $\hat{y_i} \in \hat{y}$, if $\hat{y_i} == 1$, l_i is put into SL. Until now, we receive an association of instance and label subset (x, SL). The association serves as a mobility pattern $p = \langle d, t, SL \rangle$ as discussed earlier. We predict that $l_i \in SL$ is one location the node will visit during day type d and time slot t in the future.

The binary relevance method is simple and fast but does not explicitly model label dependencies. In fact, the label correlations between labels exist in our mobility pattern extraction problem, since the occurrence of certain locations might rely on the occurrence of others. The performance of the classification can be improved if label dependency is considered. The authors of [71] present a classifier chain method (CC) to model the label dependency. The label set $\hat{y}$ for instance x is produced by the set of binary classifiers $[h_1(x), h_2(x, \hat{y_1}), ..., h_L(x, \hat{y_1}, \hat{y_2}, ...\hat{y_L})]$ for all labels as shown in eq. (3.4).

$$\begin{aligned} \hat{y} = [\hat{y_1}, \hat{y_2}, ..., \hat{y_L}] = [h_1(x), h_2(x, \hat{y_1}), ..., h_L(x, \hat{y_1}, \hat{y_2}, ...\hat{y_L})] \\ y_i \in \{0, 1\} \end{aligned} \tag{3.4}$$

In this chpater, we combine the classifier chain method with the naive Bayes classifier and present CC-NB method to advance the BR-NB discussed above. The binary classifier is constructed according to eq. (3.5).

$$\begin{aligned}\hat{y_i} &= h_i(x) \\ &= argmax_{y_i \in 0,1} P(y_i)P(d|y_i)P(t|y_i)\prod_{j=1}^{i-1} P(y_j|y_i)\end{aligned} \tag{3.5}$$

The mobility records in collection R are used to prepare for the training data. The training data set preparation process is shown in *Algorithm* 1. Afterwards, the training set is ready to be used to extract the mobility pattern with our BR-NB or CC-NB methods.

Algorithm 1 Training Data Preparation

Input:

Mobility Trace, R;
Day type Collection, D;
Time Slot Collection, T;

Output:

Training Data Set, S;

1: Extract unique locations from R: $L = \{l_1, l_2, ...\}$;
2: Divide mobility records into groups based on dates: $R = \{G_1, G_2, G_3, ..., G_j, ...\}$ where records in the same group have the same date;
3: Generate n samples in each group according to number of time slot T_n: $G_j = \{s_1, ..., s_m, ..., s_n\}$, where $s_m \in G_j$ is in the format $s_m =< d_m, t_m, L_m >$. d_m: date, t_m: time slot m, L_m: all locations visited during d_m and t_m;
4: Convert d_m into one of the day types in D for all generated samples;
5: Collect all samples into training data set: S;
6: **return** S

With the training set, we train the BR-NB or CC-NB classifier proposed in this chapter. *Algorithm* 2 shows the process of the mobility pattern extraction. For any combination of day types and time slots, we generate an instance $x = (d, t)$ where $d \in D$ and $t \in T$, its associated label subset $\hat{y}$ is predicted by the trained classifier. These prediction results produce mobility patterns in further. Pattern table $PT = \{p_1, p_2, p_3, ..., p_i, ...\}$ stores all the patterns.

Algorithm 2 Extracting Regular Mobility Pattern

Input:

Training Data Set, S;
Location Collection, L;
Day Type Collection, D;
Time Slot Collection, T;

Output:

Updated pattern table PT

```
for Each d ∈ D do
    for Each t ∈ T do
        Generate an instance: x = (d, t);
        Predict label subset SL for the instance x based on BR-NB or CC-NB;
        Produce a mobility pattern p = ⟨d, t, SL⟩;
    end for
end for
Update result table PT;
return PT
```

The time complexity of preparing for the training data set and extracting mobility pattern are roughly $O(R_n)$ and $O(D_n \times T_n \times L_n \times S_n)$ respectively. R_n is the amount of mobility records in mobility trace. D_n, T_n, L_n, S_n represent the number of day types, time slots in their collections and unique locations and samples in training data set. D_n and T_n are defined by the LOOP scheme design, while L_n and S_n are determined by the training data set.

The size of training data set plays an essential role in determining the computation cost of learning mobility patterns. We desire to minimize the size but maintain classification performance. Mobility cycle is used to denote the shortest time period that is required to collect mobility trace for generating training data. For a mobility pattern table with D_n day types and T_n time slots, it is necessary to have trace for all combinations of day types and time slots (d_i, t_j) where $i, j = 0, ..., n$. For instance, with one day type defined, the mobility cycle is a day; with two day types defined, such as weekday and weekend, the mobility cycle is at least 2 days or a week depending on the time when the trace is collected. At the beginning of LOOP when the mobility pattern table is empty, a mobility cycle is needed for collecting trace and then generating training data. Once the mobility pattern table is filled up, it can then be used in guidance of the message routing.

The mobility trace collected within one mobility cycle is enough for capturing the regularity embedded in people's mobility. It is not necessary to accumulate a large training data set because of the repeating behavior of people's movement. This conclusion is supported by our simulation results. Notice that during the bootstrapping phase when the mobility pattern table is not complete, some other forwarding algorithms could be employed to forward messages temporarily.

When new mobility record becomes available, the mobility pattern can be updated as well. Naturally, the update can be performed immediately once new mobility record is collected. This helps maintain up-to-date information, but incurs high computation cost when contacts or location changes occur frequently. Alternatively, mobility patterns can be updated periodically. It can reduce computation cost but at a cost of higher update latency. In practice, these two update policies can be combined together. It is beneficial that regular mobility patterns are updated immediately at the beginning of the mobility trace collection as we can use new mobility records in time. After that, it can be switched to periodical updates to avoid resource waste.

3.2.2 Routing Decision

Once the mobility pattern is extracted, we quantify its ability to deliver a specific message toward a location. The process for making the routing decision is detailed below.

Firstly, we evaluate the delivery probability which measures a node's ability to deliver a message to its destination location. For one location in a mobility pattern, closeness improvement metric is developed to measure the improvement in terms of distance the device can carry the message to its destination. However, a node's delivery probability for a message is not determined by one metric value, as multiple patterns may exist in the pattern table and one pattern may contain multiple locations. Correspondingly multiple metric values may be obtained. Our solution is to create a discrete probability distribution of metric values spanning all patterns and then calculate the final delivery probability. It can be generated as the expectation, median or maximum of the distribution.

Secondly, given the delivery probability, we can make the relay decision using proposed flexible comparison strategy.

3.2.2.1 Calculating Delivery Probability

Here closeness improvement metric is present to measure the progress on closeness the node can carry the message toward its destination location given one location $l_i \in SL$ in a mobility pattern $p =< d, t, SL >$. The final delivery probability is calculated based on these metric values.

Suppose there are two mobile devices that discover each other at current location lc and one of the nodes carries a message destined to location ld. We show the process of calculating delivery probability for the message in one node. The node obtains one mobility pattern $< d, t, SL >$ from its pattern table locally. $l_i \in SL$ is one location in the pattern. The closeness improvement metric is shown in eq. (3.6).

$$CIM = \frac{D(lc, ld) - D(l_i, ld)}{D(lc, ld)} \times f(t) \tag{3.6}$$

$D(lc - ld)$, $D(l_i - ld)$, calculate the distances between a pair of locations. Delay factor is implemented by decreasing function $f(t)$. It gives more weight to the nearer day type and time slot, and its definition is shown in eq. (3.7). The closeness improvement metric implies that the higher the metric value is, the closer the message can be carried toward its destination. The metric is flexible as when a mobile device is near the message destination, a small step toward the destination can be clearly reflected on the metric value.

Linear and exponential decay functions for example shown in eq. (3.7) can be employed as the decreasing function $f(t)$. Mobility patterns are sorted in increasing order of the time. Variable t refers to the index of a mobility pattern in the sorted order while y is the obtained delay factor for this pattern. We use suitable parameters to control y to satisfy$1 \geqslant y \geqslant 0$ and the metric values are limited in the range $[0, 1]$.

$$y = mt + b, y = e^{-\lambda t} \tag{3.7}$$

Delay factor has important influence on the delivery performance. The holding time of a message in each relay before forwarding to the next hop is critical in determining delivery delay. By considering delay factor in calculating the delivery probability, fast approaching mobility patterns are given higher weights. Consequently, selected relay can forward the message to the next relay without incurring high delay.

Then equation (3.6) is applied for all locations in all mobility patterns. All of the metric values are stored in collection $M = \{m_1, m_2, m_3, ..., m_i, ...m_n\}$. A metric value distribution $P\{X = m_i\}$, $i = 1, ..., n$ is created. The final delivery probability can be calculated as the expected value, median or maximum of the distribution shown in eq. (3.8).

$$\begin{aligned} prob &= E[X] = \sum_{i=1}^{n} m_i \times P_i, or \\ &= Median[X] = \mu_{1/2}(X), or \\ &= Max\{m_1, m_2, ..., m_i, ...m_n\} \end{aligned} \tag{3.8}$$

The maximum metric value reflects the highest potential of the node for delivering the message. It may perform well when the predictive model produces accurate results. In other situations, such as at the beginning phase of the mobility trace collection, the maximum metric value which depends on one location in a mobility pattern might be misleading. In this case, expected value and median are more reliable. Each node in the network can adjust its own calculation according to its current state locally.

3.2.2.2 Making Relay Decision

We now present a flexible comparison strategy to make relay decisions. The strategy is shown in eq. (3.9). $prob_{carrier}$ and $prob_{neighbor}$ refer to the delivery probabilities of the message carrier and its neighbor. Flexible comparison strategy offers more flexibility on the basis of the comparison strategy. With $rate > 1$ the strategy is strict at choosing relays as the chosen relay is required to be a better message carrier at a certain level. It can be used in networks where numerous contact opportunities are available and overhead is then reduced. While with $rate < 1$ the strategy is able to capture more relay opportunities. Neighbors that are not better than the message carrier are involved as they still have the chance to meet good nodes to benefit the message delivery. It is applicable to networks where limited contact opportunities are present and the delivery ratio is then promoted.

$$relay = \begin{cases} yes, & \frac{prob_{neighbor}}{prob_{carrier}} > rate \\ no, & otherwise \end{cases} \tag{3.9}$$

When the message reaches its destination location, it is disseminated within the region using algorithms such as epidemic [38]. The main purpose is to maintain the message in the region for a longer duration and attempt to spray it to more receivers.

3.2.3 Implementing LOOP

LOOP can be readily implemented on current mobile devices given its distributed nature. Below, we use an example to illustrate how LOOP works. Note that a node needs to collect its own mobility trace and periodically extract mobility patterns depicting regular movement of itself from this trace as described in section 3.

Assume that mobile devices carried by Alice and Bob, named N_{Alice} and N_{Bob} are in contact opportunistically. Below are the steps for LOOP implementation.

1) After neighbor discovery, N_{Alice} and N_{Bob} exchange all their messages destined to the current location. This refers to the message spreading within its destination location.

2) N_{Alice} calculates delivery probabilities for all carried messages based on its mobility patterns. The probability calculation is defined in Section 3. N_{Alice} then sends a list of headers of carried messages along with delivery probabilities to N_{Bob}. Upon receiving the header list, N_{Bob} removes the messages it carries already and then calculates its probabilities for the remaining messages. N_{Bob} requests the messages

that it decides to relay in accordance with its relay decision strategy discussed in Section 3. N_{Alice} sends messages to N_{Bob} per its request.

For simplification, we have only described how N_{Bob} determines which messages to relay. Certainly N_{Alice} can decide which messages to relay itself. In this process, only a node's delivery probabilities for messages are sent out to its neighbors for routing purpose. No information, including locations, is exchanged between two nodes. Privacy is therefore preserved. In addition, The delivery probabilities are calculated based on node's own mobility patterns, LOOP is hence scalable with the increase of network load and size.

3.3 LOOPC: LOOP Assisted With User Calendar

In this section, we further improve LOOP with the assistance of user calendars that provide reliable information of user mobility. The resulting design is termed LOOPC, short for LOOP-Calendar.

3.3.1 Exploiting User Calendars in LOOP

Calendars are commonly used to keep track of events in people's lives. In simple form, a calendar consists of scheduled events with associated dates, times, locations and descriptions. They can divulge useful information regarding user's mobility. For OppNets routing in the literature, mobility is commonly obtained by inferring them from context. The inferred mobility is sometimes inaccurate and can lead to biased relay decision. Through calendars, the mobility can be acquired straightforwardly without expensive cost. The extra traffic for collecting the context data is avoided. Besides, the learnt mobility from peoples calendar is much more trustworthy than that inferred from context information.

However, it is unreliable to depend solely on calendars to make relay decision, as only a part of people maintain personal calendars and not all activities are scheduled even there is a calendar available.

Therefore, the calendars which provide easy access to accurate mobility can be a complement to LOOP. The LOOPC, short for LOOP-Calendar, employs both mobility patterns generated from movement trace and calendar events obtained from user calendars to quantify node's ability to deliver a message to the destination and determine the proper relay.

3.3.2 Design of LOOPC

First, we define the calendar events which accurately describe user's mobility. Second, we quantify a node' ability to deliver a specific message toward a location by utilizing both mobility patterns and calendar events. The delivery probability is calculated and relay decision is determined.

In a calendar, the mobility of a person can be termed by attending events. We define calendar event as a tuple in the format $e =< d, ts, te, l >$, it describes a user activity which takes place at location l and on date d and lasts for a period time from ts to te. We then transform the calendar event into LOOP compatible format $e =< d, TS, l >$. d is the day type and TS are the time slots that the event duration falls into. Note that an event lasting for a longer duration might fit into multiple time slots. A collection of calendar events named calendar event table is maintained in LOOPC system, and is updated periodically by adding new events and deleting outdated events.

In LOOP, given a mobility pattern, closeness improvement metric is designed to measure the improvement in terms of distance the node can carry the message to its destination. The metric is shown in eq. (3.6). For a calendar event $e =< d, TS, l >$, the node's ability to carry the message toward its destination can also be quantified using this equation except that the day type, time slot and location are replaced by them of calendar event. The closeness improvement metric for calendar events is shown in eq. (3.10). $f(t)$ defined in (3.7) is a decreasing function factoring delay when measuring closeness improvement. Calendar events are sorted in increasing order of the time. Variable t refers to the index of a calendar event in the sorted order.

$$CIM = \frac{D(lc, ld) - D(l, ld)}{D(lc, ld)} \times f(t) \tag{3.10}$$

Compared to mobility patterns depicting periodic contacts, calendar events predict future movement without contact consideration despite with higher accuracy. In other words, a calendar event e shows that the user will visit a certain location at a specific time while the contact opportunity is not taken into account. To solve this issue, we define a special type of calendar events $se =< sd, sTS, sl >$, a calendar event that can also be observed in mobility patterns. There is a mobility pattern $p =< d, t, SL >$ in the pattern table with $sd = d$,$sts \in sTS = t$ and $sl = l_i \in SL$. The special calendar events combine the benefits of both mobility patterns and calendar events, taking contacts into account and accurately describing mobility. By identifying special calendar events, we recognize accurate mobility with highly possible contact opportunity. The special calendar event is given a weight $w > 1$ when the improvement toward destination is measured as displayed in eq. (3.11).

$$CIM = w \times CIM, w > 1 \tag{3.11}$$

We measure the improvement using all mobility patterns, calendar events and special calendar events. Then the following process is similar with LOOP. All of the metric values are stored in collection $M = \{m_1, m_2, m_3, ..., m_i, ...m_n\}$. A metric value distribution $P\{X = m_i\}$, $i = 1, ..., n$ is created. The final delivery probability can be calculated as the expected value, median or maximum of the distribution shown in eq. (3.8).s

Finally we make the relay decision based on the final delivery probability using our flexible comparison strategy displayed in eq. (3.9).

3.4 Analytical Study

In order to evaluate LOOP theoretically, we adopt the analytical model from [72]. We first briefly describe the general analytical framework and then adapt it to fit our scheme.

The single-copy message forwarding process is modeled as a semi-Markov chain which is a Markov process but the transition from one state to another takes a random amount of time called holding time. For a network with N nodes, there are N states in the semi-Markov chain. Each state i represents that the node i is currently holding a message. The transition p_{ij} from states i to j is the behavior of the forwarding of message from nodes i to j. The holding time T_i^{exit} denotes the time node i carries the message before it hands it over to another node.

Now we describe the semi-Markov chain by transition matrix and holding times. It is assumed that in OppNets the inter-contact time between any two nodes follows an exponential distribution with rate denoted as λ, and it is proven in [72] that with this assumption the holding time T_i^{exit} of state i also follows an exponential distribution with rate $\sum_{i=1,j\neq i}^{N} \lambda_{ij} P_{ij}^{forw}$. Here P_{ij}^{forw} is the probability of forwarding a message from node i to j in accordance with the specific forwarding algorithm. The expectation of T_i^{exit} is calculated in eq. (3.12).

$$E[T_i^{exit}] = \frac{1}{\sum_{i=1,j\neq i}^{N} \lambda_{ij} p_{ij}^{forw}} \tag{3.12}$$

For a forwarding process of message generated in node i and destined to d, the transition probability matrix is shown in eq. (3.13) where $d = N$. Obviously, $p_{di} = 0$, as destination node d will not send out the message to others.

$$P = \begin{pmatrix} 0 & p_{12} & \dots & p_{1,N-1} & p_{1,N} \\ p_{21} & 0 & \dots & p_{2,N-1} & p_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \tag{3.13}$$

The transition probability p_{ij} is given as in eq. (3.14).

$$p_{ij} = \frac{\lambda_{ij} p_{ij}^{forw}}{\sum_z \lambda_{iz} p_{iz}^{forw}} \tag{3.14}$$

The forwarding probability p_{ij}^{forw} for LOOP is defined in (3.15). We describe how to derive $prob_i$ for a node i given a message. Mobility pattern of node i is generated. Under the assumption of node's random mobility, for a time point, a node can reach any location of its movement area with equal opportunity. So the date type and time slot are redundant to mobility patterns and can be ignored. Here a pattern is one location that the node visits. For a node, all locations in its movement area are added as its mobility patterns. Then with the mobility patterns, for a given message, we evaluate the node's ability to deliver the message. The closeness improvement of all patterns can be calculated using eq. (3.6) without delay factor and delivery probability $prob_i$ is produced as the maximum of the metric distribution according to eq. (3.8). A message will be forwarded from node i to j when $\frac{prob_j}{prob_i} > rate$ where $rate = 1$. We fix one node to the destination location as the destination in order to adapt the model to our scheme.

$$p_{ij}^{forw} = \begin{cases} 1, & \frac{prob_j}{prob_i} > rate \\ 0, & otherwise \end{cases} \tag{3.15}$$

Based on the results in [72], the expected delay and number of hops for a message generated from one node i and destined to d are shown in eq. (3.16) and (3.17).

$$\begin{cases} E[D_i^d] = 0, & i = d \\ E[D_i^d] = E[T_i^{exit}] + \sum_{j \neq d} p_{ij} E[D_j^d], & \forall i \neq d \end{cases} \tag{3.16}$$

$$\begin{cases} E[H_i^d] = 0, & i = d \\ E[H_i^d] = 1 + \sum_{j \neq d} p_{ij} E[H_j^d], & \forall i \neq d \end{cases} \tag{3.17}$$

Four message forwarding policies: direct transmission (DT), always transmission (AT), Direct Acquaintance (DA) and Social Forwarding (SF) are evaluated in [72] in terms of expected delay and hop number. They fall into the three categories summarized in our related works. Direct and always transmission do not differentiate better relays; direct acquaintance is based on context information; social forwarding involves the social knowledge. Always transmission and direct acquaintance can be seen as Epidemic and Prophet algorithms except that they are based on single-copy forwarding.

In this chapter, we compare them with our scheme as well. The forwarding probabilities of the four scheme are shown in table 3.1, where $F_{i,d}^{DA} = \lambda_{i,d}, \forall i \neq d$,$F_{i,d}^{SF} = \alpha F_{i,d}^{DA} + (1-\alpha) F_{i,d}^{I}, 0 < \alpha < 1$,$F_{i,d}^{I} = f(F_{i,d}^{DA}), \forall j | \lambda_{ij} \neq 0, j \neq d$.

Community is an important attribute of OppNets, because mobile devices are carried by people who tend to belong to communities. Cooperation binds, but also divides human society into communities [53]. Therefore, we use the heterogeneous network shown in Fig. 3.1 with multiple communities for evaluation purpose. All nodes within the same community are connected with each other. Adjacent commu-

Table 3.1. Forwarding Policies

DT	$p_{ij}^{forw} = \begin{cases} 1 & j = D \\ 0 & otherwise \end{cases}$
AT	$p_{ij}^{forw} = 1, \forall i, j$
DA	$p_{ij}^{forw} = \begin{cases} 1 & F_{i,d}^{DA} < F_{j,d}^{DA} \\ 0 & otherwise \end{cases}$
SF	$p_{ij}^{forw} = \begin{cases} 1 & F_{i,d}^{SF} < F_{j,d}^{SF} \\ 0 & otherwise \end{cases}$

nities are bridged by nodes called travelers that travel across the two communities. There are 21 nodes in the whole network which move randomly in their areas. We assume that the default meeting rate is λ for each pair of nodes connected. For travelers in touch with n communities, the rate of contact with users in each of those communities is λ/n. We assign one regular node to each of the community, one traveler to each pair of the adjacent communities and analyze the performance of forwarding messages generated from any node i to node d in community $C9$.

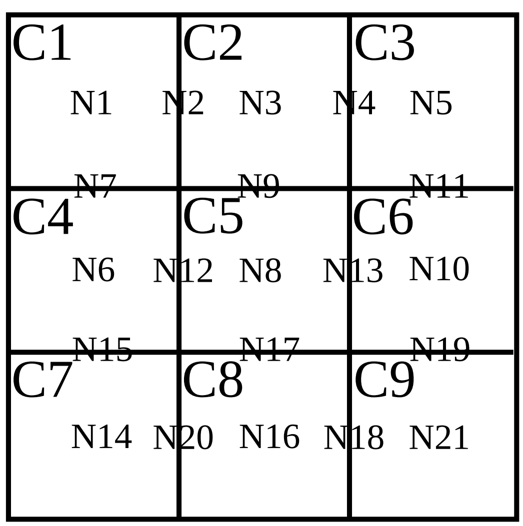

Figure 3.1. A Heterogeneous Network.

Figs. 3.2 and 3.3 show the expected delay for forwarding message generated from each node in the network using different forwarding algorithms. λ is set to be a constant number. Note that the zero delay indicates delivery failure. Delivery rate can then be derived from the figures of delays.

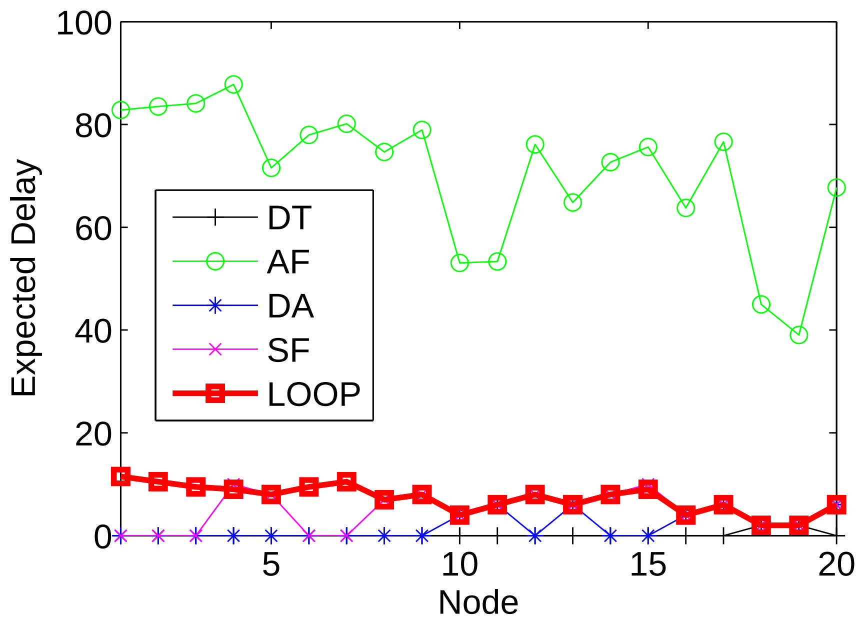

Figure 3.2. Expected Delay in Heterogeneous Network.

In direct transmission, messages can only be delivered to the destination when encountered directly. The delivery ratio is extremely low at 10%. In always transmission, message is forwarded to the first node the message carrier encounters. The process stops when the message is delivered to the destination. The delivery ratio reaches 100%. However the expected delay and hop number are the highest among all forwarding algorithms. For direct acquaintance, the message carrier forwards message to encountered nodes that have higher opportunities to meet the destination. The delivery ratio reaches 40%. For messages that can be delivered, direct acquaintance

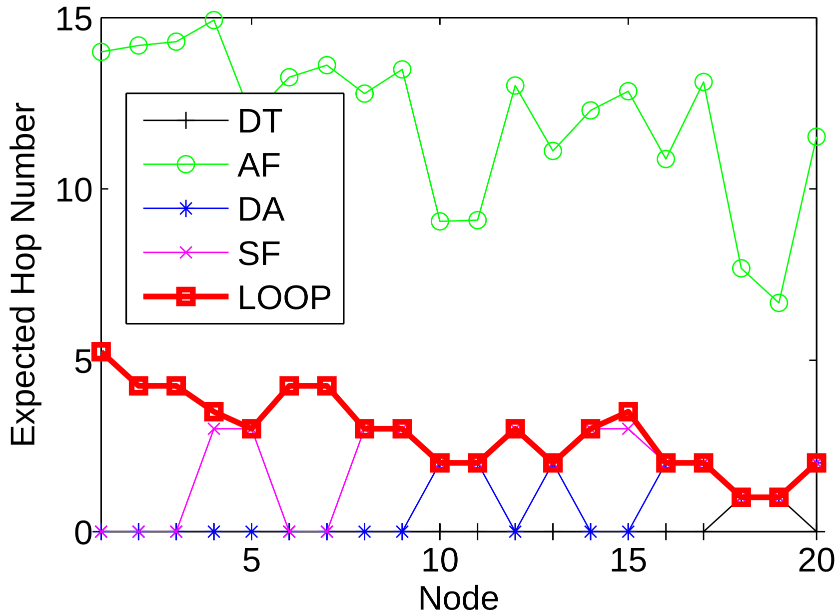

Figure 3.3. Expected Hop Number in Heterogeneous Network.

has better delay and hop number than always transmission mostly. Social forwarding is similar to direct acquaintance except that it possesses the property of acquaintance transition. More messages are delivered successfully with 75% delivery ratio. LOOP provides 100% delivery, higher than direct and social forwarding, and has similar expected delay and hop number to them.

In order to evaluate the performance of these forwarding algorithms in a network where all communities are connected together. A global traveler is added into the heterogeneous network. It travels around the whole network and hence connects all communities with the destination. The delays and hop numbers are displayed in figs. 3.4 and 3.5. The expected delay and hop number of LOOP are comparable with those of the direct acquaintance and social forwarding. The delay is slightly higher than that of social forwarding. However, LOOP can adapt itself in practice. By varying function $f(t)$ in eq. (3.6), which is omitted in the analytical study, it is allowed that

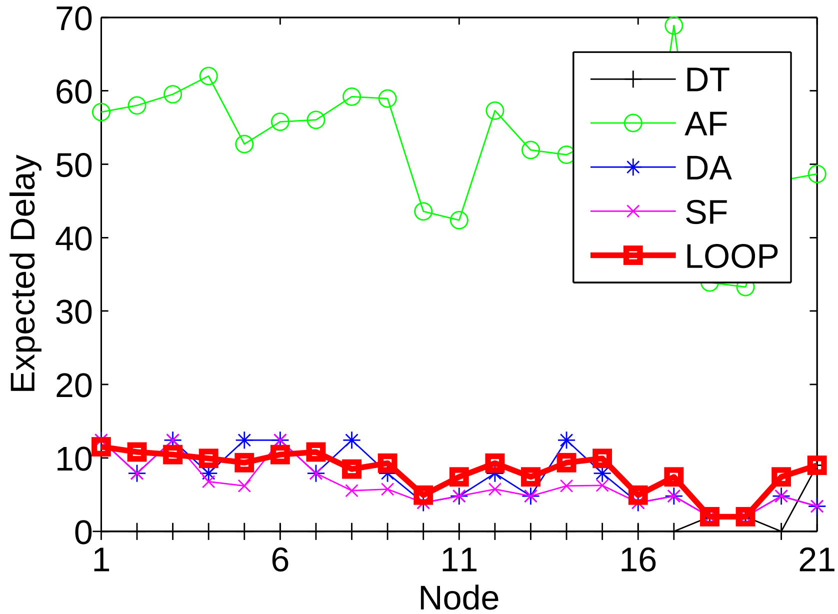

Figure 3.4. Expected Delay in Heterogeneous Network With Global Traveler.

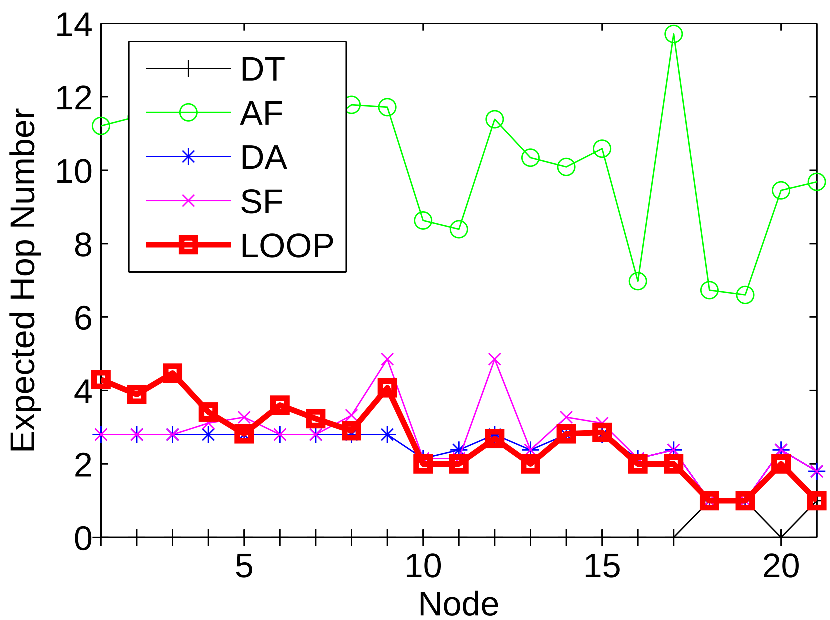

Figure 3.5. Expected Hop Number in Heterogeneous Network With Global Traveler.

$N21$ relays messages to $N18, N19$ instead of global traveler as $N18, N19$ tend to visit the destination sooner. Incorporating this into the model is our future work.

3.5 Simulation Study

In this section, we study the performance of LOOP experimentally. Opportunistic network environment simulator (ONE) [73] is employed as the simulation platform. We show the accuracy of predicting future movement by learning mobility patterns from mobility trace. The performance of LOOP is evaluated and compared with landmark routing algorithms Epidemic [38], Prophet [41] and BubbleRap [53] from the three categories summarized in related works. Simulations on LOOPC are conducted to measure its improvement on LOOP.

When calculating delivery probability, delay factor is considered to balance delivery ability and delay. We carry out simulations to display the benefits of applying delay factor in delivery probability. Besides, training data set size plays an essential role in determining the computation cost and bootstrapping time of LOOP. Simulations are conducted to support that training data generated by collecting mobility trace within one mobility cycle is enough for extracting mobility patterns.

3.5.1 Simulation Setup

Movement model is important in evaluating OppNets routing algorithm. The commonly used Random Waypoint assuming nodes move randomly is not suitable in real world. To evaluate LOOP and LOOPC, we employ Working Day Movement Model [74] which presents the everyday life of average people that go to work in the morning, spend their day at work, and commute back to their homes at evenings. The model intuitively depicts the movement pattern of people, but it is also verified that it is heterogeneous in both time and space and is able to produces similar distributions of inter-contact times and contact durations as real user traces.

The simulation scenario is on a real city map with size $10km$ by $8km$ where 400 nodes move around. The whole map is divided into four districts, about half of the nodes live and work in the same district, and the other half commute between different districts. The simulation scenario constrains node movement to predefined paths and routes derived from real map data. 50% of the nodes take public transportation for commuting between offices and homes. Nodes attend meetings during office hours and join in activities after work. 400 homes, 70 offices and 35 meeting spots are set in the scenario.

Nodes use Bluetooth as wireless interface with radio range $10m$. There is a stationary sink in a randomly chosen location to accept messages. The simulation time for one run is 72 hours and the first 24 hours are given for warm-up since Prophet, BubbleRap and LOOP need time for collecting required data for routing. One simulation run is repeated 10 times for statistical confidence.

The general parameter settings for LOOP, Epidemic, Prophet and BubbleRap are described in the following. For LOOP, we use one day type while a day is divided into 6 time slots with 4 hours per time slot. BR-NB classification approach is employed to learn mobility patterns based on collected mobility trace. The final delivery probability is produced as the maximum of distribution on closeness improvement metric values. We set parameter $rate$ as 0.9 in flexible comparison strategy for relay decision determination. For Epidemic, messages are forwarded to any encountered neighbor. For Prophet, the parameters are set as recommended in [41]. For example, $P_{init} = 0.75$, $\beta = 0.25$ and $\gamma = 0.98$. For BubbleRap, community is detected through Kclique method with $K = 3$ and $familiarthreshold = 700$. The centrality degree is calculated by S-Window method.

3.5.2 Evaluation of Mobility Pattern Extraction

BR-NB and CC-NB methods learn and extract the regular mobility pattern from mobility trace in local devices. We discuss the performance of BR-NB in this section. For a given test set $S = \{x_i, Y_i | 1 < i < p\}$, where x_i refers to the new test instance in the forms of $< daytype, timeslot >$, and Y_i denotes the predicted set of locations accordingly, the prediction of the set of labels for new instance is evaluated in terms of accuracy, precision, recall, and F-measure. The evaluation metrics are shown in eq. (3.18). For a test instance x_i, R_i is the real set of locations, while Y_i is the predicted set of locations. $< x_i, Y_i >$ denotes an extracted mobility pattern stored in the local device.

$$\begin{aligned}
Precision &= \frac{R \bigcap Y}{Y} \\
recall &= \frac{R \bigcap Y}{R} \\
F - measure &= \frac{1}{p} \sum_{i=1}^{p} \frac{2 \times P_i \times R_i}{(P_i + R_i)} \\
Accuracy &= \frac{1}{p} \sum_{i=1}^{p} \frac{P_i \bigcap R_i}{(P_i \bigcup R_i)}
\end{aligned} \tag{3.18}$$

Precision and recall are the fractions of successfully classified labels of predicted and real labels respectively. F-measure which combines precision and recall is the harmonic mean of precision and recall. Accuracy calculates the proportion of successfully classified labels of the union of predicted and real labels.

In our simulation, each node in the network collects its mobility records in the first day. The second day is selected to generate test instances $\{(x_i, R_i, Y_i) | i = 0, ..., 6\}$. x_i indicates the date and time slot. A_i and Y_i are the real and predicted set of locations associated with x_i. As a test set contains 6 instances, the network has 80 nodes and the emulation is repeated 10 times, there are a total of 4800 test instances. The metric values are calculated for all test instances. Fig. 3.6 depicts the averaged

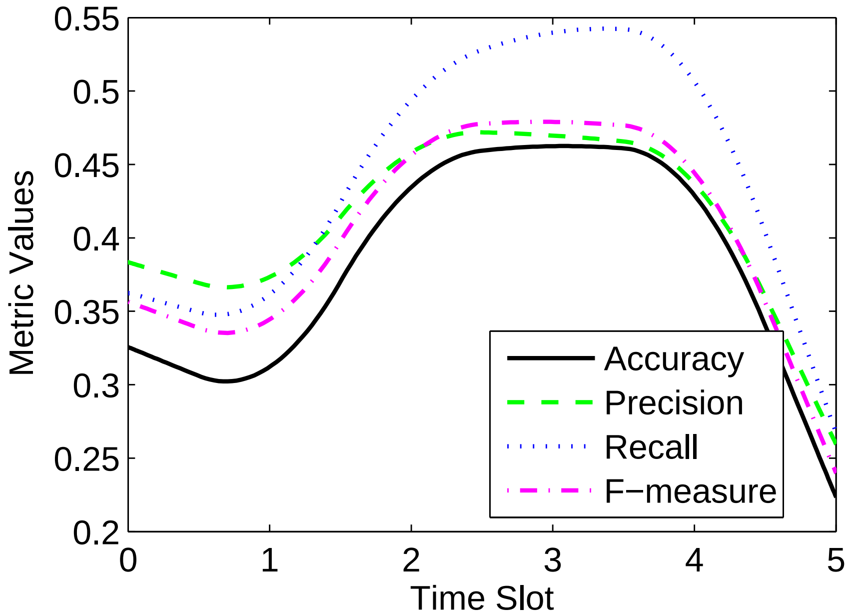

Figure 3.6. Evaluation Metric Values For Various Time Slots.

metric values of test instances with the same time slots. Notice that the evaluation metrics dictate the performance of the mobility pattern extraction: high precision value ensures that messages will not be forwarded to unsuitable relays, while high recall value means that good relay candidate will likely be discovered. The variation of performance indicates that BR-NB method generates relatively more accurate results in certain time periods, as nodes tend to move regularly in these time periods. Therefore the extracted mobility pattern for these time slots is more trustworthy to guide the message forwarding.

3.5.3 LOOP Performance Evaluation

For LOOP and compared routing schemes Epidemic, Prophet and Bubble, we study their delivery performances over offered buffer capacity, traffic load, and message TTL. The delivery performance that we are interested in is in terms of delivery success ratio, latency and traffic overhead. Delivery success ratio is defined as the proportion of messages that have been delivered out of the total unique messages created. Latency is the average time delay from the time a message is created to the time it is delivered to destination. Traffic overhead is the total number of messages (including duplicates) transmitted across the air. The control traffic is not included. [2]

3.5.3.1 Set 1 - Performance with Offered Buffer Capacity

Even though memory is cheap nowadays, messages in OppNets are usually far larger than them in conventional networks, since whole files can be accommodated in a single message. For an average message size of 1MB, with 400 messages generated in the whole network, Epidemic should reserve 400MB just for routing purpose [43]. It is worthy to study the performance of routing schemes with offered buffer capacity. The buffer size in this set varies from 80 messages to unlimited. Each node creates one message destined to the same node in one run of simulation. The message TTL is set to be 48 hours. The limitation of buffer capacity has important influence on the performance of a routing algorithm. We employ a drop tail approach when a buffer overflows. Figs. 3.7, 3.8 and 3.9 depict the performance of different routing algorithms for various buffer sizes.

From Fig. 3.7, we can see that LOOP does not require large buffer capacity to deliver messages. The limited buffer size has much less influence on its delivery performance. LOOP enables at least 98% of the messages to be delivered successfully. When the buffer is unlimited, Epidemic shows the upper bound of delivery performance and it receives the highest delivery ratio 99.8%.

[2] Epidemic, LOOP generate much less control traffic, while Prophet, BubbleRap require more for routing information exchange.

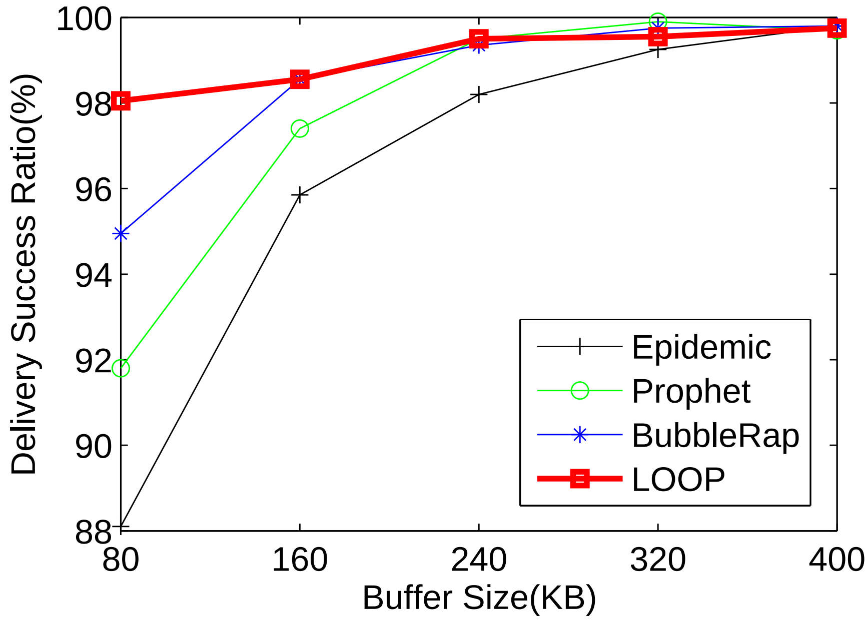

Figure 3.7. Performance with Offered Buffer Capacity:Delivery Ratio.

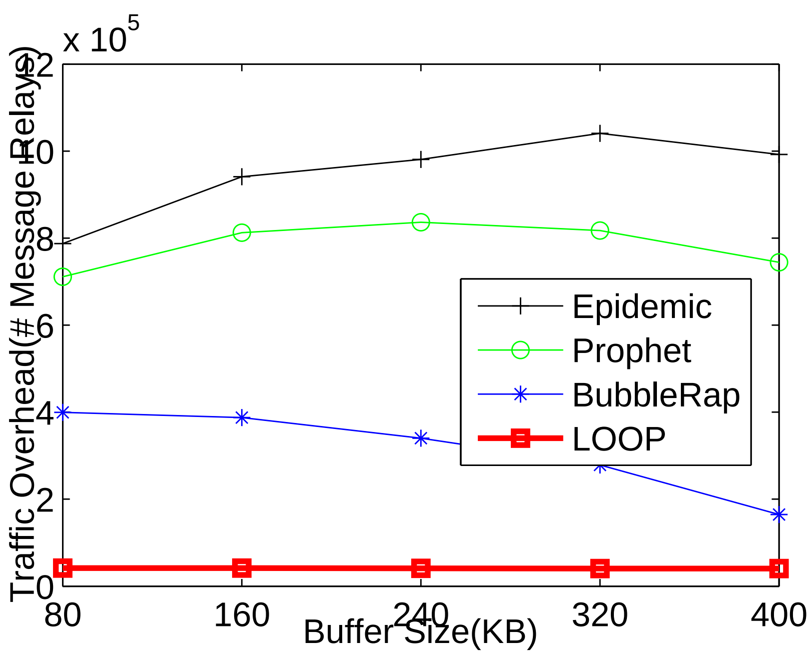

Figure 3.8. Performance with Offered Buffer Capacity:Overhead.

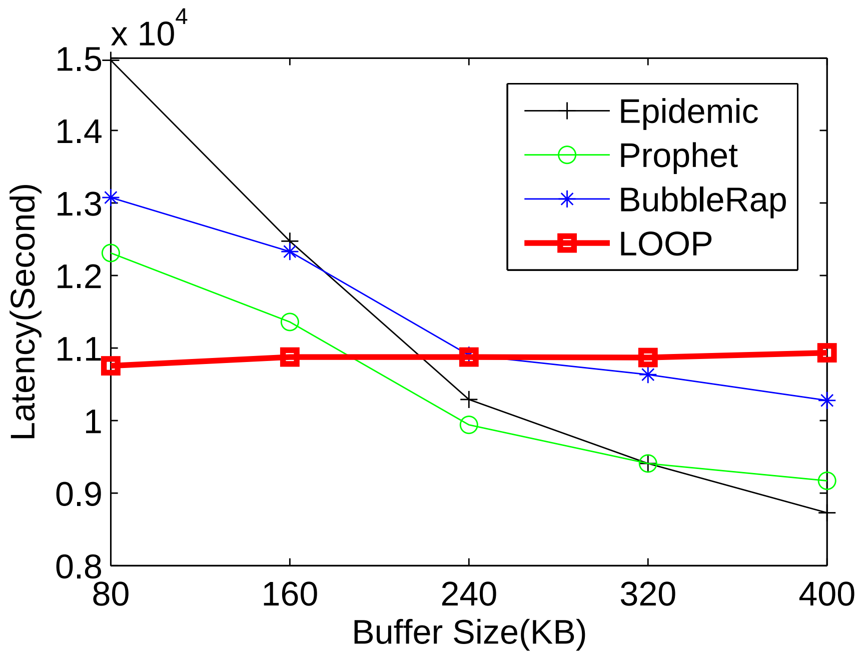

Figure 3.9. Performance with Offered Buffer Capacity:Latency.

The traffic overheads of these algorithms are shown in Fig. 3.8. It is clear that LOOP generates the least amount of overhead messages. It is about 10%, 6% and 5% of that of BubbleRap, Prophet and Epidemic respectively. As noted above, control traffic generated by Prophet and BubbleRap are not counted in our simulation. In Prophet, the delivery probability table contains entries for all the neighbors the node has met. When two nodes run into each other, the tables are exchanged. For BubbleRap, when two nodes meet, they exchange familiar sets, local community sets, and other information for the community detection.

Fig. 3.9 shows that LOOP tends to have shorter message delivery delay than Epidemic, Prophet and BubbleRap when buffer size is less than 240 KB. This advantage becomes not obvious when the buffer size increases. This might be a result of inaccurate prediction of regular mobility pattern. More accurate prediction of regular mobility pattern can improve the performance of LOOP, especially on the delivery latency.

3.5.3.2 Set 2 - Performance with Offered Traffic Load

In this set of simulations, we aim at studying the routing performance under heavy traffic load. We set 80-message buffer size and 48-hour TTL while with varied amount of messages from 100 to 1600.

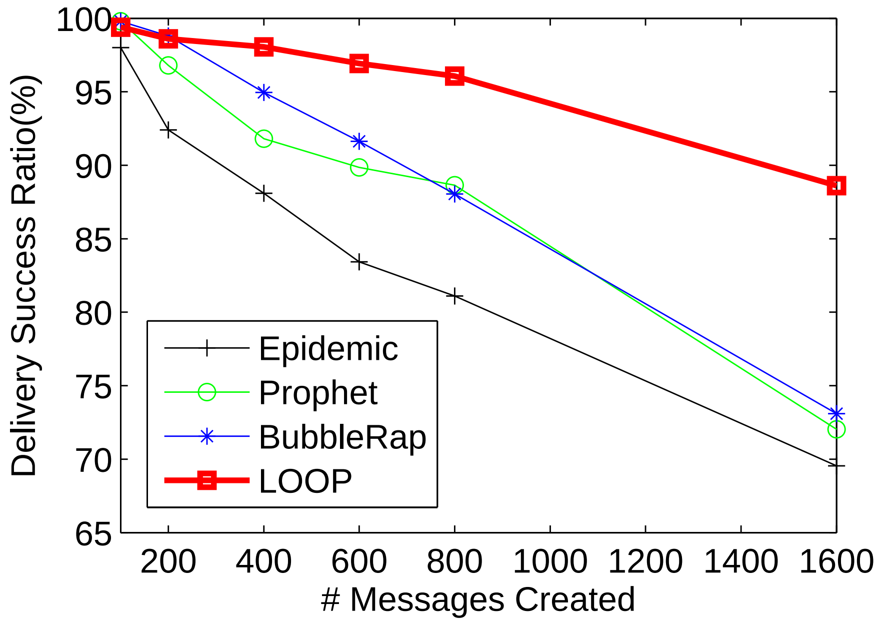

Figure 3.10. Performance with Offered Traffic Load:Delivery Ratio.

Figs. 3.10, 3.11 and 3.12 clearly show that LOOP is able to outperform others most of the time. LOOP provides much higher delivery success ratios especially when traffic load is heavier in the network. With 1600 messages in the network, LOOP can deliver about 87% of messages compared with the 69%, 72% and 73% delivery rate of Epidemic, Prophet and BubbleRap. In terms of delivery delay, LOOP has the best

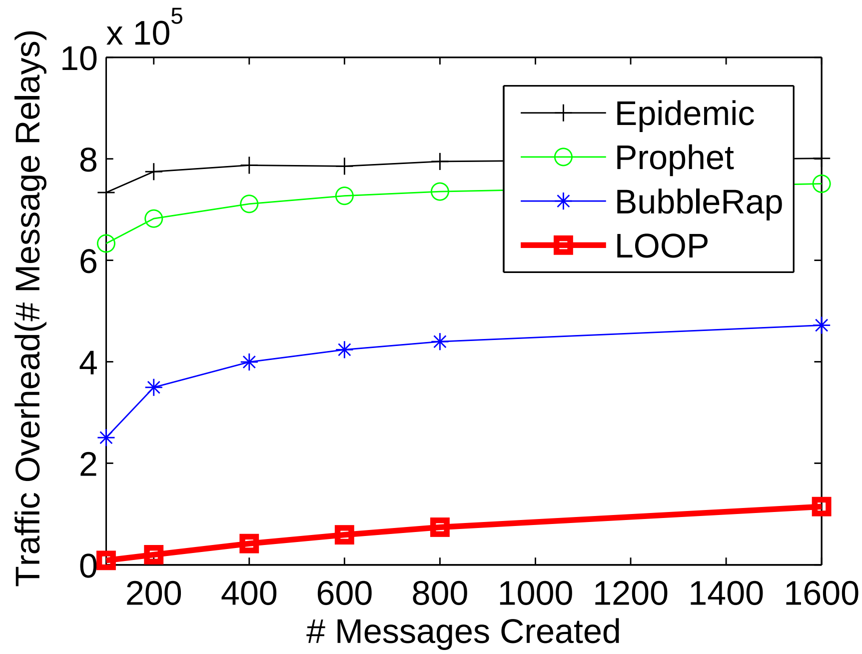

Figure 3.11. Performance with Offered Traffic Load:Overhead.

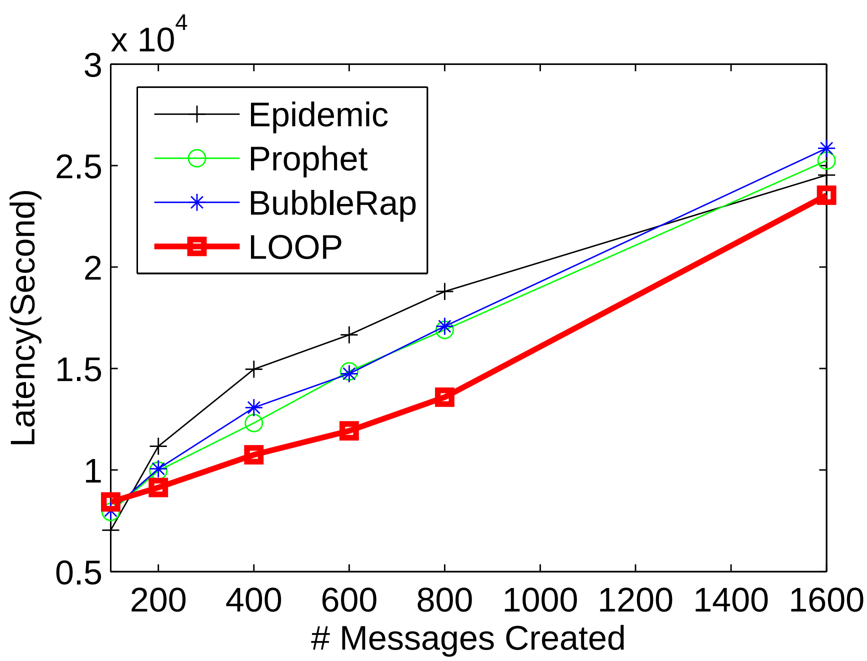

Figure 3.12. Performance with Offered Traffic Load:Latency.

performance among all routing schemes in this simulation set when the buffer size is limited.

3.5.3.3 Set 3 - Performance with Offered TTL

In the third set of simulations, we show that how LOOP and other routing schemes perform with different message TTLs. In reality, messages labeled with urgent require short delivery delay. The performance with offered TTL shows the ability of routing schemes to deliver messages within required time. Messages that can not be delivered within TTL will be dropped. In this set of simulation, buffer size is limited to 80 messages and 400 messages are created for one run of simulation. Message TTL changes from 480 minutes to 2880 minutes. Figs. 3.13 and 3.14 show that LOOP achieves the highest delivery ratio with very low traffic overhead. LOOP delivers 95% of messages in the first 8 hours, and the delivery delay is the lowest among all the routing schemes.

3.5.4 Performance Evaluation of LOOPC

In this section, we compare the performance of LOOPC to LOOP. In the simulation, 12.5% of nodes are randomly selected to maintain calendars and around 10 calendar events are created in a calendar to depict the mobility of the calendar owner. 400 messages with TTL of 24 hours are generated, and the buffer is set to be unlimited in order to isolate its effect. Simulations are conducted on LOOP and a group of LOOPC. The group of LOOPC is set with different weights for special calendar events defined in eq. (3.11). When $w = 1$, we treat all mobility patterns and calendar events equally. With the increment of w the special calendar events are identified and given higher weights than both mobility patterns and other calendar events. Their performance results in terms of delivery ratio and overhead are displayed in Figs 3.15

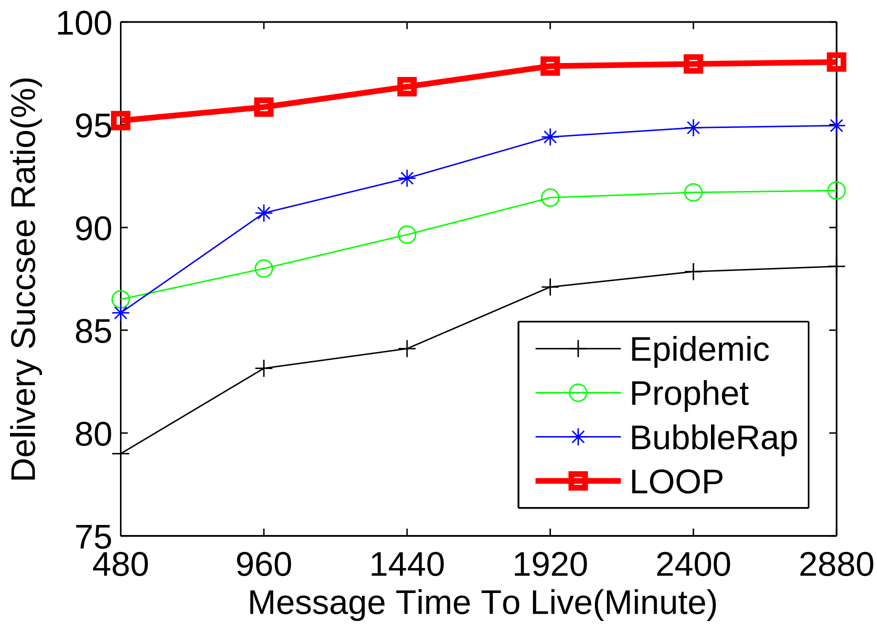

Figure 3.13. Performance with Offered TTL:Delivery Ratio.

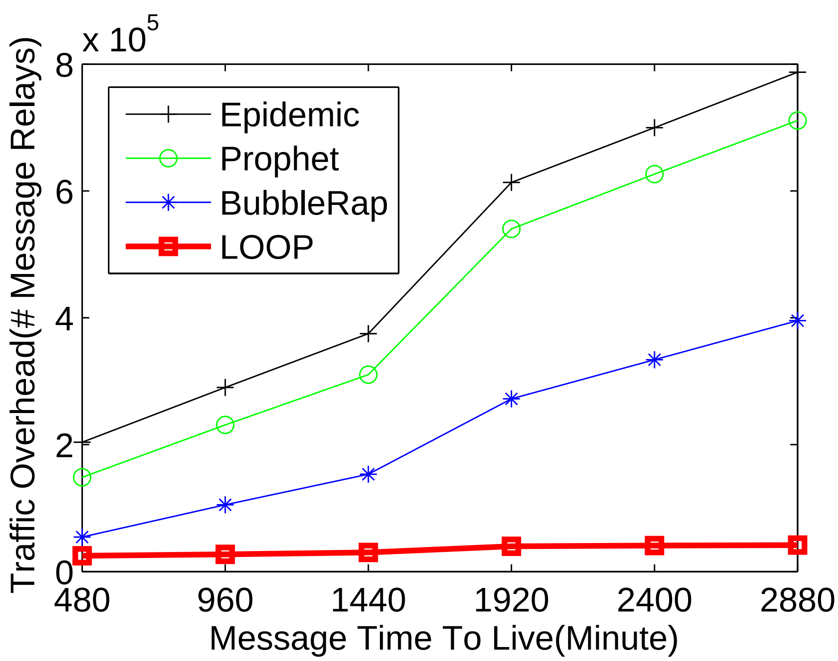

Figure 3.14. Performance with Offered TTL:Overhead.

and 3.16. On the X axis, the "N" refers LOOP algorithm which considers no calendar events.

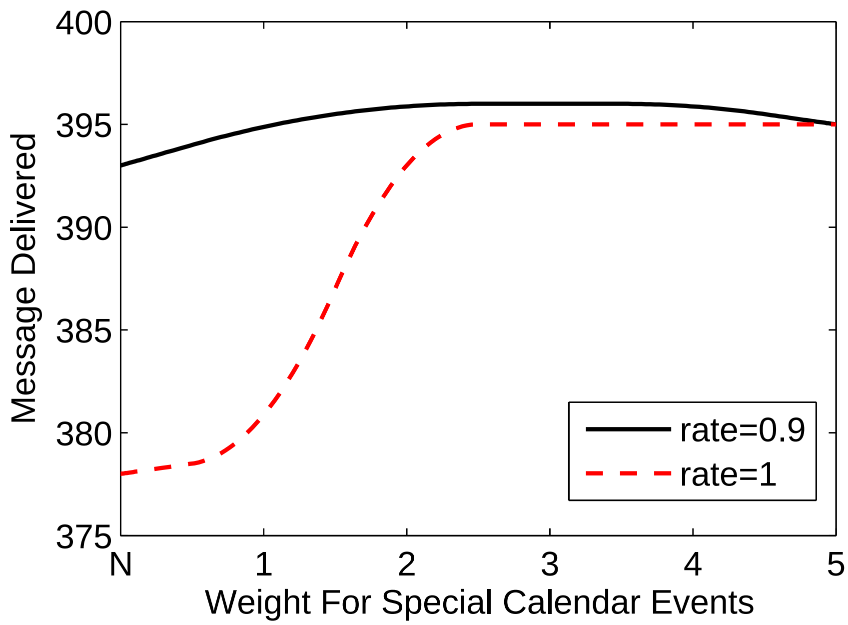

Figure 3.15. LOOPC Performance:Delivery Ratio.

Recall that in flexible comparison strategy, *rate* is a parameter which controls the strictness in choosing a relay. When $rate = 0.9$ the strategy is generous in selecting relays, results in increasing delivery ratio but more resource consumption. LOOPC helps in reducing the overhead significantly while remaining the high delivery ratio. When $rate = 1$ the strategy is strict in picking relays, leads to lower resource consumption but relatively lower delivery performance. Overall LOOPC can increase the delivery ratio while consuming less resources.

The performance of LOOPC is comparable to LOOP when we treat calendar event and mobility patterns equally, as the contact is not taken into account in calendar events. However, the performance is improved evidently if the special calendar

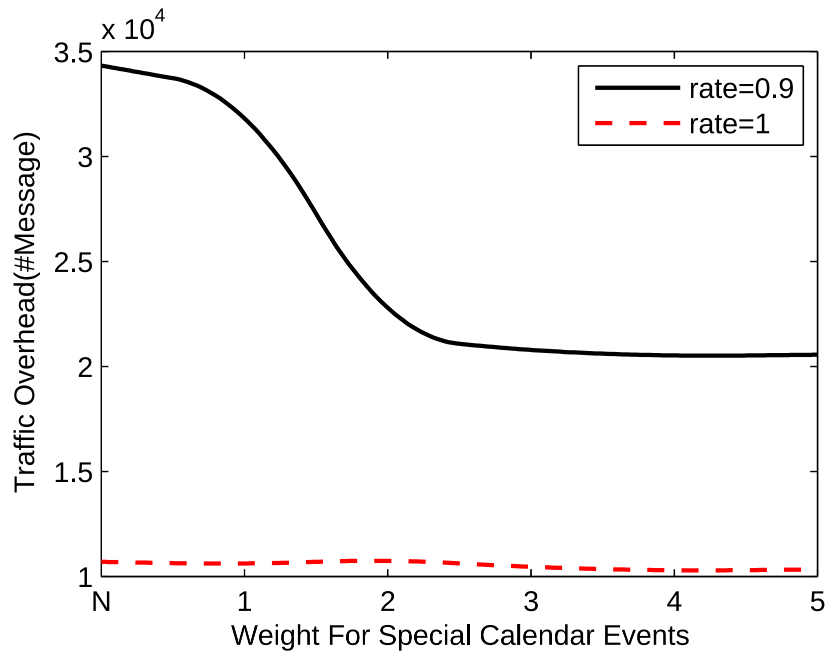

Figure 3.16. LOOPC Performance:Overhead.

events can be identified, as the accurate mobility with highly possible contact can be recognized by special calendar events.

Therefore, by identifying special calendar events, the overlapping of mobility patterns and calendar events, LOOPC can improve LOOP by providing higher delivery ratio and lower resources consumption.

3.5.5 Benefits of Delay Factor in Calculating Delivery Probability

The delay factor is produced by decreasing function defined in eq. (3.7), which gives more weight to the nearer day type and time slot. In the simulation, we tune the decreasing function of the 1 hour time slot to decrease the priority on nearer mobility patterns from very large to none to study how LOOP performs in terms of delivery and traffic load. 400 messages with TTL of 24 hours are generated, and the buffer is unlimited.

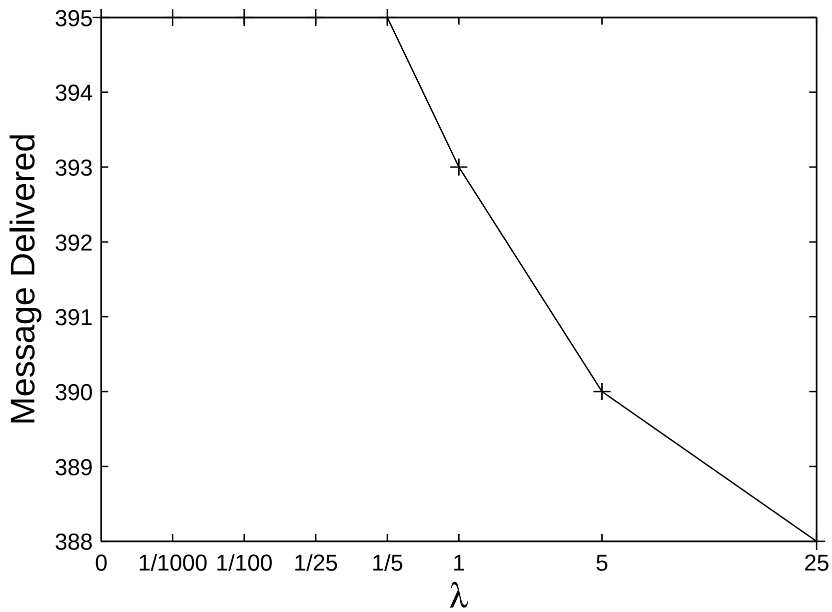

Figure 3.17. Delay Factor:Delivery Ratio.

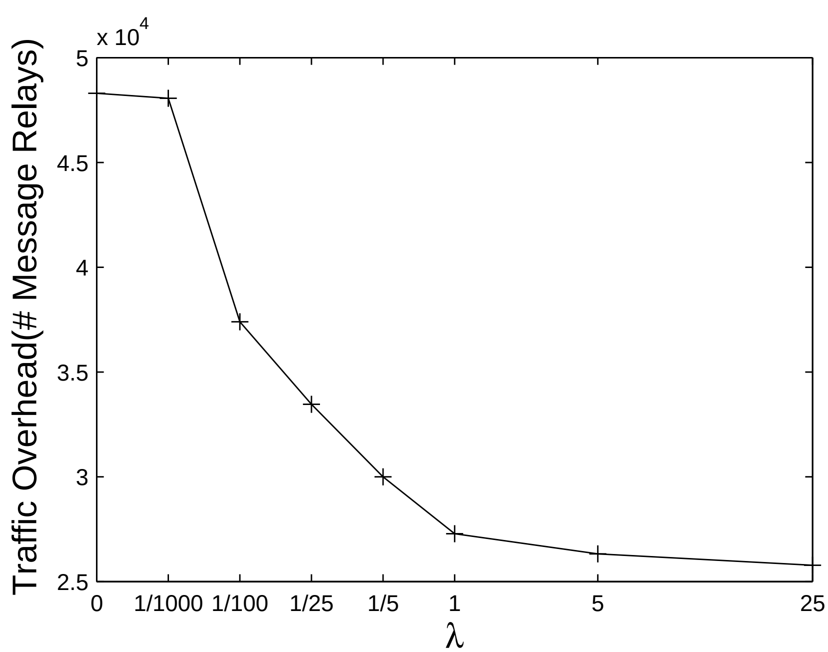

Figure 3.18. Delay Factor:Overhead.

Exponential decay defined in eq. (3.7) is utilized. Figs. 3.17 and 3.18 show that larger λ delivers comparable amount of messages before expiration by triggering less relays. Larger λ results in steep decay curve and the delivery probability calculation relies more on nearer mobility patterns. We are selective in choosing relays who will hold the message in a shorter period. As a result, it consumes less resources. With smaller λ and thus flat decay curve, we are less selective in choosing relays, and more network resources are consumed. Note that 1 hour time slot degenerates into 24 hour time slot with $\lambda = 0$.

3.5.6 Determine Training Data Set Size

We conduct simulations to study the performance of LOOP with varied training data set sizes. 400 messages with TTL of 24 hours are generated, and the buffer is unlimited. The results are exhibited in figs. 3.19 and 3.20. During the bootstrapping phase, Epidemic is employed as a temporary routing scheme.

According to our simulation settings where one day type and six time slots are defined, the mobility cycle is thus a day. As seen in Figures, after one mobility cycle, the LOOP can achieve satisfied delivery performance. Subsequently, with the growth of training data set size, we do not observe clear increase trend of performance with regard to delivery rate and traffic load.

3.6 Summary

In this chapter, we have proposed LOOP, a new location based routing scheme for large-scale opportunistic networks. By forwarding messages to specified location instead of a targeted node, LOOP can serve as the underlying routing protocol for a plethora of pervasive applications. The location based routing is adapted and applied in the context of opportunistic networks. Our results show that node's mobility

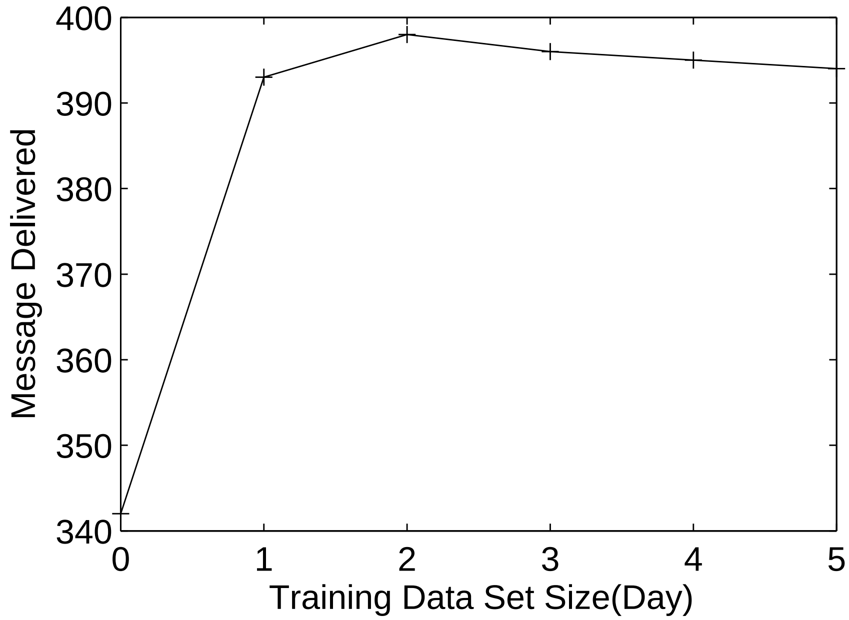

Figure 3.19. Training Data Set Size:Delivery Ratio.

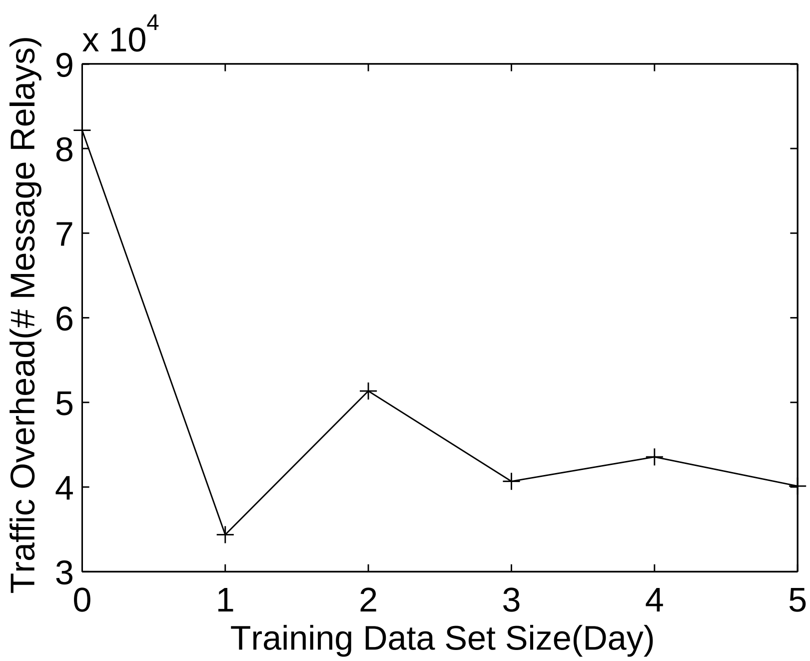

Figure 3.20. Training Data Set Size:Overhead.

patterns that are learnt from mobility trace can be effectively employed in message forwarding.

LOOPC further improves LOOP through the acquisition of mobility information from user calendars. By identifying special calendar events which disclose accurate mobility with highly possible contact opportunity, LOOPC can improve LOOP through delivery performance and resources consumption.

We evaluate performance of LOOP and compare it with well known protocols including Epidemic, Prophet and BubbleRap. The simulation results indicate that our scheme is able to deliver messages at a high ratio, drastically reduces network load and nodes' buffer occupation, especially when more messages are involved in the network.

CHAPTER 4

Geoopp: Geocasting For Opportunistic Networks

This chapter presents Geoopp which provides geocasting service for large-scale opportunistic networks. Location targeted routing is also named Geocasting and aims to deliver information to all nodes within a geographic area rather than an arbitrary group of nodes. Supporting geocasting in the context of opportunistic networks where nodes are not well-connected is still an open problem.

Geoopp is similar to LOOP, as both adapt geographic greedy routing and fit it in the context of OppNets. However, they differ in determining the future locations of nodes. Geoopp offers a novel way to characterize node mobility by introducing the concepts of inter-visiting time and contact availability per visit. Chebyshev's inequality is employed to capture the regularity embedded in visit and contact and compute the probabilities that a node will visit a region and have contact inside.

We first introduce the idea of Geoopp in Section 4.1 and then present the scheme of Geoopp in Section 4.2. We study the performance of Geoopp and compare it with LOOP and other well known routing schemes in Section 4.3.

4.1 Overview

A wide range of geocasting protocols are present for mobile ad hoc networks (MANETs). However, there the assumption that nodes are well-connected in MANETs restricts these protocols from being applied to opportunistic networks. In this chapter, we propose a routing algorithm to provide *GEOcasting* service for large-scale

OPPortunistic networks for the purpose of forwarding messages toward destination region, termed Geoopp.

Geoopp combines unicasting and flooding by first forwarding a message to the specified geographic region and then flooding the message to all nodes inside the region. Forwarding a message toward a region is clearly the key issue to solve. Geoopp works in a greedy manner for the purpose of forwarding messages to their destination regions. Traditionally, geographic greedy routing tries to forward a message closer to the destination in each step. Each node chooses the neighbor that is geographically closer to the destination to relay a message, and "closer" is decided by the routing metric specified in the greedy routing algorithm. It is suitable in well-connected networks where a message flows in the direction of destination hop by hop. Unfortunately, it cannot be applied directly to OppNets with sparse nodes and intermittent connectivity arise in the course of human mobility. A message in OppNets is routed in a store-carry-forward style and the message is carried by a node until it meets the next relay. In this chapter we adapt geographic greedy routing for OppNets where each node chooses the neighbor that can carry the message closer to destination. A progress within radius metric (PWRM) is introduced to measure the geographic progress a node can make carrying a message toward its destination region given one of its future visited regions.

Next we determine the future visited regions in Geoopp. Note that future visited regions should be correlated with visiting time for the sake of minimizing message delivery delay. Future mobility pattern can be inferred by the past movement. In real world, people are not likely to move around randomly, but rather move in a predictable fashion based on repeating behaviors. For example, if a node has visited a location several times before, it is likely that it will visit that location again. We attempt to capture the regularity in a node's movement and determine its future visits. A new concept of inter-visiting time is introduced in this chapter and it refers to the time duration between the two consecutive visits of a node to a region. In Geoopp, a node keeps track of its own visiting history in terms of inter-visiting time to all regions. We then utilize Chebyshev's inequality to study the visiting history and derive the visiting probability that the node will visit a region within a certain time duration.

The contact regularity embedded in human movement is exploited in addition to regular visits by Geoopp. It is stated in [75] that for geographic greedy routing to guarantee delivery it must be the case that for every node carrying a message, there exists a closer neighbor. That is wherever a message is in the network, there is always a next hop that gets the message closer to its ultimate destination. Hence in Geoopp, when a message carried by a node reaches a closer region, the availability of next relay candidates, i.e., contact opportunity, should be considered. For acquiring contact regularity to determine future contacts, a node maintains its own contact history in terms of contact availability per visiting to a region. We then employ Chebyshev's equality to study the contact history and obtain the contact probability that the node will have contact if visiting a region.

Geoopp is a multi-copy-based routing approach to overcome nondeterministic mobility and connectivity. Geoopp only uses a node's own visiting and contact information, and does not require any extra routing information to be exchanged across the network. Therefore, privacy is preserved and it is highly scalable as network size increases.

4.2 The Scheme of GEOOPP

In this section, we define Geoopp, a geocasting algorithm for large-scale opportunistic networks. We start with an overview of the algorithm. The calculation of the delivery probability function is then detailed. Finally we use an example to explain the protocol of Geoopp.

In Geoopp, we make the following assumptions. The network field is divided into cells/regions. It is assumed that each node in the network is able to determine its location in a cell (e.g., using GPS). It is assumed that nodes initially know nothing of its own mobility regularity, and learn it only online through the mechanisms of the protocol.

Geoopp forwards messages in greedy manner. A node is considered as a good relay if it can carry the message closer to its destination region. Future visited regions and contact chance in the visited regions can be predicted by observing the past movement history. We now describe the calculation of the delivery probability in general, as when two nodes encounter each other, the relay decision of carried messages is made by comparing their delivery probabilities for the messages.

We establish a delivery probability function to evaluate a node's ability/probability p_i to carry a message toward its destination considering the node visiting cell i. The delivery probability function p_i has three factors: $p_i(m)$ the geographic progress the node can make carrying the message to cell i measured by progress within radius metric; $p_i(v)$ probability the node visiting cell i within a certain duration; and $p_i(c)$ probability the node having contact opportunity when visiting cell i. Intuitively, we take into account not only i) how close a node can carry a message toward its destination by visiting the cell; but also ii) the probability that the node will visit the cell; and iii) the probability that the node will have contact when visiting the cell. In other words, if a node is able to visit a closer region and meet nodes inside the region with high probabilities, the node can be considered as a proper relay for the message.

We calculate the node's ability/probability p_i to carry a message toward its destination considering the node visiting every cell in the network field $(p_1, ..., p_i, ..., p_n)$.

The final delivery probability of the node for the message is derived as the maximum $max(p_i)$. It represents the highest potential the node possesses for carrying the message closer toward its destination.

Next, we elaborate the calculation of delivery probability function in detail.

4.2.1 Calculation of Delivery Probability Function

A node N's probability p_i to carry a message toward its destination considering N visiting cell i is calculated as the product of $p_i(m)$ the geographic progress N can make carrying the message to cell i, $p_i(v)$ the probability N visiting cell i within a certain duration; $p_i(c)$ the probability N having contact opportunity when visiting cell i. The calculation of p_i is shown in eq. (4.1).

$$p_i = p_i(m) \cdot p_i(v) \cdot p_i(c) \tag{4.1}$$

The derivations of $p_i(m)$, $p_i(v)$ and $p_i(c)$ are then detailed in the following.

4.2.1.1 Computation of $p_i(m)$

We develop the progress within radius metric (PWRM) to measure $p_i(m)$ the progress that node N can make carrying a message toward its destination by visiting cell i. The PWRM is illustrated in fig. 4.1. The current location of N is denoted by C while D is the center of the destination region of the message. I is the center of cell i.

$p_i(m)$ is computed as indicated in eq. (4.2) where $||C, D||_2$ is the Euclidean distance between the current location and the center of the destination region and $||I, D||_2$ is the Euclidean distance between the center of cell i and the center of the destination region. $||C, D||_2 = \sqrt{(x_C - x_D)^2 + (y_C - y_D)^2}$ and $||I, D||_2 = \sqrt{(x_I - x_D)^2 + (y_I - y_D)^2}$

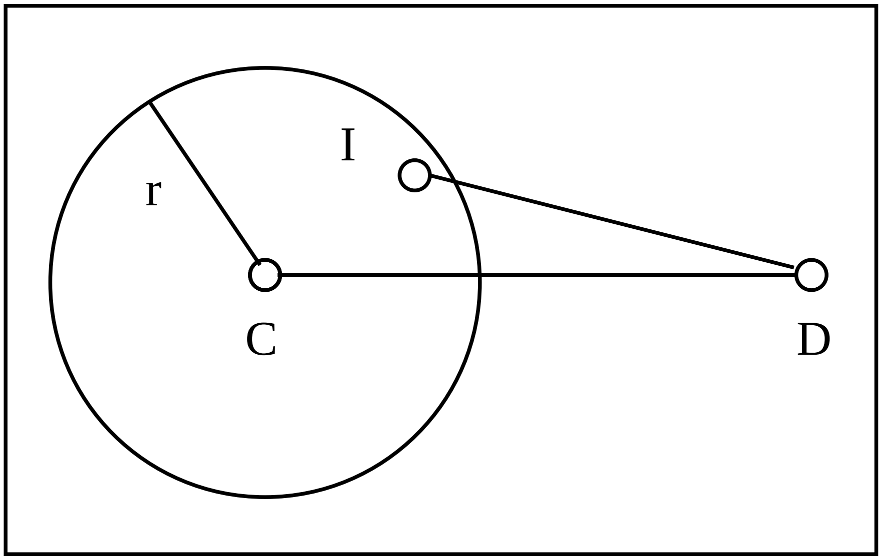

Figure 4.1. Progress Within Radius Metric.

The metric implies that the higher $p_i(m)$ is, the closer the message can be carried toward its destination by N visiting cell i.

$$\frac{||C,D||_2 - ||I,D||_2}{||C,D||_2} \tag{4.2}$$

4.2.1.2 Computation of $p_i(v)$

In Geoopp, a node N's visit to a cell is characterized by inter-visiting time to the cell. Inter-visiting time $X_i(v)$ is defined as the time duration between the two consecutive visiting of N to cell i. Node N collects its own inter-visiting time to all cells in its moving area. The study of inter-visiting time can divulge useful visit information. The intuitive idea is that when N will reach a cell is predicted by how often N visits the cell before. We employ Chebyshev's inequality to study the distribution of recorded inter-visiting time and derive $p_i(v)$ the probability that N visiting cell i before a message expires. N visits cell i before the message expires is desirable, as the message can be carried by N to a closer region and waiting for the next relay before it expires. If not, the message will get dropped before it reaches the closer region and N barely contributes to the delivery of the message.

Chebyshev's inequality characterizes the dispersion of data away from its mean. It bounds the probability that the data is far from its mean. Chebyshev's has no restriction on the shape of an underlying distribution. It only assumes that the underlying distribution has a mean and that the average size of the deviations away from this mean not be infinite. The calculation of $p_i(v)$ is displayed in eq. (4.3). Let random variable $X_i(v)$ denote the inter-visiting time of N to cell i. μ and σ^2 are the mean and variance of $X_i(v)$. t_{exp} is the message's expiration time while t_v is the most recent visiting time of N to cell i. μ and σ^2 can be approximated by the average and variance of N's historical inter-visiting times to cell i. If the historical data regarding cell i is empty, we define $p_i(v) = 0$. It indicates that N has never reached cell i in the history. Chebyshev's inequality requires that $t_{exp} - t_v - \mu > 0$, thus the equation is valid under the condition that $t_{exp} - t_v > \mu$. We define $p_i(v) = 0$ if $t_{exp} - t_v \leq \mu$. The value of $p_i(v)$ should be bounded by 1. $p_i(v)$ is actually a lower bound on the probability that the node visiting cell i before a message expires.

$$\begin{aligned} p_i(v) &= P\left\{X_i(v) \leq t_{exp} - t_v\right\} \\ &= 1 - P\left\{X_i(v) \geq t_{exp} - t_v\right\} \\ &= 1 - P\left\{X_i(v) - \mu \geq t_{exp} - t_v - \mu\right\} \\ &\geq 1 - \frac{\sigma^2}{(t_{exp} - t_v - \mu)^2} \end{aligned} \tag{4.3}$$

4.2.1.3 Computation of $p_i(c)$

When a node N carrying a message reaches a closer region, neighbors should be available to enable the next hop for the message toward its destination region. In Geoopp, N's contact in a cell is characterized by contact availability per visiting to the cell. Contact availability per visiting to cell i $X_i(c)$ is defined as the availability of contact opportunity N having during its one-time visit to cell i. $X_i(c) = 1$ if N has contact while $X_i(c) = 0$ if not. Node N collects its own contact availability per visiting to all cells in its moving area. The study of contact availability per visiting can disclose useful contact information.

Chebyshev's inequality is again utilized to calculate $p_i(c)$ shown in eq. (4.4). Let random variable $X_i(c)$ denote the contact availability per visiting to cell i. μ and σ^2 are the mean and variance of $X_i(c)$. μ and σ^2 can be approximated by the average and variance of N's historical contact availability per visiting to cell i. If the historical data regarding cell i is empty, we define $p_i(c) = 0$. It means that N has never had contact chance in cell i. Chebyshev's inequality requires that $1 - \mu > 0$, thus the equation is valid under the condition that $\mu < 1$. There is a possibility that $\mu = 1$ if N has contact chance every time it reaches cell i. We then define $p_i(c) = 1$ if $\mu = 1$. The value of $p_i(c)$ should be bounded by 1.

$$\begin{aligned} p_i(c) &= P\{X_i(c) \geq 1\} \\ &= P\{X_i(c) - \mu \geq 1 - \mu\} \\ &\leq 1 - \frac{\sigma^2}{(1-\mu)^2} \end{aligned} \tag{4.4}$$

$p_i(c)$ is actually an upper bound on the probability the node having contact opportunity when visiting cell i. It intuitively means that the node with higher average contact availability per visiting to cell i will be likely to encounter available neighbors during next visit to cell i.

$$P_d = max(p_1, ...p_i, ...p_n) \tag{4.5}$$

Upon receiving node N's probabilities to carry a message toward its destination considering N visiting all cells, the final delivery probability of N for the message is derived as the maximum shown in eq. (4.5) where n cells are divided in the network field. It indicates the best N can achieve for carrying the message closer toward its destination.

4.2.2 The Protocol of Geoopp

We now use an example to illustrate the protocol of Geoopp. The following steps show how Geoopp works when two nodes Alice and Bob carrying messages meet each other. For similarity, we only describe how Bob determines which messages to relay. Alice does so in similar ways.

1) After neighbor discovery, node Alice and Bob exchange all their messages destined to the current region, this refers to the message flooding within destination region.

2) Alice calculates delivery probabilities for all her carried messages. Alice then sends a list of headers of her carried messages along with her delivery probabilities to Bob. Upon receiving the header list, Bob removes the messages he carries already and then calculates his probabilities for the remaining messages. Bob requests the messages that he has higher delivery probabilities with than Alice. Alice sends messages to Bob per Bob's request.

4.3 Evaluation

The proposed Geoopp is evaluated using Opportunistic Network Environment (ONE) simulator [73]. We compare the performance of Geoopp against three other algorithms, Epidemic [38], Direct delivery and LOOP [8]. Epidemic which floods messages in the whole network shows an upper bound on the network connectivity. As a lower bound, Direct delivery delivers messages by visiting the destination directly. LOOP has the same purpose with Geoopp, aiming at delivering messages toward destination region. The delivery performance is measured over two sets of simulations and the impact of network size and network load is studied in this section.

4.3.1 Simulation Setup

Movement model plays an essential role in evaluating OppNets routing algorithm. The commonly used Random Waypoint assuming nodes move randomly is not suitable in real world. To evaluate Geoopp, we employ Working Day Movement Model [74] which presents the everyday life of average people that go to work in the morning, spend their day at work, and commute back to their homes at evenings. The model intuitively depicts the movement pattern of people, but it is also verified that it is heterogeneous in both time and space and is able to produces similar distributions of inter-contact times and contact durations as real user traces.

The simulation scenario is on a real city map with size $10km$ by $8km$. The whole map is divided into four districts, about half of the nodes live and work in the same district, and the other half commute between different districts. The simulation scenario constrains node movement to predefined paths and routes derived from real map data. 50% of the nodes take public transportation for commuting between offices and homes. Nodes attend meetings during office hours and join in activities after work.

Nodes use Bluetooth as wireless interface with radio range $10m$. There is a stationary sink in a randomly chosen location to accept messages. The simulation time for one run is 32 hours and the first 24 hours are given for warm-up since LOOP needs at least 24 hours for extracting mobility patterns. Messages are generated during 24 to 28 hours with exponential inter-arrival times. A generated message will get expired in 4 hours. In Geoopp, network map is divided into cells and we set one cell in the size of $500m$ by $500m$. Nodes are equipped with unlimited storage.

4.3.2 simulation Results

We are interested in delivery performance in terms of delivery ratio and traffic overhead. Delivery ratio is defined as the proportion of messages that have been delivered out of the total unique messages created. Traffic overhead is the total number of relays divided by number of nodes.

We first study the delivery performance over offered network sizes. The network size varies from 80 nodes to 320 nodes. Figs. 4.2 and 4.3 show the effect of offered number of nodes on message delivery rates and traffic overhead for Epidemic, Geoopp, LOOP and Direct delivery algorithms.

When more nodes are added into the network, they increase the network load while provide more message carriers. For instance, a non-existence route from source to destination might become available and thus enable the delivery of messages created by the source. Or the new route might be faster approaching and hence shorten the delivery delay. Fig. 4.2 indicate that the delivery rates increase with the growing of network size. Epidemic gives the maximum achievable delivery rate. Geoopp can deliver at roughly 80% of the maximum achievable delivery rate. At the same time, LOOP can deliver at around 60%. Both Geoopp and LOOP use geographic greedy

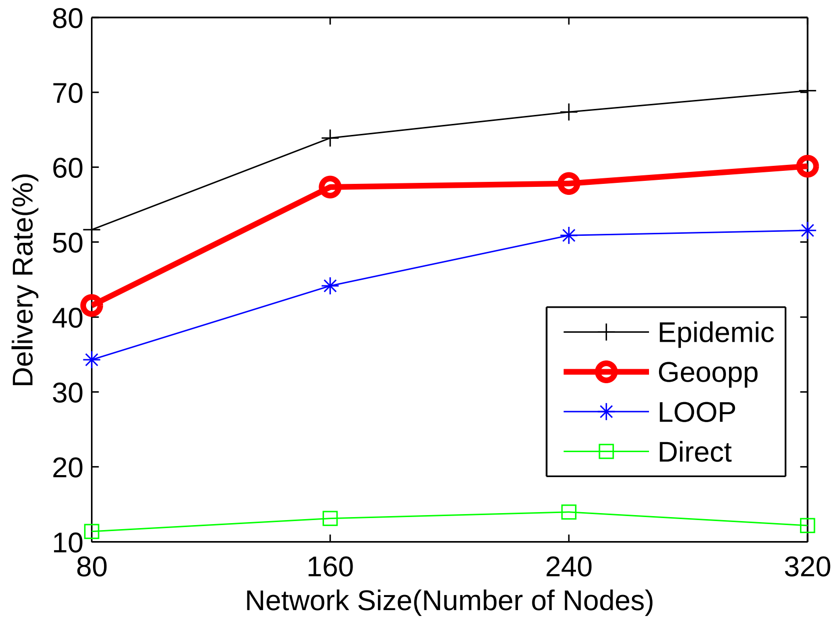

Figure 4.2. Impact of Network Size:Delivery Ratio.

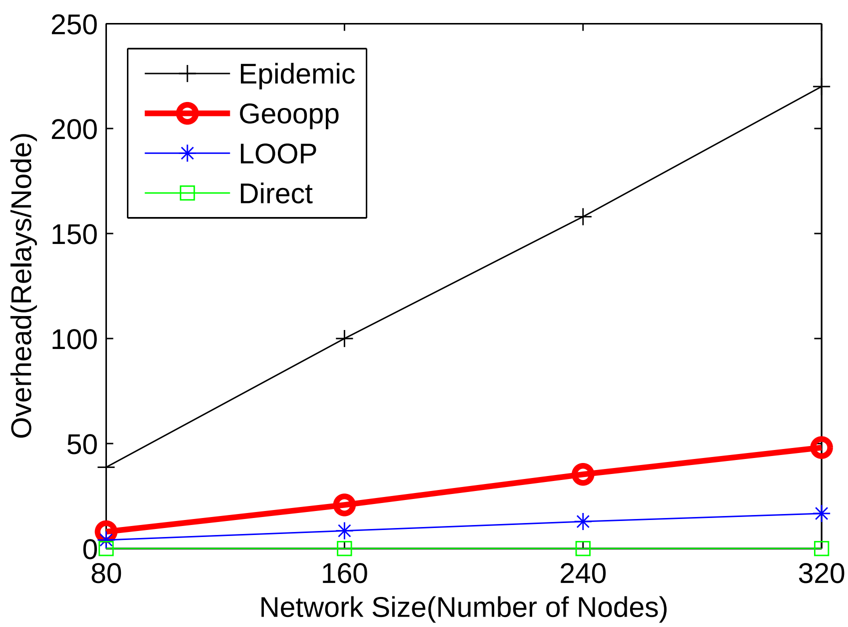

Figure 4.3. Impact of Network Size:Overhead.

routing. The results then indicate that Geoopp captures node' mobility more precisely and therefore gains better delivery rates.

When the capacity of message carriers enlarges, new routes from source to destination might emerge and hence more nodes are involved in delivering a message. Fig. 4.3 displays that overhead increments with more nodes present in the network. Epidemic exhibits the maximum consumable relays. When 320 nodes participate in the network, Geoopp consumes nearly 20% of the maximum consumable relays while LOOP consumes about 10%. Geoopp should be improved in further to reduce the overhead, such as using a more sophisticated method to generate the visiting and contact probability more exactly.

We then study the delivery performance over offered network loads. The inter-arrival times of message follow exponential distribution according to a specific mean varied from 20 minutes to 100 minutes. Figs. 4.4 and 4.5 show the effect of offered inter-arrival times on message delivery rates and traffic overhead for Epidemic, Geoopp, LOOP and Direct delivery algorithms. In fig. 4.4, when inter-arrival time is reduced, in another word messages are generated more frequently, delivery rates drop down as a result. Geoopp remains slightly lower delivery rate than Epidemic while LOOP shows a clear gap with Epidemic. Fig. 4.5 displays that overhead grows with the decreasing of inter-arrival time. Geoopp consumes much less relays than Epidemic but more than LOOP.

4.4 Summary

This chapter presents Geoopp a routing algorithm to support geocasting in large-scale OppNets. Geoopp adapts geographic greedy routing to the context of OppNets. A node's mobility is characterized by inter-visiting time and contact availability per visiting and Chebyshev' inequality is employed to capture embedded reg-

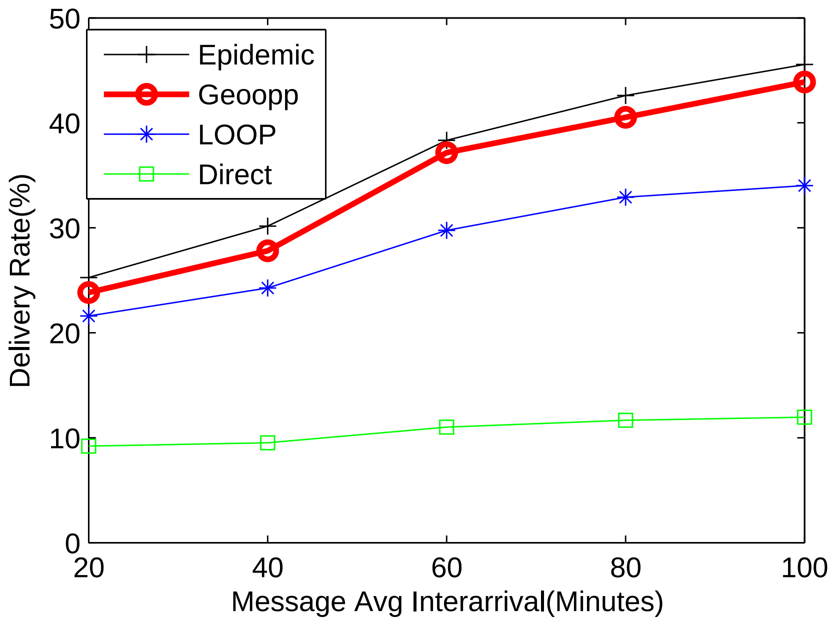

Figure 4.4. Impact of Network Load:Delivery Ratio.

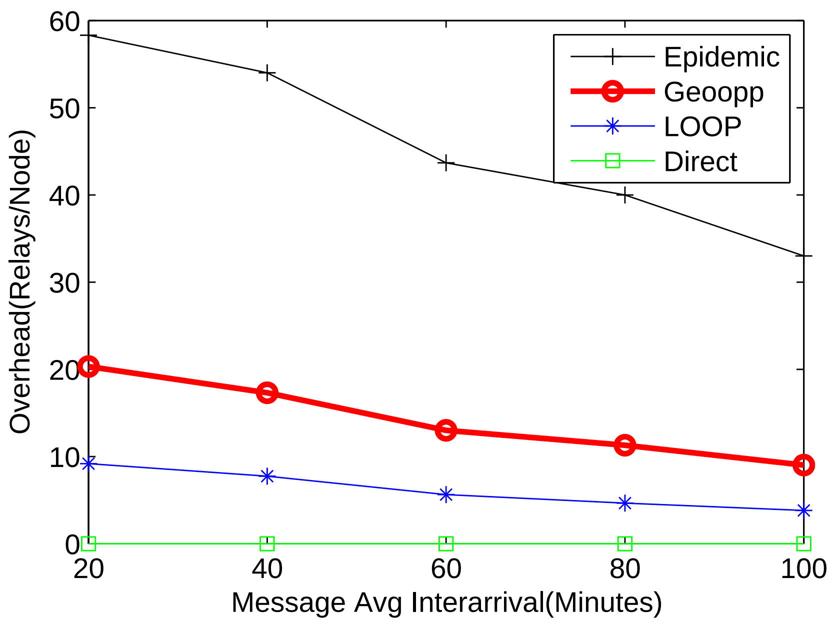

Figure 4.5. Impact of Network Load:Overhead.

ularity. The simulation results demonstrate that Geoopp consumes far less resources to attain a large extent of the maximum achievable delivery performance.

CHAPTER 5

Exploring Public Online Information For Routing In Opportunistic Networks

In the literature, extensive research works have been presented for node targeted routing for OppNets. Different from most of the work targeting at this problem, instead of inferring people's movement from context or social knowledge, this chapter studies utilizing online public information to learn node's mobility and facilitate the message forwarding for large-scale OppNets.

We first discuss the feasibility and benefits of utilizing public online information in message routing and forwarding in large-scale OppNets in Section 5.1. We then propose a *CAlendar based RouTing scheme* termed CART by disclosing information regarding human mobility from public calendars in Section 5.2. The performance of CART is evaluated in simulations and compared against well known routing schemes in Section 5.3. The evaluation results show that this scheme achieves message delivery with extremely low resource consumption. Finally we discuss the limitations of this scheme in Section 5.4.

5.1 Overview

In the literature a significant number of works have been proposed for node targeted routing in OppNets. In general they attempt to infer nodes' mobility and then choose the proper relays for messages by utilizing context information or social knowledge. However, the inferred mobility is on the basis of the local information and usually suffers from inaccuracy. For example, a message carrier relays a message to its neighbor because the neighbor has higher similarity score on social profile and then mobility pattern with the destination node. But there is still a probability that the relay will never meet with the destination. Therefore, most of the routing schemes are struggling with the tradeoff between increasing message delivery probability and lowering resource consumption. In order to have more chances the message is delivered to its destination in a short time, more message replicates are sprayed out to reach good relays, in turn the more network resources it wastes.

In this chapter, we study this message forwarding problem of OppNets in a novel direction. Instead of relying on knowledge acquired from other nodes directly during contacts, we seek nodes' mobility information from public sources such as the Internet. The mobility information can be acquired and trusted without inference if it is announced in public places. Additionally, nodes can acquire the mobility information of all other nodes that share it online. In other words, a global view of the network nodes' mobility can be obtained. Therefore, the chance to discover a path from the message source to the destination can be significantly increased by searching a sequence of nodes starting and ending with the source and destination nodes respectively and adjacent nodes in the node sequence have contact opportunity for at least once. In this case, all the intermediate nodes are determined and the message can be forwarded along these nodes on the path. Only nodes present in the path are involved in the forwarding. Therefore the message delivery and resource saving can both be mostly guaranteed if one path can be found. This can be particularly useful for mission oriented OppNets where certain performance guarantee is desired.

Following the above idea, the key issues we strive to address in the chapter are the following: i) How to obtain nodes' mobility knowledge from Internet; ii) How to discover the path from message source to destination given nodes' mobility; and iii) How to handle the message relay when the path is not available.

In this chapter, we will focus on using calendar as available public information in our design. Other public information can be utilized as well and will be considered in future studies. Calendars are commonly used to keep track of events in lives. People tend to publish their calendars online if they would like to share their schedules with the world. For example, a professor may announce his calendar on his personal web page for his students' convenience. These online calendars divulge useful mobility information about the calendar owners. Possibly we can find the node sequence from the source to destination by searching and mining the public calendars. However it is difficult to search and extract useful mobility information from calendars throughout the Internet, since there are millions of online calendars that usually have diverse formats and styles.

Google Calendar is widely used to maintain personal calendars for users. Calendars can be shared with the whole world if they are set to be public calendars, and also they can be embedded into web pages to be published online. Searching public Google calendars is feasible as the calendars content is stored in their server machines in specifically defined formats. The only drawback is that they removed the option of searching content in Google public calendars they used to provide. In this chapter, we will use Google calendar as an example for available public information. We emulate the calendar search function and mine the returned calendars to attempt to gain mobility knowledge to benefit the message forwarding.

With the calendar search, a path discovery algorithm is present to find the node sequence which links the source to destination. In the sequence, every pair of adjacent nodes attend at least one overlapped event, which enables them to perform message exchanges. The discovered path is applied to guide the message forwarding. At the same time, backup forwarding algorithms are necessary when the path is not available. Also we need to detect the condition where the path becomes ineffective in reality as people might change their schedules temporarily. In such case, the path needs to be renewed and applied to the message forwarding all over again.

The most essential feature that distinguishes CART from the rest works is that our scheme utilizes online public calendar information to learn the movement of people (and hence mobile devices) instead of inferring them from the context or social information. The main advantages exhibit in two folds. First, the learnt mobility from people's public calendar is much more trustworthy than that inferred from context information, and network traffic can be reduced by avoiding collecting context or social information for mobility inference. Second, public online calendars are accessible from any node in the network. In contrast, nodes can only collect context information of nodes that it has contact with. Although context information can be propagated, it will inevitably increase network load. More importantly, we can obtain global view of the network nodes' mobility and therefore the path from source to destination can be searched and determined.

5.2 Public Calendar Based Routing Scheme

In this section, we study how to utilize the public calendars to forward messages in OppNets. The first step is to obtain the public calendars. As Google Calendar does not support the public calendar search, we have to emulate the function of search engine. However, the Google calendar database is not accessible from the outside world, therefore we define our own public calendar database and provide the public calendar search function in this chapter. Table 5.1 shows the structure of a calendar in our database. In a calendar, the mobility of a person can be termed by attending events. Event $E(l, ts, te)$ is human activity which takes place at location l and lasts for a period time from ts to te.

Table 5.1. Calendar

Event	Location	Starting Time	Ending Time
E	E.l	E.ts	E.te

Event overlapping is a core concept in CART. There is an event $E1$ in a calendar with time slot starting from $E1.ts$ to $E1.te$ and in location $E1.l$ while another event $E2$ in another calendar with time slot starting from $E2.ts$ to $E2.te$ in location $E2.l$. We could say the two events overlap with each other if $min(E1.te, E2.te) - max(E1.ts, E2.ts) > 0$ and $E1.l == E2.l$ are satisfied at the same time. The overlapped time and location are calculated according to Eq. 5.1.

$$\begin{aligned} OverT = min(E1.te, E2.te) - max(E1.ts, E2.ts) \\ OverL = E1.l = E2.l \end{aligned} \tag{5.1}$$

Since all calendars are stored in the server machines, it is possible to implement global addressing. A unique 16-digit address is assigned to each public calendar in the database. A calendar is associated with a mobile device, and hence the calendar address is used to address the corresponding mobile device in our scheme. We can say that the mobility of a mobile device is described by its associated calendar.

Suppose a path is generated from a source mobile device to a destination mobile device, and in the path device a with address 0000.0000.0000.0001 is followed by device b with address 0000.0000.0000.0002, both share overlapped event with each other. Device a needs to discover the subsequent device b and relay the message to it during the overlapped event. According to the device discovery and connection establishment scheme proposed in [7], the device a can stuff device b's address 0000.0000.0000.0002 in its beacon during the overlapped event to announce its searching target. In this way, it avoids the frequent and costly communication with irrelevant mobile devices.

With the public calendar database, the emulated calendar search engine can search and return all the other calendars which have events that are overlapping with a given event E specified by a location and a time slot. With the emulated search engine, a path discovery algorithm is proposed to find the node sequence which links the source to destination. The naive idea behind the algorithm is illustrated in Fig. 5.1. The top vertical lines show the time elapse. Each row represents a part of a person's calendar where a line segment indicates an event for a certain period of time. The overlapped event expressed by bold lines between two calendars implies that the two persons have an opportunity to contact with each other during the overlapped time at overlapped location. The message can then flow from the source mobile device A to the destination mobile device D during the overlapped events.

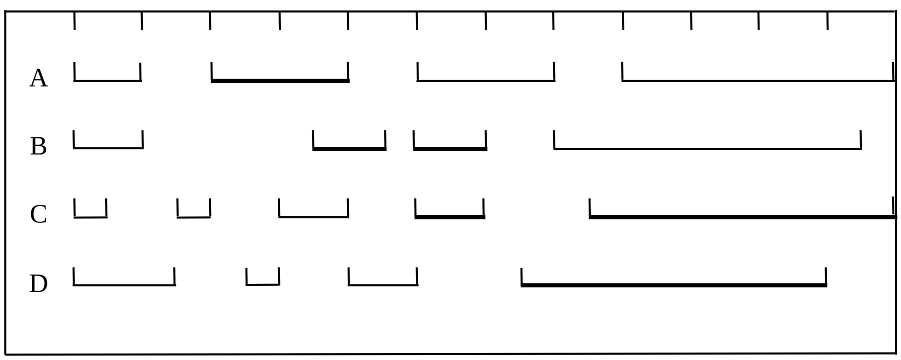

Figure 5.1. Illustration of Calendar Based Routing.

The path discovery algorithm shown in algorithm 1 is a modified breath search algorithm. The inputs are the source and destination mobile device addresses and the message to be delivered. The goal of the algorithm is to discover a path, a node sequence starting from source to destination where the adjacent two nodes share at least one pair of overlapped events. A data structure Path_Node shown in table 5.2

is defined to assist the path discovery algorithm. A list containing path_nodes will be constructed to record the path from source to destination. A path_node N stores related information of a mobile device in the path. $N.C$ is the address of the associated calendar and also the mobile device. $N.prevN$ refers to the previous path_node in the path. $N.prevN$ shares an overlapped event with the current device. $N.overT$ and $N.overL$ store the overlapped time and location information of the overlapped event between the two path_nodes.

Table 5.2. Path_Node N

Calendar Address	Previous Node	Overlapped Time and Location
N.C	N.prevN	N.overT, N.overL

An the beginning, a source path_node is created and added into the empty list. Subsequently we iterate every path_node in the list in order. Specifically for each path_node $currN$ in the list and each event $currE$ in the node' calendar, we employ the emulated search engine to search for all calendars that share overlapped events with the current event. It should be noticed that the search is time-sensitive, as only events that occur after the current time and before the message expiration time is worth considering. A path_node is created for each satisfactory calendar and then appended into the list meanwhile $currN$ is recorded as previous path_node and overlapped time and location information are stored in the newly created path_node. After the search of one path_node, we check whether the destination path_node is already in the list. The path is found if true, otherwise, the search continues. When all path_nodes in list have been iterated the search stops and the path discovery is announced to be failed.

Algorithm 3 Path Discovery Algorithm

Input:

Source Calendar Address, SC;
Destination Calendar Address, DC;
Message M, Creating Time, TTL of Message, $M.ct$, $M.ttl$;

Output:

Path from source to destination P

Source path_node SN, $SN.C = SC$;
Destination path_node DN, $DN.C = DC$;
P.add(SN)
for Each path_node $currN \in P$ **do**
    for Each event $currE \in currN.C$ **do**
        if $(currE.ts > currT) and (M.ct + M.ttl > currE.ts)$ **then**
            Search for any calendar C which contains at least one event E that shares overlapped time and location with $currE$;
            Create path_node N where $N.C = C$;
            $N.prevN = currN$;
            $N.overT$ = the overlapped time of $currE$ and E;
            $N.overL$ = the overlapped location of $currE$ and E;
            Append all Ns into P if they do not exist
        end if
    end for
    if $PcontainsDN$ **then**
        We found the path, break
    end if
end for
Failed to find the path, clear P;

After the path is found and stored in the list, we could retrieve the reversed path that is starting from destination to source as each path_node in the list stores the previous path_node. We do not detail this trivial part of algorithm in this chapter. With the retrieval, the path starting from source to destination is obtained. The source node will then start the message forwarding according to the path. Table 5.3 shows a typical path from source device 0000.0000.0000.0001 to destination device 0000.0000.0000.0003.

Table 5.3. Discovered Path

Calendar	0001	0002	0003
Previous Calendar	Null	0001	0002
Overlapped Time	Null	03.24.2012:11-13	03.24.2012:15-16
Overlapped Location	Null	location2	location3

The computation cost is not an issue for mobile devices as the path discovery can be performed at the server machines. The source mobile device sends path discovery request to the server and the discovered path will be returned by server to the requested device after the computation. Besides, there are some strategies that can be applied to reduce the computation complexity, such as that only calendars in the specific town are considered when source and destination nodes are from the same town.

In reality, the path might be unavailable as some people do not maintain calendars or they keep their calendars private or the link between the source and destination never exists. In such situation, it is necessary to have another forwarding plan such as Epidemic which disseminates the message to any encountered mobile devices. The path might become ineffective after a certain time, as people might cancel their schedule temporarily or the pair of adjacent nodes attend the overlapped event but did not discover each other and then fails to forward the message to the next relay. When the path is not available or becomes ineffective, any device that received the message will request a path from itself to the message destination from the server and follow the new obtained path to forward the message.

5.3 Evaluation

Simulations are conducted to evaluate the performance of the proposed scheme. 40 nodes are involved in a simulation map of size 1000*1000 square meters. A calendar is designed and constructed for each node. The movement of a node is driven by its calendar. The simulation time is set to be 6 hours and each node creates a message that is destined to a randomly picked destination node. Each simulation scenario is repeated 10 times for statistical confidence.

For the first set of simulations, CART is compared with several commonly known forwarding schemes in opportunistic networks: Epidemic [38], Prophet [41] and BubbleRap [53] described in section II. Epidemic is chosen as the backup forwarding plan for the CART. The parameters for Epidemic, Prophet and BubbleRap are set as recommended in their published papers. The delivery performance is evaluated with various message Time-To-Lives from 0.5 to 4 hours.

For CART, it is observed in the simulation that a part of source nodes can not find the path to the destination. But after several relays using Epidemic scheme the path from the relay node that currently holds the message to destination can always be discovered. The evaluation results seen in Figs. 5.2, 5.3, 5.4 and 5.5 show that the CART achieves high delivery ratio while consumes extremely low traffic and buffer cost compared with Epidemic, Prophet and BubbleRap. However, one drawback is the relatively high latency the reason of which is that our path discovery algorithm is a breath search on the events of calendars. In other words, it tends to involve less relays in the path but leads to higher delay for the message to reach its destination.

For the second set of simulations, the percentage of public calendars, the fraction of nodes who maintain public calendars, is controlled during the evaluation. The message Time-To-Live is set as infinite. The delivery performance of various public calendar percentages is displayed in Figs. 5.6, 5.7, 5.8 and 5.9. It is clearly shown

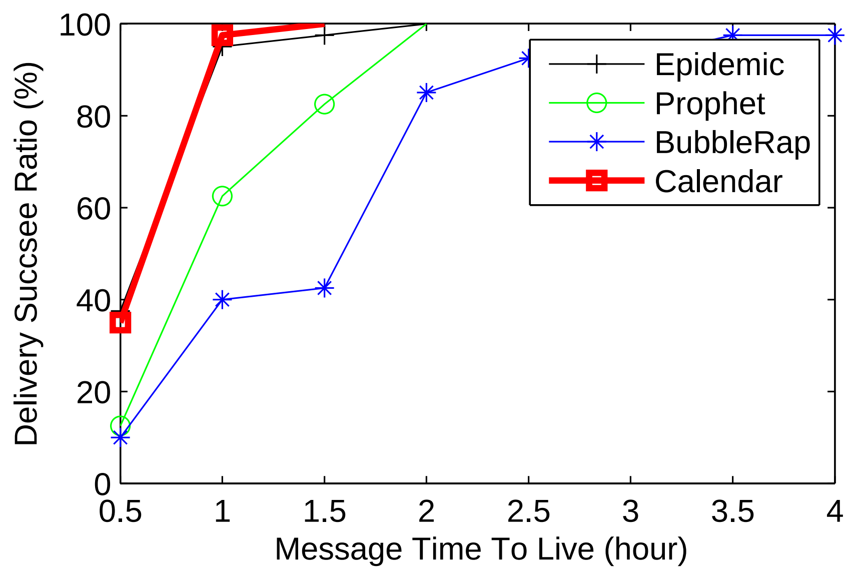

Figure 5.2. Effect of TTL - Delivery Rate.

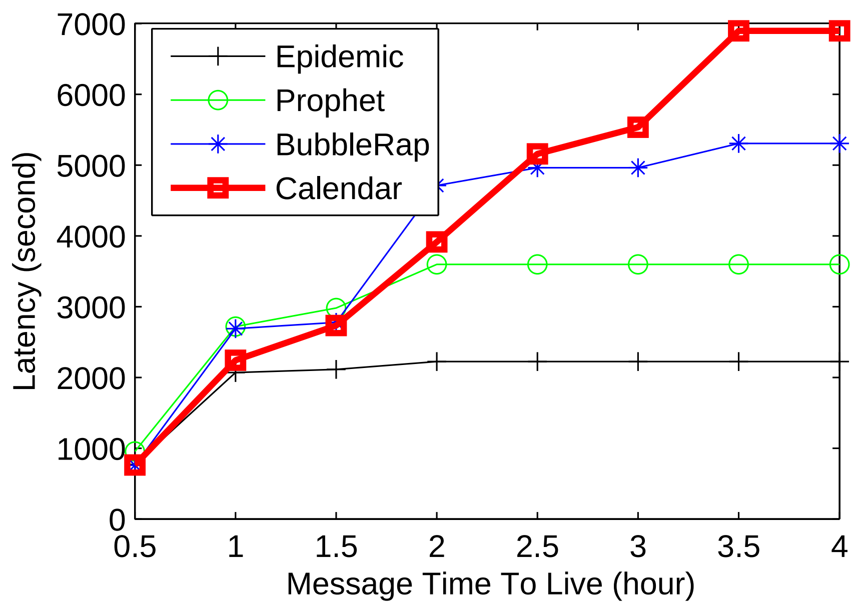

Figure 5.3. Effect of TTL - Latency.

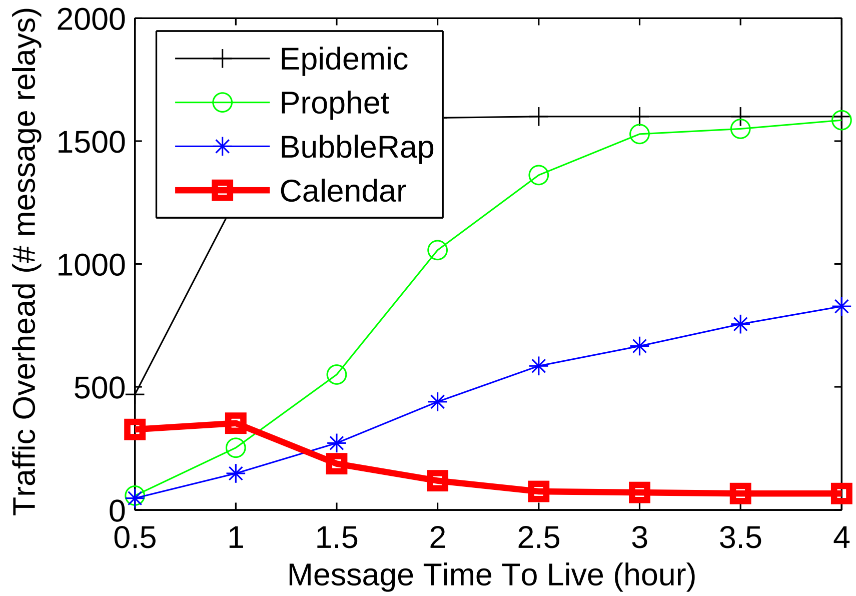

Figure 5.4. Effect of TTL - Traffic Overhead.

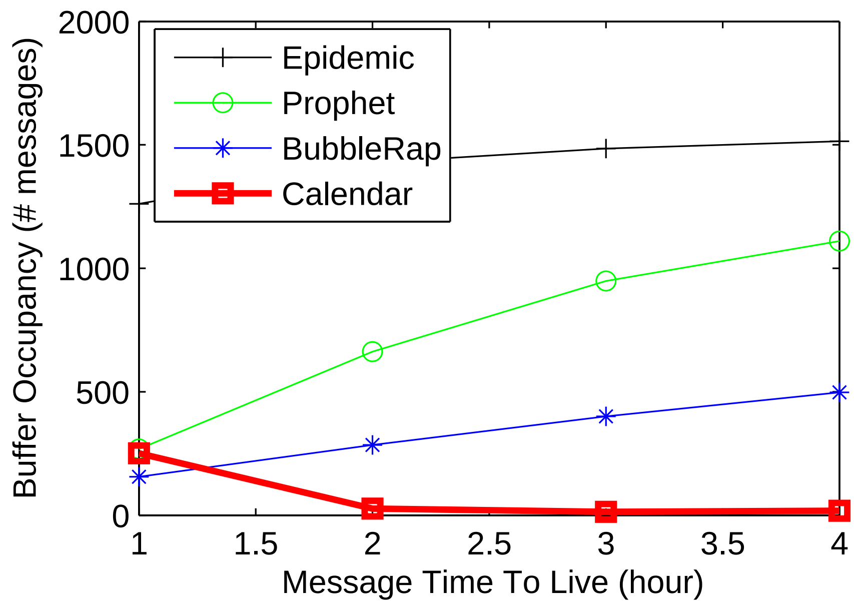

Figure 5.5. Effect of TTL - Buffer Occupancy.

that the performance continues to be improved with the increase of public calendar percentage in the network. As we have a backup forwarding plan when the path is not available, the delivery ratio remains high for all public calendar percentages.

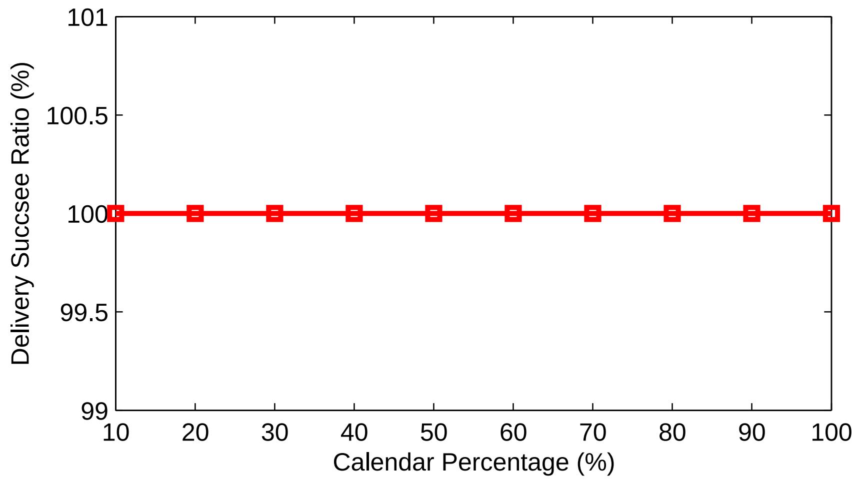

Figure 5.6. Effect of Calendar Percentage - Delivery Rate.

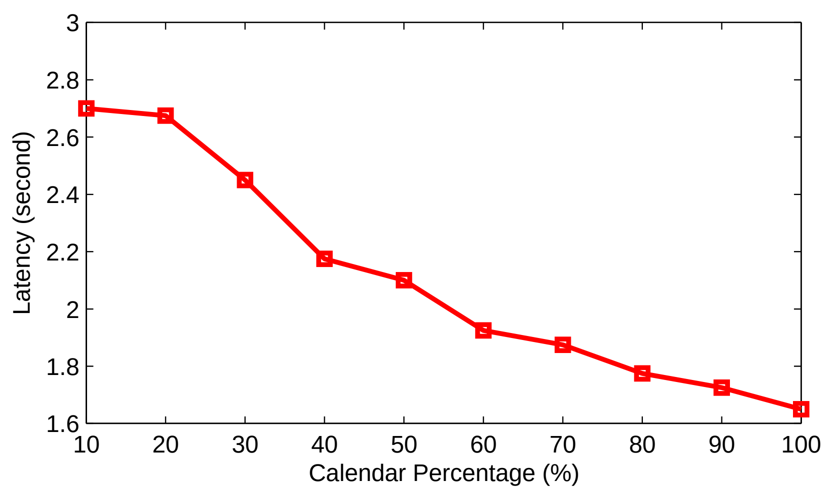

Figure 5.7. Effect of Calendar Percentage.

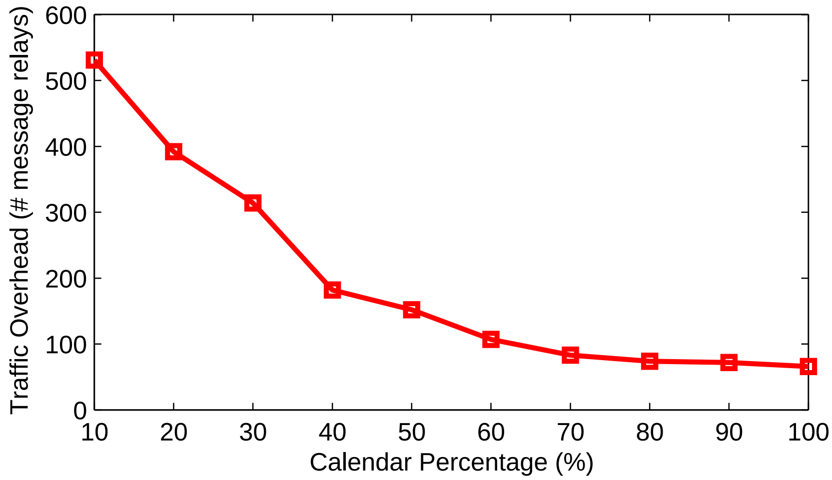

Figure 5.8. Effect of Calendar Percentage.

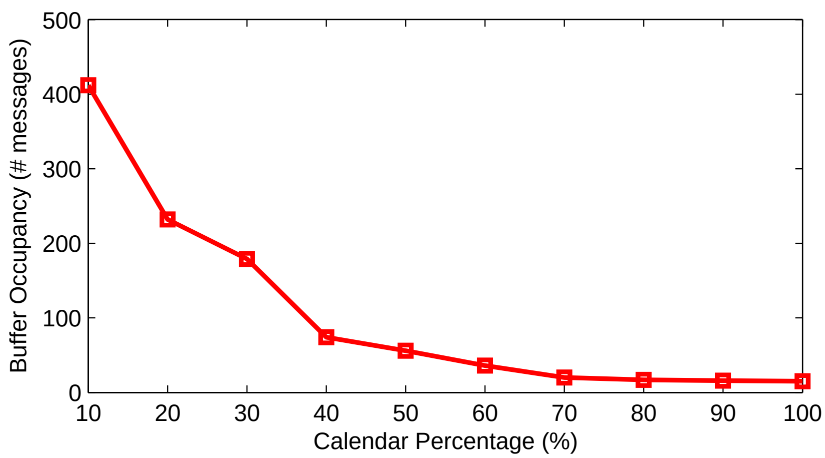

Figure 5.9. Effect of Calendar Percentage.

5.4 Discussion and Summary

This chapter explores the online public information to gain people's mobility knowledge and therefore facilitate the message forwarding in OppNets. A calendar based routing scheme is proposed to calculate a path, a node sequence where the pair of adjacent nodes share overlapped events, starting from source to destination on the basis of the online public calendar. The evaluation results show that CART achieves high delivery ratio while tremendously reduces the cost of communication and mobile device storage.

There are certain limitations for our scheme. First, the availability of public calendar search needs to be addressed. We are currently studying potential approaches to extract such information online by different means. Second, a study is required on the percentage of people who actually maintain public calendars and also the percentage of events scheduled in calendar of all activities for a person. These are essential factors that would largely affect the performance of CART. Third, evaluation on real calendar movement data would generate more convincing results. In the future, we plan to work on these limitations and improve the scheme further.

Despite the limitations of the CART in the current version, we believe it is a promising approach to facilitate the message forwarding in OppNets by exploring useful online public information including public calendars. Particulary it can be useful for networks with certain performance guarantee requirement. It can also be used as a complement to existing message forwarding schemes.

CHAPTER 6

Device Discovery and Connection Establishment

In this chapter, we propose a device discovery approach and connection establishment scheme for large-scale OppNets using Ad-Hoc Wi-Fi. This approach provides much longer communication range compared to commonly used Bluetooth. As a result, it enables a mobile device discover all available devices within a location and supports the stabler and longer connections among mobile devices inside a location.

We first introduce the approach in Section 6.1 and then present the design of the approach and the scheme in Section 6.2. We discuss the experiments and analysis in Section 6.3 and summarize this chapter in Section 6.4.

6.1 Overview

In OppNets content is forwarded between mobile devices in the absence of global connectivity by taking advantage of communication opportunities that arise in the course of user mobility [69]. This feature makes device discovery an essential technique in the field of OppNets. Any device in OppNets should have the ability to explore its neighborhood and discover available devices to enable further communication between the device and its discovered neighbors. A novel methodology that can provide a relatively long communication range is needed for supporting the device discovery in large-scale OppNets.

This chapter proposes a device discovery approach and a connection establishment scheme for OppNets using Ad-Hoc Wi-Fi technique for communication. 802.11 Wi-Fi provides communication range around 1-2 hundred meters [76] and allows devices to discover and connect with remote devices in a much larger range compared with Bluetooth technology. 802.11 Wi-Fi can work in Ad-Hoc mode without the presence of an access point nearby which facilitates peer-to-peer communication required in OppNets. In Ad-Hoc mode devices can communicate with each other in the same IBSS (Independent Basic Service Set) with the same SSID (Service Set Identifier). The first device actively establishes the IBSS and sends out beacons to broadcast SSID periodically. Other devices nearby can join the IBSS after receiving the beacons [77] and communication can begin.

In this chapter we study device discovery and connection establishment using Wi-Fi as the underlying communication scheme in OppNets. We employ beacon stuffing method for a device to announce its existence by broadcasting beacons stuffed with useful information in the field of SSID to remote devices. It allows peers to decode beacons to obtain the information, and then identify the device ID and the application running on there. Employing this approach, multiple devices can be discovered at the same time by one device through the reception of beacons from different peers without attempting to establish connection with. Furthermore as the device discovery operation is performed in the MAC layer only, application layer communication can be avoided entirely, which can significantly reduce overhead.

As scanning in device discovery is a significant source of energy consumption, we design a score-based scanning scheme to schedule device scanning operations at proper times with the objective to achieve high energy-efficiency and timely device discovery. This scheme is context aware: it can identify active events (and hence likely active devices) around the device quickly and increase scanning frequency accordingly; it can also recognize quiet environment and reduce the scanning frequency consequently.

Once the neighbor has been discovered, we present a connection establishment scheme between the two devices. The key feature of our scheme is that before connection established, both of the two devices switch to a new IBSS termed session IBSS identified by session SSID which has different format with the SSID a device sends out to broadcast its existence. Then devices that receive session SSID can identify the sender as in an ongoing session. This prevents other devices from attempting to create connection with the device that is already in a session. Consequently the ongoing session between two devices will not be disturbed.

We have carried out extensive experiments on HP IPAQ 910 mobile phone to evaluate our approach. The results show that our approach can achieve high energy efficiency and satisfactory peer connection speed and quality.

6.2 Device Discovery Approach and Connection Establishment Scheme

A Wi-Fi based mobile device shall periodically send out beacons to announce its very existence, scan available devices in its proximity and establish the connection finally if the neighbor is discovered. Our overall objective is to achieve high energy efficiency while providing device discovery probability demanded by the network and building satisfactory connection with the neighbor. Our approach can be divided into three components: beacon stuffed broadcasting, score-based scanning schedule and connection establishment scheme.

6.2.1 Beacon Stuffed Broadcasting

It is well known that 802.11 Wi-Fi can operate at two modes: infrastructure mode and ad-hoc mode. Obviously, for OppNets, ad-hoc mode is the key operation mode to enable direct mobile-mobile connection. To refresh, in ad-hoc mode, devices within one another's communication range communicate directly without going through an access point. These devices thus form an IBSS, which is identified by an SSID [78]. The first ad hoc station actively establishes an IBSS and starts sending beacons, which are needed to maintain synchronization among the stations. Other ad hoc stations can join the network after receiving a beacon and accepting the IBSS parameters found in the beacon frame.

In our approach, any device that is not connected with any remote device sends out beacons periodically to announce its existence. These beacons are stuffed with useful information that can be decoded by peers within their communication ranges. Specifically, an application ID can be stuffed in a beacon which facilitates a peer to identify running applications on the device without establishing a connection. As a device can attain more than one beacon during channel scanning, it is possible to discover multiple senders and corresponding applications in its surrounding neighborhood during one scan. Table 6.1 depicts the structure of a stuffed beacon frame provided by [79] with an 8 bytes application ID in the SSID field. Based on the stuffed beacon frame, we define our own specific SSID as shown in table 6.2.

Table 6.1. Beacon Frame

Beacon Interval	Time Stamp	SSID	Supported Rates	Capacity Info	Infomation Element	BSSID

Table 6.2. SSID

Device Identifier	Field Separator	Application Identifier

It is worth noting that there is another type of SSID defined in our approach: session SSID as shown in table 6.3 which identifies an ongoing connection between two devices in vicinity. In this case, a searching device ignores corresponding devices and excluding them from connection establishment attempts.

Table 6.3. Session SSID

Separator	Application Identifier	Separator	Part of Device Identifier	Separator	Part of Device Identifier

6.2.2 Score-Based Scanning Schedule

There are also two types of scanning for a Wi-Fi device to discover surrounding devices: passive scanning and active scanning. In our approach we choose passive scan due to its energy efficiency, as it performs only listening instead of transmitting. As a trade off, passive scan can incur higher latency in device discovery, but the latency can be tolerated in OppNets. Since scanning is a significant source of energy consumption for Wi-Fi enabled devices [80], there exists a tradeoff between scanning frequency, consequently timely device discovery and energy consumption.

In this chapter we control a device's scanning frequency based on the context where opportunistic contacts can potentially happen. Depending on the context, user contact period and corresponding scanning frequency can be classified into the following three categories.

Event: It is easy to find available devices in such scenario where some people stay together for a certain period, for example, students having courses in classroom, people commuting in subway, working in an office, two friends hanging together for a while etc. As we use Wi-Fi technique with large communication range, it benefits a device from covering almost all places in an event, and then discovering almost all available devices presented in the event, consequently indicating the beginning and ending of an event. Oppositely, it is not possible for Bluetooth device to monitor the whole event with just several meters range. In an event, the device discovery interval should be short enough to avoid wasting time to sleep and wait.

Occasional contact: The time duration for occasional contact is much less than it in events, such as two people are walking and entering each other's communication range for just several seconds. It is very hard to predict the happening of occasionally contact. Sometimes the short occasionally contact is harmful for device, since the pair of devices can go out of each other's communication range before beginning the actual data message transmission. In this chapter, we don't focus on contacts that happen occasionally.

Quiet time: no contact happens and the device should not scan frequently.

Hence, the overall objective of the score-based scanning schedule is to reveal the event occurrence and quiet time, adjust the scanning frequency accordingly, and then ensure the high frequency for not missing contact opportunities in events and maintain the much lower frequency in quiet time for energy-saving. The score-based scanning mechanism grades the current environment based on potential contact opportunities. Environment with higher score indicates higher contact opportunities and hence higher scan frequency should be used. The score of the current environment is calculated after a device search or a connect operation as depicted in eq. (6.1). The variables are subject to $\alpha_{rew} > \alpha_{pun}$, $DevN \geq 0$ and $ConN \geq 0$.

$$Score_{curr} = \begin{cases} min(Score_{max}, Score_{last} + \alpha_{rew} \times ConN^{\beta}), & DevN \geq 1, ConN > 1 \\ min(Score_{max}, Score_{last} + \alpha_{rew}^{\sim} \times Score_{last}), & DevN \geq 1, ConN = 1 \\ min(Score_{min}, Score_{last} - \alpha_{pun} \times ConN^{\beta}), & DevN = 0 \end{cases} \tag{6.1}$$

The next scanning interval is then calculated as in eq. (6.2).

$$NextInterval = Score_{max} - Score_{curr} + \gamma, \gamma \geq 0 \tag{6.2}$$

Where γ is a constant no less than zero. $Score_{curr}$ is the calculated score of current environment, and $NextInterval$ represents the next scanning interval determined by our schedule. $ConN$ is the number of consecutive actions. For example, $ConN = 3$ if a device discovers a neighboring device available three times consecutively or finds no device three times consecutively. Additionally, we count a successful connection as a successful device discovery. $DevN$ stands for the number of available devices discovered. Reward and punish factors are represented by α_{rew} and α_{pun} respectively. The reward factor should be larger than punish factor since it is desirable to recognize the occurrence of an event much more quickly than the environment which is inclined to be quiet. The exponential factor β is an important factor that could have impact on score. It forces the adjustment to be exponentially enlarged when more consecutive actions happen and then be able to indicate events occur or quiet time quickly. We limit the range of the score to be within $(Score_{max}, Score_{min})$, so the interval range can be controlled. And γ is a parameter to control the minimum scanning interval.

There are two key advantages employing the above scheme in determining the scan frequency. Firstly, event can be recognized faster. When an available device has been found, the score of the current environment will become higher; and the score grows exponentially when successful device discovery occurs consecutively. This makes the mechanism adaptive to the current context and helps to identify events rapidly. Secondly, it can avoid certain harmful contacts. During quiet time, the larger interval helps to reduce the amount of short contacts that happen occasionally. Since we treat the first discovery in consecution ($ConN = 1$) specially, the interval will not even decrease for one single contact under low score environment. We do not utilize the current discovered available device number to grade the score here, as discovered device number may send wrong signals. For instance, a quick moving device may be able to find numerous devices for one time, but all of the neighbors may disappear shortly.

6.2.3 Connection Establishment Scheme

Any device broadcasts beacons stuffed with application ID and device ID in the field of SSID acting as a server. At the same time, it also acts as a client scanning available devices in its neighborhood according to the Score-based scanning schedule presented in the previous section. The client will choose the connection candidate, send out session request, and then switch to the session IBSS. The server receives session request from client and switch to the same session IBSS. When the client and server are in the same session IBSS, they begin to broadcast invitations to get each other's IP information, the connection will begin if they both receive replies. The connection establishment scheme can be described in algorithms 1 and 2 in detail. Note that both mobile devices disable their performances of scanning after they discover neighbor or receive session request from neighbor. When the communication finishes or the connection fails new scanning interval is obtained in accordance with the score-based scanning schedule. The scanning will be performed when the new interval is satisfied by time.

Traditionally after selecting the neighbor, the client can send out invitation to the server, and start the communication in server's IBSS directly. But we choose to create a new session SSID, and both client and server switch to the session IBSS. An advantage to this technique is that it prevents other devices from attempting to

Algorithm 4 Connection Establishment Process: Server Side

```
Broadcast stuffed beacons periodically and Listen to session request from client;
if session request received then
    Wait for a while;
    if find session SSID in neighborhood then
        Switch to session IBSS;
        Broadcast invitation message to get client's IP and Port number;
        if neighbor replies then
            Connection established;
        else
            Begin to listen to session request again;
        end if
    else
        Begin to listen to session request again;
    end if
else
    Keep broadcasting and listening;
end if
```

Algorithm 5 Connection Establishment Process: Client Side

```
Scan device in its neighborhood;
if find no available device then
    Sleep for a certain period according to the score-based scanning schedule;
else
    Switch to the discovered neighbor's IBSS;
    Send session quest with SSID to the server;
    Switch to the session IBSS;
    Broadcast invitation message to get server's IP and Port number;
    if neighbor replies then
        Connection established;
    else
        Begin to scan device again;
    end if
end if
```

create connection with the device that is already in a session. As session SSID has different format with the SSID a device sends out to broadcast its existence. Then devices that receive session SSID can identify the sender as in an ongoing session. Consequently the ongoing session between two devices will not be disturbed.

6.3 Experiments and Analysis

Our approach is implemented on HP IPAQ 910 mobile phone equipped with Wi-Fi and Bluetooth interfaces. In this section, its performance is evaluated and discussed by means of extensive experiments carried out on a group of mobile phones. We first present results on the score-based scanning schedule and then the connection establishment scheme.

6.3.1 Score-Based Scanning Schedule

Firstly, we study our schedule on two cases: event case and quiet time case to show how it performs for different situations in detail. Secondly, a large set of experiments is carried out to evaluate the score-based scanning schedule on statistical results. For case 1 and 2, we assign the parameters as follows: $Score_{min} = 0$, $Score_{max} = 60$, $\gamma = 0$, $\alpha_{rew} = 3$, $\alpha_{pun} = 0.5$, $\beta = 2$ and $\alpha_{rew}^{\sim} = 1$.

Case 1: We suppose the initial score is 0, and then we record how our schedule behaves in a 10 minutes-event. During the event there are two devices that run our schedule and stay in the neighborhood, the interval variation for one device has been traced and shown in Table 6.4. It clearly indicates that our schedule can identify event occur rapidly and adjust the scanning frequency accordingly.

Here the operation types are described here. 1. Get session request from neighbor and communicate for a while; 2. Send session request to neighbor and commu-

Table 6.4. An Example of Interval Variation With One Neighbor In Vicinity

Operation type	1	4	2	3	5	1	4	2
Communication Time(sec)	88		148			119		91
Next Scan Interval(sec)	60	48	21	21.5	21.5	0	0	0

nicate for a while; 3. Scanning and find nothing; 4. Scanning and find at least one device; 5. Get session request before the scanning starts.

Case 2: The interval varies as follows in Table 6.5 when a device is alone by itself for most of the time in a period of 10 minutes, and an occasional 10 seconds-contact happens one time. We can see that our schedule can achieve quite low scanning frequency in quiet time and the one single occasional contact would not affect the scanning frequency very much.

Table 6.5. An Example of Interval Variation Without Neighbor In Vicinity

Operation type	3	3	3	3	3	3	3	3	3	1	3	3	3	3
Communication Time(sec)										10				
Interval(sec)	0	0.5	2.5	7	15	27.5	45.5	60	60	60	60	60	60	60

After those two case studies, we conduct a larger set of experiments to compare our schedule with the fixed interval schedules. We assign the following parameters to our schedule: $Score_{min} = 2$, $Score_{max} = 60$, $\gamma = 0$, $\alpha_{rew} = 3$, $\alpha_{pun} = 0.5$, $\beta = 2$ and $\alpha^{\sim}_{rew} = 1$.

The three fixed interval schedules compared are outlined below in equation eq. (6.3), (6.4) and (6.5). For schedule A, the next interval will be fixed to 5 seconds if the device discovers one or more neighbors in this scanning, otherwise it will be set to a random number within (2, 8). The scheme of schedule B and C are similar to schedule A, but with different intervals.

Fixed interval schedule A defined in eq. (6.3):

$$NextInterval = \begin{cases} 5, & found\ device \\ (5-3, 5+3), & no\ device \end{cases} \tag{6.3}$$

Fixed interval schedule B defined in eq. (6.4):

$$NextInterval = \begin{cases} 10, & found\ device \\ (10-3, 10+3), & no\ device \end{cases} \tag{6.4}$$

Fixed interval schedule C defined in eq. (6.5):

$$NextInterval = \begin{cases} 30, & found\ device \\ (30-3, 30+3), & no\ device \end{cases} \tag{6.5}$$

The connection establishment operation is blocked in the experiment setup in order to measure the scanning frequency fairly, as the connection establishment can take different durations for different situations. Event is defined as the scenario where two or more devices stay in the communication range for a period of time. The event starts when at least one device comes into neighborhood and ends when all neighbors are gone out of communication range. During the event, some devices may go out of range occasionally, but devices can discover available neighbors for most of the time. And quiet time refers to the scenario where only one device stays in the communication range for a while with occasional contacts sometimes.

The whole experiment set contains 40 single experiments in total. Each experiment involves a group of devices, each running a certain schedule. The single experiment lasts for 10 to 60 minutes. During the period for one single experiment, events and occasional contacts occur randomly. In this case, it is not fair to collect and compare absolute scanning amount as even in the same experiment the context for each device is different. For example, some devices are having event occurence meanwhile the others are going through quiet time, and then the event and quiet time duration can be different for each device. Thus we measure the scanning frequency instead. The scanning frequency in event, quiet time and total is calculated as the scanning times divided by the duration of event, quiet time and total defined in eq. (6.6).

$$ScanningFrequencyInEvent(QuietTime, Total) = \frac{scanning\ times}{event(quiettime, total)\ duration} \tag{6.6}$$

We calculate the three types of scanning frequency for every schedule in every experiment, and then measure the average scanning frequency and standard deviations for each schedule throughout all experiments. The results are depicted in Fig. 6.1.

From Fig. 6.1, we can see that overall our approach has much higher scanning frequency during events and quite lower frequency in quiet time. In detail, the high deviation about scanning frequency in event indicates that our approach is more adaptive to the current environment. Since sometimes all the devices are connected with each other in an event, we cannot find any device, and then our approach can decrease the scanning frequency temporarily. The total average frequency and deviation depend on the duration ratios of event and quiet time in the whole period. With longer quiet time our approach can achieve very low scanning frequency without sacrificing contact opportunities in events.

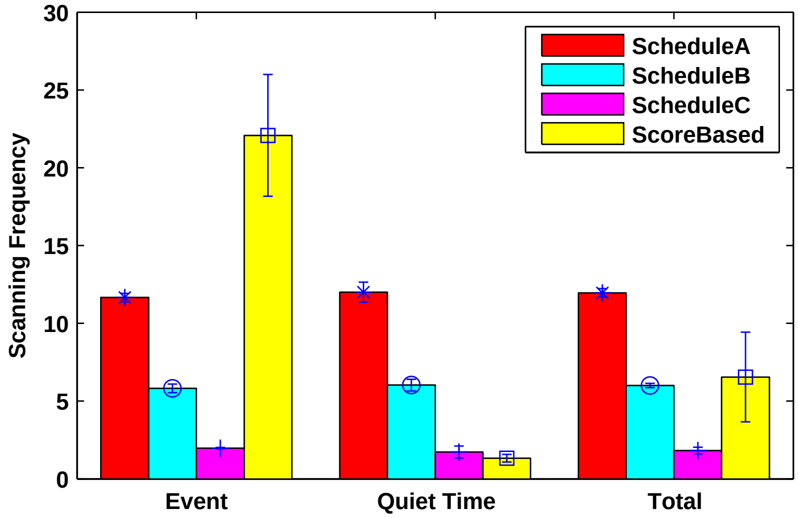

Figure 6.1. Comparison of Scanning Schedules.

6.3.2 Connection Establishment

The time consumed for a typical process of connection establishment is shown in table 6.6 and 6.7.

Table 6.6. An Example of Connection Establishment:Server Side

Event	Time Consumed (sec)
Get session request from client and switch to session IBSS	12
Broadcast invitation request till get replied	46

Table 6.7. An example of Connection Establishment:Client Side

Event	Time Consumed (sec)
Find the neighbor, then switch to neighbor's IBSS	7
Broadcast session request	2
Switch to session IBSS	6
Broadcast invitation to locate server till get replied	58

In broadcasting phase, the local device tries to send out invite request or invitation several times with 15 seconds interval. If the local device cannot get reply from remote device within a certain period, the broadcasting will be stopped and the connection establishment will be seen as failed. It is also worth noting that any ongoing connection will not be disturbed by the third device in the neighborhood.

The durations for all of these operations vary with different conditions. Experiments have been conducted to measure the average time taken for the important operations in connection establishment process. These experiments involve multiple devices, and last for 10 hours with hundreds of connections established in total.

The numbers on the x-axis in Fig. 6.2 denote different operations, as detailed below: Server side: 1. Switch to session IBSS; 2. Broadcast invitation request till get replied; 3. Total Time; Client side: 4. Switch to neighbor's IBSS; 5. Broadcast session request; 6. Switch to session IBSS; 7. From discovering neighbor to switch to session IBSS; 8. Broadcast invitation to locate server till get replied; 9. Total Time.

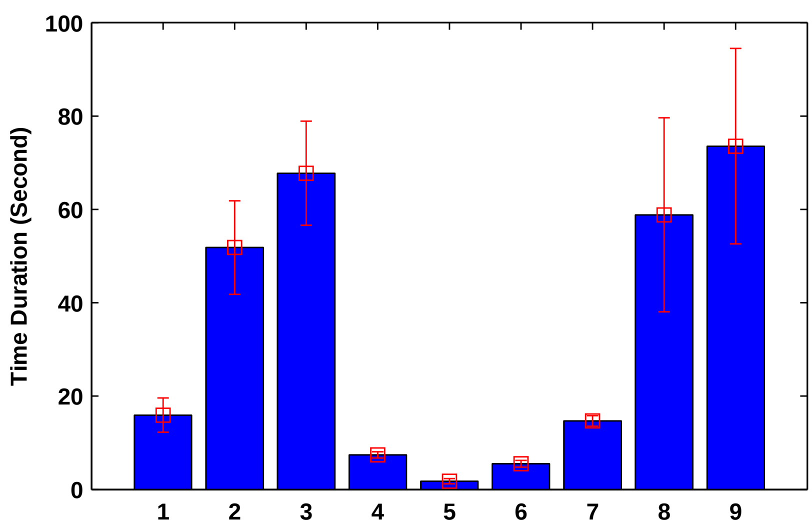

Figure 6.2. Connection Establishment Process (Measure Server and Client Separately).

The numbers on the x-axis of Fig. 6.3 denote different operations as described below: 1. From beginning of connection to switching to session IBSS; 2. Broadcast invitation/invite request till get replied; 3. Total time.

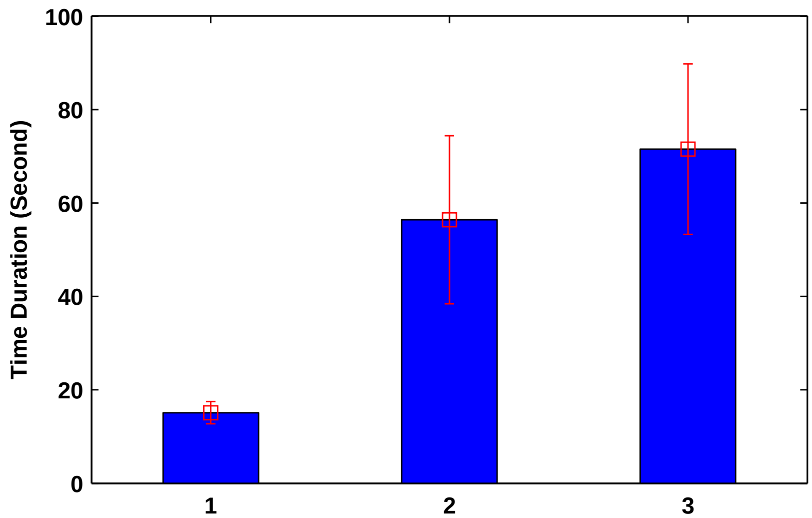

Figure 6.3. Connection Establishment Process (Measure Server and Client Together).

As shown in Fig. 6.2 and 6.3 the time taken for devices to switch to session IBSS is quite constant. In contrary, it is rather dynamic to broadcast invitations to neighbor in the session IBSS. Since the contacts occur randomly in our experiments, connection establishment might fail for devices becoming out of each other's communication range before connection has been built successfully. According to our approach, devices try to broadcast invitations at most 5 times with interval of 15 seconds if no reply is received. Hence the longest period for broadcasting is 75 seconds, after which the connection establishment is announced to be failed.

6.4 Summary

We proposed a device discovery and connection establishment approach using Ad-Hoc Wi-Fi technique for OppNets. The approach consists of three components: beacon stuffed broadcasting, score-based scanning schedule and connection establishment scheme. The score-based scanning schedule achieves quite higher scanning frequency in events and rather lower frequency in quiet time compared to the fixed interval schedules. The connection establishment scheme creates connection between two devices with average duration of 65 seconds.

CHAPTER 7

Conclusions and Future Works

OppNets have attracted a lot of attention from the wireless and mobile network research community, while little has been paid to the OppNets in large-scale scenario. This dissertation proposes a two-layer architecture for the location centered large-scale opportunistic network. The base layer handles connection which provides underlying technique to enable the device discovery and connection establishment. The upper layer deals with routing which delivers messages to targeted nodes/locations. In the following, we briefly draw conclusions and correspondingly shed a light on future works.

(1) Both LOOP and Geoopp adapt geographic greedy routing and fit it in the context of OppNets. Geographic routing is naturally the choice for routing messages toward destined locations. In the adapted geographic greedy routing, it is assumed that each node knows its future positions. The relay decision is made based on its current and future positions. A message will be forwarded to a node, if it can carry the message and make the most progress on closeness toward destination. The adapted geographic routing is well suited in large-scale OppNets.

(2) LOOP formulates the mobility pattern mining as a multi-label classification problem and constructs a Bayes' predictive model to explore a node's mobility history and learn its mobility pattern. This mobility pattern will then be used to predict the node's future mobility. It is effective to learn mobility patterns and employ them in message routing in large-scale OppNets. LOOP shows evident performance gain over existing routing schemes for OppNets. LOOPC further improves LOOP

through the acquisition of mobility information from user calendars. By identifying special calendar events which disclose accurate mobility with highly possible contact opportunity, LOOPC can improve LOOP through delivery performance enhancement and resources consumption reduction.

(3) Geoopp introduces a novel approach for characterizing node mobility. It characterizes a node's mobility by inter-visiting time and contact availability per visiting to capture the regular visits and contacts in a specific region. Chebyshev's inequality is employed to compute the probabilities that a node visiting a region and having contact inside. It is simple yet efficient in facilitating message routing in large-scale OppNets. We demonstrate that Geoopp can attain 80% of the maximum achievable delivery rate at a cost of 20% of the maximum consumable relays.

(4) CART studies utilizing online public information to learn node's mobility. By exploring public calendars CART can discover a message path at source node. By forwarding messages along the determined path, CART achieves message delivery at remarkably low cost. Despite the limitations of the CART in the current version, we believe it is a promising approach to facilitate the message routing in large-scale OppNets by exploiting useful online public information. CART can also be used as a complement to existing routing schemes.

(5) A device discovery approach and connection establishment scheme using ad-hoc Wi-Fi is proposed in this dissertation. The approach demonstrates that the equipped Wi-Fi interface can be employed to support the device discovery within locations. Compared to Bluetooth, it provides much longer communication range. This approach is further enhanced with energy efficiency using the score based scanning schedule.

Inspired by the works in this dissertation, there are some future works discussed as following:

(1) For routing scheme LOOP, there are some future works that can be considered. First, the performance of LOOP can be further improved by learning more accurate mobility patterns. Bayes' predictive model is used in the current scheme, while some other data mining approach can also be employed in learning the mobility patterns, such as using association rule to learn the relevance among visited locations as the occurrence of one location might imply the occurrence of another one. Second, the current analytical model does not include the time regularity of LOOP. We can incorporate it into the model and then consider delay factor which helps differentiate the better relays when measuring closeness improvement. As a result, we can better study the performance of LOOP theoretically. Third, we present flexible comparison strategy to make relay decision in LOOP. In our simulation study, we found out that the different settings of the key parameter threshold in flexible comparison strategy lead to dramatically different results. The efficiency of LOOP can be further enhanced if we can adaptively schedule the threshold to fit the environment. Forth, more convincing results can be received if we evaluate the performance of LOOP over real trace data.

(2) CART searches the first available path from source to destination. We would like to design efficient algorithm to produce the optimal path with minimum price. Besides, we want to explore more public online information to aid the routing in OppNets, such as the online social information from the populous social web sites.

(3) Large-scale OppNets using populous locations as backbone to form the network are more feasible than existing OppNets relying on transient contacts. Envision more potential applications in large-scale OppNets and identify the characteristics and challenges in the targeted environment are amongst the future works.

REFERENCES

[1] M. Conti and M. Kumar, "Opportunities in opportunistic computing," *Computer*, vol. 43, no. 1, pp. 42–50, 2010.

[2] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[3] A. Lindgren and P. Hui, "The quest for a killer app for opportunistic and delay tolerant networks," in *Proceedings of the 4th ACM workshop on Challenged networks*. ACM, 2009, pp. 59–66.

[4] A. Garyfalos and K. Almeroth, "Coupons: A multilevel incentive scheme for information dissemination in mobile networks," *IEEE Transactions on Mobile Computing*, pp. 792–804, 2008.

[5] B. Zhang, J. Teng, X. Bai, Z. Yang, and D. Xuan, "P3-coupon: A probabilistic system for prompt and privacy-preserving electronic coupon distribution," in *Proceedings of PerCOM*, 2011, pp. 93–101.

[6] A. Pietiläinen, "Opportunistic mobile social networks at work," Tech. Rep., 2010.

[7] S. Lu, S. Shere, Y. Liu, and Y. Liu, "Device discovery and connection establishment approach using ad-hoc wi-fi for opportunistic networks," in *Proceedings of PERCOM Workshop on MP2P*, 2011, pp. 461–466.

[8] S. Lu, Y. Liu, Y. Liu, and M. Kumar, "Loop: A location based routing scheme for opportunistic networks," in *Proceedings of the 9th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2012, pp. 118–126.

[9] S. Lu, Y. Liu, and M. Kumar, "Using public online information to facilitate message forwarding in opportunistic networks," in *Proceedings of the first ACM*

international workshop on Mission-oriented wireless sensor networking, 2012, pp. 45–50.

[10] M. Gonzalez, C. Hidalgo, and A. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[11] N. Aschenbruck, A. Munjal, and T. Camp, "Trace-based mobility modeling for multi-hop wireless networks," *Computer Communications*, 2010.

[12] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 6, pp. 606–620, 2007.

[13] T. Karagiannis, J. Le Boudec, and M. Vojnovic, "Power law and exponential decay of intercontact times between mobile devices," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 10, pp. 1377–1390, 2010.

[14] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 3, pp. 630–643, 2011.

[15] T. Azevedo, R. Bezerra, C. Campos, and L. de Moraes, "An analysis of human mobility using real traces," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, 2009, pp. 1–6.

[16] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[17] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[18] P. Nain, D. Towsley, B. Liu, and Z. Liu, "Properties of random direction models," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, 2005, pp. 1897–1907.

[19] B. Liang and Z. Haas, "Predictive distance-based mobility management for pcs networks," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 1999, pp. 1377–1384.

[20] F. Bai, N. Sadagopan, and A. Helmy, "Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, 2003, pp. 825–835.

[21] C. Papageorgiou, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, "An obstacle-aware human mobility model for ad hoc networks," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on*, 2009, pp. 1–9.

[22] J. Ghosh, S. Philip, and C. Qiao, "Sociological orbit aware location approximation and routing (solar) in manet," *Ad Hoc Networks*, vol. 5, no. 2, pp. 189–209, 2007.

[23] A. Mei and J. Stefa, "Swim: a simple model to generate small mobile worlds," in *INFOCOM 2009, IEEE*, 2009, pp. 2106–2113.

[24] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 758–766.

[25] L. Hu and L. Dittmann, "Heterogeneous community-based mobility model for human opportunistic network," in *2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2009, pp. 465–470.

[26] F. Morlot, S. Elayoubi, and F. Baccelli, "An interaction-based mobility model for dynamic hot spot analysis," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1–9.

[27] E. Hyytia, P. Lassila, and J. Virtamo, “A markovian waypoint mobility model with application to hotspot modeling,” in *Communications, 2006. ICC’06. IEEE International Conference on*, vol. 3, 2006, pp. 979–986.

[28] Q. Zheng, X. Hong, and J. Liu, “An agenda based mobility model21,” in *Proceedings of the 39th annual Symposium on Simulation*, 2006, pp. 188–195.

[29] F. Ekman, A. Keränen, J. Karvo, and J. Ott, “Working day movement model,” in *Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, 2008, pp. 33–40.

[30] M. Musolesi and C. Mascolo, “Designing mobility models based on social network theory,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 3, pp. 59–70, 2007.

[31] C. Boldrini, M. Conti, and A. Passarella, “Users mobility models for opportunistic networks: the role of physical locations,” *Proc. of IEEE WRECOM*, 2007.

[32] D. Fischer, K. Herrmann, and K. Rothermel, “Gesomo-a general social mobility model for delay tolerant networks,” in *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, 2010, pp. 99–108.

[33] V. Borrel, F. Legendre, M. Dias de Amorim, and S. Fdida, “Simps: Using sociology for personal mobility,” *Networking, IEEE/ACM Transactions on*, vol. 17, no. 3, pp. 831–842, 2009.

[34] S. Yang, X. Yang, C. Zhang, and E. Spyrou, “Using social network theory for modeling human mobility,” *Network, IEEE*, vol. 24, no. 5, pp. 6–13, 2010.

[35] S. Gaito, G. Grossi, and F. Pedersini, “A two-level social mobility model for trace generation,” in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, 2008, pp. 457–458.

[36] M. Musolesi and C. Mascolo, “Mobility models for systems evaluation. a survey,” *Middleware for Network Eccentric and Mobile Applications*, pp. 43–62, 2009.

[37] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network.* ACM, 2004, vol. 34, no. 4.

[38] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," Tech. Rep., 2000.

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

[40] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on.* IEEE, 2007, pp. 79–85.

[41] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[42] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in dtns," in *Proceedings of INFOCOM*, 2009, pp. 846–854.

[43] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "Hibop: a history based routing protocol for opportunistic networks," in *Proceedings of World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2007, pp. 1–12.

[44] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *INFOCOM*, vol. 6, 2006, pp. 1–11.

[45] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobyspace-based routing strategies in delay-tolerant networks," *Wireless communications and mobile computing*, vol. 7, no. 10, pp. 1171–1182, 2007.

[46] M. Musolesi and C. Mascolo, "Car: context-aware adaptive routing for delay-tolerant mobile networks," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 2, pp. 246–260, 2009.

[47] I. Cardei, C. Liu, J. Wu, and Q. Yuan, "Dtn routing with probabilistic trajectory prediction," in *Wireless Algorithms, Systems, and Applications*, 2008, pp. 40–51.

[48] J. Link, D. Schmitz, and K. Wehrle, "Geodtn: Geographic routing in disruption tolerant networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–5.

[49] M. Kalantari and R. La, "A dtn packet forwarding scheme inspired by thermodynamics," *Information Sciences and Systems*, pp. 1216–1221, 2008.

[50] I. Parris and T. Henderson, "The impact of locations privacy on opportunistic networks," in *Proceedings of World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2011, pp. 1–6.

[51] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proceedings of Pervasive Computing and Communications Workshops*, 2007, pp. 65–70.

[52] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, 2007, pp. 32–40.

[53] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," *IEEE Transactions on Mobile Computing*, 2010.

[54] A. Mtibaa, M. May, C. Diot, and M. Ammar, "Peoplerank: Social opportunistic forwarding," in *Proceedings of INFOCOM*, 2010, pp. 1–5.

[55] J. Pujol, A. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proceedings of INFOCOM*, 2009, pp. 837–845.

[56] C. Maihofer, "A survey of geocast routing protocols," *Communications Surveys & Tutorials, IEEE*, vol. 6, no. 2, pp. 32–42, 2004.

[57] Y. Ma and A. Jamalipour, "Opportunistic geocast in disruption-tolerant networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1–5.

[58] B. Burns, O. Brock, and B. N. Levine, "Mv routing and capacity building in disruption tolerant networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, 2005, pp. 398–408.

[59] Y. Ma and A. Jamalipour, "Opportunistic geocast in disruption-tolerant networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–5.

[60] M. Piorkowski, "Mobility-centric geocasting for mobile partitioned networks," in *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*. IEEE, 2008, pp. 228–237.

[61] K. Lee, U. Lee, and M. Gerla, "Geo-opportunistic routing for vehicular networks [topics in automotive networking]," *Communications Magazine*, vol. 48, no. 5, pp. 164–170, 2010.

[62] J. LeBrun, C. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of Vehicular Technology Conference (VTC)*, vol. 4, 2005, pp. 2289–2293.

[63] I. Leontiadis and C. Mascolo, "Geopps: Geographical opportunistic routing for vehicular networks," in *Proceedings of World of Wireless, Mobile and Multimedia Networks(WoWMoM)*, 2007, pp. 1–6.

[64] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "Geodtn+ nav: geographic dtn routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[65] G. Kuijpers, T. Nielsen, and R. Prasad, "Optimizing neighbor discovery for ad hoc networks based on the bluetooth pan profile," in *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, vol. 1. IEEE, 2002, pp. 203–207.

[66] G. Záruba, "Accelerated neighbor discovery in bluetooth based personal area networks," in *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA02)*, 2002.

[67] A. Brodt, A. Wobser, and B. Mitschang, "Resource discovery protocols for bluetooth-based ad-hoc smart spaces: Architectural considerations and protocol evaluation," in *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. IEEE, 2010, pp. 145–150.

[68] M. Rónai and E. Kail, "A simple neighbour discovery procedure for bluetooth ad hoc networks," in *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, vol. 2. IEEE, 2003, pp. 1028–1032.

[69] C. Drula, C. Amza, F. Rousseau, and A. Duda, "Adaptive energy conserving algorithms for neighbor discovery in opportunistic bluetooth networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 1, pp. 96–107, 2007.

[70] H. Yoon, J. Kim, and R. Hsieh, "idls: Inter-bss direct link setup in ieee 802.11 wlans," in *Communications and Information Technologies, 2007. ISCIT'07. International Symposium on*. IEEE, 2007, pp. 1015–1020.

[71] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning and Knowledge Discovery in Databases*, pp. 254–269, 2009.

[72] C. Boldrini, M. Conti, and A. Passarella, "Modelling social-aware forwarding in opportunistic networks," Tech. rep., IIT-CNR, Tech. Rep., 2010.

[73] A. Keranen, J. Ott, and T. Karkkainen, “The one simulator for dtn protocol evaluation,” in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, p. 55.

[74] F. Ekman, A. Keranen, J. Karvo, and J. Ott, “Working day movement model,” in *Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, 2008, pp. 33–40.

[75] T. Leighton and A. Moitra, “Some results on greedy embeddings in metric spaces,” *Discrete & Computational Geometry*, vol. 44, no. 3, pp. 686–705, 2010.

[76] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, “Ieee 802.11 b ad hoc networks: Performance measurements,” *Cluster Computing*, vol. 8, no. 2, pp. 135–145, 2005.

[77] G. Anastasi, M. Conti, and E. Gregori, “Ieee 802.11 ad hoc networks: protocols, performance and open issues,” *Mobile Ad hoc networking*, pp. 69–116, 2004.

[78] B. Xing, K. Seada, and N. Venkatasubramanian, “An experimental study on wi-fi ad-hoc mode for mobile device-to-device video delivery,” in *INFOCOM Workshops 2009, IEEE.* IEEE, 2009, pp. 1–6.

[79] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, “Beacon-stuffing: Wi-fi without associations,” in *Mobile Computing Systems and Applications, 2007. HotMobile 2007. Eighth IEEE Workshop on.* IEEE, 2007, pp. 53–57.

[80] A. Gupta and P. Mohapatra, “Energy consumption and conservation in wifi based phones: A measurement-based study,” in *In Sensor and Ad Hoc Communications and Networks (SECON).* IEEE, 2007, pp. 121–131.

BIOGRAPHICAL STATEMENT

Shanshan Lu was born in Shijiazhuang, Hebei, China, in 1983. She received a Bachelor of Computer Science degree from Hebei University, China, in 2005, a Master of Computer Science degree from Beijing University of Chemical Technology, China, in 2008. Shanshan Lu joined the Department of Computer Science and Engineering at the University of Texas at Arlington in Fall 2009 for her Ph. D study.