Realistic, Efficient and Secure Geographic Routing in Vehicular Networks

by

Lei Zhang
B. Eng., China University of Geosciences, 2010

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

Realistic, Efficient and Secure Geographic Routing in Vehicular Networks

by

Lei Zhang
B. Eng., China University of Geosciences, 2010

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Kui Wu, Departmental Member
(Department of Computer Science)

Dr. Issa Traore, Outside Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Kui Wu, Departmental Member
(Department of Computer Science)

Dr. Issa Traore, Outside Member
(Department of Electrical and Computer Engineering)

Abstract

It is believed that the next few decades will witness the booming development of the Internet of Things (IoT). Vehicular network, as a significant component of IoT, has attracted lots of attention from both academia and industry in recent years. In the field of vehicular networks, Vehicular Ad hoc NETwork (VANET) is one of the hottest topics investigated. This dissertation focuses on VANET geocast, which is a special form of multicast in VANET. In geocast, messages are delivered to a group of destinations in the network identified by their geographic locations. Geocast has many promising applications, i.e., geographic messaging, geographic advertising and other location-based services. Two phases are usually considered in the geocast process: phase one, message delivery from the message source to the destination area by geographic routing; phase two, message broadcast within the destination area.

This dissertation covers topics in the two phases of geocast in urban VANETs, where for phase one, a data-driven geographic routing scheme and a security and privacy preserving framework are presented; and for phase two, the networking connectivity is analyzed and studied. The contributions of this dissertation are three-fold.

First, from a real-world data trace study, this dissertation studies the city taxicab mobility. It proposes a mobility-contact-aware geocast scheme (GeoMobCon)

for metropolitan-scale urban VANETs. The proposed scheme employs the node mobility (two levels, i.e., macroscopic and microscopic mobilities) and contact history information. A buffer management scheme is also introduced to further improve the performance.

Second, this dissertation investigates the connectivity of the message broadcast in urban scenarios. It models the message broadcast in urban VANETs as the directed connectivity problem on 2D square lattices and proposes an algorithm to derive the exact analytical solution. The approach is also applied to urban VANET scenarios, where both homogeneous and heterogeneous vehicle density cases are considered.

Third, this work focuses on the security and privacy perspectives of the opportunistic routing, which is the main technique utilized by the proposed geographic routing scheme. It proposes a secure and privacy preserving framework for the general opportunistic-based routing. A comprehensive evaluation of the framework is also provided.

In summary, this dissertation focuses on a few important aspects of the two phases of VANET geocast in urban scenarios. It shows that the vehicle mobility and contact information can be utilized to improve the geographic routing performance for large-scale VANET systems. Targeting at the opportunistic routing, a security and privacy preserving framework is proposed to preserve the confidentiality of the routing metric information for the privacy purpose, and it also helps to achieve the anonymous authentication and efficient key agreement for security purposes. On the other hand, the network connectivity for the message broadcast in urban scenarios is studied quantitatively with the proposed solution, which enables us to have a better understanding of the connectivity itself and its impact factors (e.g., bond probability and network scale).

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my deepest appreciation to my advisor professor Dr. Jianping Pan, who has been an excellent supervisor for me. Thank him for the guidance and advice provided throughout my time as his student. His support on both the research and my personal life has been and will be priceless to my whole life. Thank Dr. Jun Song from China University of Geosciences (Wuhan), who was my mentor for my undergraduate study and provided great support to my later study abroad. Thank Dr. Lin Cai from Department of Electrical and Computer Engineering. We had a successful collaboration, in which her help and wisdom is greatly appreciated. Thank Dr. Jun Tao from Southeast University and Dr. Zhidong Shen from Wuhan University, who shared their valuable life experience with me and made me learn a lot.

Thanks to the graduate students and my friends from Computer Science and Electrical Engineering departments, such as Le Chang, Maryam Ahmadi, Liang He, Yanyan Zhuang, Min Xing, Xuan Wang, Lei Zheng, Fei Tong, Tianming Wei, Boyang Yu, S. Dawood Sajjadi, Maryam Tanha, etc. Their friendships and supports have made my experience at University of Victoria both educational and fun.

I would also like to express my gratitude to my committee members, professor Kui Wu and professor Issa Traore. Thank you for generously giving your time and expertise to better my work.

A special thanks to my family. Words cannot express how grateful I am to my mother, father and the families who are extremely supportive to me. Your prayer for me was what sustained me thus far.

Lei Zhang, Victoria, BC, Canada

Dedication

To My Dear Mentors, Family and Friends.

Chapter 1

Introduction

1.1 VANETs

Mobile Ad hoc NETwork (MANET) is constructed with mobile devices through wireless communications. Because each node has the ability to move in any directions, MANETs can provide information access with fewer constraints on geographic position and are easy to set up since pre-existing infrastructures are not necessary.

As an important category of MANETs, Vehicular Ad hoc NETwork (VANET), where vehicles serve as network nodes, is becoming a significant component of the future Intelligent Transportation Systems (ITSs), which are important parts of Internet of Things (IoT). As a special form of MANET, VANET aims at providing drivers and passengers with information services [1], i.e., safety services (e.g., emergency alert services), and infotainment services (e.g., location-based advertising services), etc. Compared with traditional wireless communication networks, VANET has its own unique features. Usually, VANETs have fewer constraints on power and buffer size, since we assume vehicles always carry sufficient power supply and on-board storage. Wireless Access for Vehicular Environments (WAVE) [2], as an approved amendment to the IEEE 802.11, is the industry standard to support vehicular communication systems. It includes IEEE 802.11p (Physical and MAC layer standards) and IEEE 1609 (the upper layers standards).

Typically supported by Dedicated Short-Range Communications (DSRC) technology, two types of communication modes coexist in VANETs, i.e., Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) [3]. With the infrastructure support, vehicles can easily obtain information from or deliver information to nearby

infrastructures (e.g., Road Side Unit, RSU), i.e., V2I. However, for areas lacking infrastructures, one vehicle needs to communicate with its nearby vehicles to achieve the message exchange and dissemination, i.e., V2V, which is more challenging than the V2I communication. With the high mobility of vehicles, it is hardly likely that there exists a stable and permanent end-to-end path from the message source to destination. As a result, vehicles have to rely on the opportunistic contacts with other vehicles for the message dissemination.

In this dissertation, we focus on the challenging message dissemination with V2V communication. As a result of opportunistic contacts, the message dissemination in VANETs occurs in multi-hop [4] and store-carry-and-forward manners, where vehicles exchange data when they are within the wireless communication range of each other [5] and vehicles can also carry the data while there are no message transmission opportunities. Therefore, the opportunistic vehicular contact behaviors (i.e., the contact frequency and duration) and the mobility of the vehicles ultimately determine the performance of such networks.

1.2 Geocast in Wireless Networks

In communication networks, geocast is a technique to deliver messages to nodes identified by their geographic locations. Geocast protocols can be divided into two categories: proactive geocast and reactive geocast. Proactive geocast protocols determine the message forwarders before the message dissemination starts, which means a forwarding path is first built before the start of the message transmission. On the other hand, reactive geocast protocols only decide the next-hop forwarder at each hop when there is a need of the message transmission for the message carrier. And the decision is usually based on a distributed contention among the neighbors of the current message carrier.

For proactive protocols, acquiring and maintaining the routing information to maintain the forwarding path is costly as it involves additional message transmissions which require energy and bandwidth consumptions. Especially in mobile wireless networks, it is much more challenging because of the frequent change of the node position and network topology. On the other hand, the fact that neither routing tables nor route discovery activities are necessary makes reactive geocast attractive for dynamic networks such as wireless ad hoc networks. In this dissertation, because of the distributed nature of VANETs, we focus on the study of the reactive geocast.

In fact, not only position information can be used for geographic routing, the other information, such as the visiting frequency to the target location, can also be utilized for the routing decision. We call such information, which helps to make routing decisions, the routing metric in the rest of the dissertation. The geographic routing algorithms can be applied under the following assumptions: 1) a node can determine its own routing metric; 2) a node is aware of its neighbors' routing metrics; 3) the position of the destination is known. Routing metric information (e.g., position information) of an individual node should always be easily self-maintained, e.g., through an on-board GPS device. The second assumption can be achieved by short-range wireless communications with neighboring nodes who are within the communication range. Only with the routing metric information of neighbors, the message carrier is able to determine the next hop with a higher chance to deliver the message. The third assumption can be met by means of a location service that maps network addresses to geographic locations. If all the three assumptions are met, geocast is applicable for the routing in wireless and mobile networks.

1.3 Research Objectives and Contributions

In this dissertation, we focus on the geocast in VANETs, which can support various kinds of promising applications, such as urban data (e.g., the traffic or environment information, etc.) collection and Location-Based Services (LBS) (e.g., location-based notification or advertising, etc.).

Typical geocast usually involves two phases: phase one, after messages are generated, they are forwarded towards the destination region; phase two, once messages reach the destination region, they are broadcast within the destination region. The research objectives of this dissertation focus on the two phases. For phase one, we are interested in designing an effective geographic routing scheme which supports the geocast in the metropolitan-scale urban VANET environment. Since user privacy-sensitive information is usually utilized for routing as the routing metric information in such opportunistic networks, the protection of the security and user privacy is also one of our goals. For the second phase, we focus on the network connectivity analysis of the broadcast within the destination region, with considerations of the urban environment, e.g., network topology, vehicle density, etc.

1.3.1 Geographic Routing towards Destination Regions

Designing effective routing schemes is a typical research topic in MANETs. Lots of existing work of geocast is based on geometric distance-based approaches [6–9], where the distance to the destination is taken as the routing metric. However, they are not suitable for the large-scale urban VANETs for the following reasons: first, because of the high node mobility and complex road network structure, the distance relations of nodes change frequently and quickly, causing the reduction in the performance of the distance-based schemes; second, the schemes which require network topology information such as GeoTORA [6] and GeoGrid [8], need to frequently update their knowledge of the network, which can cause tremendous overhead in a large-scale network, e.g., the urban VANET with thousands of nodes.

On the other hand, the routing in Delay-Tolerant Networks (DTNs), which specializes on intermittent connectivity, as a feature of VANETs, is extensively studied [10–12]. We find that, even not designed for geocast, many existing DTN routing schemes can adapt to geocast with minor modifications. However, these DTN routing schemes are originally designed for relatively small-scale networks, e.g., with up to hundreds of nodes. Concerning about the scale, existing schemes either fail to achieve an acceptable performance due to the flooding-like mechanisms [10], or introduce enormous communication and computation overhead, such as the maintenance of the pair-wise node contact history information [11, 12].

Instead of the traditional geometry-based approaches, we extend our previous work [13, 14] and propose a mobility-contact-aware geocast scheme (GeoMobCon) for metropolitan-scale urban VANETs from the DTN perspective, through Vehicle-to-Vehicle (V2V) communications. Different from some of the most efficient DTN routing schemes [11, 12], which are based on the expensive pair-wise contact probability calculation and sharing, our scheme employs the node mobility information at different levels, i.e., macroscopic and microscopic mobility, in addition to a relatively simple use of the contact information. The macroscopic mobility describes the traffic trend of all vehicles in a city, while the microscopic mobility captures the mobility patterns of individuals. Because the macroscopic mobility for a city is relatively stable and the microscopic mobility is completely self-maintained by each vehicle, this mobility hierarchy makes our scheme distributed, simple, scalable and communication and computation-efficient when compared with existing solutions.

The two levels of mobility are extracted from the real-world GPS traces of taxicabs

and buses in Shanghai, China. To facilitate the mobility modeling, we divide the city into regions, each of which contains considerable traffic volumes. Traffic flows among regions are extracted and utilized as the macroscopic mobility pattern. The volume of the traffic flows can indicate how well the regions are "connected" through vehicles and how reliable the message dissemination between regions can be via vehicular communications. The massive data trace also allows us to investigate each individual's mobility pattern, which serves as the routing criterion. The proposed scheme also employs the contact information of vehicles with the targeted regions. Considering practical restrictions, i.e., the limited buffer size and transmission bandwidth, an efficient buffer management is introduced.

1.3.2 Connectivity within the Destination Region

Once a message reaches the destination region, it is broadcast among all the nodes within the region. The connectivity between the message source and a node at an arbitrary position can be used to evaluate the effectiveness of the broadcast. Connectivity has been extensively studied in ad hoc networks [15–19]. Connectivity is defined as the probability of delivering the message to the destination at a certain time or within a time duration.

The study of connectivity in two-dimensional (2D) ad hoc networks has attracted lots of attention in the community, most recently with geometrical probability, stochastic geometry, and percolation theories [20–22]. In urban VANET V2V scenarios, messages are propagated along the roads by vehicles. For simplicity and versatility, we use Manhattan grid to model the urban road structure. The network then can be modeled as a 2D square lattice, where percolation theory has been used. Initially in statistical physics, percolation theory studies the process of liquid filtering through porous materials [23]. The process can be modeled by vertexes (sites) and edges (bonds) in certain dimensions. Assuming an infinite number of vertexes and edges, percolation occurs when there exists an infinite connected giant component (and an infinite number of finite components). Percolation is more likely to occur with a larger bond probability p, so when p varies from 1 to 0, percolation either occurs or not, exhibiting a sharp phase transition at the so-called *critical* probability p_c. If the filtering directions are given, it is called directed percolation (DP).

In this dissertation, we study a related but different problem: directed connectivity (DC), i.e., given a starting vertex and the bond probability to connect neighbor

vertexes on a square lattice, what is the probability for the message to reach an arbitrary vertex following certain directions?

Despite the effort in more than half a century, DP and many related problems are mainly solved numerically by simulations. The most related work determined the critical probability analytically of a square lattice where the vertical bond probability is p_y and the horizontal probabilities are 1 and p_x interleaved at different layers [24]. Conceptually, DC problems are even harder than DP. However, by extending our previous work on 2D ladder connectivity [25] and by using a new recursive decomposition approach, we have obtained the analytical expression for the DC problem on square lattices. The approach shall be extended to lattices with different horizontal and vertical bond probabilities and arbitrary shapes.

In this dissertation, the work on the connectivity analysis makes the following contributions [26–28]. First, to the best of our knowledge, it is the first time in literature to give an exact analytical solution to the DC problem on square lattices and can quickly determine the network connectivity without lengthy simulations. Even though the majority of the results are based on square lattices, they can offer valuable insights when clustering and aggregation are possible in full 2D networks. We also show that the approach is applicable for other shape lattices, e.g., triangular lattice. Second, we explore the obtained analytical expressions and analyze the impact of the bond probability, and the lattice size and ratio on network connectivity, in addition to determining the complexity of the proposed approach. Third, we apply the approach to the urban VANET scenarios to show the extensibility of the approach. Inspired by existing work [25, 29], we carefully map the urban VANET message propagation to the DC problem. Both homogeneous and heterogeneous network node density cases are discussed and valuable insights are discovered about how applications can benefit from the results.

1.3.3 Security and Privacy Protection in Geographic Routing

In VANETs, the message propagation is usually conducted based on the opportunistic contacts. The routing is called opportunistic routing. Different from the traditional topology-based routings, opportunistic routings make routing decisions based on each node's local information, making them more applicable for networks with large scales and high dynamics [14]. Opportunistic routing has been extensively studied in

DTNs [11]. Because of the high-dynamic nature, VANET belongs to the family of DTN. Therefore, these routing techniques are also applicable in VANETs. In most opportunistic routing algorithms, messages are forwarded to the nodes with a higher delivery chance to the destination. Nodes in opportunistic routings have to broadcast, exchange and compare their local or individual information, e.g., the distance or visit frequency to the destination. In this dissertation, we call such information the *routing metric*.

However, from the privacy and security perspectives, opportunistic routing can raise critical issues. A serious threat is the traffic analysis, where the network traffic can be observed by a malicious node and then the malicious node uses the information gathered to launch attacks. Besides, the routing metrics, e.g., geographic location or contact history, are highly privacy-sensitive. Without a proper protection, severe privacy problems can occur.

Although the routing metric information is very privacy-sensitive, most of the current work on the security and privacy of both VANET and DTN has neglected the protection of it. Lots of work [30–33] focuses on the node identity anonymity, using techniques such as pseudonyms, group signature, and identity-based encryption, etc. On the other hand, the recent work [34] takes the privacy issue of the "metric" information into consideration in social-based DTNs. However, because of their social relationship-based nature, such work does not provide node identity anonymity, which is essential for VANETs.

To address the concerns, we propose an advanced secure and privacy-preserving framework [35] especially for opportunistic routings, integrating the following three properties: 1) Confidentiality of the routing metric. Protected by cryptographic tools, the routing metric is known only to its owner. However, to perform message routing, the framework allows a node to compare its own routing metric with others' without knowing the exact values of others' routing metrics. This is achieved by integrating a solution to the "Yao's millionaire problem" [36]. The protection of the routing metric, thus enhancing the node privacy, is the key feature which distinguishes our design from others. 2) Anonymous authentication. Authentication is the fundamental mechanism for various security properties, i.e., data integrity, authenticity and non-repudiation. For the strong requirement of identity privacy in VANETs, anonymity is another essential property that must be provided. In this dissertation, we adopt a group signature-based scheme to achieve the anonymous authentication. 3) Efficient key agreement. In ad hoc networks, it is desirable for each pair of nodes to share

a unique session key to achieve the pair-wise confidentiality. Considering the total number of the session keys and the lack of central control in VANETs, an efficient key management is crucial. In this dissertation, we adopt an efficient pairing-based key agreement scheme and integrate it seamlessly into the message routing process without creating much overhead.

A comprehensive evaluation of the proposed framework is provided. We first analyze the security of our design, and then evaluate the performance with cryptographic implementation specifications and event-driven simulations. These evaluations show the security and feasibility of the framework for the targeted network environment. Moreover, our framework is not limited to VANETs. It can be applied to any opportunistic routing scenarios (MANETs or DTNs).

1.4 Dissertation Organization

This dissertation covers topics in the two phases of geocast in urban VANETs, including phase one, where a data-driven routing design and related security and privacy design are presented; and phase two, where the networking connectivity is analyzed and discussed. The rest of this dissertation is organized as follows.

In Chapter 2, we introduce the real-world data traces (i.e., GPS traces of taxicabs and buses from Shanghai) we used, and perform data analysis, which particularly focuses on the study of vehicle mobility. The knowledge discovered provides us with good insights for the design of geographic routing in Chapter 3.

In Chapter 3, we propose a mobility-contact-aware geocast scheme (GeoMobCon) for metropolitan-scale urban VANETs. The proposed scheme employs the node mobility (two levels, i.e., macroscopic and microscopic mobilities) and contact history information. Considering practical restrictions, i.e., the limited buffer size and transmission bandwidth, a buffer management scheme is introduced which further improves the performance of our scheme.

Chapter 4 focuses on the connectivity analysis of the message broadcast. It models the message broadcast in urban VANETs as the directed connectivity problem on 2D square lattices. The proposed algorithm gives the exact analytical solution without lengthy simulations. It is also applied to the urban VANET scenario, where both homogeneous and heterogeneous vehicle density cases are discussed and valuable insights are discovered about how the applications can benefit from the results.

Chapter 5 focuses on the security and privacy perspectives of the opportunis-

tic routing. We propose a secure and privacy-preserving framework for the general opportunistic-based routing, to which VANET geocast belongs. A comprehensive evaluation of the framework is also provided.

Chapter 6 concludes the dissertation with further research issues.

Chapter 2

Data Analysis based on Real-World Traces

2.1 Overview

Because VANETs are featured with the store-carry-and-forward message propagation, it is of great importance to understand the mobility of the basic network components, i.e., vehicles. To be realistic, we conducted the majority of our work based on real vehicle GPS traces collected in a modern city, Shanghai, China. There are two main benefits of introducing the real trace: first, the real-world traces enhance the reliability of our scheme by providing the realistic user mobility; second, the analysis of the traces provides us more insights of vehicle behaviors, which can be utilized in the network design for better performance. In this chapter, we focus on the introduction of the traces and perform data analysis to extract insightful knowledge regarding the vehicle mobility.

2.2 Introduction of Vehicle Traces

The traces we used (partially available at http://www.cse.ust.hk/scrg) were collected from vehicles in Shanghai, including 2,299 taxicabs from Jan. 31, 2007 to Feb. 27, 2007 and 2,500 buses of 103 routes from Feb. 24, 2007 to Mar. 27, 2007. Each bus reported a GPS report every one minute, while taxicabs reported every 15 seconds if there was no customer on board and every one minute when with customers. The information contained in the trace includes the vehicle ID, the latitude and longi-

tude location, timestamp, vehicle moving speed and heading direction. In addition, taxicabs also reported whether they are hired by customers. For buses, the reports also contain the route ID that the bus is operating on, and whether the bus is at the terminal station. We are interested in different types of vehicles, i.e., taxicabs and buses, since the diversity of different node mobility pattern can potentially benefit the message propagation scheme design. The mobility patterns considered are summarized as follows:

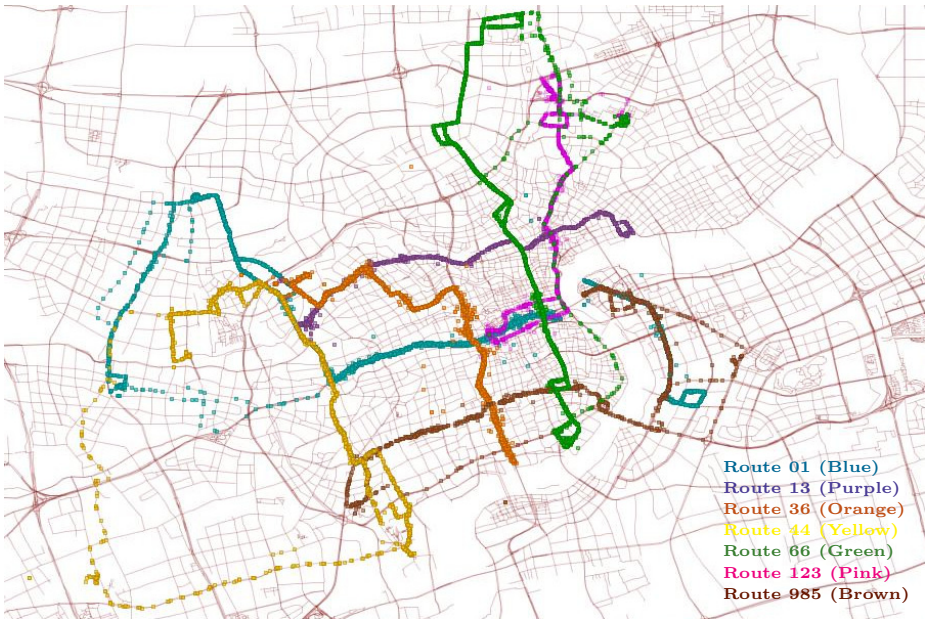


Figure 2.1: Selected Bus Backbone.

Buses Each bus has a limited spatial and temporal coverage, i.e., moving along fixed routes during a certain period of time, which implies a very distinct mobility pattern. And this can be helpful for the geographic routing since the node mobility is highly predictable. Given a message destination as a location, i.e., area or region, the buses whose routes cross that area or region should be preferred as message forwarders. Figure 2.1 shows seven selected bus routes, where different colors correspond to the bus traces on different routes. Most routes go through the urban area of Shanghai, where the major financial and tourism districts, universities and train stations are located. These routes cover the major roads that carry significant amount of the traffic in the city, and form a grid-like backbone.

Taxicabs Compared with buses, the pattern of taxicab mobility is less distinct with a much larger spatial and temporal coverage. By intuition, the taxicab mobility is driven by two factors: 1) customer demands and 2) taxicab drivers' driving habits. If one taxicab is occupied by customers, the mobility is mainly determined by the customer destination. The driver may pick the shortest path or a path with least congestion. If the taxicab is not occupied, the mobility depends on the taxi driver's driving habits and "customer hunting" preference.

Figure 2.2: Heat Map of Taxicab Traffic[1].

All the traces cover a large portion of Shanghai city, with the horizontal span of 70 km from the west-most record point (Qingpu District) to the east-most point (Pudong Airport), and 45 km vertical span from the north-most point (Baoshan District) to the south-most one (South of Minhang District). Such a square area has a size of around $3,150$ km^2. However, such an area is not spread with vehicles everywhere because of the city and road structures. Public vehicles (i.e., buses and taxicabs) appear more often in hot social spot areas, such as transportation hubs, commercial areas and regions connecting the hot spots. For example, by counting the number of the GPS records of all taxis, we plot Fig. 2.2 which shows the traffic condition of the city. Red color indicates a high traffic volume while green color indicates a lower volume. Although the study is based on the trace of public vehicles, i.e., buses and taxis, the mobility feature of private vehicles is even more affected by the city geographical characteristics [37], therefore, the similar approaches can be applied to

[1]This figure plots all the GPS record locations on the map, using Google Map API. From color green, to yellow, and to red, the record density becomes higher and higher. The "heat" is just an estimation of the density, so no numerical scale is shown.

and similar conclusions can be drawn on the private vehicles. The mobility pattern for private vehicles is stronger since the daily driving trajectories of private vehicles are usually more distinct, e.g., a normal private vehicle user commutes mostly between his home and work place.

2.3 Vehicle Mobility Modeling

Because of the store-carry-and-forward feature of VANETs message dissemination, it is important to understand how vehicles move in the city and where they can "carry" the messages. In this section, we focus on the study of the vehicle mobility with the help of the traces. Note that, the mobility of the buses are stable and predetermined by their routes, therefore, we focus on the mobility study of the taxis, which is more random.

2.3.1 "Hot" Region Identification

Data Pre-processing

To simplify the data processing, we grid Shanghai map into small unit square regions of size 1 km $\times$1 km each. We treat each unit square as the minimum composition unit for the geographic region.

During the data processing, we observe that the traces contained errors. A common error found in these traces is the distance between two consecutive GPS reports exceeds the maximum possible distance traveled at the maximum allowed speed. In this work, we set the maximum allowed travel speed at 120 km/h, a practically enforced speed up-limit in Shanghai. The reasons for these errors can be the inaccurate time synchronization of device clocks or the disturbance from the environment, etc. To guarantee the accuracy of the following analysis, we omitted the trace of the particular vehicle in that particular day in which the error is detected.

Traffic Indicator

"Hot" regions refer to the geographical regions with remarkable vehicle motion properties and are composed of unit squares with very noticeable traffic load. To locate such unit squares, we adopt two traffic metrics to express the traffic load in each unit

square: the Vehicle Kilometers Traveled (VKT) and the Accumulative Residence Time (ART).

VKT is a widely-used traffic evaluation metric in transportation engineering, which refers to the distance traveled by the vehicles on the roads. It is usually considered as an indicator of the traffic pressure (or traffic demand) and is used to describe mobility patterns and travel trends. The VKT value for each unit square is calculated as:

$$VKT_i = \sum_{k=1}^{N_i} v_k \cdot t_k, \tag{2.1}$$

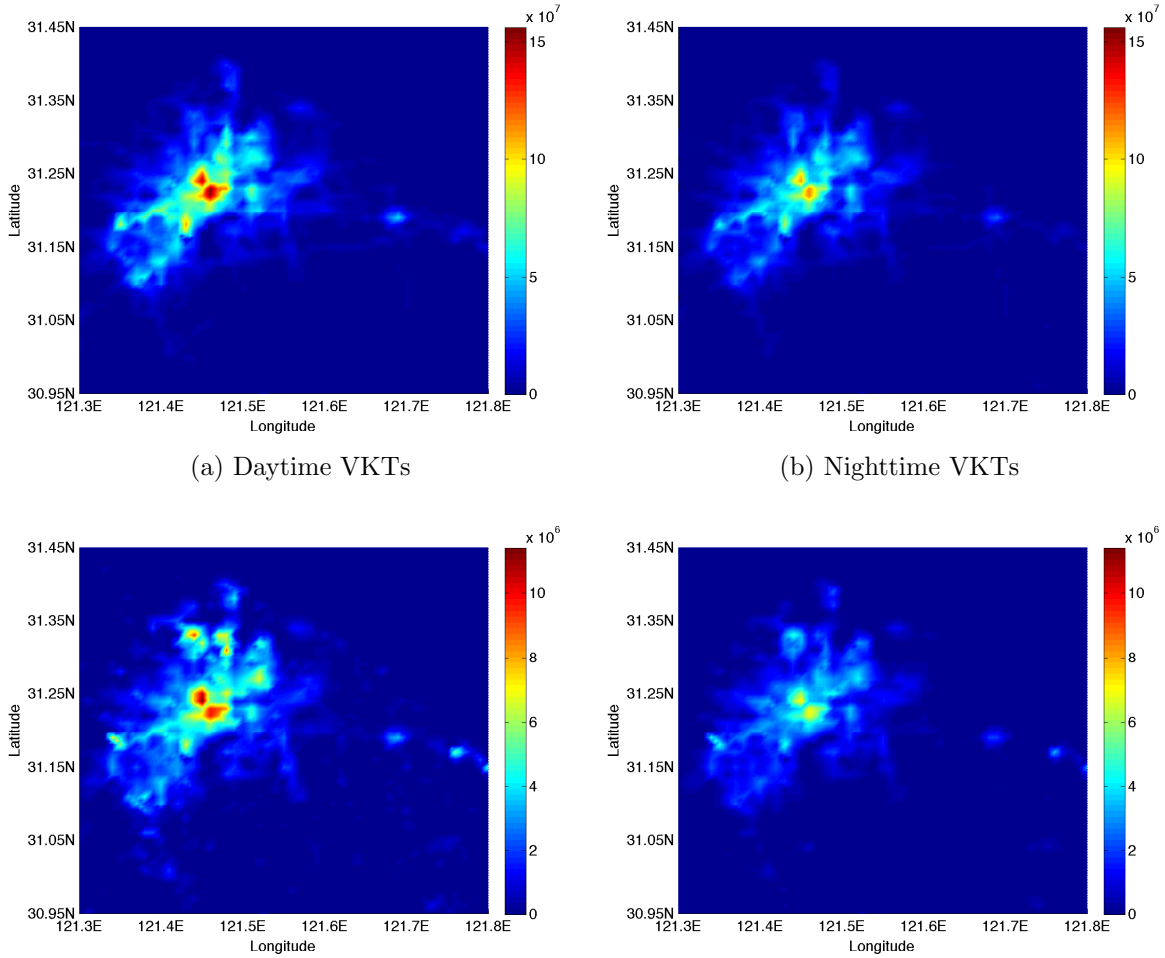
where N_i is the total number of taxis once appeared in unit square i, and v_k and t_k are the average travel speed and time duration taxi k spends in square i, respectively. Higher speed and longer staying time imply more traffic pressure.

In modern cities, different areas may exhibit different traffic properties. Some areas, e.g., downtown, have higher traffic flow rates, which make the traffic patterns in those areas more dynamic. On the other hand, areas such as airports usually demonstrate more static property, since taxis tend to stay until they are hired by new customers. To reflect the static side of taxicab mobility, we also calculate the ART of each unit square i, which is the sum of the residence time for all taxis appeared in square i.

Figure 2.3: Traffic Distribution (VKT) of Shanghai[2].

[2]The color in the figure indicates the values of the traffic indicator VKT. The numerical scale can be referred to Fig. 2.4(a).

After calculating VKT and ART values for all the unit squares, we can identify the traffic attraction areas of Shanghai, where the majority of the recorded traffic is reported. Figure 2.3 gives an overview of the taxi traffic distribution over the city of Shanghai in terms of VKT. A warmer color (e.g., red over yellow) implies higher traffic load. As demonstrated, the aforementioned "hot" regions contribute to the most of the taxicab traffic in the city and become the areas we are more interested in when considering the vehicle mobility.



(a) Daytime VKTs

(b) Nighttime VKTs

(c) Daytime ARTs

(d) Nighttime ARTs

Figure 2.4: Traffic Load during Daytime and Nighttime.

Besides the spatial taxicab traffic distribution, we also consider the impact of time in a day. Figure 2.4(a) and (b) illustrate the VKTs in daytime hours (6 am to 6 pm) and nighttime hours (6 pm to 6 am), respectively. While Fig. 2.4(c) and (d) show the ARTs in daytime and nighttime, respectively. As expected, less traffic is reported

during nighttime hours. From the figures we can see that, for either VKT or ART, the traffic distributions over the map are quite consistent in daytime and nighttime hours. The stable distribution indicates the consistency of the region division over time in the next section.

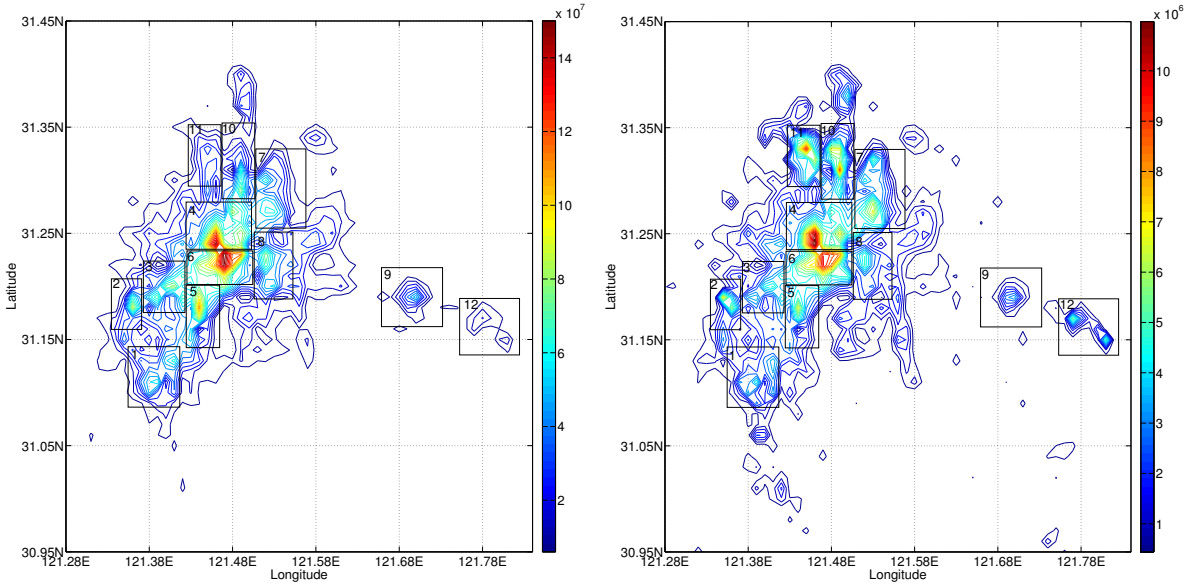
Region Division

We concentrate on popular regions having a large amount of traffic, i.e., with remarkable VKT or ART values. By clustering the unit squares combined with the geographical and social information of Shanghai, such as the locations of large commercial districts or transportation hubs, we convert the map of unit squares into a graph, whose nodes are referred to as independent "hot" regions in terms of traffic density.

Table 2.1: Identified Hot Regions

ID #	Region Name	Description
1	Xingzhuang district	Transportation hub & commercial area
2	Hongqiao airport	Transportation hub
3	Xinjingzhen	Hi-tech development zone
4	Shanghai railway station	Transportation hub
5	South railway station	Transportation hub
6	City centre	Commercial area
7	Wujiaochang district	Commercial area
8	Pudong district	Commercial area
9	Chuansha district	Nearby town
10	Gaojinzhen	Taxi company
11	Gongfuxincun	Transportation hub
12	Pudong airport	Transportation hub

Table 2.1 gives each region's sequence ID, name and a short description. As shown in Fig. 2.5, we plotted the contour of the VKT and ART values and picked 12 regions with considerable traffic densities. Most "hot" regions are active in both VKT in Fig. 2.5(a) and ART in Fig. 2.5(b). However, region 11 (Gongfuxincun) and region 12 (Pudong airport) appear more distinct in terms of ART than their VKTs, which means the taxicab mobility in these two regions is more static. This is because, for the region 12, it is Pudong International Airport, which is far away from the city center and region 11 was a subway terminal when the data was collected. In both regions, the taxi drivers prefer to stay longer to wait for new customers.



(a) Division according to VKTs

(b) Division according to ARTs

Figure 2.5: The Division of Popular Regions.

2.3.2 Macroscopic Mobility Modeling

From a macroscopic perspective, we can describe the vehicle mobility as the movement among different "hot" regions. To characterize the mobility patterns, we study the transition residence time and transition probability of vehicle movement among regions.

Transition Residence Time between Regions

The transition residence time is defined as the travel time within one region before the taxi leaves for the next one. After collecting the transition residence time from the traces, we investigate the distribution of the transition residence time. We use the statistics toolbox in Matlab to generate some distributions to fit our data samples, among which both exponential and log-normal distributions show good fits. However, as we can observe from Fig. 2.6, exponential distribution performs better than log-normal in fitting the samples with small residence time but high frequency, i.e., the first data bin in the figure. The samples with higher frequency dominate the whole distribution, and thus we prefer to adopt the exponential distribution as the transition residence time distribution approximation. We also perform the Chi-square tests to

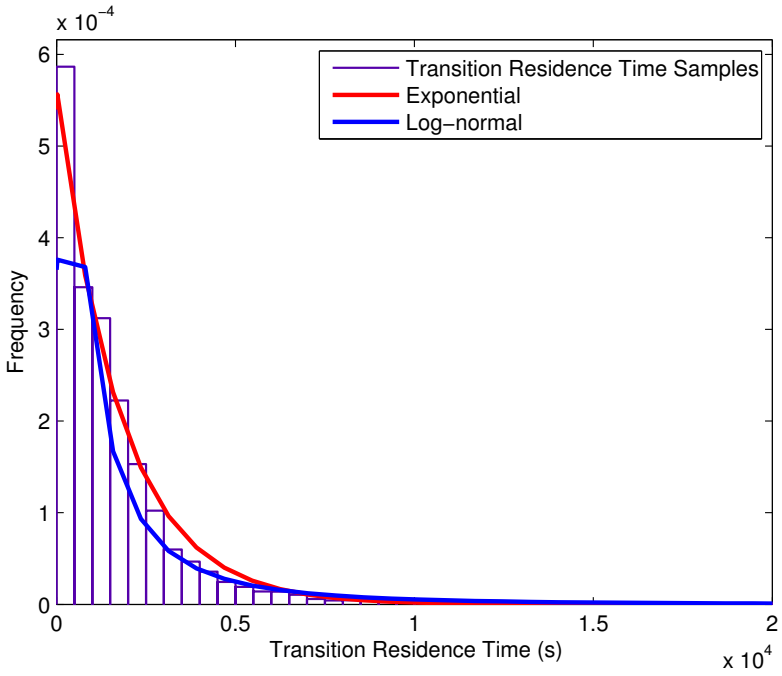
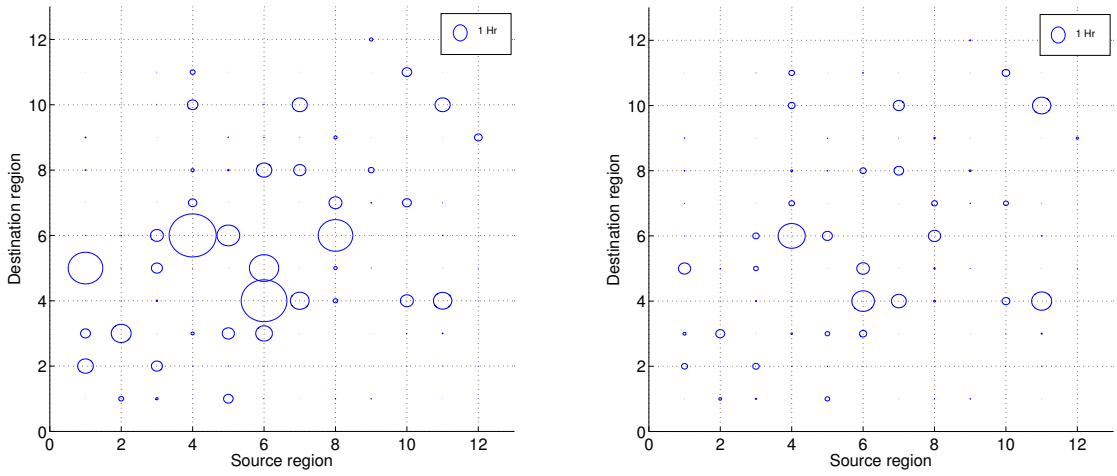


Figure 2.6: Distribution of Transition Residence Time from Region 6 to 5.

verify our hypothesis. The results[3](in Fig. 2.6) show that the fit of our sampled data to an exponential distribution is accepted at the level of significance $\alpha = 0.05$. Different from the log-normal distribution observed in [38] for pedestrian mobility on campus, we believe exponential distribution is more capable of capturing taxicabs' motion property. As for pedestrians, when they enter a region, i.e., a building, they probably stay there for a while until finishing working or shopping, etc. So the data bin having the highest frequency residence time falls in somewhere between the minimum and maximum values. But for taxis, it is more probable that they just pass through a region in a short time when they do not need to take new customers. Thus, the frequency of small residence time samples is much higher. Because the exponential distribution fits, we also claim that the transition residence time of all traffic appearing in a region is memoryless.

To understand the relationship between the transition residence time and the taxicab travel trajectory, we plot the transition residence time from one region to another in Fig. 2.7. These figures tell us the average transition residence time that

[3]The chi-square goodness-of-fit test does not reject the null hypothesis (i.e., that the data comes from a population of a certain distribution) at the α significance level for the exponential distribution, while it rejects the null hypothesis at the α significance level for the lognormal distribution.



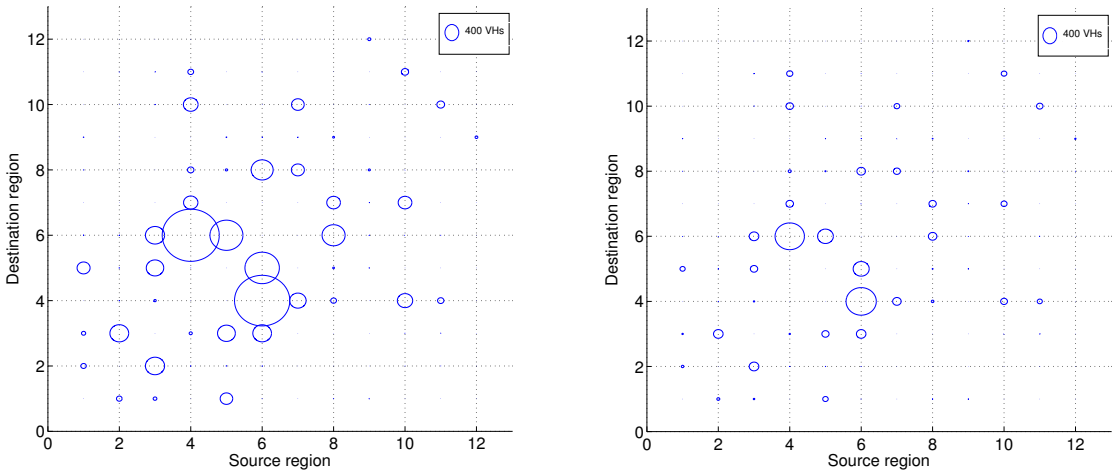
(a) Daytime Hours

(b) Nighttime Hours

Figure 2.7: Average Transition Residence Time (indicated by the Circle Radius).

the taxis spend in the current region before moving to the next. In each figure, the x-axis and y-axis represent the current region and the next region, respectively. A larger radius of the circle indicates a larger average transition residence time in the current region (x coordinate of the circle center) before moving to the next (y coordinate of the circle center). In other words, by looking at this figure we can compare the transition residence time of different region pairs. We observe that the transition residence time within one region depends on the next region that vehicles move to and we can also consider it as the inter-region transition residence time. For instance, we consider taxis in region i are leaving for region j, and the transition residence time in region i follows distribution Pt_{ij}. We find that Pt_{ij} is not identical to Pt_{ik}, if $j \neq k$, i.e., in Fig. 2.7, the size of circle(i, j) is different from that of circle(i, k). Such a phenomenon in fact reflects the geographic and social features of different regions. If the traffic flow between two regions is very smooth, e.g., with less chance of traffic jams, we can expect a shorter transition residence time; otherwise, the transition residence time will be longer. Moreover, we compare the transition residence time in different time periods, and find that the transition residence time during nighttime hours is shorter than that in daytime. This indicates the change of the traffic load during a day. During the day time, with more vehicles on the roads, traffic jam is more likely to happen, which increases the travel time within regions.

Transition Probabilities between Regions



(a) Daytime Hours (b) Nighttime Hours

Figure 2.8: Number of Transitions (indicated by the Circle Radius).

To reflect the motion pattern of taxis traveling among different regions, we count the total number of transitions between any two regions, and plot them in Fig. 2.8. In this figure, the radius of a circle represents the number of such transitions. The number of transitions demonstrates similar patterns between daytime and nighttime, despite the fact that there are much fewer transitions during nighttime. If such transition numbers are normalized by the total number of all transitions from the same region, we have the transition probabilities shown in Fig. 2.9, which are expressed by bars. The lengths of the bars with the same x coordinates add to 1. According to our results, the transition probabilities remain quite stable during the daytime and nighttime, so we summarize all the statistics in one figure.

Mobility Modeling

Given the transition probabilities and transition residence time, the movement of vehicles among different regions can be modeled as a random process where the vehicles spend in each region for a certain amount of time. The transition probability of vehicles moving from region i to j within time t could be expressed as follows:

$$P_{ij}(t) = Pr\{X(s+t) = j | X(s) = i\}, \text{ for all } s, \tag{2.2}$$

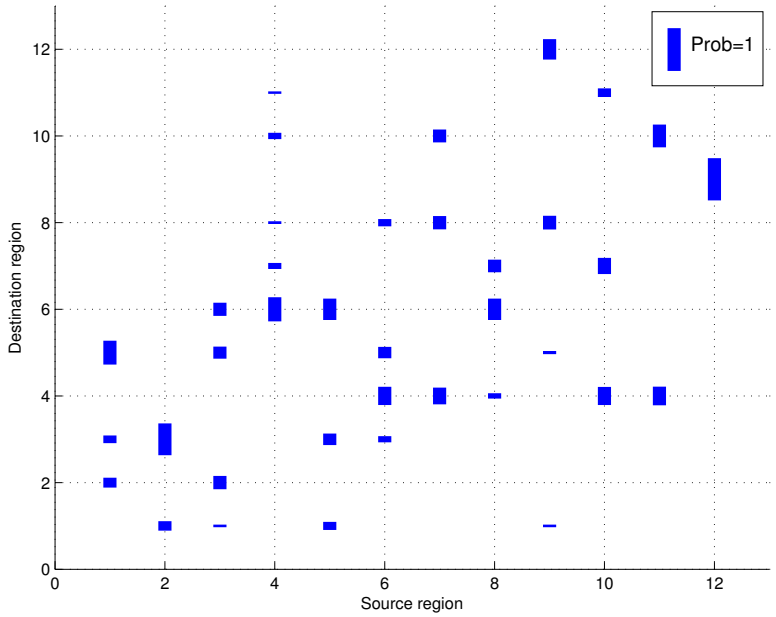


Figure 2.9: Transition Probabilities (indicated by the Bar Length).

and the general transition probability from region i to region j

$$P_{ij} = \int_0^\infty P_{ij}(t)\mathrm{d}t. \tag{2.3}$$

Since the time spent in each region, say i, can be modeled as exponential distribution with parameter $\frac{1}{v_i}$, then the vehicle departure rate from this region is v_i. The transition rate from state i to state j is denoted as q_{ij}, where

$$q_{ij} = v_i \cdot P_{ij}. \tag{2.4}$$

And we let q_{ii} indicate the total incoming traffic rate of region i as opposed to the outgoing traffic rate q_{ij},

$$q_{ii} = -\sum_{j \neq i} q_{ij} = -v_i, \tag{2.5}$$

then the transition rate matrix can be written as

$$\mathbf{Q} = \begin{pmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,j} & \cdots \\ q_{1,0} & q_{1,1} & \cdots & q_{1,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ q_{i,0} & q_{i,1} & \cdots & q_{i,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}. \tag{2.6}$$

For a region which has a relatively stable traffic condition, say j, the outgoing flow rate equals the incoming flow rate:

$$\sum_{i \neq j} q_{j,i}\pi_j = \sum_{i \neq j} \pi_i q_{i,j} \text{ or } v_j \pi_j = \sum_{i \neq j} \pi_i q_{i,j} \implies \bar{\pi} \cdot \mathbf{Q} = \mathbf{0}. \tag{2.7}$$

Because $\sum_{0 \leq i \leq n} \pi_i = 1$, we can express this as following:

$$\bar{\pi} \cdot \mathbf{E} = \mathbf{e}, \tag{2.8}$$

where $\mathbf{E} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$, and $\mathbf{e} = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}$then

$$\bar{\pi} \cdot (\mathbf{Q} + \mathbf{E}) = \mathbf{e} \Longrightarrow \bar{\pi} = \mathbf{e} \cdot (\mathbf{Q} + \mathbf{E})^{-\mathbf{1}}. \tag{2.9}$$

With the knowledge of transition probability and exponentially distributed residence time, we take the transition probabilities and the distributed residence time as inputs, to model the taxicab mobility and make simulations to generate synthetic trace data of the movement of vehicles. We verify our modeling using the stationary distribution, which is the spacial location distribution of taxicabs over all regions. As shown in Fig. 2.10, the green curve reflects the result calculated from the model. By arbitrarily picking a time point, we derive the stationary distributions of taxicabs at that time point, from both the synthetic simulation traces and real world traces, and we represent them with blue and red curves, respectively. As we can see, the stationary distributions match well with each other, and it proves the accuracy of our modeling. There are some differences between the green (the model) and blue (the synthetic trace) curves. This is because the synthetic traces are generated following the certain distributions described in the model. The sampling of a certain

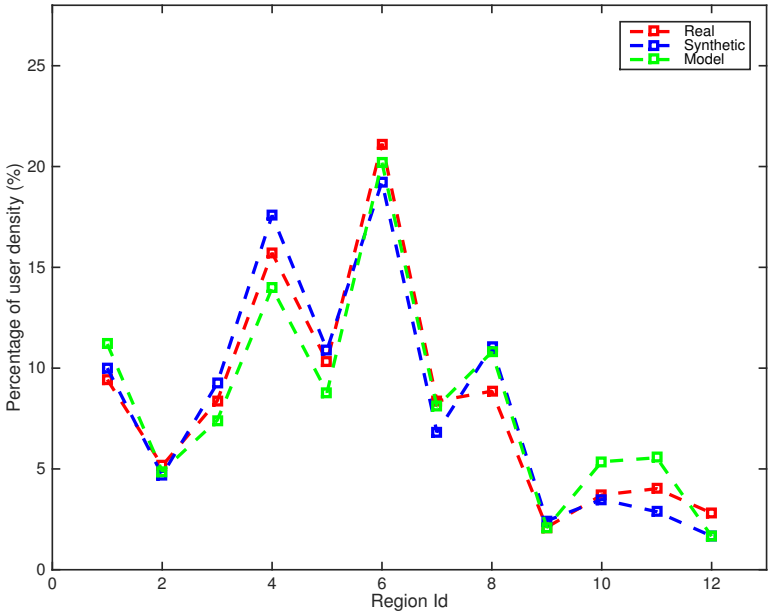


Figure 2.10: Taxicab Stationary Distribution.

distribution usually introduces randomness, leading to the difference.

2.4 Conclusions

In this section, we have introduced and analyzed the real-world GPS trace data of Shanghai. After a short introduction of the trace, we concentrate on the analysis of vehicle mobility from two aspects: transition residence time and transition probability, based on which we further proposed a model for vehicle mobility. In the following geographic routing design in Chapter 3, we further explore the transition property of individual vehicles, and utilize it as the microscopic mobility pattern for the routing decision purpose.

Chapter 3

Mobility-Contact-based Geographic Routing

3.1 Overview

The study of the vehicle trajectory trace inspires us for the message propagation protocol design. Specifically, the study of the vehicle mobility pattern, e.g., the transition behaviors of vehicles among different regions, inspires us to the design of new geographic location-based routing schemes, i.e., geocast. In this chapter, we propose a mobility-based geocast routing scheme and further extend it with the consideration of the contact history information between individual nodes and destinations, which helps to improve the routing performance. The details of the new geocast design are presented and discussed.

3.2 Related Work

3.2.1 Traditional Geocast Schemes

The study of geocast has a relatively long history since 1987 [39]. In [6, 39], only unicast is considered. To further improve the delivery ratio, multicast, e.g., directed flooding, is widely adopted by many schemes. [7] proposed two schemes based on directed flooding, where one defines a rectangular forwarding zone between the source and destination and the other forwards messages to all the neighbors who have shorter distances to the destination. [40] introduced Voronoi diagram for the forwarding zone

selection. It selects the nodes in a Voronoi region towards the destination as the next hop. [41] vertically divides the vehicle transmission coverage region into two half-circular sections and selects all the border nodes as the next hop from the half section towards the destination. [42, 43] select nodes near road junction within the transmission range as next hop vehicles. Another category of geocast schemes uses group-based approaches. In GeoGrid [8], the network is partitioned into logic grids and each partition selects one single gateway node as a group representative to forward messages. [9] introduced additional infrastructures as the gateways, collecting data from mobile nodes and forwarding them to the destination. The maintenance of gateways introduces extra overhead. [44, 45] deploy navigation system for geocast purposes.

Specifically for VANETs, different geocast schemes are proposed utilizing different information about the vehicles and the traffic to improve the performance. [42, 46] utilize the vehicle density and traffic load information to select the propagation route. Traffic lights information is used in [47]. One-hop link quality and degree of vehicle connectivity is considered in [48]. In [49], multiple metrics (i.e., distance to the destination, vehicle density, vehicle trajectory and communication bandwidth) are considered in the routing protocol design. In [50], the vehicle movement prediction in a grid road structure is used for geographic routing. However, the acquisition of such information itself could be very challenging. Some applications of the geocast in VANETs are discussed in [51–53].

3.2.2 DTN Routing Schemes

DTNs enable communications where the source to destination connectivity cannot be always sustained. VANET is a typical delay-tolerant network. Compared with the conventional geocast algorithms, DTN routing is more capable of dealing with high node mobility and transient node connectivity. For such reasons, we propose a geocast solution from the DTN's point of view in this dissertation.

Flooding is a very popular technique in DTN routing. Epidemic [54] allows nodes to exchange messages whenever there is a chance. For a better scalability, some controlled flooding schemes are proposed. Two-Hop-Relay [55] limits the number of hops each message can travel. Spray-and-Wait [10] limits the number of message copies that can be forwarded during each transmission. None of these schemes include any relay node selection mechanism, leading to a poor delivery performance.

Besides flooding-based schemes [10,54], another very important DTN routing category is the contact information-based routing, where a smarter relay node selection is made. Spray-and-Focus [56] is the follow-up protocol of Spray-and-Wait, introducing the relay selection phase. In Prophet [11], each node maintains the encounter history with other nodes, and the routing decision is made based on the encounter probability. MaxProp [12] also utilizes the encounter information to estimate the cost of a virtual end-to-end path to the destination and uses it as the metric for routing decisions. MaxProp also takes into account realistic issues such as buffer size and bandwidth limitation. However, in [12], MaxProp is only tested on bus traces. [57,58] further take into account the inter-contact time, and thus are much more complicated.

3.3 Vehicle Mobility Description

In the previous chapter, the vehicle mobility among the "traffic" dense areas was discussed, where the significant unit areas are identified and the regions are manually selected. However, the choice of the areas is not fixed since the main idea is to give a general description of the vehicle mobility. In this chapter, we use a more precise and rigorous method to identify those areas. The vehicle mobility, i.e., the transition behaviors at two different levels, will be our main focus and used in the routing design. The concept "mobility entropy" is introduced to demonstrate the activeness of the vehicle mobility and the mobility difference between individual vehicles.

3.3.1 Clustering-based Region Identification

Due to the large scale of the map, it is hard to outline all details of the regions which have distinct traffic volumes. Thus, following the method mentioned in the previous chapter, we first discretize the map as a tiling of unit square regions with a size of 1 km $\times$ 1 km each, and each vehicle trajectory trace is converted to a sequence of unit squares. We focus on the unit squares with a considerable amount of traffic by counting its GPS report frequency.

With the unit squares identified, we cluster them into regions. The identification of regions helps to describe vehicle mobility in a proper granularity concisely. Different from the previous chapter, i.e., dividing the regions manually, we adapt a more precise method, i.e., using a clustering algorithm to form the regions. We apply the k-means clustering algorithm to these unit squares. The value of k determines how many

regions can be formed. Since the total number of unit squares is fixed, with a larger value of k, more regions will be formed, but with a smaller size each. For generality, we set 40 as our default number of regions, so each region nearly covers a distinct area whose size is similar to the regions identified in Chapter 2. Notice that the clustering is only one of the methods to define the geographic regions and 40 is used as an example. Depending on different applications, the size, shape and location of these regions of interest can be customized. But the devision of regions will not have an obvious impact on the network performance since the total number of regions is limited and the region information will be locally used by each vehicle as described in the following routing design.

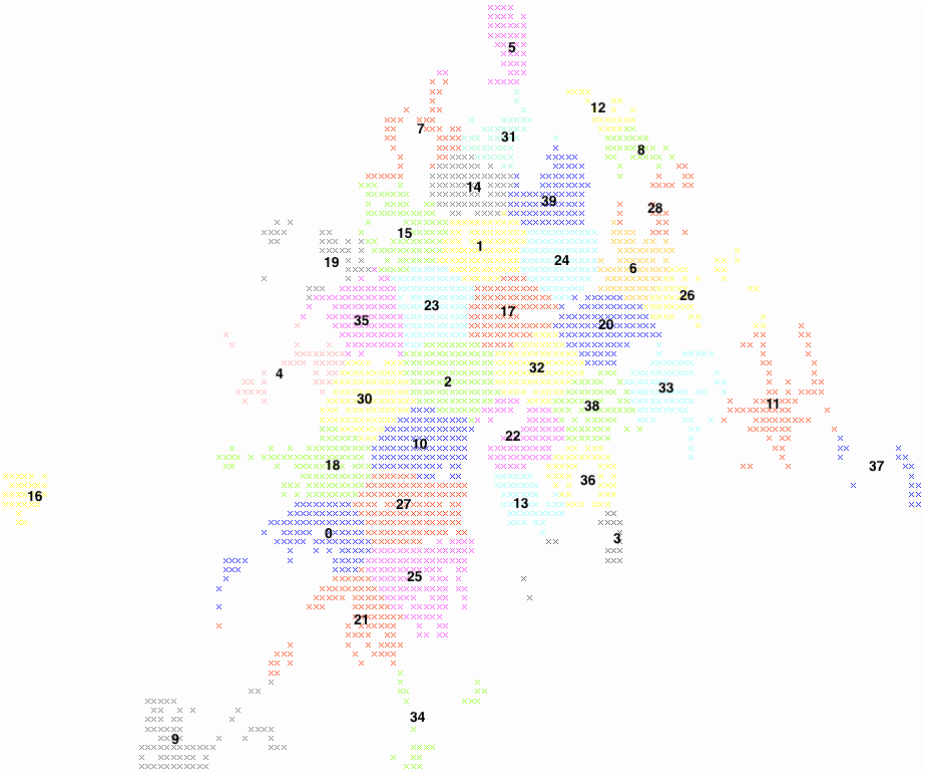
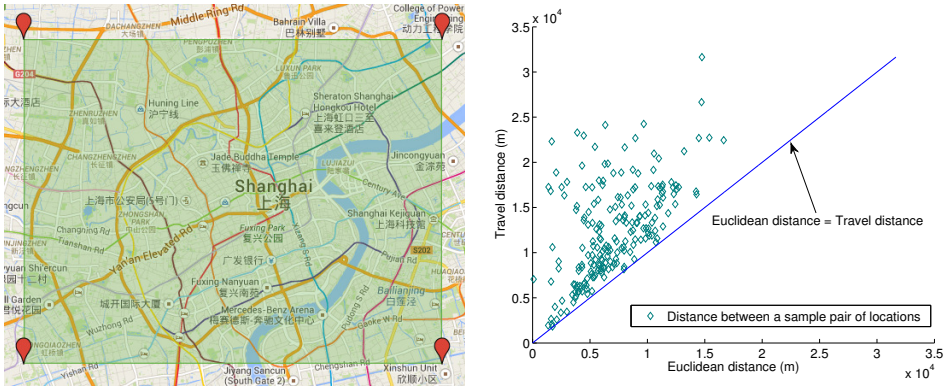


Figure 3.1: Clustered Regions based on the Travel Distance.

Traditional clustering algorithm is based on the Euclidean point distance but we use the travel distance of two locations instead. This is because we take the real-world road structure into consideration to reflect the actual reachability between any two locations. Travel distances can be obtained through online map services, e.g., Google maps. Figure 3.2 shows a sample study of the difference between travel distance and Euclidean distance. We select a sample area, with size of 199 km^2 in downtown,



(a) The Selected Area (b) Distance Difference of Samples

Figure 3.2: Difference of Euclidean and Travel Distances .

Shanghai, shown in Fig. 3.2 (a). Two hundreds pairs of points are randomly selected and the Euclidean and travel distances are compared in Fig. 3.2 (b). In Fig. 3.2 (b), each dot represents a sample pair of nodes and its x and y coordinates represent the Euclidean and travel distances, respectively. And the difference is very obvious.

The clustering result is shown in Fig. 3.1, where all regions are identified by different numbers and colors. Note that the colors here do not reflect density.

3.3.2 Two-level Mobility

For mobile ad-hoc networks, node mobility plays a significant role in opportunistic forwarding-based routing protocols. Especially for the geographic routing such as geocast, a proper understanding and utilization of the node mobility can help improve the performance. Two levels of mobility patterns, macroscopic and microscopic, are extracted from the real-world vehicle traces.

Macroscopic Mobility Pattern

Macroscopic mobility pattern reflects the overall traffic transition among regions. It can be either provided by the urban planning or transportation department or obtained by counting the number of vehicle commuting between regions. More of such vehicle transitions between two regions imply a stronger traffic flow, which further implies the higher reliability to transfer messages between two regions via vehicles. The macroscopic mobility characterizes the traffic flows between any pair of neighbor regions in the city. It reflects how regions are connected by vehicle traffic and how

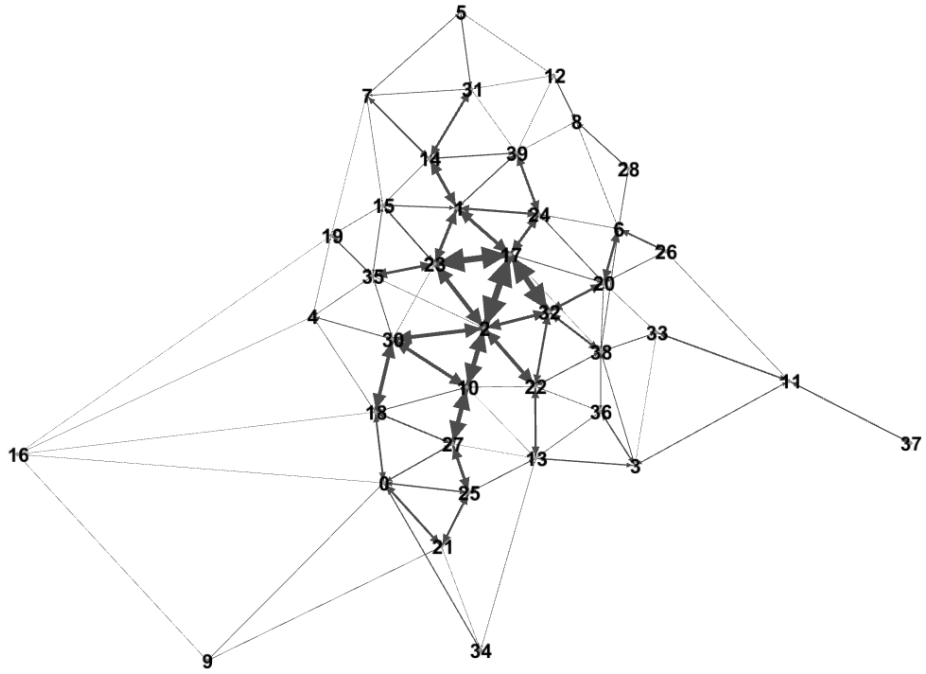
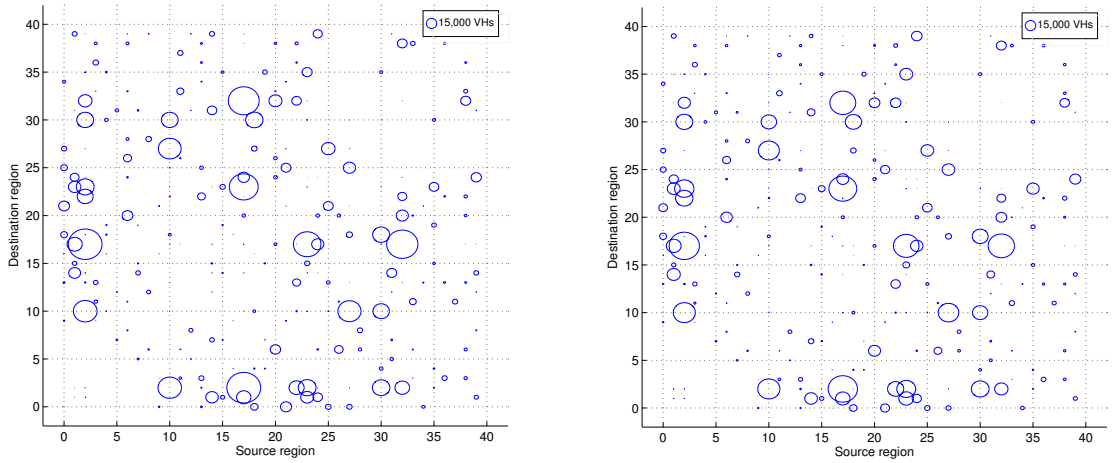


Figure 3.3: Macroscopic Mobility Patterns.

strong each connection is. Inspired by the mobility modeling mentioned in the previous chapter, we express the macroscopic mobility pattern by a weighted directed graph, $MacMP(\mathbf{V}, \mathbf{E}, \mathbf{w})$, as shown in Fig. 3.3, where the vertices, $\mathbf{V}$, represent the regions shown in Fig. 3.1, the directed edges, $\mathbf{E}$, represent the traffic flows and the thickness, $\mathbf{w}$, of edges represents the amount of the vehicle traffic, i.e., the strength of the connection. We can observe the strong region connections in downtown areas, e.g., between region 2, 17, 23 and 32, and weaker connections on the periphery of the city, e.g., at airports (region 16 and 37). Another property of the macroscopic mobility is that such patterns are relatively stable for the whole city, see Fig. 3.4. Figure 3.4 follows the same style as Fig. 2.8 in Chapter 2 but based on the data of a longer time duration, where the radius of the circle indicates the number of vehicle transitions from the source region to the destination region. In big cities, without a major change of the districts development or the transportation systems, such a traffic status is usually stable. This is an attractive feature as such information does not need frequent updates, which greatly reduces the scheme complexity.



(a) Daytime Hours

(b) Nighttime Hours

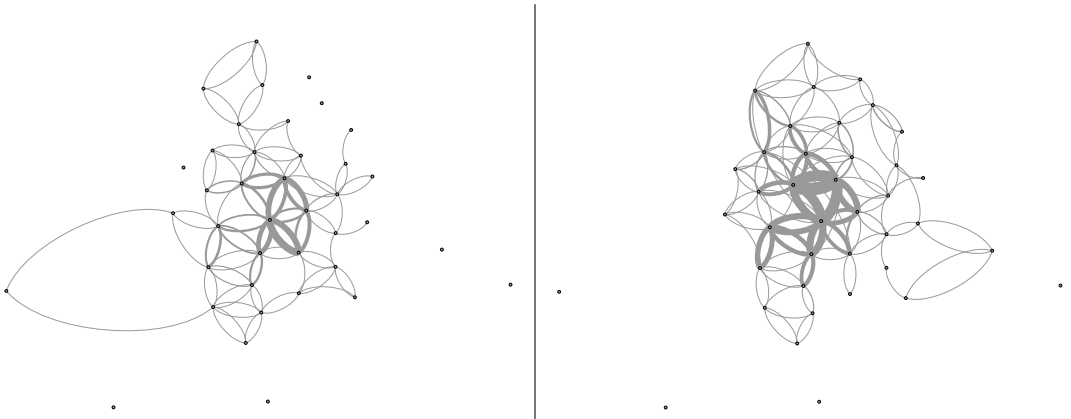
Figure 3.4: Number of Transitions (indicated by the Circle Radius).

Microscopic Mobility Pattern

Microscopic mobility pattern, on the other hand, captures the motion patterns of individual vehicles. For buses, the ones which belong to different routes, have different mobility coverage. For taxis, individual taxis have different mobility pattens caused by different driving behaviors of the drivers, e.g., some drivers prefer to work in downtown areas while others prefer to take longer-distance businesses, such as to airports. The mobility pattern can also be shown using weighted graph similar to Fig. 3.3. Figure 3.5(a) and (b) depict the patterns of two taxis over the data collection period. An obvious difference can be observed, i.e., Taxi 0094 was active in only downtown areas while Taxi 01292 showed more activities in more regions. For a specific vehicle v, the microscopic mobility pattern can be presented as a set of conditional probabilities,

$$MicMP_v = \bigcup P(f_i|h_n h_{n-1} \cdots h_1 h_0), f_i, h_j \in \mathbf{V}, n \leq N,$$

which records all the transition probabilities to certain regions given the history information as a sequence of regions. Here f_i indicates the region which is possible for v to go to, given that it has come from regions $h_n, h_{n-1}, \cdots, h_1, h_0$ and h_0 indicates the current region of the vehicle. If the history contains the n previous regions, we call it the nth-order conditional probability. Given a threshold N, we represent



(a) Taxi 0094 (b) Taxi 01292

Figure 3.5: Microscopic Patterns for Individual Taxis.

each vehicle's microscopic mobility pattern with all its possible nth-order conditional probabilities where $n \leq N$. And its space complexity is on the order of $|\mathbf{V}| * N_B^{n-1}$, where $|\mathbf{V}|$ represents the total number of regions and N_B is the average number of neighbors for each region.

A very desirable feature is that the microscopic mobility pattern is totally self-maintained by a vehicle itself because it only depends on its own movement. No information sharing between vehicles is needed. Each vehicle only needs to perform a statistic analysis of its history trajectory to obtain its own microscopic mobility pattern (i.e., the conditional probabilities).

3.3.3 Mobility Entropy

To a large extent, our proposed routing scheme depends on the microscopic mobility patterns of individual taxis. It is important to understand how strong the microscopic mobility patterns are for different vehicles and how they change over time during a normal work day. To quantitively show the activeness of individual mobility, the mobility model can be expressed in terms of mobility entropy, exploiting the mobility patterns of mobile nodes in MANETs. Similar concepts can be found in [59–62]. An example is given as follows.

With clustered regions, the trip of a taxi during a certain time period can be

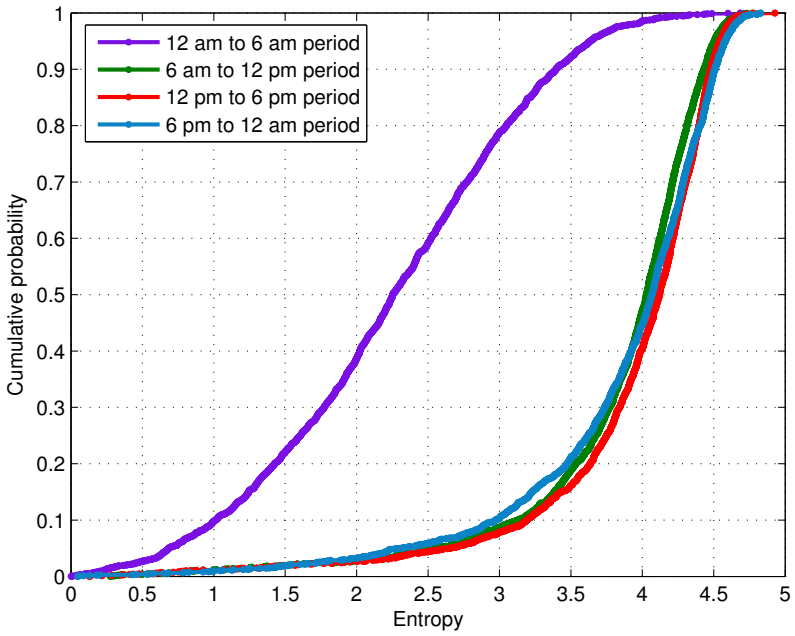


Figure 3.6: Mobility Entropy Distributions.

represented as a sequence of geographic regions, e.g., for taxi A and B,

$$T_A = r_3, r_2, r_2, r_3, r_5, r_2,$$
$$T_B = r_1, r_2, r_3, r_5, r_4, r_1.$$

During this short period, for taxi A, the visiting frequencies of regions r_3, r_2, r_5 are $\frac{2}{6}, \frac{3}{6}, \frac{1}{6}$, respectively. And for taxi B, the visiting frequencies of regions r_1, r_2, r_3, r_4, r_5 are $\frac{2}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}$, respectively. We can observe that taxi A has a more distinct movement preference as it moves in a limited number of regions, i.e., only three regions. On the other hand, the trace of B has higher randomness with 5 regions. Thus, introducing the similar concept of entropy from communication theory, the mobility entropy can be calculated:

$$E_A = -\frac{2}{6}\log\frac{2}{6} - \frac{3}{6}\log\frac{3}{6} - \frac{1}{6}\log\frac{1}{6} = 0.439,$$
$$E_B = -\frac{2}{6}\log\frac{2}{6} - \frac{1}{6}\log\frac{1}{6} * 4 = 0.678.$$

Taxi A has a more predictable pattern than B, which can be shown as $E_A < E_B$. We studied the traces of all taxis over the whole data collection period. With different

time granularity, we are able to obtain the activeness of a taxi at time periods with different lengths. For simplicity, we divide a day, i.e., 24 hours, into equal-length periods, saying two 12-hour periods scenario, three 8-hour periods scenario and four 6-hour periods scenario. We compare the mobility entropy distributions of different periods in each scenario. We have found that the most distinct difference of different periods appears in the 6-hour period scenario, and we plot the cumulative distribution function (CDF) of all vehicles entropy in Fig. 3.6. The following conclusions can be obtained: 1) For the first 6-hour period, from 0 to 6 am, the entropies are much smaller than those of other periods, implying taxis are a lot less active during this period and the mobility pattern is very deterministic. 2) The entropy distributions of the other three periods are very similar, meaning the activeness of the taxis during their usual work hours, i.e., 6 am to 12 am, is very similar. 3) From 6 am to 12 am, the majority (70%) of taxis has an entropy falling between 3.5 to 4.5, showing that most taxis have a similar activeness during that period of time, although the individual difference exists as shown in Fig. 3.5. The mobility entropy study provides us with more insights of the taxicab mobility pattern, i.e., how strong the patterns are, how they change over time and how the pattern is distributed among all taxis. The mobility entropy reflects the mobility frequency information, while the mobility sequence information is reflected by conditional probabilities described in the microscopic mobility. Both kinds of information are used in the routing decision logic in our proposed geocast scheme.

3.4 Mobility-Contact-aware Geographic Routing

In this section, we propose a geocast scheme in a city scenario via taxicabs and buses. As a basic assumption in VANETs, all vehicles are cooperative for message transmission. Generated from the source node, a message is first forwarded towards the destination area through multiple hops. Once it reaches the target area, it is simply broadcast within that area. Because the message broadcasting within the target area is relatively simple in terms of routing, in this chapter, we mainly focus on the geographic routing from the source to the target area. The message forwarding scheme consists of two parts: message forwarding strategy and buffer management. The two levels of mobility pattens extracted and the contact history are utilized in the message forwarding strategy. The macroscopic mobility pattern is used in the forwarding path selection, while microscopic mobility patterns and the contact

Table 3.1: Notations

Notation	Explanation
FR	A list of a vehicle's future regions
TG	A list of a message's ToGo regions
M_i	Message with ID i
LM_m^v	Vehicle v's mobility-based forwarding likelihood of message m
LC_r^v	Vehicle v's contact-based visiting likelihood of region r
N	Threshold for the number of history regions
$h_n h_{n-1} \cdots h_1 h_0$	The nth-order history regions of a vehicle
C_{age}	Aging constant for the contact-based visiting likelihood
$C_{encounter}$	Initialization constant for the contact-based visiting likelihood

history are used in routing decision making.

3.4.1 Mobility-Contact-based Routing Algorithm Design

We propose a mobility-contact-aware geocast scheme, which we call GeoMobCon for short. The key idea of GeoMobCon message routing is to utilize the two levels of vehicle mobility and the contact history information. An optimal (i.e., strong region-to-region connectivity via vehicle traffic) routing path, which is a sequence of regions towards the destination, is selected based on the macroscopic mobility when a message is generated. Then vehicles try to forward this message along such a path towards the destination. A vehicle's microscopic mobility and its contact history with the destination determine whether it can help to forward the message to its next one or a few hops on the optimal path. Notations used are given in Table 3.1.

Macroscopic Mobility-based Forwarding Path Selection

The larger traffic volume towards a region usually implies a better connectivity to that region. The chance of successfully delivering a message to that region is also higher. Thus, we start our geocast routing by first selecting an optimal forwarding path leading to the destination with the best connectivity based on the graph shown in Fig. 3.3. Once a message is generated or propagated into a new region, the message carrier calculates the optimal path leading to the destination. Such an optimal path is a region sequence which has the best connectivity (i.e., in terms of the traffic volume) from the origin to the target destination and the path information is embedded in the message header once the path is calculated. To find such a path, Dijkstra algorithm is

applied to the weighted graph in Fig. 3.3, where the weight on the edge is determined by the total traffic volume between regions. All vehicles should pre-load the overall traffic weight graph to be capable of calculating the optimal path for each message. Since the city traffic pattern does not change dramatically over time, there is no need to frequently update this graph. Vehicles try their best to forward the messages along their optimal paths, to increase the delivery ratio. The message is transferred in a region-by-region manner, i.e., once the message enters one region, suitable vehicles are selected to forward it to the next region along the optimal path.

Microscopic Mobility-based Routing Decision

Each vehicle also maintains its own mobility pattern, i.e., the microscopic mobility pattern. Individual mobility pattern is used for making explicit routing decision when two vehicles encounter with each other. Such a pattern is represented as a collection of conditional probabilities. These conditional probabilities can be obtained and updated from the movement history of vehicles among the regions.

According to the current vehicle location, relay vehicles will try to forward a message to the next optimal region according to the optimal path stored in the message. We use the term *likelihood* to determine how capable a vehicle is to forward a message to the next optimal region. Assume the optimal path for message m is $r_2 \rightarrow r_3 \rightarrow r_7 \rightarrow r_5 \rightarrow r_6$, where r_6 indicates the destination region. When two vehicles encounter with each other in region r_5, one of them, say v_i is carrying m and the other, v_j, is not. Then v_i first requests v_j's mobility-based delivery likelihood LM_m^j for message m. If $LM_m^j > LM_m^i$, v_i forwards the message to v_j, otherwise, the contact information will be used for the routing decision making, which will be explained in Section 3.4.1. To increase the delivery ratio, the message source does not delete the message after it being forwarded. The message is only deleted from the local buffer when its LM drops below zero or by the buffer management mechanism introduced in Section 3.4.2. Such a process is expressed in Alg. 1.

A vehicle maintains the mobility-based likelihood for each message it is carrying, and these likelihoods are used to compare with other vehicles' likelihoods to make the routing decision. Algorithm 2 gives the details of how the likelihood is calculated and updated. In Alg. 2, we utilize the near-future location information if it is available. This is because we consider the mobility features of city buses and taxis, whose near-future location may be easily accessible. We denote the near-future regions of a

Algorithm 1 Routing Strategy

```
1: procedure RoutingStrategy(vehicle S, R, message m)
2:     LM_m^S = LikelihoodMobUpdate(S, m)
3:     LM_m^R = LikelihoodMobUpdate(R, m)
4:     D is the destination region of m
5:     LC_D^S = LikelihoodConUpdate(S, D)
6:     LC_D^R = LikelihoodConUpdate(R, D)
7:     if LM_m^S < LM_m^R then
8:         if LM_m^R > 0 then
9:             S forwards m to R
10:            if LM_m^S < 0 then
11:                Drop m
12:            end if
13:        else
14:            S does not forward m to R
15:        end if
16:    else
17:        if LC_D^S < LC_D^R then
18:            S forwards m to R
19:            if LM_m^S < 0 then
20:                Drop m
21:            end if
22:        else
23:            S does not forward m to R
24:            if LM_m^S < 0 then
25:                Drop m
26:            end if
27:        end if
28:    end if
29: end procedure
```

vehicle, if available, as "future regions" $\mathbf{FR} = \{fr_1, fr_2, \cdots, fr_n\}$ and the regions the message still has to go through according to its optimal path as "ToGo" regions $\mathbf{TG} = \{tg_1, tg_2, \cdots, tg_n\}$.

If the encountered vehicle's near-future information is known, such information is utilized. For taxis carrying customers, their destinations are determined by the customers on board. For buses, because they have pre-fixed routes, so their future regions over a short period are predictable. For such vehicles, we only need to see if their future locations have any intersections with the message's desired future optimal regions. If yes, these vehicles are very likely to be helpful to forward that message along the optimal path. There are also vehicles whose destinations are unknown, such as taxis without customers. At this point, the taxi's microscopic mobility pattern

becomes the source for location prediction. According to v_j's historical mobility information, it replies to v_i with the probability of going to the next region along the optimal path given the current trajectory history, as v_j's forwarding likelihood, according to the conditional probability from its microscopic mobility pattern. If v_j has a larger probability than v_i to go to any of the future regions along the optimal path of the message, v_i will forward the message to v_j.

Algorithm 2 Mobility-based Likelihood Update

```
1: procedure LikelihoodMobUpdate(vehicle V and message m)
2:     if V's FR is known then
3:         if V's FR intersects with m's TG then
4:             Find the intersection region I closest to the destination
5:             Record I's index in TG idx
6:             LM_m^V = 1 + idx / TG.length
7:         else
8:             LM_m^V = −1
9:         end if
10:    else
11:        Start nth order prediction, n = N
12:        while nth order pattern does not exist in pattern set do
13:            n = n − 1
14:            if n < 2 then
15:                LM_m^V = −1
16:                Break
17:            end if
18:        end while
19:        LM_m^V = P(TG_1 | h_n h_{n−1} ··· h_1 h_0)
20:    end if
21:    Return LM_m^V
22: end procedure
```

Contact History-based Routing Decision

Some vehicles may not show a high chance of going to the expected future regions according to their current mobility, i.e., current history trajectory. But these vehicles can also have a high visiting frequency to the destination according to their long-term history records. However, these vehicles will be excluded by the mobility-only based routing process mentioned before. In order to grab such forwarding opportunities, we introduce another routing decision metric, contact history, in addition to the microscopic mobility-based routing decision making strategy. The microscopic mobility is based on the vehicle's short-term moving trajectory, while the contact pattern reflects

the vehicle's long-term motion pattern. As shown in Alg. 1, when a message carrier v_i compares the mobility-based likelihoods of itself and the encounter v_j, and notices that $LM_m^j < LM_m^i$, it will request v_j's contact history-based likelihood $LC_{destination}^j$. If $LC_{destination}^j > LC_{destination}^i$, v_i still forwards the message to v_j since v_j has a higher visiting frequency to the destination.

Algorithm 3 Contact-based Likelihood Update

1: **procedure** LIKELIHOODCONUPDATE(vehicle V and region D)
2: k is the time units elapsed since the last time V visited D
3: $LC_{D(new)}^V = LC_{D(old)}^V * C_{age}^k$
4: **if** V enters D **then**
5: $LC_{D(new)}^V = LC_{D(old)}^V + (1 - LC_{D(old)}^V) * C_{encounter}$
6: **end if**
7: Return $LC_{D(new)}^V$
8: **end procedure**

Each vehicle maintains its contact history information with each region of interests. Such history information is updated in real time, meaning whenever it visits a region of interest, the contact information with that region is updated. We adopt the contact updating method [11], with the consideration of the aging factor. Define LC_d^j as the contact-based visiting likelihood of vehicle v_j and region of interest r_d. The initial value of it is set to be 0. Whenever v_j visits r_d, it updates the value of LC_d^j:

$$LC_{d(new)}^j = LC_{d(old)}^j + (1 - LC_{d(old)}^j) * C_{encounter}, \tag{3.1}$$

where $C_{encounter} \in (0, 1)$ is an constant and according to [63] we set it to be 0.75. We also consider the aging factor, which means the value of LC_d^j is reduced with time elapsing since v_j's last visit of r_d. And this can be represented in the following equation,

$$LC_{d(new)}^j = LC_{d(old)}^j * C_{age}^k, \tag{3.2}$$

where $C_{age} \in (0, 1)$ is the aging constant and set to be 0.98 [63]. k represents the number of time units[1] elapsed. This updating process is summarized in Alg. 3. Note that, the contact information is updated in real time, which means whenever a vehicle visits a region, it updates the contact information to that region. Since the number of the regions of interest is limited, such information maintenance is scalable with space complexity on the order of $|\mathbf{V}|$, where $\mathbf{V}$ is the set of the regions of interest as

defined in Section 3.3.2.

3.4.2 Buffer Management

To be realistic, we assume each vehicle carries a limited-size buffer on board and has limited wireless transmission bandwidth. Thus when two vehicles encounter, not all the messages in the buffer can be transmitted at one contact. To effectively utilize the buffer space and each contact opportunity, the buffer management of our routing scheme involves the following mechanisms:

First, after a message carrier V forwards the message m to another vehicle, it will check its likelihood LM_m^V. For each message m and vehicle V pair, LM_m^V changes with the location of the vehicle. If LM_m^V becomes too small, say below zero, and it has already been forwarded to others, the vehicle deletes the message to reduce the extra storage and communication overhead.

Second, each vehicle prioritizes the messages in the transmission queue. When vehicle v_i encounters v_j, v_i may have more than one message suitable to be forwarded to v_j. Before the transmission starts, v_i first sorts the messages in its transmission queue according to the likelihood of LM_m^j descendingly. Note that the messages are sorted according to the message delivery likelihood LM_m^j from v_j's point of view, so that the message which can take the most benefit from v_j gets first transmitted.

Third, the acknowledgments of the received message are flooded in the network to remove the redundant copies of the message. We introduced a similar idea from MaxProp [12], where each acknowledgement is a 128-bit hash of the received message ID, source and destination. The cost is little if the acknowledgment is small compared to data messages, which is 512 Kbits in our simulation setting. In addition to the small buffer size overhead, it also costs little communication overhead, which has been shown in [12] that no more than 1% of the average connection duration is spent on sending acknowledgments.

[1]We use the default time unit, which is 30s, defined in our simulator theONE [65].

3.5 Performance Evaluation

3.5.1 Protocols Comparison

We compare our scheme with six other geocasting protocols. These protocols include traditional distance-based geocast scheme, DTN-based schemes and mobility-based schemes related. Although the DTN-based ones [11, 12, 12, 54] are not originally designed for geocast, we can easily convert them for geocast purpose by setting the destination with stationary mobility. To distinguish from the original schemes, we add a "Geo" prefix to the original scheme names. Short descriptions about the schemes are given as follows:

• **GeoEpidemic [54].** Whenever two vehicles encounter with each other, they exchange as many messages as they can, which is subject to buffer and bandwidth constraints. This is a simple flooding.

• **GeoProphet [11].** Prophet makes routing decisions based on the history of contacts information. Messages are forwarded to the nodes who have a higher contact frequency with the destination. A *transitive* property is considered when updating the contact frequency. If A meets B frequently while B meets C frequently, then it is believed that A and C should have a relatively high contact frequency.

• **GeoMaxProp [12].** Similar to Prophet, MaxProp also utilizes the history of contact information. Each node maintains a contact graph based on the history of contact information of both its own and its encounters'. Routing decisions are made according to the cost of a delivery path going through a specific neighbor. A low-cost delivery acknowledgement scheme is adopted and the transmission queue is prioritized considering both message hop counts and delivery probabilities.

• **GeoDist.** We compare our scheme with distance-based geocast scheme (i.e., GeoDist), which is the key and most representative algorithm for many existing geocast schemes. Messages are forwarded to the nodes with a shorter distance to the destination area. Similar to [64], for possible transmissions, the messages in the transmission queue are prioritized according to the "improvement" which can be possibly achieved by a transmission. The "improvement" here means the reduction of the distance to the destination. The greater the "improvement", the higher the priority assigned and the sooner the message can be transmitted.

• **GeoMob [14].** GeoMob [14] is the precursor of the currently proposed GeoMobCon. In GeoMob, the same buffer management scheme as GeoMobCon is

adopted. However, for the routing decision, only the two-level mobility model is utilized. Contact information between vehicles and regions is not utilized. This is the main difference from its enhanced version, GeoMobCon.

• **GeoMobCon-NoBufMgt.** Lastly, in order to show the effectiveness of the buffer management in our proposed scheme. We add a modified version of GeoMobCon, i.e., GeoMobCon-NoBufMgt, which removes the buffer management mechanism, for comparison. It has the mobility-based and contact information-based routing strategy, but no buffer management.

Table 3.2: Comparison among Multiple Schemes

Routing algorithm	**Abbreviation**	**Knowledge for routing decision**	**Buffer management**
GeoEpidemic [54]	GE	None	No
GeoProphet [11]	GP	Contact history	No
GeoMaxProp [12]	GMx	Contact history	Yes
GeoDist	GD	Distance to the destination	Yes
GeoMob [14]	GMb	Nodes mobility	Yes
GeoMobCon	GMC	Node mobility & contact history	Yes
GeoMobCon-NoBufMgt	GMCNoBM	Node mobility & contact history	No

A scheme comparison is shown in Table 3.2. Flooding-based schemes, e.g., Epidemic, do not make any routing decisions to select relay nodes. The other schemes carefully select relay nodes based on different information, i.e., contact history or distance information, leading to a controllable amount of copies for each message. The amount of the copies varies in different situations. For instance, using GeoProphet, if the destination is active and has rich contact history with many nodes, a large number of copies are expected to be forwarded to those nodes. However, if the destination is not very popular, a limited number of copies will be generated. Among all schemes, our scheme, as well as GeoMaxProp, GeoDist and GeoMob, has taken into account the practical restriction of the buffer size and transmission bandwidth, which is reflected by introducing the buffer management mechanisms.

We provide a further comparison between other schemes and ours. First, traditional distance-based geocast routing, e.g., GeoDist, has limitation in a large-scale urban vehicular network. With the complex road structure and high vehicle mobility, the relative position of vehicles changes quickly. A vehicle, which is currently closer to the destination, may be farther away in the next second, because either it takes a detour towards another area or it is moving in the opposite direction, or it just stops. Thus taking the distance as the only routing criterion is not sufficient and suitable for the city vehicle networks environment.

Second, as compared with routing schemes Prophet and MaxProp, our scheme is more distributed and requires much less information from other nodes. As described, in Prophet, the routing decision for one node is solely based on its contact history with the destination. To maintain its own encountering probability with the destination, each node has to frequently acquire its neighbors' encountering probabilities, whenever there is a chance. Similarly, MaxProp requires sharing every node's contact information among all nodes to maintain the contact graph up to date. Remember that the contact information is always changing as nodes move and meet each other, thus an enormous communication overhead will be introduced, not to mention if the network scale is large such as with thousands of nodes in the city case. To manage and update all the contact information in the local buffer, e.g., frequently searching, updating, etc., also introduces computation overhead. Most existing work did not count it as the scheme overhead since it is on the control level. However, the complexity issue can become severe in the real world.

Our scheme also requires global information, i.e., the macroscopic mobility pattern, however, the city macroscopic mobility pattern stays quite stable over time according to our observation of the trace. Unless there are some major changes happening, e.g., a new highway is built which changes the city traffic dramatically, there is no need of frequently updating the macroscopic mobility pattern. The other information needed in our scheme is the microscopic mobility and local contact history information, which is totally self-maintained by each node. This is a very attractive feature for a large-scale distributed network. GeoMob and GeoMobCon-NoBufMgt are also listed. Comparing to GeoMobCon, they do not have the contact-based routing strategy and buffer management, respectively.

3.5.2 Simulation Setup

We use a Java-based open-source simulator theONE [65] for simulation. To be practical, we randomly select 200 taxis and 300 buses, importing their traces for the node mobility.

According to the specifications of IEEE 802.11p [66, 67], we set the vehicle transmission speed to 6 Mbps. The buffer size is set to $2,000$ MB. We test our protocol with different simulation settings to give a comprehensive understanding of the impact of different factors. For different transmission power and urban environment, we test our protocol performance under different transmission ranges, i.e., ranging

among 50 m, 250 m, 500 m and 1,000 m. Different applications may have different message generation rates and Time-To-Live (TTL) periods, thus we test our protocols with different message generation intervals, varying among 10 s, 60 s and 300 s, and different message TTLs, varying among 60 min, 120 min and 180 min.

Messages are generated randomly among all taxis and the destination region is randomly chosen. We set the simulation time to 24 hours with the first 6 hours as a warm-up period.

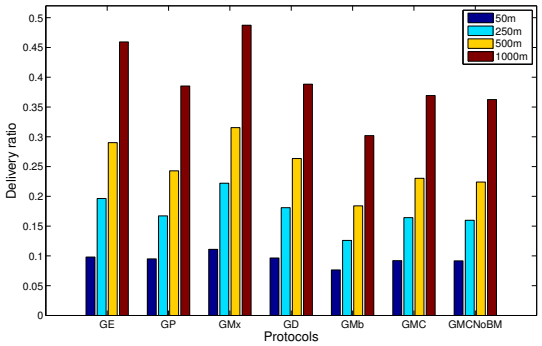
3.5.3 Message Delivery Performance

With different parameter combinations, the performance changes. To understand the overall performance, we study two extreme cases: *pessimistic case*, all parameters are set to restrain the message transmission, saying the transmission range is set to the smallest 50 m, the message generation interval is the lowest 10 s and the TTL is set to be the lowest 60 min; *optimistic case*, all parameters are set to smooth the message transmission, say the transmission rage is set to be the largest 1,000 m, the message generation interval is the largest 300 s and the TTL is set to be the highest 180 min. We are interested in how the system performs under these two extreme conditions.

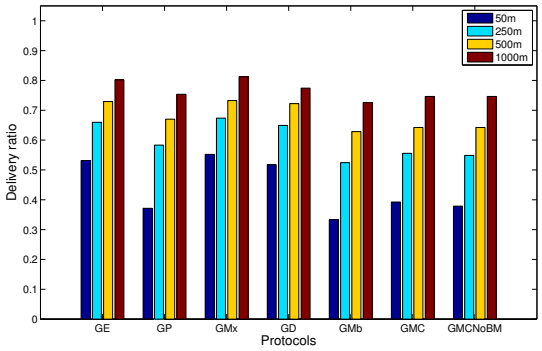
For comparison, four performance metrics are discussed, including: delivery ratio, calculated as $\frac{N_D}{N_G}$, where N_D is the total number of delivered messages and N_G is the total number of generated messages; overhead ratio, calculated as $\frac{N_R - N_D}{N_D}$, where N_R is the total number of message relays; average latency, the average delay for successful deliveries; average hop count, the average hop count for the delivered messages. Because the performance of the geocast is dominated by the geographic forwarding process (i.e., from the source to the target region), the above performance metrics are all calculated for this phase. Once a message reaches the target region, we assume the same flooding scheme will be applied to all schemes within the destination region. So the flooding performance should be the same for all schemes.

Impact of Transmission Range

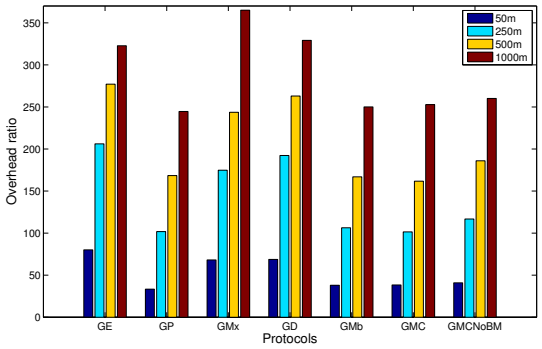
Figure 3.7 depicts the comparison results with the change of the transmission range, with the left column of figures illustrating the results for the pessimistic case and the right one for the optimistic case. As we can see, the delivery ratio, overhead ratio, and average hop count for all protocols in both cases increase with the rise of the transmission range. This is because, the larger transmission range, the further



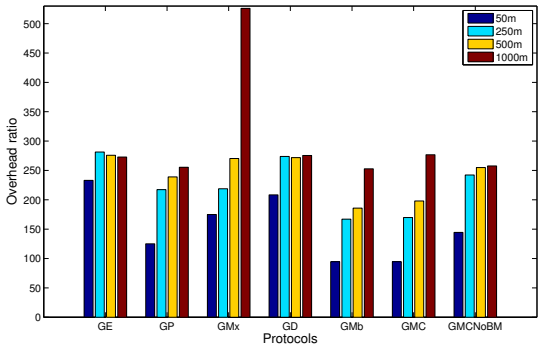
(a) Delivery Ratio, Pessimistic Case



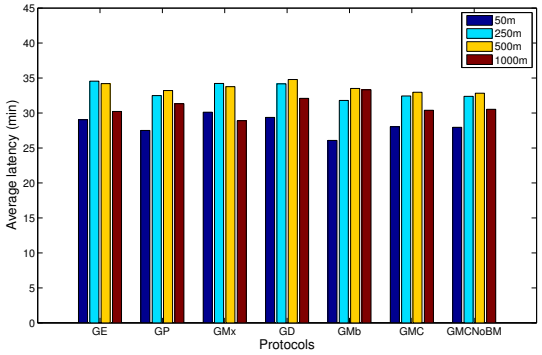
(b) Delivery Ratio, Optimistic Case



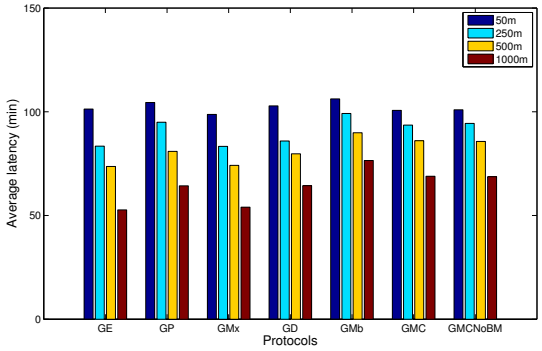
(c) Overhead Ratio, Pessimistic Case



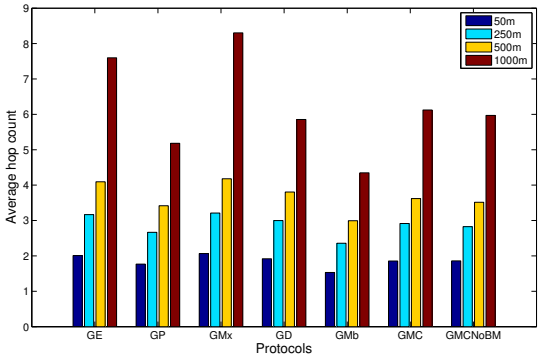
(d) Overhead Ratio, Optimistic Case



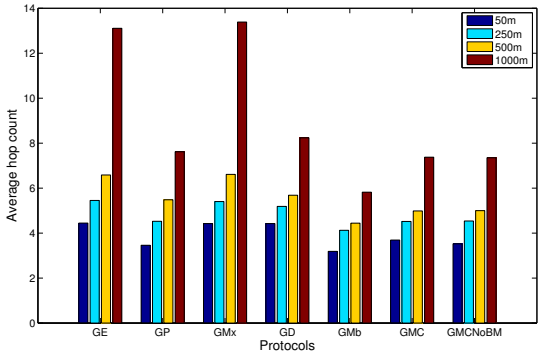
(e) Average Latency, Pessimistic Case



(f) Average Latency, Optimistic Case



(g) Average Hop Count, Pessimistic Case



(h) Average Hop Count, Optimistic Case

Figure 3.7: Effect of Transmission Range on Performance.

the message can be forwarded to within one hop. Thus the message can reach the destination more easily. This leads to the increase of the delivery ratio. With a larger range, more possible relay candidates show up for each transmission, leading to an increase of the overhead. With the increase of the delivery ratio, more messages are delivered to the destination than the scenario with a smaller transmission range. However, many of these messages are delivered through more hops of transmissions, leading to the increase of the average hop count.

In terms of the average latency, for the pessimistic case, we see for all protocols, the latency first increases then decreases. When the transmission range is small, vehicles have much fewer contacts with other vehicles, therefore, a large portion of the delivered messages are those whose destinations are close to their sources. Especially when the TTL is short, which is the fact in the pessimistic case, carry-and-forward is the main approach to deliver these messages. Thus these messages are easily delivered to their nearby destinations with a relatively short latency. But the delivery ratio in such a case is very low. However, with the increase of the transmission range, more messages can be delivered, but with a longer travel distance and a longer latency. Thus the average latency increases. When the transmission range continues increasing, more messages can be delivered with multi-hop transmissions, which is much faster than carry-and-forward transmission, making the average latency drop. On the other hand, for the optimistic case, the average latency decreases with the increase of the transmission range. This is because with a larger TTL (i.e., 180 min for the optimistic case), messages have longer time to be propagated to the destinations, even when the transmission range is small. Then for a larger transmission range, messages get more chance to be delivered, resulting in a shorter latency.

We can also observe that, in terms of the delivery ratio and average latency, for different transmission ranges, our proposed protocol GeoMobCon performs slightly worse than GeoEpidemic and GeoMaxProp. However, we have to emphasize that, unlike GeoMaxProp, utilizing global control information which implies a large control overhead, our protocol only utilizes the local information (mobility and contact information) of each node, making it more scalable and practical. Besides, different from GeoEpidemic and GeoDist with a flooding fashion, our protocol makes smarter relay selection, so that it achieves very low overhead ratio of all seven protocols. Our proposed protocol also achieves a low average hop count. This implies that our protocol requires fewer transmissions, leading to less power consumption and cause less interference. The proposed GeoMobCon also outperforms its precursor, GeoMob, in

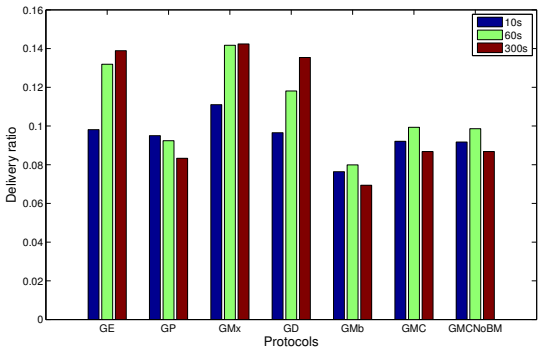
delivery ratio and average latency, but still maintains a similarly low overhead ratio.

Impact of Network Traffic Intensity

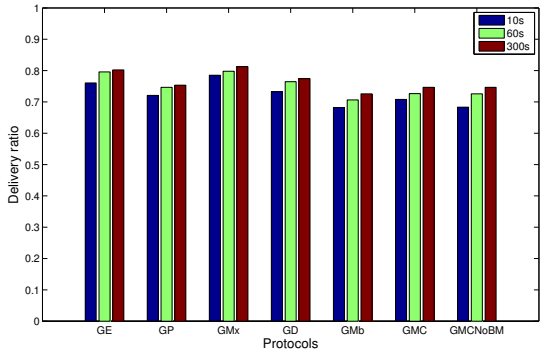
Figure 3.8 shows the impact of the network traffic intensity, which is represented by the message generation interval, from 10 s per message, 60 s per message to 300 s per message. With the increase of the message generation interval (i.e., decrease of the message generation rate), we see dramatic changes in the left column figures (the pessimistic case). With less traffic intensity, there is less competition for message transmissions, leading to the increase of the delivery ratio. However, when the parameter setting is harsh for the message transmission, contact opportunity becomes precious, and mobility-based routing decision making strategy (e.g., GeoMob, GeoMobCon and GeoMobCon-NoBufMgt) can suppress the efficiency of using the contact opportunities, leading to the fluctuation of the delivery ratio, Fig. 3.8(a).

From Fig. 3.8(c) and (d), we see the overhead ratio increases with the increase of the message generation interval. This is because the network overhead is defined as $\frac{N_R - N_D}{N_D} = \frac{N_R}{N_D} - 1$. When the message generation interval increases, the network traffic is reduced. Therefore, with lower traffic competition, each message gets more chance to be transmitted to more vehicles, which increases the total number of relays. The trend for average latency results differs for the pessimistic case and the optimistic case. For the pessimistic case, because of the harsh setting for the message transmission, when the traffic intensity is high, most of the delivered messages are those easy to be delivered, such as those whose source locations are close to the destinations. When the traffic intensity is low, each message has a higher chance to be forwarded through more hops (i.e., the increase of the average hop count and the overhead ratio), leading to the increase of average latency. However, in the optimistic case, because of the larger transmission range and longer TTL, the majority of the messages get delivered. Therefore the delivery ratio, overhead ratio, average latency and average hop count all get increased compared the results of the pessimistic case. Especially for the latency, lower traffic intensity reduces the transmission competition, leading to the decrease of the average latency in Fig. 3.8(f).

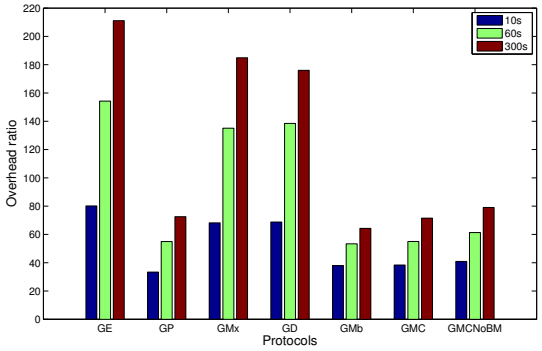
In terms of the protocol comparison, for the similar reasons mentioned before, i.e., being local-information based and more scalable, our protocol performs slightly worse in terms of delivery ratio and average latency than GeoEpidemic and GeoMaxProp, but much better than GeoMob. Besides, its simplicity and scalability make it achieve



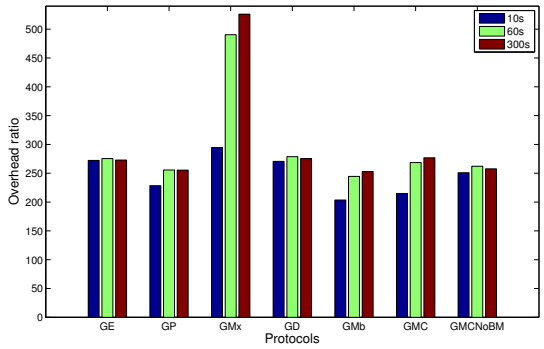
(a) Delivery Ratio, Pessimistic Case



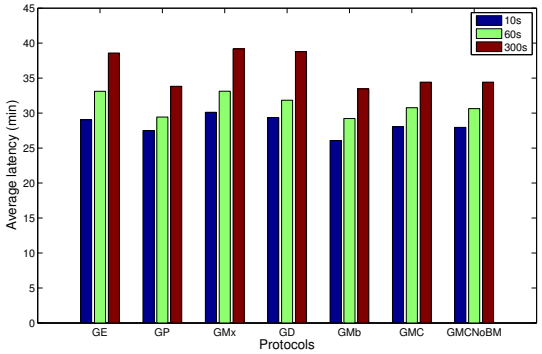
(b) Delivery Ratio, Optimistic Case



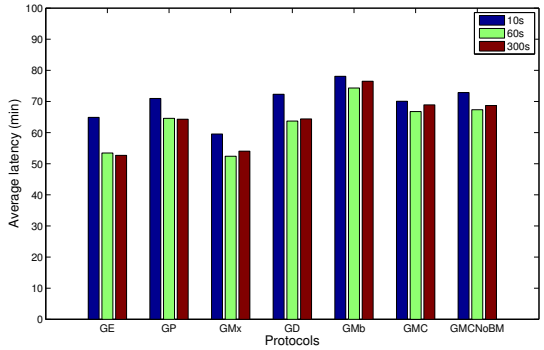
(c) Overhead Ratio, Pessimistic Case



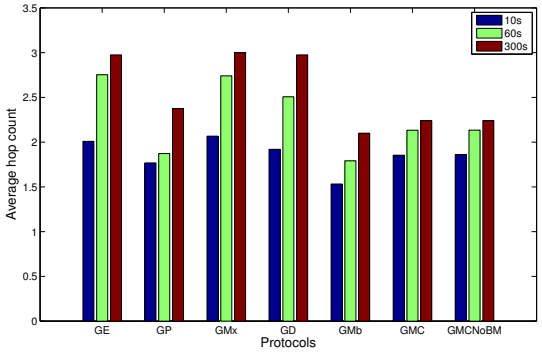
(d) Overhead Ratio, Optimistic Case



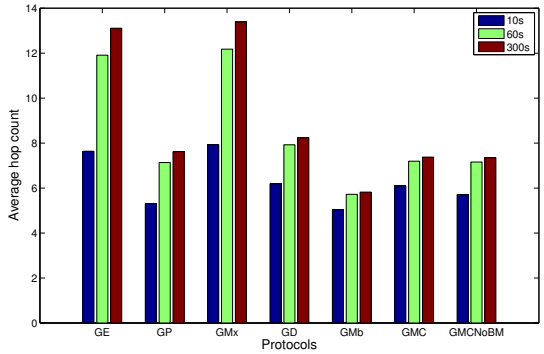
(e) Average Latency, Pessimistic Case



(f) Average Latency, Optimistic Case



(g) Average Hop Count, Pessimistic Case



(h) Average Hop Count, Optimistic Case

Figure 3.8: Performance with Network Traffic.

the lower overhead ratio and average hop count than GeoEpidemic and GeoMaxProp.

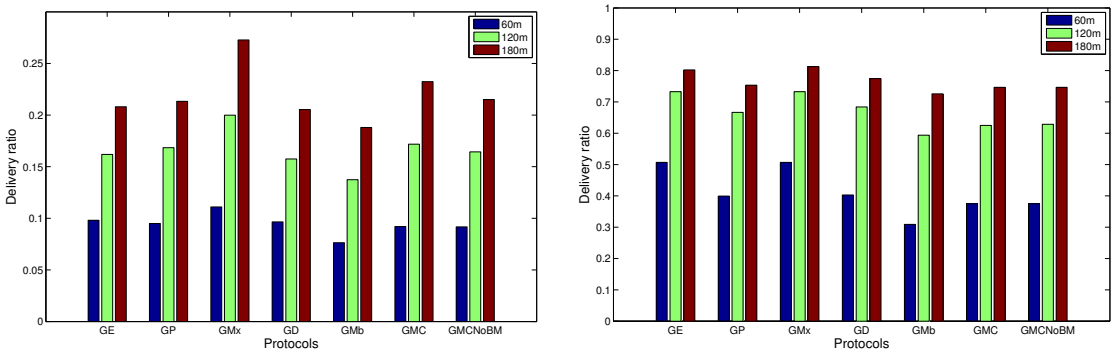
Impact of Message TTL

The impact of TTL is demonstrated in Fig. 3.9. With a longer TTL, messages have longer time to stay in the network and be transmitted, leading to a higher delivery ratio, larger average latency and average hop count. With more messages delivered, the delivery efficiency becomes higher, leading to the drop of the overhead ratio, shown in Fig. 3.9(c) and (d). Comparing the two cases, with less congested traffic and larger transmission range, the results of the optimistic case show a higher delivery ratio, overhead ratio and average hop count. However, the average latency is reduced since the optimistic case has more contact opportunities for the message propagation. To compare the performance of different protocols, similar conclusions as previous discussions on the other impacting factors can be drawn.

Evaluation of Proposed Design Components

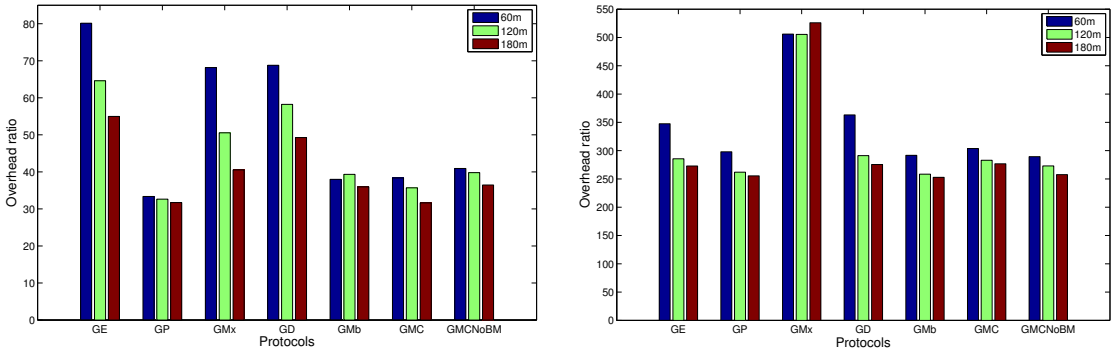
The GeoMob, GeoMobCon and GeoMobCon-NoBufMgt are all the mobility-based routing protocols. Comparing with other protocols, the mobility-based protocols are featured with the low overhead and highly distributed manner. GeoMob is the original protocol proposed in our previous work [14], including two major components: mobility-based routing strategy and buffer management. On top of it, we propose its extended version GeoMobCon, considering the contact history for routing. As we can see from Fig. 3.7, 3.8, 3.9, in most scenarios, GeoMobCon achieves the same low overhead ratio as GeoMob, which is the lowest of all. But at the same time, it outperforms GeoMob in terms of delivery ratio and average latency. The improvement shows the advantages of the contact history-based routing strategy. Because the contact history of each vehicle is also self-maintained, so it does not conflict with the highly-distributed feature.

On the other hand, when comparing the GeoMobCon with GeoMobCon-NoBufMgt, we can see the benefit of the buffer management. From the figures, we can observe that, GeoMobCon-NoBufMgt performs very close to GeoMobCon in different scenarios. However, in optimistic cases, GeoMobCon has less overhead and higher delivery ratio than the GeoMobCon-NoBufMgt. This is mainly because, buffer management can effectively remove the redundant messages (i.e., the messages with low delivery likelihood and the extra copies of the delivered messages) in the system. This ad-



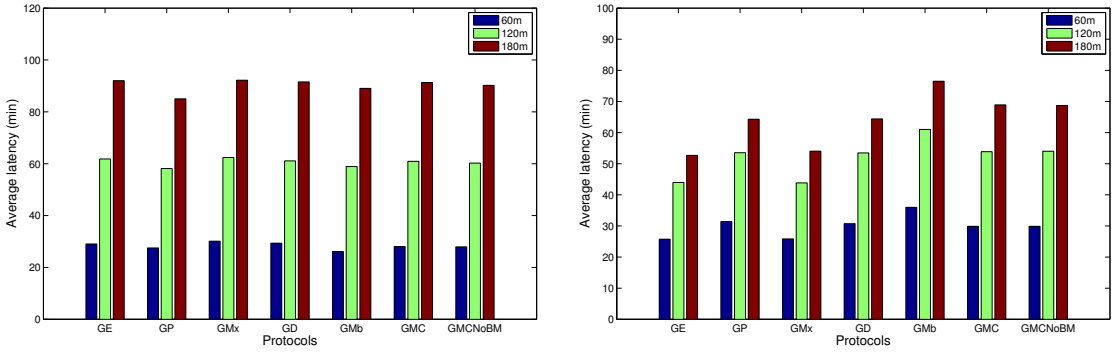
(a) Delivery Ratio, Pessimistic Case

(b) Delivery Ratio, Optimistic Case



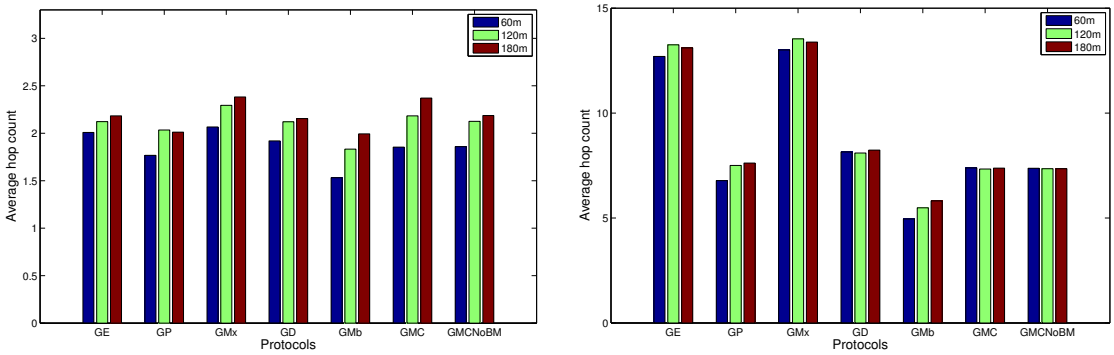
(c) Overhead Ratio, Pessimistic Case

(d) Overhead Ratio, Optimistic Case



(e) Average Latency, Pessimistic Case

(f) Average Latency, Optimistic Case



(g) Average Hop Count, Pessimistic Case

(h) Average Hop Count, Optimistic Case

Figure 3.9: Performance with TTL.

vantage is more apparent for optimistic cases. Without removing those redundant messages, the number of total message relays easily gets high, leading to a high overhead. By prioritizing the transmission queue, it also helps to utilize the transmission opportunities more effectively to achieve a better delivery ratio.

3.6 Conclusions

In this chapter, an advanced mobility-contact-aware geographic routing scheme (GeoMobCon) is proposed for large-scale urban VANETs. This new scheme has many new features compared with the existing geocast schemes: first, it is designed from a DTN perspective, enabling it to deal with the high mobility and transient connectivity in VANETs; second, vehicle mobility information (with different levels) and contact history information is employed, making GeoMobCon distributed, simple, scalable and communication and computation-effective; third, practical issues are well considered by introducing the real-world trace analysis, trace-driven simulations and efficient buffer management. Extensive performance comparisons with other protocols have shown its great advantages.

Chapter 4

Vehicular Message Dissemination in Two-Dimensional City Blocks

4.1 Overview

As mentioned in the previous chapters, in geocast, messages are forwarded towards specific geographic destination regions and broadcast within the regions. In Chapter 3, we focused on the message propagation process towards the target region. In this chapter, we focus on the propagation within the target region. Because the simple broadcast is performed, we put our focus on the broadcast efficiency analysis by studying the network connectivity from the broadcast source location to any destination location within the target region.

4.2 Related Work

4.2.1 Directed Percolation

Percolation has been observed in many natural and man-made systems, initially motivated by the process of liquid filtering through porous materials [23]. The process can be modeled by vertexes (sites) and edges (bonds) in certain dimensions, and depending on whether to occupy a site or bond, two processes are defined: site or bond percolation. In a homogeneous bond percolation, the bond probability $p \in [0, 1]$ connecting two neighbor vertexes (e.g., the liquid filters through the bond) is considered. Assuming an infinite number of vertexes and edges, percolation occurs when

there exists an infinite connected giant component (and an infinite number of finite components). Percolation is more likely to occur with a larger p, so when p reduces from 1 to 0, percolation either occurs or not, exhibiting a sharp phase transition at the so-called *critical* probability p_c.

A rich set of research outcome has appeared for various percolation scenarios in different dimensions. Depending on whether the bonding between neighbor vertexes is directional or not, percolation can be further classified into (isotropic) percolation and directed percolation (DP). Although mostly by numerical approaches, the critical probabilities for many lattice (discrete) and continuum models have been found or approximated. For example, for 2D square lattices, p_c is 0.5 by proof for bond percolation, and 0.59 by approximation for site percolation. For other 2D regular tiling, both the bond and site percolation thresholds can be determined for triangles, but only the bond percolation threshold for hexagons. However, the *directed* version of them turns out to be much harder and only numerical results are available, even for regular triangles, squares and hexagons. Besides critical probabilities, the convergence behaviors around p_c are also heavily investigated.

The most related work is the directed percolation on a square lattice where the vertical (or one of the two dimensions) bond probability is p_y and the horizontal (i.e., the other dimension) bond probabilities are 1 and p_x interleaved at different layers [24]. By defining the "wet" and "primary wet" edges on each layer, the critical probability is found analytically when the connectivity from the origin to the farthest vertex in a finite square lattice of a given aspect ratio α is transitioning from 0 to 0.5 and to 1. When $p_x = 1$, this model degenerates into a quasi-2D model with results known long ago. Similarly, when $p_x = 0$, it degenerates into the quasi model with vertical bond probability p_y^2. Most recently, the author also discussed the convergence behaviors of such an interleaved model [68].

4.2.2 Connectivity in Ad Hoc Networks

On the other hand, connectivity has also been extensively studied in ad hoc networks, mostly in the 2D Euclidean spaces [15–19]. Without fixed communication infrastructures, nodes in ad hoc networks have to rely on their neighbors or leverage the mobility of them to deliver messages to the destination, often in a multi-hop manner through wireless communications and/or short-range contacts, so the connectivity has to be characterized probabilistically. A wide variety of ad hoc networks

exist, ranging from stationary (sensor networks [69]) to mobile ones. VANET is a special type of the latter, where vehicles are involved as the communication source, destination and relay [25, 70, 71]. High vehicle velocity introduces more challenges to connectivity, but the predictable mobility also offers new opportunities. For example, along a highway, vehicles travel in one dimension, possibly also communicating with the vehicles on the reverse direction. In a city block scenario, a 2D square lattice is often used to approximate the road grid.

Analytical and algorithmic tools in graph theory and computational geometry have been widely used in the modeling and analysis of connectivity in ad hoc networks [72], together with geometrical probability, stochastic geometry, and percolation theories in recent years. For example, a connected dominating set is introduced in ad hoc networks to create a virtual backbone for the network [73]. Geometrical probability tools offer the characterization of distance distributions among nodes in and between different geometry shapes (e.g., triangles, rectangles and hexagons [74–76]), and stochastic geometry tools further introduce the time line in the random process of node coverage and connectivity [77]. Additional nodes can be deployed, some even mobile, to improve the connectivity.

Most recently, percolation theory has found a wide range of applications in networking research, particularly on the connectivity in ad hoc networks [78–80]. Many networking scenarios can be adequately modeled as percolation on square lattices, either individually (e.g., VANET in city blocks) or after clustering and aggregation (cluster heads in wireless sensor networks). Although square lattices are most widely used, other 2D regular tilings can also be used (e.g., hexagons for cellular systems and rhombuses or triangles for cells with directional antennas). For messages with a given destination, or vehicles traveling in certain directions, geographical forwarding is often used to minimize the network overhead due to flooding [25]. Thus directed percolation becomes a premier model in such scenarios, and most existing work applies the results from isotropic or directed percolation on square lattices.

This dissertation studies the directed connectivity (DC) problem on square lattices, motivated by the VANET connectivity in city scenarios. We first try to establish the analytical expression for the directed connectivity from a given message source to any possible destinations in the network. This problem is related to DP but more microscopic, as DP is only concerned about the existence of an infinite giant component, but the coexistence of an infinite number of finite components also indicates (although not characterizing quantitatively) that some destinations are not connected. To the

best of our knowledge, this is the first analytical result in the literature, other than our previous work on a special case (i.e., 2D ladders [25]). Although we assume an identical bond probability in this dissertation, our work can be easily extended to more general DC problems with variable bond probabilities. With the analytical results on the DC problem, we also hope to shed some new light on the half-century old DP problem.

4.3 Message Propagation Model

In an urban VANET system, many applications are based on message broadcast, e.g., collision or traffic congestion messages can be propagated to notify drivers blocks away to detour well in advance; parking lots, hotels and restaurants can advertise their availability to potential customers, reducing the extra time and fuel wasted when the drivers are looking for empty spots. The efficiency of the broadcast is greatly impacted by the network connectivity with multi-hop relaying from the message source to the destination. Because message relaying only happens when the relay node falls within the transmission range of the transmitting node, the vehicle density along the road highly determines the multi-hop relaying efficiency.

For simplicity and versatility purposes, we consider the most typical and general urban city structure, i.e., Manhattan-like city structure, which is composed of horizontal and vertical streets. Such a structure can be modeled as a square lattice where each intersection is a site in the lattice and each road segment is the bond between sites. Vehicles move in two directions on each road segment between any two adjacent intersections. We assume that the inter-vehicle distances follow an independent and identical distribution which can be derived mathematically or obtained empirically through measurement. Recent work [70, 81] made statistic analysis of empirical data collected from real world and found that an exponential distribution, taking vehicle density as the only parameter, can well capture the characteristics and variation of the vehicle traffic in terms of the inter-vehicle distance and inter-contact time [5]. Considering the same speed for vehicles in the same road segment, the exponential model for inter-vehicle distance is actually equivalent to that the count of passing vehicles at a certain road point follows a Poisson process. In other words, we say the arrival of vehicles at the observation point follows the Poisson process.

By the multi-hop transmission of vehicles, the probability for a message reaching one intersection from its adjacent preceding intersection is denoted as p, i.e., the bond

probability. Our previous work [25] studied the derivation of p. A "connected vehicle cluster" along a road was first investigated, within which all vehicles can reach each other by wireless transmissions and relays. With the assumption of the exponentially distributed inter-vehicle distances, the distribution of the cluster size, i.e., the distance from the first vehicle to the last vehicle in the same cluster, is approximated with a Gamma distribution. Therefore, bond probability p can be modeled as the probability that the cluster size is larger than the distance between two adjacent intersections. In the derivation of p, several factors are considered, including the message transmission range, the distance between neighbor intersections (i.e., the block size), vehicle density and shadowing effect. Once we have p, the connectivity of any two intersections in such an urban VANET can be explored, previously by simulation in [25], and now with the analytical expressions obtained by the new approach.

Depending on the nature of a message, the message may be propagated among vehicles in the same street (e.g., highway), which is the one-dimensional connectivity on a line. If the message can be disseminated to all the intersections in the downtown area (e.g., parking availability), then the lattice connectivity applies. In the following section, we assume the message origins at the intersection $(0, 0)$ and the connectivity to all other intersections will be obtained using our approach.

4.4 Connectivity Analysis on Square Lattice

In this section, we first give the system model for directed connectivity, and then present the analytical framework and derivation results for lattices with size varying from $1 * 1$ and $2 * 2$ to $m * n$.

4.4.1 System Model

As shown in Fig. 4.1(a), we consider a 2D lattice $L(m, n)$, with edges parallel to the x and y axes for notation convenience. A message is generated at the origin $O = (0, 0)$ at time $t = 0$, and propagated along the lattice edges in the directions indicated by arrows. Assuming the bond probability p of any two neighbor vertices, what we want to know is the connection probability from the origin to (m, n), as a function of p. The derivation of bond probability p varies in different applications. For instance, in Section 4.6, we consider p as the probability of any two adjacent road intersections in urban areas being connected and the detailed calculation could be referred to [25]. To

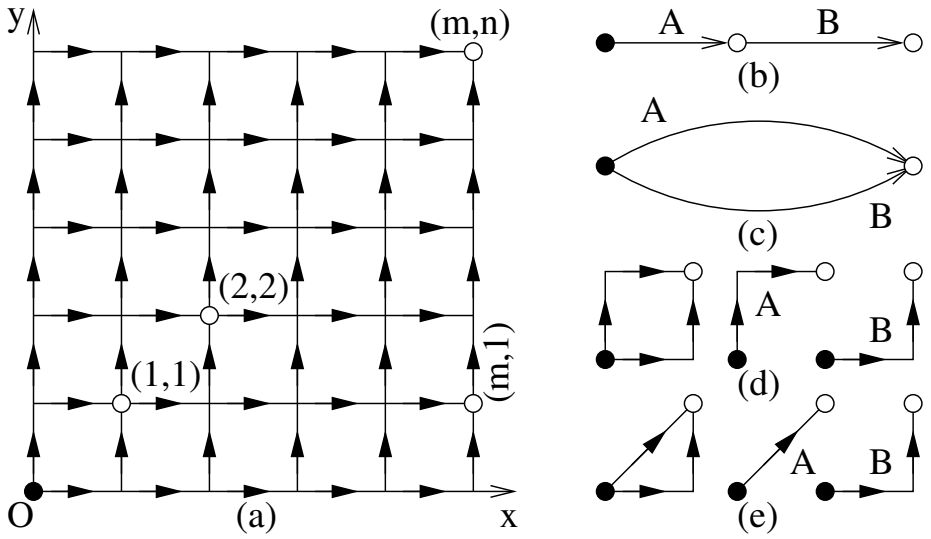


Figure 4.1: System Model and Basic Principles.

be more general, bond probabilities in a lattice network could be different from edge to edge. But for the simplicity of presenting our approach, we assume a homogeneous p in this section. Heterogeneous case is presented as an extension in Section 4.6.

Even with the simplified model, it is still a hard problem to derive the connection probability at vertex (m, n), denoted as $P(m, n)$. For example, to reach (m, n), the message can go through $(m-1, n)$ or $(m, n-1)$ as the last hop. However, even if $P(m-1, n)$ and $P(m, n-1)$ were known, it is still difficult to derive $P(m, n)$, as the paths from $(0, 0)$ are not independent before they reach the last hop. A brute-force approach has to enumerate all possible paths and overlapping (i.e., when different paths share the same edges) and its complexity suffers the combinatorial explosion on the exponent. This is also the reason why the exact result of DC remains unsolved for so many years.

To facilitate the presentation, we also illustrate some basic principles and simple cases in Fig. 4.1. First, if there are two directed paths A and B connected by a common vertex *serially* as shown in Fig. 4.1(b), the end-to-end connectivity is $P(AB) = P(A)P(B)$, as A and B are always independent with directed edges. Here, we define $P(A)$ and $P(B)$ as the probabilities that path A and B are connected, respectively. Second, if there are two *parallel* paths A and B connecting the source and destination as shown in Fig. 4.1(c), the source-to-destination connectivity is $P(A+B) = P(A)+P(B)-P(AB)$ according to the Principle of Inclusion and Exclusion (PIE). These two principles can be used to solve the $1*1$ lattice problem as shown

in Fig. 4.1(d): $P(A) = P(B) = p \cdot p = p^2$, and $P(1,1) = P(A) + P(B) - P(A)P(B) = 2p^2 - p^4$ as A and B are independent and not mutually exclusive. Later we find that we also encounter a triangular cell as shown in Fig. 4.1(e), and the end-to-end connectivity in this case is $P_T(1,1) = p + p^2 - p \cdot p^2 = p + p^2 - p^3$. The cases become more complicated when A and B are also dependent.

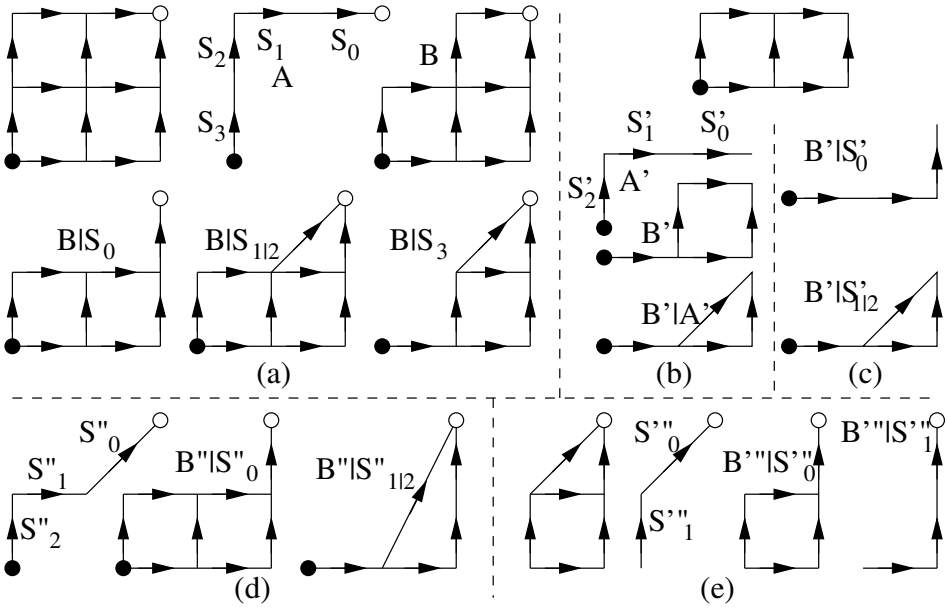


Figure 4.2: The Decomposition of a $2 * 2$ Lattice.

4.4.2 $2 * 2$ Square Lattices

Following the same principles, we attempt to solve the $2 * 2$ lattice problem as shown in Fig. 4.2. Similarly, the top-leftmost path A and the union of all other paths, $\mathcal{B}$, are identified. But the difference from Section 4.4.1 is that they are no longer independent (as A and $\mathcal{B}$ have many overlapping edges). A naive approach is to consider each edge along A separately and check the impact of their connection status on $\mathcal{B}$. Depending on each edge being connected or not, there are 2^4 cases of a single path A and many more cases will be introduced in $\mathcal{B}$. If using the PIE principle, after the first level of decomposition, to further decompose $\mathcal{B}$, which has more than two layers, is very difficult, if not impossible. This is because most edges, other than the bottom-rightmost two, are shared by many paths. This also renders our previous approach [25] on one-layer ladders not applicable to multi-layer lattices. Observing A is a single path (i.e., no branches possible), we can have a simple partition on it. As

shown in Fig. 4.2(a), define S_i as the event that the last i edges along A leading to the destination are all connected, but the last $(i+1)$-th one is not, so $P(S_i) = p^i(1-p)$ for $0 \leq i \leq m+n-1$. For the origin and destination to be connected, we then have $m+n+1$ mutually exclusive cases, including $\mathcal{B}|S_i$ and A being connected where $P(A) = p^{m+n}$. Define the probability that $\mathcal{B}$ is connected given S_i as $P(\mathcal{B}|S_i)$, we have

$$\begin{aligned}
P(m,n) &= P(A+\mathcal{B}) = 1 - P(\overline{\mathcal{B}+A}) \\
&= 1 - P(\bar{\mathcal{B}}\bar{A}) && (4.1) \\
&= 1 - P(\bar{\mathcal{B}} \bigcup_{i=0}^{m+n-1} S_i) && (4.2) \\
&= 1 - \sum_{i=0}^{m+n-1} P(\bar{\mathcal{B}}|S_i)P(S_i) && (4.3) \\
&= 1 - \sum_{i=0}^{m+n-1} (1 - P(\mathcal{B}|S_i))P(S_i) \\
&= P(A) + \sum_{i=0}^{m+n-1} P(\mathcal{B}|S_i)P(S_i), && (4.4)
\end{aligned}$$

where (4.1) is due to De Morgan's law, (4.2) due to $\bigcup_{i=0}^{m+n-1} S_i = \bar{A}$, (4.3) due to S_i being mutually exclusive, and (4.4) due to $\sum_{i=0}^{m+n-1} P(S_i) = P(\bar{A}) = 1 - P(A)$, i.e., A and S_i partition and constitute the entire event space in *total probability*.

For $L(2,2)$, given S_0, no end-to-end connection is possible via vertex $(0,2)$ or $(1,2)$, so we can discard the edges adjacent to the two vertices and have $\mathcal{B}|S_0$ as shown in Fig. 4.2(a). Given S_1, it implies that $(1,2)$ and $(2,2)$ are connected, and $\mathcal{B}$ does not include any edges from $(0,2)$, so we can merge $(1,2)$ with $(2,2)$ in $\mathcal{B}$ to obtain $\mathcal{B}|S_1$. Since S_1 and S_2 have the same effect, they are illustrated as $\mathcal{B}|S_{1|2}$ in Fig. 4.2(a). Given S_3, no connection is possible through $(0,1)$, so the edges adjacent to it have to be removed; it also implies that $(0,1)$, $(0,2)$, $(1,2)$ and $(2,2)$ are connected sequentially, so they can be merged, as $\mathcal{B}|S_3$ in Fig. 4.2(a).

After this decomposition, we have $\mathcal{B}|S_{0..3}$. Using the serial principle, $B|S_0$ can be decomposed into a $2*1$ lattice (or ladder) and an edge. Figure 4.2(b) shows how we further decompose the ladder into A', $\mathcal{B}'$ and $\mathcal{B}'|A'$ following the conditional probability approach that we previously proposed for ladders specifically [25], while Fig. 4.2(c) shows the new total probability approach with $\mathcal{B}'|S'_{0..2}$, which can both be solved directly using the serial principle, $P(1,1)$ and $P_T(1,1)$: the results are the same, but

the new approach is simpler, especially when we have multi-layer lattices. Similarly for $\mathcal{B}|S_{1|2}$, they are decomposed in Fig. 4.2(d) to components of known connectivity (e.g., $\mathcal{B}''|S_0''$ is the same as $\mathcal{B}|S_0$), and part of $\mathcal{B}|S_3$ is decomposed in Fig. 4.2(e), where the serial principle and $P(1,1)$ can be applied. Using the total probability approach, the connectivity of the decomposed components can be reassembled,

$$P(A) = p^4,\ P(\mathcal{B}|S_0) = p^8 - p^7 - 2p^6 + 3p^4,$$
$$P(\mathcal{B}|S_{1|2}) = -p^9 + 3p^8 - 3p^6 - 3p^5 + 3p^4 + 2p^3,$$
$$P(\mathcal{B}|S_3) = p^7 - 2p^6 - p^5 + 2p^4 + p^3.$$

Then $P(2,2)$ can be recovered as follows

$$\begin{aligned} P(2,2) &= P(A) + \sum_{i=0}^{3} P(\mathcal{B}|S_i)P(S_i) \\ &= p^{12} - 4p^{11} + 2p^{10} + 4p^9 + 2p^8 - 4p^7 - 6p^6 + 6p^4. \end{aligned}$$

4.4.3 $m * n$ Square Lattices

Following the new total probability approach, we attempt to solve the generic $m * n$ lattice problem as shown in Fig. 4.3(a). For clarity, we have omitted the arrow on edges in the following figures. Similar to $\mathcal{B}|S_i$ in Section 4.4.2, we can first remove the edges or merge the vertices on the top row of an $m * n$ lattice, as $\mathcal{B}|S_{0..m-1}$, and then remove the edges or merge the vertices along the leftmost column of the lattice, as $\mathcal{B}|S_{m..m+n-1}$, by considering the top-leftmost path A and events $S_{0..m+n-1}$, as well as their impacts on $\mathcal{B}$.

After this decomposition, similar to Section 4.4.2, we need to further decompose the components eventually to the ones of known connectivity. For example, $\mathcal{B}|S_0$ contains an $m*(n-1)$ lattice and an edge, which leads to a recursion among lattices. $\mathcal{B}|S_{m+1}$ is further decomposed in Fig. 4.3(b) and similarly for all the other $\mathcal{B}|S_i$s. Although all $\mathcal{B}|S_i$s and their decomposed components have different structures, we have found the similarities between these structures during the decomposition process, and we can introduce a generic structure called *Tower* to formulate the recursions among them.

Figure 4.4(a) shows the generic structure of Tower $\mathcal{T}$. All the decomposed components of the $m*n$ lattice, plus the lattice itself, can be regarded as a special case of

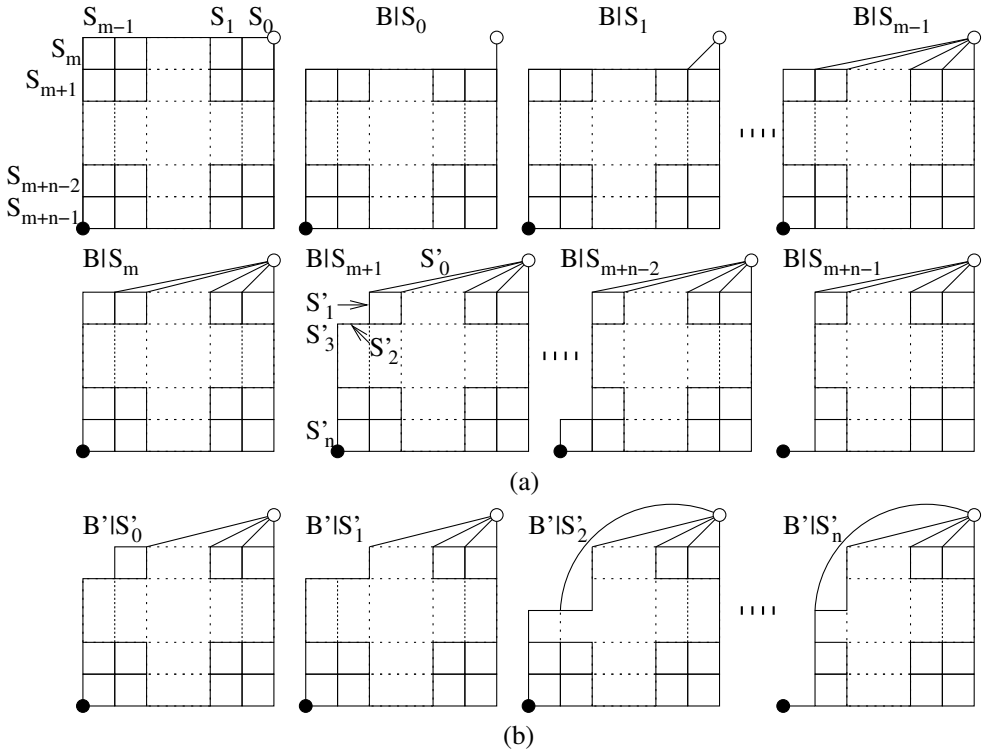


Figure 4.3: The Decomposition of an $m * n$ Lattice.

$\mathcal{T}$. $\mathcal{T}$ has a layered structure, with more blocks near the bottom, as we remove edges and merge vertices gradually along the top-leftmost portion of the tower. The source is the bottom-leftmost vertex at $(0, 0)$, and the destination is the top-rightmost one at (m, n). On each layer, there are two types of building blocks: *triangles* and *rectangles*. Each triangle $\triangle$, highlighted in blue on layer K in Fig. 4.4, is composed of two shortcuts to (m, n) and one or two ordinary lattice edges. A rectangle $\square$, highlighted in red, is the block originally in the lattice and not affected by the decomposition process yet. Depending on the number (t_i) of $\triangle$s and that (r_i) of $\square$s, we can denote layer i by (t_i, r_i), except for the base b which is represented by the number of the bottom edges. Taking into account all the layers in a configuration for the tower, we can denote it by $\mathcal{T}((t_0, 0), \sqcup_{i=1}^{K}(t_i, r_i), \sqcup_{i=K+1}^{N}(0, r_i), b)$, where $\sqcup$ represents a series of layers. Be aware that we distinguish four types of layers: 1) the top layer 0 with $\triangle$s only; 2) the mixed ones of both $\triangle$s and $\square$s from layer 1 to K; 3) the ones with $\square$s only from layer $K+1$ to N; 4) the base b. For example, $\mathcal{T}(\sqcup_{i=0}^{n-1}(0, m), m)$ is the original $m*n$ lattice, $\mathcal{T}((1, 0), \sqcup_{i=1}^{n-1}(0, m), m)$ is the $\mathcal{B}|S_1$ in Fig. 4.3, $\mathcal{T}((m-1, 0), (0, m-1), \sqcup_{i=2}^{n-1}(0, m), m)$ is $\mathcal{B}|S_{m+1}$, and $\mathcal{T}((m-1, 0), \sqcup_{i=1}^{n-1}(0, m-1), m))$ is $\mathcal{B}|S_{m+n-1}$.

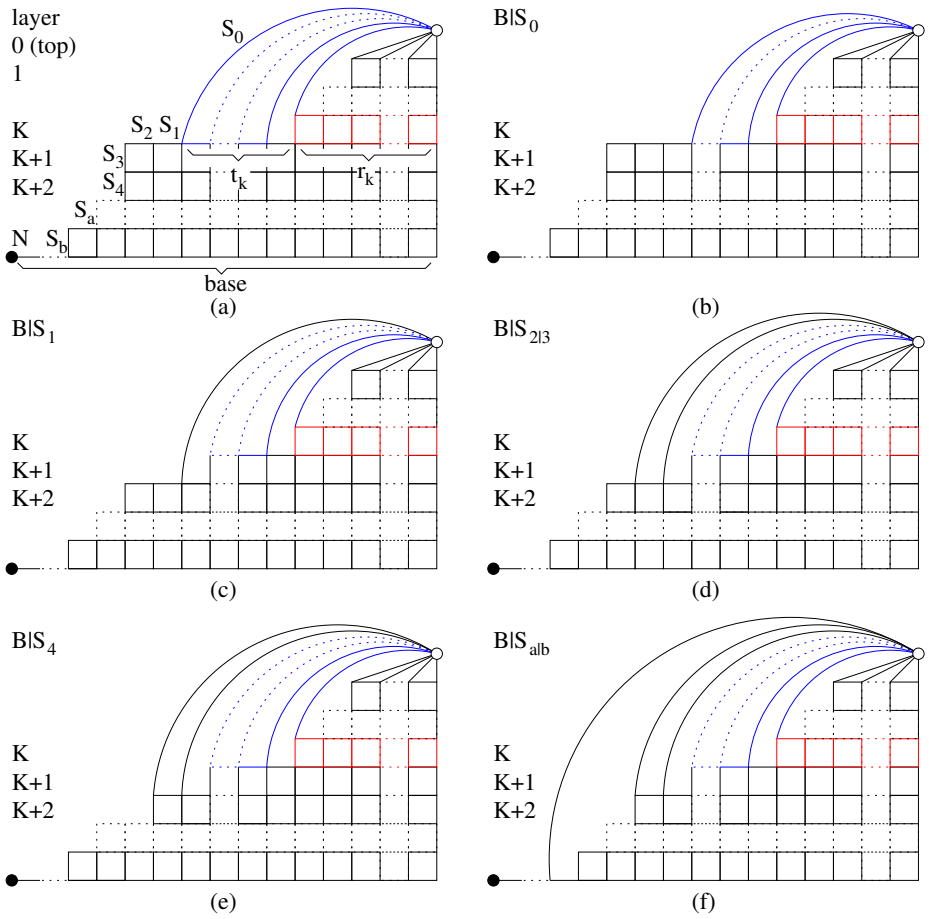


Figure 4.4: The Decomposition of a *Tower*.

For a generic tower as shown in Fig. 4.4, we can identify the top-leftmost path as A and a series of events S_i. Recall that S_i means the last i edges along A leading to (m, n), including the original lattice edges (either horizontal or vertical) and shortcut edges, are connected, but the last $(i+1)$-th one is not. Let s_i represent the last broken edge toward (m, n) and the edge s_i corresponds to the event S_i. It is important to recall that the decomposition happens serially from event S_0 to S_b, corresponding to Fig. 4.4(a). Each decomposition (e.g., S_i's) is performed on the tower obtained from the previous decomposition (i.e., S_{i-1}'s). Essentially we have four types of edges along A: 1) one shortcut edge (e.g., s_0 in Fig. 4.4(a)) on layer K; 2) horizontal edges (e.g., s_1 and s_2), of which there are $r_{i+1} - t_i - r_i$ on each layer for $i \geq K$; 3) at most one vertical and topmost corner edge (s_3 or s_a or s_b) on each layer for $i > K$ and $r_i > t_{i-1} + r_{i-1}$; 4) at most one vertical but not topmost edges (s_4) on each layer for $i > K$ and $r_i = r_{i-1}$. In the following, we will show how each type of S_i can reduce a

tower to another of less complexity.

The Shortcut Edge along A

For example, s_0 is in A but not in $\mathcal{B}$, so whether it is broken or not does not affect $\mathcal{B}$, and $\mathcal{B}|S_0 = \mathcal{B}$ as shown in Fig. 4.4(b). Using the tower notation, $\mathcal{T}(\cdots, (t_K, r_K), \cdots) \xrightarrow{S_0} \mathcal{T}'(\cdots, (t_K - 1, r_K), \cdots)$ with the absence of the shortcut edge at layer K, and no changes in other layers, so the tower complexity is reduced.

The Horizontal Edges along A

For s_1, it is in both A and $\mathcal{B}$, and if it is broken but s_0 is connected, it will remove all the horizontal edges left to it on layer K and introduce a shortcut to (m, n) directly on layer $K + 1$, as shown in Fig. 4.4(c). Using the tower notation, $\mathcal{T}(\cdots, (t_K, r_K), (0, r_{K+1}), \cdots) \xrightarrow{S_1} \mathcal{T}'(\cdots, (t_K - 1, r_K), (1, t_K + r_K - 1), \cdots)$, i.e., one $\triangle$ on layer K is removed, but one $\triangle$ on layer $K+1$ is introduced. However, the number of $\square$s on layer $K+1$ has been reduced to $r_K + t_K - 1$. Recall that $r_K + t_K \leq r_{K+1}$ for a valid tower, the tower complexity is reduced overall as well.

For s_2, if it is broken but s_1 and s_0 are connected, it will also remove all the horizontal edges left to it on layer K and introduce a shortcut on layer $K + 1$, as shown in Fig. 4.4(d). In fact, all horizontal edges along A will have the same behavior, and since they always remove at least one $\square$ on the next layer and only introduce one $\triangle$ on the next layer, therefore, the tower complexity keeps decreasing with $s_{1|2}$-like edges.

The Vertical and Topmost Corner Edges along A

For s_3 on layer $K + 1$, if it is broken but $s_{0..2}$ are connected, it will have the same effect as S_2, since s_3 is the topmost edge of a vertical path segment and there are no branches between s_2 and s_3, so the reduction is shown as $\mathcal{B}|S_{2|3}$ in Fig. 4.4(d).

The Vertical but Not Topmost Edges along A

For s_4 on layer $K + 2$, if it is broken but $s_{0..3}$ are connected, it will remove a $\square$ from the same layer, without introducing any $\triangle$ in any layer, as shown in Fig. 4.4(e). Using the tower notation, $\mathcal{T}(\cdots, (0, r_{K+2}), \cdots) \xrightarrow{S_4} \mathcal{T}'(\cdots, (0, r_{K+2} - 1), \cdots)$. In fact, all vertical but not topmost edges along A will have the same behavior, and

since they always remove one $\square$ without introducing a $\triangle$, the tower complexity is further reduced.

The Base

On the base line, serial and parallel principles can be applied to reduce the tower complexity. For example, as shown in Fig. 4.4(f), if layer N has t_N $\triangle$s and r_N $\square$s, it implies that the base layer has $b-(t_N+r_N)$ edges along a single path of connectivity $p^{b-(t_N+r_N)}$, so $P(\mathcal{T}(\cdots,(t_N,r_N),b)) = p^{b-(t_N+r_N)}P(\mathcal{T}'(\cdots,(t_N,r_N),t_N+r_N))$ using the serial principle. For the t_N $\triangle$s, each of them implies two parallel paths: one by the shortcut to the destination directly, and another through a horizontal edge and then a smaller tower. Since these two paths are mutually exclusive, the PIE principle applies as $P(\mathcal{T}'(\cdots,(t_N,r_N),t_N+r_N)) = p+(1-p)P(\mathcal{T}''(\cdots,(t_N-1,r_N),t_N+r_N))$. The PIE principle can be applied repeatedly until the base becomes $\triangle$ free. After that, another top-leftmost path A' and layer K' can be identified and the above procedures can be repeated to further reduce the tower complexity, until the decomposition leads to the components of known connectivity.

The Overall Recursion

According to Fig. 4.3 and Eqn. (4.4), the recursion process can by summarized using the following theorem:

Theorem 1. *For a lattice with size $m*n$, the connection probability at vertex (m,n),*

$$P(m,n) = P(A) + \sum_{i=0}^{m+n-1} P(\mathcal{B}|S_i)P(S_i),$$

where $P(S_i) = p^i(1-p)$ *and* $P(\mathcal{B}|S_i)$ *are*

$$\begin{aligned}
P(\mathcal{B}|S_0) &= p \cdot P(m, n-1), \\
P(\mathcal{B}|S_1) &= P(\mathcal{T}((1,0), \sqcup_{i=1}^{n-1}(0,m), m)), \\
&\cdots \\
P(\mathcal{B}|S_{m-1}) &= P(\mathcal{T}((m-1,0), \sqcup_{i=1}^{n-1}(0,m), m)), \\
P(\mathcal{B}|S_m) &= P(\mathcal{T}((m-1,0), \sqcup_{i=1}^{n-1}(0,m), m)), \\
P(\mathcal{B}|S_{m+1}) &= P(\mathcal{T}((m-1,0), (0,m-1), \sqcup_{i=2}^{n-1}(0,m), m)), \\
&\cdots \\
P(\mathcal{B}|S_{m+n-2}) &= P(\mathcal{T}((m-1,0), \sqcup_{i=2}^{n-2}(0,m-1), (0,m), m)), \\
P(\mathcal{B}|S_{m+n-1}) &= P(\mathcal{T}((m-1,0), \sqcup_{i=2}^{n-1}(0,m-1), m)),
\end{aligned}$$

with termination conditions given in Section 4.4.1 and 4.4.2.

The correctness of the theorem will be proved in the following section. Note that the tower complexity is always reduced by each recursion, so the entire decomposition process will terminate for sure, and then the components can be reassembled, as well as the connectivity back to $P(m, n)$.

4.5 Performance Evaluation

In this section, we first offer the complexity analysis of the proposed method to obtain the connectivity expressions of $m * n$ lattices. We then verify its correctness by both symbolic analysis and simulation. The impact of bond probability and lattice size on the end-to-end connectivity is also discussed. In addition, we exhibit some connectivity expressions of lattices with various sizes. Because the connectivity expression is a function of bond probability, by analyzing the expressions, we uncover more insights into the impact of bond probability.

4.5.1 Computational Complexity

As being aforementioned, one existing approach to calculating the exact square lattice connectivity with size $m * n$ is to use the PIE principle. $P(m, n)$ can be obtained by enumerating all possible source-destination paths (i.e., $\binom{m+n}{n}$ paths of $m+n$ segments each), crosschecking their overlapping segments, and calculating the probabilities for

each combination of them. The complexity is dominated by the total number of path combinations is

$$\sum_{i=1}^{\binom{m+n}{n}} \binom{\binom{m+n}{n}}{i} = 2^{\binom{m+n}{n}} - 1. \tag{4.5}$$

Because the source-destination connectivity on a lattice is symmetric along the diagonal, i.e., $P(x, y) = P(y, x)$, the total complexity of the PIE approach is $O(2^{\binom{m+n}{n}-1})$.

In our proposed approach, the $m*n$ lattice is decomposed into towers. Each tower is further decomposed into towers of smaller scales. Thus the total complexity of our approach is determined by the total number of components generated from the entire decomposition process. For each *Tower*, we define a *Stair* which is constructed with all □s of the *Tower*. In other words, a *Stair* is the remaining part of a *Tower* when removing its top and all △s. The total number of different *Stair*s with height i is $\left[\binom{m+i}{i} - \binom{m+i-1}{i-1}\right]$. Upon each *Stair*, there can be △s with height from 1 to $n - i$. Thus for each *Stair* with a height i, there can be $(m-1)*(n-i)+1$ cases of △s on top of the *Stair*. Each case, together with the corresponding *Stair*, forms a *Tower*. Thus for the total number N_t of the intermediate towers, which determines the complexity of the proposed algorithm, for an $m * n$ lattice,

$$\begin{aligned} N_t &= \sum_{i=1}^{n-1} \left[\binom{m+i}{i} - \binom{m+i-1}{i-1}\right] \cdot [1 + (m-1)(n-i)] \\ &= \sum_{i=1}^{n-1} \binom{m+i-1}{i} \cdot [1 + (m-1)(n-i)]\,. \end{aligned} \tag{4.6}$$

For strip lattice cases where $m \gg n$, N_t can be calculated explicitly, e.g., when $n = 2$ and $n = 3$, the complexity can be expressed as

$$\begin{aligned} N_t(n=2) &= m^2, \\ N_t(n=3) &= \frac{1}{2}m^3 + \frac{5}{2}m^2 - m. \end{aligned}$$

which shows the feasibility of our approach for strip lattices with the drop of the complexity from exponential sense (i.e., $O(2^{\binom{m+n}{n}-1})$ of the PIE approach) to polynomial sense.

Because of the symmetry of the lattice structure and for the ease of the complexity

expression, analysis and comparison, let $m = n$, then,

$$\begin{aligned} N_t &= \sum_{i=1}^{n-1} \binom{n+i-1}{i} \cdot [1 + (n-1)(n-i)] \\ &= \sum_{i=1}^{n-1} \frac{(n^2 - (n-1)(i+1))\,(n+i-1)!}{(n-1)!i!} \\ &= \frac{n(2n^2 - n + 1)(2n-1)! - (n^3+1)(n!)^2}{n!(n+1)!} \\ &= \frac{(2n^2 - n + 1)(2n)!}{2 \cdot n!(n+1)!} - (n^2 - n + 1). \end{aligned} \tag{4.7}$$

Applying the big O notation, the complexity of our algorithm is $O(n^2 \cdot \frac{(2n)!}{n!(n+1)!})$. We can observe that $\frac{(2n)!}{n!(n+1)!}$ is the n^{th} *Catalan number*, which can be used to represent the number of monotonic paths along the edges of a lattice with $n * n$ square cells, without passing above the diagonal. Asymptotically, the n^{th} Catalan number grows as $C_n \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$. Thus, we claim that the complexity of our algorithm to derive the directed connectivity expression for an $n * n$ lattice is $O(\sqrt{n} \cdot 4^n)$.

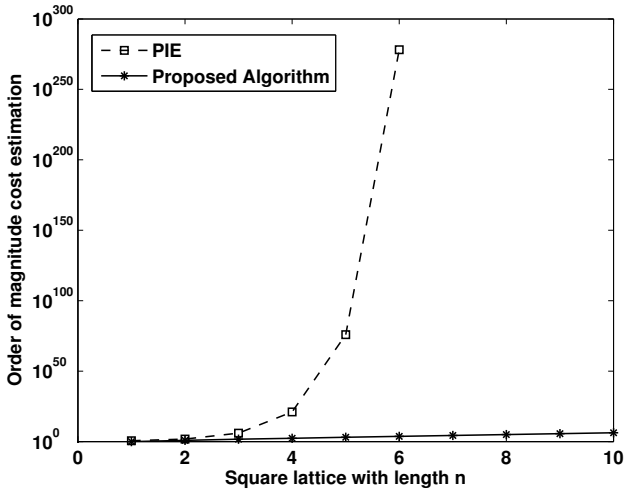


Figure 4.5: The Cost Estimation for $n * n$ Lattices.

Verified by the simulation running time, our new approach is much more efficient than the PIE approach that has the combinatorial on the exponent. Shown in Fig. 4.5, for a $6 * 6$ lattice, the number of PIE basic operations, i.e., path enumerations, has already reached a magnitude over 10^{250}, while the total number of *Tower* enumerations that our approach needs has a magnitude below 10^4. Our approach is considered to be

viable, especially in many engineering applications where one dimension of the lattice is limited, even though the other dimension can grow to a large number, e.g., in a VANET city block scenario with certain traffic flow directions. In addition, due to the recursive manner of our approach, when we obtain $P(m,n)$, we have also obtained all $P(x,y)$ for $x \leq m$ and $y \leq n$ as a byproduct, so the complexity should be amortized over all $m*n$ lattices. Further, during the recursion process, the connectivity of pre-calculated components can be stored for table lookup in new decomposition branches, which will greatly reduce the recursion depth and running time.

4.5.2 Symbolic Verification

2D Ladders

In [25], we derived the connectivity for 2D ladders, which is the connectivity from $(0,0)$ to $(x,1)$ on lattices, using another decomposition approach that is not extensible to lattices of more than one layer. However, we can use that approach to verify the new one. According to [25], the following recursive expressions can be defined for $P(x,1)$

$$\begin{aligned} P(x,1) &= p[p^x + P(x-1,1) - p^x\theta(x)],\ x \geq 1, \\ \theta(x) &= p[p + \theta(x-1) - p\theta(x-1)],\ x \geq 1, \end{aligned}$$

where $P(0,1) = p$ and $\theta(0) = 0$. By simplifying these recursions, we can obtain the symbolic, non-recursive expression of the 2D ladder connectivity as follows

$$\begin{aligned} P(x,1) &= (p^{x+1}(-p^{x+3}(1-p)^{x+1} - p(p((p-2)x + p - 3) \\ &\quad +2(x+1)) + x + 1))/((p-1)p+1)^2. \end{aligned} \tag{4.8}$$

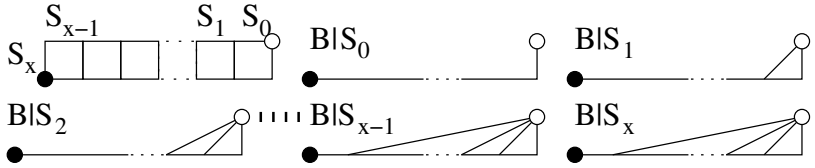


Figure 4.6: The Decomposition of a *Ladder*.

With the new approach, as shown in Fig. 4.6 for illustration purposes, we have the following recursions according to the decomposition process of towers (essentially

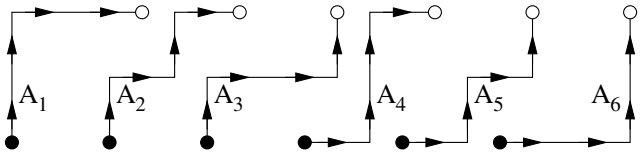


Figure 4.7: All Source-destination Paths of a $2 * 2$ Lattice.

of $\triangle$s)

$$\begin{aligned} P(\mathcal{T}((i,0),x)) &= p^{x-i} \cdot P(\mathcal{T}((i,0),i)), \\ P(\mathcal{T}((i,0),i)) &= p + p \cdot P(\mathcal{T}((i-1,0),i-1)) - p^2 \cdot P(\mathcal{T}((i-1,0),i-1)), \end{aligned}$$

with $P(\mathcal{T}((1,0),1)) = P_T = p + p^2 - p^3$. Then with the total probability in the new approach,

$$\begin{aligned} P(x,1) &= p^{x+1} + \sum_{i=0}^{x-1} P(\mathcal{T}((i,0),x))p^i(1-p) \\ &\quad + P(\mathcal{T}((x-1,0),x))p^x(1-p), \end{aligned} \tag{4.9}$$

which comes to the same expression as (4.8). For example, $P(0,1) = p$, obviously, $P(1,1) = 2p^2 - p^4$, the same as that obtained in Section 4.4.1 using PIE, and

$$\begin{aligned} P(2,1) &= p^7 - p^6 - 2p^5 + 3p^3, \\ P(3,1) &= -p^{10} + 2p^9 + p^8 - 2p^7 - 3p^6 + 4p^4, \\ &\cdots \quad . \end{aligned} \tag{4.10}$$

2*2 Lattices

Because the approach used in [25] is not capable for lattices of more than one layer, we have to use the PIE principle. Here we use a $2 * 2$ lattice as an example. To use the PIE principle, we first identify all the paths from the source to the destination. For the case of a $2 * 2$ lattice, there are 6 paths in total, as shown in Fig. 4.7.

Using the PIE principle,

$$\begin{aligned} P(2,2) &= P(A_1 + A_2 + A_3 + A_4 + A_5 + A_6) \\ &= \sum_{k=1}^{6} (-1)^{k+1} \left(\sum_{1 \le i_1 < \cdots < i_k \le 6} (A_{i_1} \cdot \ldots \cdot A_{i_k}) \right) \\ &= p^{12} - p^{11} + 2p^{10} + 4p^9 + 2p^8 - 4p^7 - 6p^6 + 6p^4. \end{aligned} \tag{4.11}$$

where $(-1)^{k+1}$ indicates the inclusion and exclusion. Simplifying (4.11), we obtain the same result as that in Section 4.4.2 with the new approach.

4.5.3 Simulation Verification

For ladders of more than one layer, however, there are no symbolic results in the literature, and the PIE complexity grows extremely quickly due to the combinatorial factor on the exponent. Thus we have to rely on the simulation results to verify the new approach. In addition, based on the calculation results, we try to understand the end-to-end connectivity from the lattice size and ratio perspective. In the next subsection, we will look at the connectivity problem from the viewpoint of bond probability by analyzing the connectivity expressions.

Symmetric Lattices

Figure 4.8 shows the connectivity of symmetric lattices whose length equals to their width. With the same bond probability, the connectivities of lattices with different size are plotted. The bond probabilities we choose here are from $p = 0.35$ to 0.95. For all the following figures in this subsection, the lines indicate the calculation results by using the obtained connectivity expressions, and the point markers show the results from the simulation. As shown in the figure, the new approach gives very accurate numerical results, which have a very good match with the simulation results, but without lengthy simulations.

Obviously, for the same lattice, the higher the bond probability, the better the connectivity, because any adjacent vertices have a higher chance to be connected. With the same bond probability, however, the increase of lattice size does not have the same impact on the end-to-end connectivity. For small bond probabilities, i.e., from 0.35 to 0.65, a clear drop of connectivity can be observed. This actually corresponds to the conclusion in [82], where the end-to-end connectivity shows an exponential

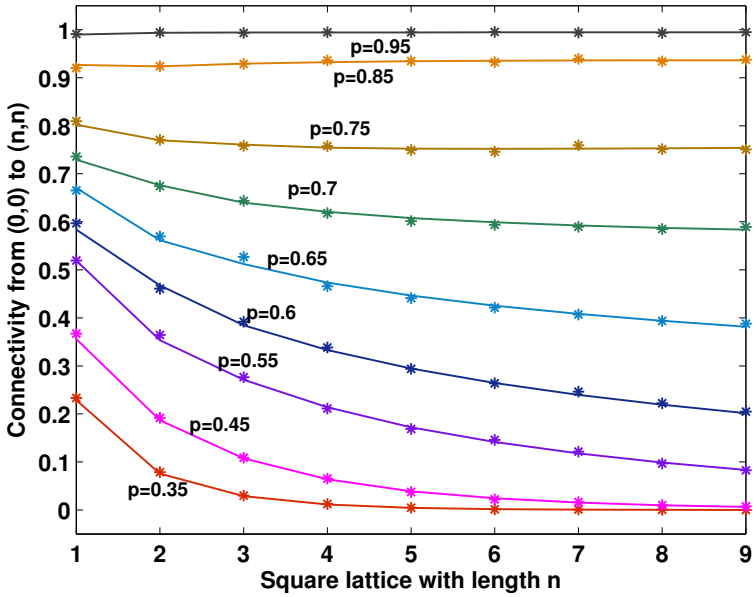
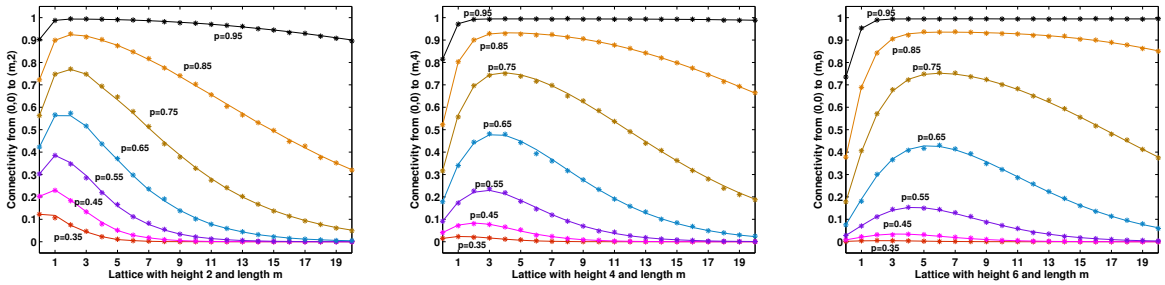


Figure 4.8: The Connectivity of $n * n$ Lattices.

decay and the exponent is determined by how far away the bond probability is from the critical bond probability, i.e., around 0.6447 for the directed bond percolation on square lattices. But when the bond probability is reasonably large, i.e., higher than 0.65, the connectivity remains stable.

It is also interesting to investigate how the end-to-end connectivity increases with the rising of the bond probability p for the same size lattice. When p is small, e.g., increasing p from 0.35 to 0.45, only the connectivity for small $n * n$ lattices (e.g., $n < 3$) increases considerably, and the increase diminishes very quickly for larger n. However, when p is reasonably large, e.g., increasing from 0.55 to 0.65, even though the end-to-end connectivity still decreases with a large n, the increase due to an increased p actually amplifies as n increases. When p is further increased, e.g., from 0.65 to 0.75, the end-to-end connectivities are no longer to decrease with n (more obviously when $p = 0.85$ or 0.95). Recall that percolation occurs around $p = 0.6447$ on an infinite lattice, the end-to-end connectivity on a finite lattice also shows the deepest gradient when p is around 0.65, illustrated by the gap between curves of different bond probabilities in Fig. 4.8.



(a) Lattice with $n = 2$ (b) Lattice with $n = 4$ (c) Lattice with $n = 6$

Figure 4.9: The Connectivity of $m * n$ Lattices.

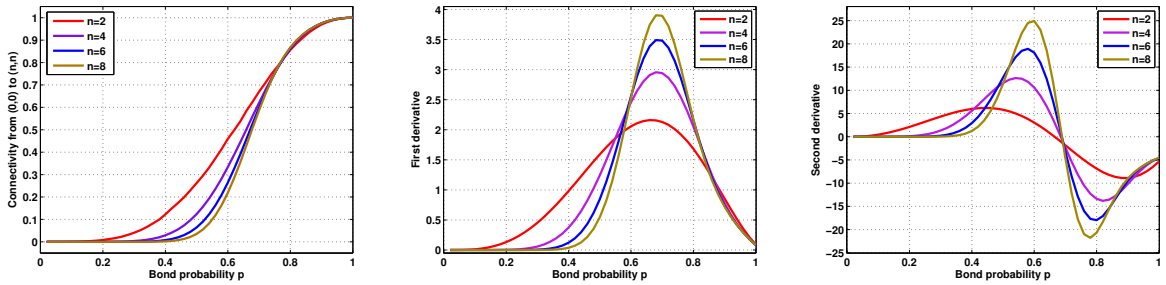
Strip Lattices

In many engineering fields (e.g., VANET in a city block scenario), we are more interested in propagating messages along certain directions (or traffic flows). In this sense, we shall focus more on the lattices with certain width, or strips, which are less computationally complex in terms of the connectivity expression derivation. Figure 4.9 shows the connectivity of lattices with different widths (n), when $n = 2$, 4 and 6 as examples. For any lattices with size $m * n$, we can observe that the higher the bond probability, the better connectivity. For each bond probability, with the fixed lattice width and increasing length, the connectivity first increases, followed by an eventual decrease. The wider the lattice, the further the peak will occur. These non-monotonic curves are very interesting to observe and very important in engineering fields to determine the optimal m, n and p for given applications. It shows that there is a trade-off between the total number of available paths and the length of each path. For a lattice with the given width, when the lattice length increases, the number of paths will increase, which brings more possibilities of connections between the source and destination. However, the length of these extra paths increases as well, leading to a lower probability to connect the source and destination along each path. For the overall end-to-end connectivity, the increase of path diversity has a positive effect, while the increase of path length has a negative one. Considering the curves shown in the figure, before the peak occurs, the positive effect of the path diversity is stronger than the negative one of path length increase, leading to the increase of connectivity probability. However, after the peak occurs, the negative effect of path length increase becomes dominating, which leads to the decrease of the overall connectivity. The peak occurs around the cases where the lattice length equals the width, which

implies a symmetric $n * n$ lattice.

In all figures of this subsection, the numerical results from the new approach are very accurate when compared with the simulation ones. The analytical expressions obtained from the decomposition process can be used for further manipulation, e.g., derivatives, probability distribution functions and higher-order moments. Discussion on the expressions is presented in the next subsection.

4.5.4 Analysis on Connectivity Expressions



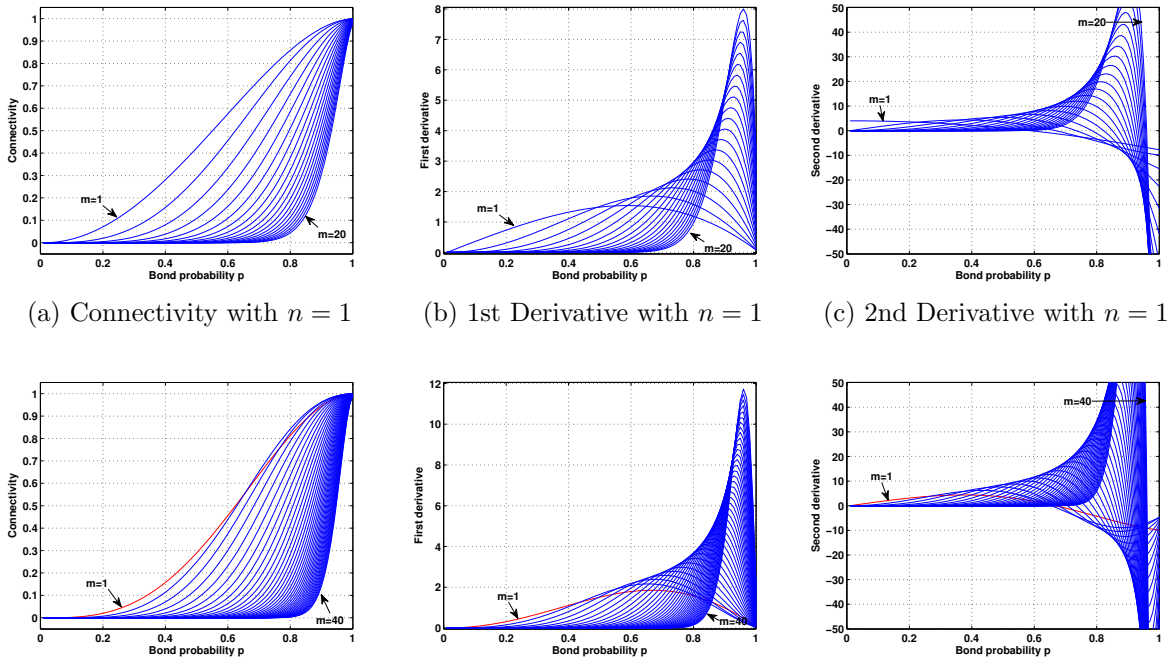
(a) Connectivity of $n * n$ Lattice (b) 1st Derivative (c) 2nd Derivative

Figure 4.10: Analysis of the $n * n$ Lattice Connectivity Expressions.

Our proposed approach provides us with the connectivity expressions of $m * n$ square lattices [83], which are polynomial functions of bond probability p, e.g.,

$$\begin{aligned}
P(2,3) &= p^{17} - 7p^{16} + 15 * p^{15} - 6 * p^{14} - 9 * p^{13} - 5 * p^{12} \\
&\quad +11 * p^{11} + 8p^{10} + 4p^9 - 9p^8 - 12p^7 + 10p^5, \\
P(3,3) &= -p^{24} + 12p^{23} - 56p^{22} + 124p^{21} - 116p^{20} + 34p^{18} \\
&\quad +40p^{17} + 11p^{16} - 68p^{15} - 22p^{14} + 16p^{13} + 25p^{12} \\
&\quad +24p^{11} + 12p^{10} - 24p^9 - 30p^8 + 20p^6, \\
P(4,2) &= p^{22} - 10p^{21} + 37p^{20} - 58p^{19} + 23p^{18} + 20p^{17} \\
&\quad +15p^{16} - 34p^{15} - 16p^{14} + 6p^{13} + 15p^{12} + 16p^{11} \\
&\quad +7p^{10} - 16p^9 - 20p^8 + 15p^6.
\end{aligned}$$

More are available at [83]. The first and second derivatives of the polynomials indicate the change rates of the connectivity and the first derivatives with regard to the bond probability, respectively. By computing the first and second derivatives of such connectivity expressions, we can reveal more insights quantitatively.



(a) Connectivity with $n = 1$ (b) 1st Derivative with $n = 1$ (c) 2nd Derivative with $n = 1$

(d) Connectivity with $n = 2$ (e) 1st Derivative with $n = 2$ (f) 2nd Derivative with $n = 2$

Figure 4.11: Analysis of the $m * 1$ and $m * 2$ Lattice Connectivity Expressions.

Symmetric Lattices

Figure 4.10(a) shows the $n * n$ lattice connectivity over different bond probabilities. For the same lattice, higher bond probabilities always help achieve better connectivity. For most values of p, smaller lattices always have higher connectivity. However, when the bond probability is large enough, e.g., $p > 0.8$, it is even possible for larger lattices to have higher connectivity due to many more paths.

All the connectivities increase significantly when the bond probability is between 0.4 and 0.8, where a slight increase of the bond probability will greatly increase the connectivity over lattices. The sharp transition corresponds to the same phenomenon in percolation on an infinite lattice, but here we have more microscopic results on the connectivity to any vertex on the lattice. For each curve in Fig. 4.10(a), the increasing slope, i.e., the first derivative of connectivity polynomial, is large when the bond probability is within a certain transition range. To quantitively define the transition range, we calculate the two inflection points of the first derivative curve. We call the range bounded by the two inflection points the *critical transition range* in this work. Following the same legend as shown in Fig. 4.10(a), Fig. 4.10(b) and

(c) show the first and second derivatives of connectivity expressions of $n * n$ lattices, where n is 2, 4, 6 and 8.

From Fig. 4.10(b), the difference of critical transition ranges for lattices of different size can be observed, e.g., for $2 * 2$ lattice, the connectivity has a significant increase when the bond probability falls in the range of $[0.43,\ 0.9]$. However, with the increment of the lattice size, the critical transition range shrinks, from $[0.43,\ 0.9]$ for a $2*2$ lattice to $[0.6,\ 0.79]$ for an $8*8$ lattice. Another property that can be noted from the figure is that, the larger the lattice is, the higher the connectivity increase amplitude it has, which means a sharper rise. These two observations imply that the larger the lattice size, the less sensitive the connectivity is to the low bond probability, i.e., $p \in [0,\ 0.4]$, but more sensitive to the reasonably high bond probabilities, i.e., $p \in [0.6,\ 0.8]$.

By calculating the solution of the second derivative equaling to 0, we can obtain a very important bond probability, i.e., $p_c(m, n)$, where the connectivity curve achieves the sharpest increase. This bond probability is also the probability where the maximum value of the first derivate curve occurs. By calculation, we find that the values of $p_c(n, n)$, shown in Table. 4.1.

Table 4.1: Critical Bond Probability for $n * n$ Lattices.

$n * n$	$p_c(n, n)$	$n * n$	$p_c(n, n)$
1*1	0.57735	5*5	0.66747
2*2	0.64541	6*6	0.66849
3*3	0.65986	7*7	0.66888
4*4	0.66517	8*8	0.66895

An asymptotic behavior of bond probability could be observed since the difference between two consecutive critical bond probabilities reduces with the increment of the lattice size. The critical transition range can also be obtained from the second derivative, indicated by the range of the bond probability from the maximum second derivative to the minimum.

Strip Lattices

Similarly, we investigate the connectivity expressions of strip lattices whose width is fixed, but with variable lengths, e.g., $m * 1$, $m * 2$, etc. In Fig. 4.11, the two rows

of figures show the connectivity expressions and their first and second derivatives for lattices with width of 1 and 2, respectively. We choose these two lattice widths for illustration purposes and the conclusion drawn from these figures could represent other strip lattices. For the first row where $n = 1$, 20 curves of lattices with length from 1 to 20 have been plotted. To make a fair comparison, we let the largest lattice in each row have the same length to width ratio, e.g., 20. Thus in the second row where $n = 2$, 40 curves of lattices with length from 1 to 40 have been plotted.

For lattices with the same width (i.e., in the same row), conclusions similar to $n * n$ lattices could be drawn. The connectivity increases as the bond probability increases. However, the connectivity critical transition range shrinks dramatically with the increase of the lattice length, while the connectivity increase amplitude rises significantly. In the second row of figures, the curves of lattices with $m = 1$ are plotted in red color. Specifically in Fig. 4.11(d), only the curve with $m = 1$ has intersections with other curves, which means that for smaller p, $P(2,1) > P(2,m)$, when $m > 1$ and for higher p, $P(2,1) < P(2,m)$. For any $m_1 > m_2 \geq 2$ with a given p, $P(2,m_1) > P(2,m_2)$ always holds. This could be further generalized that given a lattice width n and bond probability p, the monotonic increasing property of $P(m,n)$, i.e., the connectivity from origin to (m,n), only exists when $m \geq n$.

The exact critical bond probability $p_c(m,n)$ can be calculated with the connectivity polynomial obtained. For lattices with width 1 and 2, the critical bond probabilities are shown in Table 4.2.

Table 4.2: Critical Bond Probability for $m * n$ Lattices.

$m*n$	$p_c(m,n)$	$m*n$	$p_c(m,n)$	$m*n$	$p_c(m,n)$	$m*n$	$p_c(m,n)$
1*1	0.57735	6*1	0.84026	1*2	0.65818	6*2	0.75278
2*1	0.65818	7*1	0.86133	2*2	0.64541	7*2	0.77616
3*1	0.72418	8*1	0.87770	3*2	0.66700	8*2	0.79624
4*1	0.77495	9*1	0.89072	4*2	0.69639	9*2	0.81345
5*1	0.81246	10*1	0.90130	5*2	0.72594	10*2	0.82826

By studying the curves in Fig. 4.11 and numbers in Table 4.2, asymptotic behaviors can be observed as the lattice size grows and general properties for lattices with size $\varphi n * n$, can be summarized. When the lattice length to width ratio, i.e., φ, is fixed, the larger the n is, the narrower the critical transition range is but the larger the increase amplitude is. The critical bond probability $p_c(\varphi n * n)$ is also higher as

shown in the table. Similar observations are found in the symmetric lattices as well, where φ could be considered as one. On the other hand, when n is fixed, the change of φ, implies the change of the percolation direction. We see with a larger φ, the critical bond probability $p_c(\varphi n * n)$ becomes higher.

4.6 Application in Urban VANETs

We apply our approach to obtaining the connectivity of a realistic 2D network for urban VANETs, where a Manhattan-like road structure is considered and each road segment can be represented by an edge in the square lattice as shown in Fig. 4.13.

4.6.1 Problem Description

We consider the most general Manhattan-like city road structure, which can be modeled as a square lattice where each intersection is a site in the lattice and each road segment is the bond between sites. Vehicles move on each road segment between any two adjacent intersections. By multi-hop transmissions of vehicles, the probability for a message reaching one intersection from its adjacent preceding intersection is denoted as p, i.e., the bond probability. The detailed derivation of p under the Manhattan-like city structure is given in the following subsection. Once we have p, the directed connectivity of any two intersections in such an urban VANET system can be explored. In the real world, the bond probability p reflects the connection condition between two neighbor intersections. Besides the following mathematical derivation, p can be also obtained empirically with more realistic constraints.

4.6.2 Bond Probability

To derive the bond probability p in the urban VANET system, we borrow the method from [25]. We start by investigating the size of the "connected vehicle cluster", within which all vehicles are connected via wireless transmissions, on the one-dimensional road. We denote the size of the cluster as a random variable (RV) C in the following derivation. Because we consider the one-dimensional case, the cluster is formed by vehicles distributed in a linear road. So, C is the sum of several inter-vehicle distance RVs, which are assumed to be exponentially distributed. It has been widely accepted that the sum of exponential distributed RVs follows a Gamma distribution, so the

probability density function of C can be expressed as:

$$f_C(x) = x^{k-1}\frac{e^{-x/\theta}}{\theta^k \Gamma(k)}, \quad \text{for } x > 0, \tag{4.12}$$

where k and θ are the distribution parameters, $E[C] = k\theta$, $Var[C] = k\theta^2$, and $\Gamma(k)$ is the Gamma function evaluated at k. Therefore, the distribution parameters k and θ can be calculated with the first and second moments of C as:

$$k = (E[C^2]/E[C]^2 - 1)^{-1} \text{ and } \theta = E[C]/k.$$

Then the goal is left to obtain $E[C]$ and $E[C^2]$.

Let X_1 denote the inter-vehicle distance between the first and second vehicles in a cluster and R denote the communication range of vehicles. Then the expectation of the cluster size can be expressed as:

$$E[C] = E[C|X_1 < R] \times \Pr\{X_1 < R\}. \tag{4.13}$$

The conditional expectation $E[C|X_1 < R]$ can be expressed as:

$$E[C|X_1 < R] = E[X_1|X_1 < R] + E[C'], \tag{4.14}$$

where $E[C']$ is the expectation of the cluster size without counting the distance X_1. Because the cluster size is composed of i.i.d. RVs of the inter-vehicle distance, $E[C'] = E[C]$. We know the inter-vehicle distance X_1 follows the exponential distribution, i.e., $\Pr\{X_1 < R\} = 1 - e^{-\lambda R}$. Let $\overline{X_1'}$ denote $E[X_1|X_1 < R] = \int_0^R \lambda x e^{-\lambda x}/(1 - e^{-\lambda R})\mathrm{d}x$. Therefore, Eqn. (4.13) can be simplified as

$$E[C] \;=\; \left(\overline{X_1'} + E[C]\right) \times \Pr\{X_1 < R\}, \tag{4.15}$$

and

$$\begin{aligned} E[C] &= \overline{X_1'} \times \frac{\Pr\{X_1 < R\}}{1 - \Pr\{X_1 < R\}} \\ &= \frac{1 - e^{-\lambda R}(\lambda R + 1)}{\lambda e^{-\lambda R}}. \end{aligned} \tag{4.16}$$

Similarly, for the second-order moment of RV C, we have

$$\begin{aligned} E[C^2] &= \Pr\{X_1 < R\} \times E[(C' + X_1)^2 | X_1 < R] \\ &= \frac{1 - e^{-\lambda R}}{e^{-\lambda R}} \times \left(2E[C]\overline{X_1'} + \overline{X_1'^2} \right), \end{aligned} \tag{4.17}$$

where $\overline{X_1'^2} = E[X_1^2 | X_1 < R] = \int_0^R \lambda x^2 e^{-\lambda x}/(1 - e^{-\lambda R}) \mathrm{d}x$.

With $E[C]$ and $E[C^2]$ obtained, the Gamma approximation in Eqn. (4.12) can be derived.

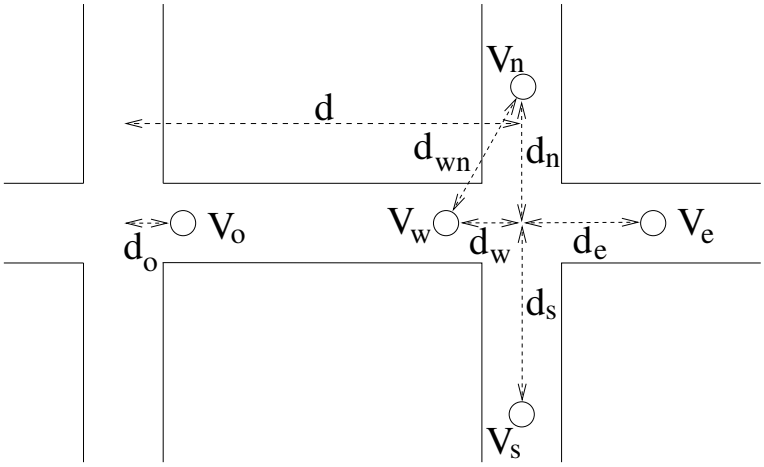


Figure 4.12: Bond Probability Illustration.

p, as defined in previous subsection, is equivalent to the "bond probability" in percolation theory. In the urban VANET scenario, p represents the connection probability of any two neighbor intersections via wireless communication. Shown in Fig. 4.12, assume the distance between any two neighbor intersections is d. Let V_e, V_s, V_w, and V_n denote the vehicles which locate closest to the right intersection on their road segments, respectively. Their distances to the right intersection are d_e, d_s, d_w, and d_n, respectively. Assume the directed connection starts from the left intersection to the right one and V_o is the closest vehicle on its road segment to the left intersection with distance d_o. To make V_o connected to the left intersection, the distribution of d_o is a truncated exponential function $\lambda e^{-\lambda t}/(1 - e^{-\lambda R})$, for $0 \le t \le R$.

In order to connect two neighboring intersections, the cluster, starting from V_o, should include at least one vehicle of V_e, V_s and V_n, in order to start new transmissions on the other road segments. Therefore, depending on whether V_e is connected to V_o, two disjoint cases need to be considered:

Case One: V_e is connected to V_o, which means the cluster originating from V_o has a size larger than $d - d_o$. Also with the consideration of V_o's location, the probability in this case is

$$p_1 = \int_0^R \int_{d-t}^{\infty} f_C(x)\mathrm{d}x \frac{\lambda e^{-\lambda t}}{1 - e^{-\lambda R}}\mathrm{d}t. \tag{4.18}$$

Case Two: V_e is not connected to V_o, which means the cluster size is smaller than $d - d_o$. Let V_w be the last vehicle of the cluster. To connect to the right intersection, either V_s or V_n or both of them need to connect to V_w. Denote the cluster size, i.e., the distance from V_o to V_w as x, d_o as t, and the distance from V_w to the right intersection as d_w, then at least one of d_n and d_s should be shorter than $\sqrt{(\eta R)^2 - (d - x - t)^2}$, where we consider $\eta \in (0, 1)$ as the shadowing effect for signal transmissions to other perpendicular streets. Therefore, the probability that at least one of V_s and V_n is connected to V_w is $(1 - e^{-2\lambda\sqrt{(\eta R)^2 - (d-x-t)^2}})$. Then the probability for this case is

$$p_2 = \int_0^R \int_{d-t-\eta R}^{d-t} (1 - e^{-2\lambda\sqrt{(\eta R)^2 - (d-x-t)^2}}) f_C(x)\mathrm{d}x \frac{\lambda e^{-\lambda t}}{1 - e^{-\lambda R}}\mathrm{d}t. \tag{4.19}$$

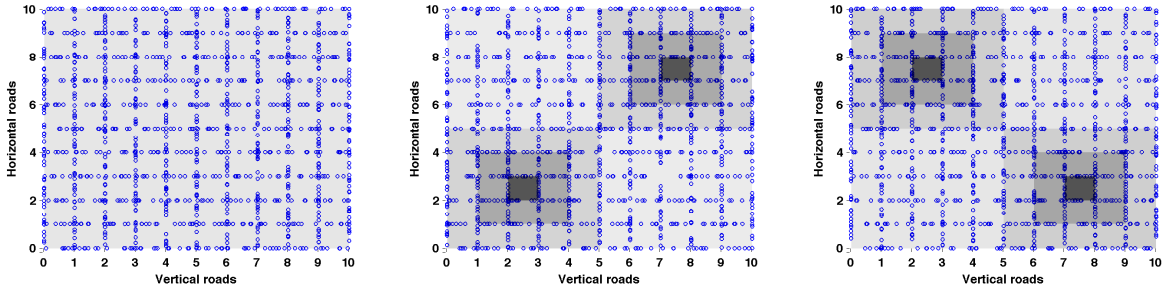
Considering the above two disjoint cases, p is given by

$$p = p_1 + p_2. \tag{4.20}$$

The above derivation has been verified through extensive simulation in our previous work [25]. The bond probability under different applications can have different definitions. It is the foundation of all the technical contribution and the analysis above shows the feasibility and practicability of our solution. We can also notice that the above derivation does not take the time into consideration, which means it does not consider the change of p over time. However, the proposed algorithm can still apply with a dynamic p. Similar to the study of the vehicle mobility in Chapter 2 and 3, the traffic change on the roads as a stationary ergodic random process, and the time average equals the ensemble average. Thus, the bond probability can represent both the percentage of time two intersections being connected and the likelihood that the two intersections being connected at a given time instant. Another issue we would like to discuss here is that the current p is derived only based on the multi-hop connection between neighbor intersections at a time instance. If we consider the carry-store-and-forward message transmission, we can slightly change the definition of p as the connectivity probability of two neighbor intersections over a period of time, during

which messages can be forwarded between the two intersections. However, this will not affect the proposed algorithm and all the technical solutions still apply.

4.6.3 Connectivity of Heterogeneous Lattices

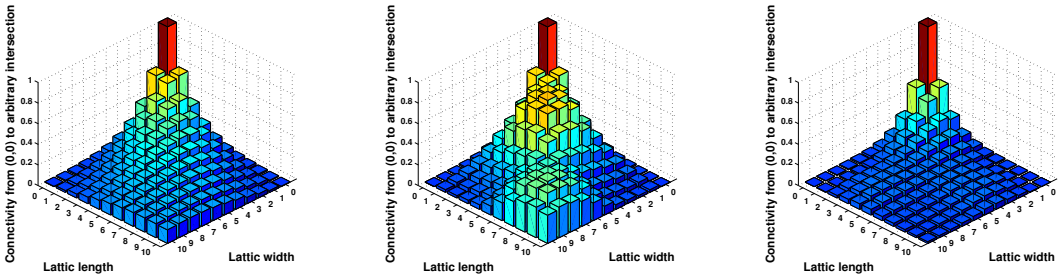


(a) Homogeneous Vehicle Density (b) Heterogeneous Vehicle Density, Case One (c) Heterogeneous Vehicle Density, Case Two

Figure 4.13: Vehicle Density Distribution of Urban VANETs.

The vehicle density λ on each road segment plays an important role affecting the bond probability. We investigate two heterogeneous cases to study the impact of different vehicle density distributions on the lattice connectivity, as shown in Fig. 4.13(b) and (c), respectively, where in the simulation, we assume the distance between two adjacent intersections is 500 m, with the wireless transmission range of 200 m. All road segments are categorized into different tiers, indicating the regions with different traffic "popularity". The darker the region, the more "popular" it is, implying a higher vehicle density, i.e., a larger λ. We assume the message origin always locates at the intersection $(0, 0)$. Two social spots, e.g., commercial area or transportation hub, are set at different locations with regard to the message origin in different cases, shown as the darkest regions in each figure. Different vehicle densities determine the heterogeneous bond probability of each road segment. We refer to [25] for the mapping between the vehicle density and bond probability, so road segments with different vehicle density have corresponding bond probabilities. For the four grey-scale regions in each heterogeneous figure, the vehicle densities are 0.02, 0.016, 0.012, and 0.01 vehicles per meter, respectively, from dark regions to light regions. And the corresponding bond probabilities are 0.89, 0.76, 0.59, and 0.47. With $1,332$ vehicles in total for each figure, Fig. 4.13(a) demonstrates the homogeneous case with the identical density 0.012 and identical bond probability 0.59 over all road segments for

comparison purposes.



(a) Homogeneous Vehicle Density (b) Heterogeneous Vehicle Density, Case One (c) Heterogeneous Vehicle Density, Case Two

Figure 4.14: The Connectivity from $(0, 0)$ of Urban VANETs.

Our approach can not only calculate the connectivity of lattices with homogeneous bond probabilities, it is also applicable to lattices with heterogeneous bond probabilities. The message is sent out from the origin to all other intersections within the lattice in a multi-hop manner by vehicles. The connectivity probabilities from $(0, 0)$ to every other intersection are calculated and plotted in Fig. 4.14(a), (b), and (c), corresponding to Fig. 4.13(a), (b), and (c), respectively. X and Y coordinates indicate the locations of intersections and the height of each bar represents the value of connection probability between the corresponding intersection and the source $(0, 0)$.

With the homogeneous vehicle density, the closer the intersection to the message origin, the higher the connection probability it has. The closer the intersection to the diagonal of the whole lattice, the higher chance it is connected to the message origin because there are more paths between itself and the source according to the nature of the directed propagation. Similar conclusions are discussed in Section 4.5.

However, for the heterogeneous vehicle density cases, the connectivity distribution can be very different. For the first case, i.e., Fig. 4.13(b), the origin locates near one of the social spots and the other social spot locates on the diagonal of the lattice. Thus the connectivity of social spot areas is greatly increased. For the second case, however, the connectivities of all intersections are considerably low because both social spots locate far from either the origin or the diagonal. The overall connectivity is even worse than the homogeneous case, because the social spots attract more vehicles, leaving the other regions with smaller vehicle densities than the average level in the homogeneous case. Thus the high density of the social spots in this case does not help in improving the connectivity.

The connectivity analyses on heterogeneous bond probability distributions give us unique and valuable insights about some implementation details of the message propagation. First, the choice of the message source location can have a great impact on the end-to-end connectivity. As observed from the experiment, the closer the social spot is to the message source, the better the connectivity can be achieved; the closer the social spot locates to the diagonal of the two message propagation directions, the better connectivity can be achieved. This provides us a guideline where to choose the message source location, i.e., the location-fixed infrastructures or mobile vehicles which can determine the location to start broadcasting, for a better connectivity.

On the other hand, vehicles can help to increase the bond probability actively and wisely. Recall that the bond probability is affected by the vehicle transmission range. Therefore, vehicles can actively increase the transmission power to enlarge the transmission range, and further increase the bond probability. However, vehicles do not need to do so anytime anywhere which may cause more interference and a waste of energy. Only when a vehicle senses the environment and detects the bond probability of the current road segment is low, it can consider to tune up the transmission power. It can further consider the critical transition range of the bond probability discussed in the previous section. Only when the current bond probability falls within the critical transition range, it tunes up the transmission power in order to achieve a dramatic increase in the overall connectivity; otherwise, the improvement brought by the increased bond probability may not be significant.

4.7 Connectivity of Other Lattice Topologies

However, not all road structures can be modeled as a square grid. Many more are irregular. To demonstrate the versatility of the proposed algorithm, we use triangular lattice as an example in this section. The topology and its decomposition process is shown in Fig. 4.15.

4.7.1 Connectivity Analysis on Triangular Lattices

The square lattice can be further evolved to the triangle lattice shown in Fig. 4.15, which is more compact and able to provide stronger connections. To analyze the connectivity of the triangle lattice, the idea of decomposition can still be applied. As shown in Fig. 4.15, which uses a 3×3 triangle lattice as an example, the 2-D lattice

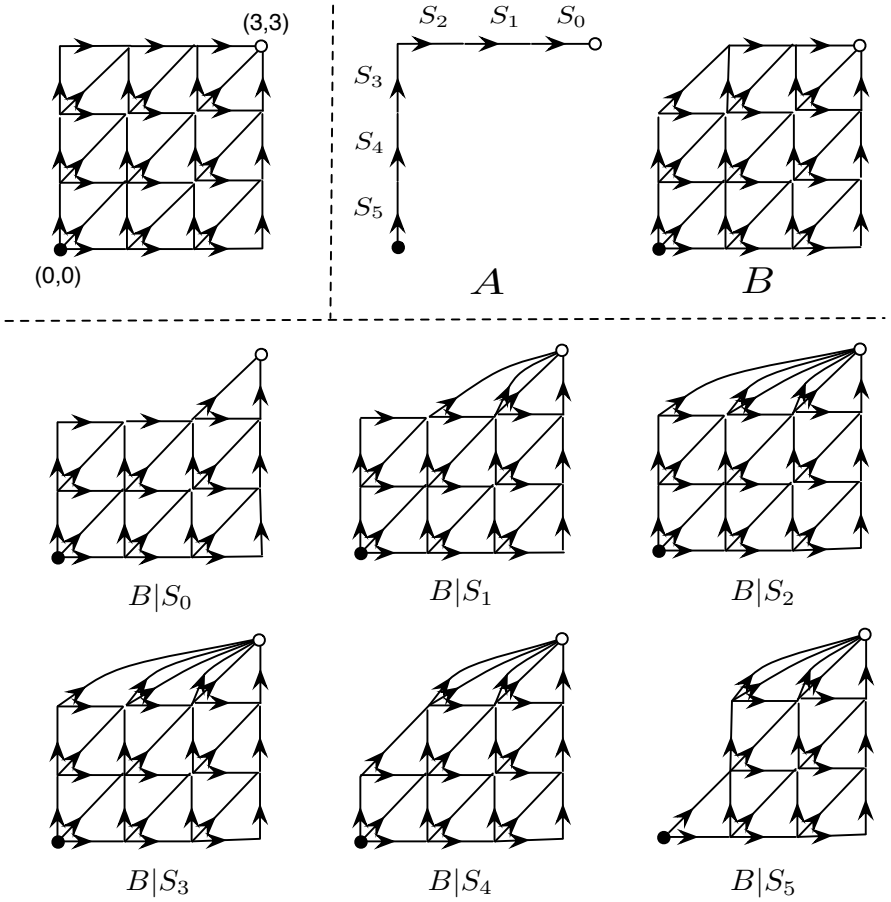


Figure 4.15: The Decomposition of a 3*3 Triangle Lattice.

can be firstly decomposed to the top-left most connection path A and the union of all other paths $\mathcal{B}$. For an easy description, we define S_i as the event that the last i edges along A leading to the destination are all connected, but the last $(i+1)$-th one is not. Hence, $\Pr\{S_i\} = p^i(1-p)$ for $0 \leq i \leq m+n-1$. Intuitively, for the origin and destination to be connected, there are $m+n+1$ mutually exclusive cases, including one case where A is connected (with probability $\Pr(A) = p^{m+n}$), and $m+n$ cases where A is not connected and the connection is through $\mathcal{B}$, i.e., $\mathcal{B}|S_i$, $i = 0, 1, \cdots, m+n-1$. Define the probability that $\mathcal{B}$ is connected given S_i as $\Pr\{\mathcal{B}|S_i\}$, so the connectivity probability from $(0,0)$ to (m,n), $P_{m,n}$, can be derived as follows:

$$P_{m,n} = P(A) + \sum_{i=0}^{m+n-1} \Pr\{\mathcal{B}|S_i\} \cdot \Pr\{S_i\}. \tag{4.21}$$

And the derivation is the same as the one shown in Section 4.4.2

The construction of $\mathcal{B}|S_i$ is demonstrated in Fig. 4.15. Given S_0, i.e., edge $(2,3) \rightarrow (3,3)$ is broken, for $\mathcal{B}$, the edges $(2,2) \rightarrow (2,3)$, $(1,2) \rightarrow (2,3)$, $(1,2) \rightarrow (1,3)$, $(1,3) \rightarrow (2,3)$ and $(0,2) \rightarrow (1,3)$ will not be able to contribute to the end-to-end connection. So all the five edges can be removed from $\mathcal{B}$ to obtain $\mathcal{B}|S_0$. Given S_1, i.e., $(2,3) \rightarrow (3,3)$ is connected but $(1,3) \rightarrow (2,3)$ is broken, for $\mathcal{B}$, the connection from $(1,2)$ to $(1,3)$ and $(0,2) \rightarrow (1,3)$ become useless. And the connections from $(2,2)$ to $(3,3)$ through $(2,3)$ only depends on the bond from $(2,2)$ to $(2,3)$. So we can remove the edge $(2,3) \rightarrow (3,3)$, and replace the edge $(2,2) \rightarrow (2,3)$ with a single "bond" $(2,2) \rightarrow (3,3)$. Similarly, the connection from $(1,2)$ to $(3,3)$ through $(2,3)$ only depends on the bond from $(1,2)$ to $(2,3)$. And we can remove the edge $(2,3) \rightarrow (3,3)$, and replace the edge $(1,2) \rightarrow (2,3)$ with a single "bond" $(1,2) \rightarrow (3,3)$. Now, we have obtained $\mathcal{B}|S_1$. The other $\mathcal{B}|S_i$s can be derived in a similar way.

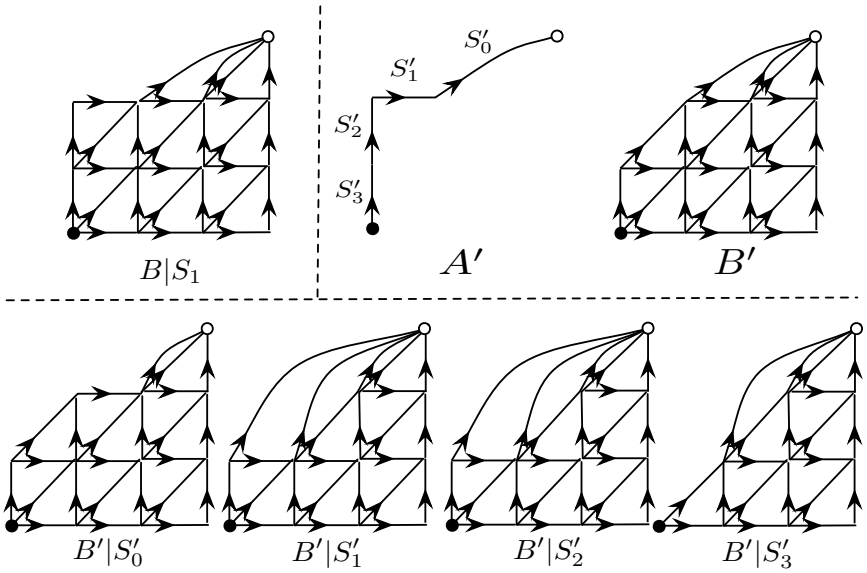


Figure 4.16: Decomposition of $B|S_1$ from Fig. 4.15.

The same decomposition process can be iteratively applied on each $\mathcal{B}|S_i$ and its further decomposed shapes, until the connectivity is easy enough to be calculated directly. For instance, the decomposition process of $\mathcal{B}|S_1$ (in Fig. 4.15) is shown in Fig. 4.16. Two sets of paths connecting the source and destinations are first identified, i.e., the left-top most single path A' and the union of the rest paths B'. Given S'_0, i.e., edge $(1,2) \rightarrow (3,3)$ is broken, for $\mathcal{B}'$, the same connection is broken. But this does not affect the other parts. So $\mathcal{B}'|S'_0$ is obtained as shown in Fig. 4.16. Given S'_1,

edge $(0,2) \rightarrow (1,2)$ is broken, but the edge $(1,2) \rightarrow (3,3)$ is connected, making point $(1,2)$ the alternative destination. So the connection from $(1,1)$ to $(1,2)$ becomes the connection from $(1,1)$ to $(3,3)$ and the connection from $(0,1)$ to $(1,2)$ becomes the connection from $(0,1)$ to $(3,3)$. Then the shape of $\mathcal{B}'$ is updated as $\mathcal{B}'|S'_1$ in Fig. 4.16. Similar process is done for $\mathcal{B}'|S'_2$ and $\mathcal{B}'|S'_3$.

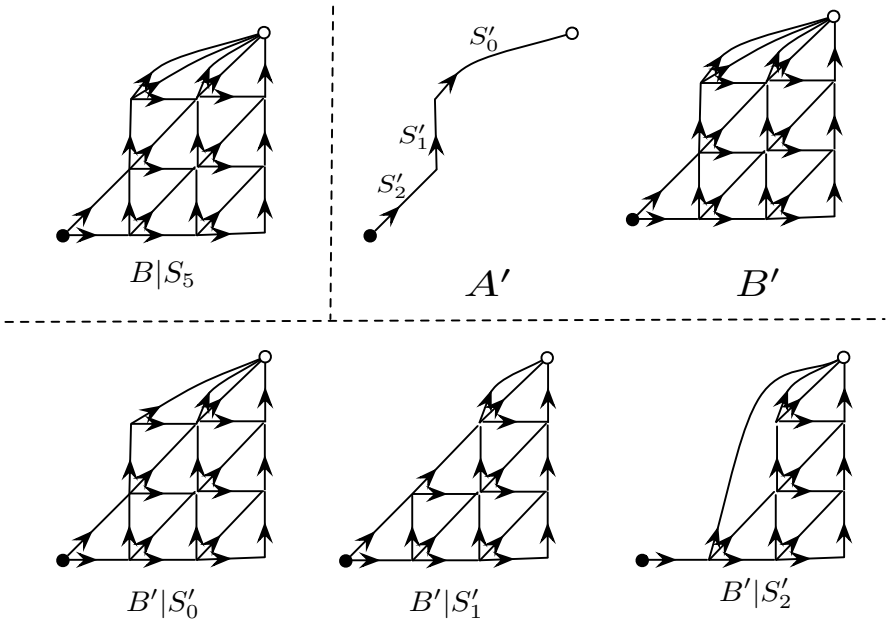


Figure 4.17: Decomposition of $B|S_5$ from Fig. 4.15.

For another example, $\mathcal{B}|S_5$ in Fig. 4.15, the decomposition process can be shown in Fig. 4.17. Different from Fig. 4.15, B' here contains the whole path A'. This is because all edge segments on A' can contribute to other paths included in B'. The decomposition is just similar as described before.

4.7.2 Verification of 2D Triangle Lattice Connectivity

In this section, we present the analytical results of the 2D triangle lattice connectivity, with the decomposition method mentioned above.

Figure 4.18 shows the connectivity of 4 different lattices whose length equals to the width. In this figure, the lines indicate the calculation results by our analytical approach, and the points show the results from the simulation. As we can see, our approach produces very accurate numerical results, which have a very good match with the simulation results.

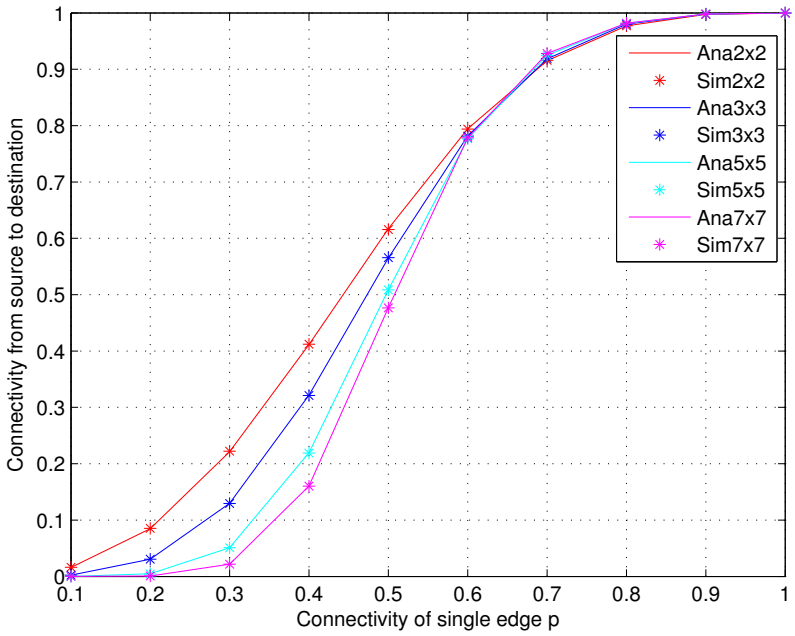


Figure 4.18: Connectivity of $n * n$ Triangle Lattices.

It also exhibits the impact of the bond probability on the lattice connectivity. Larger bond probabilities always achieve higher connectivity. However, the increase is not obvious when the bond probability is less than 0.2 or above 0.7, but the increase is very significant when it is between 0.2 and 0.7, which means that a slight increase of the bond probability within this range will greatly increase the connectivity over lattices. This actually corresponds to the "sharp" transition phase in percolation theory. We can also observe that when n increases, the transition from low connectivity to high connectivity becomes sharper. It is even possible to have higher connectivity for larger lattices when the bond probability is large enough, e.g., $p > 0.65$, although for most small p, smaller lattices have higher connectivity.

4.8 Conclusions

In this chapter, we presented an algorithm to obtain the directed connectivity on square lattices (as used to model the network topology of urban VANETs) in a recursive manner, by proposing a decomposition approach based on the mutually exclusive events and total probability. The results are given in polynomial expressions as a function of the bond probability on each edge. The approach and the obtained results

are validated with the existing approaches and numerical results, which confirm the correctness of the new approach and the accuracy of the analytical results without lengthy simulations. Analysis of the connectivity expressions and the application of these expressions in urban VANETs have been conducted to provide more insights into the directed connectivity in two-dimensional lattice networks. Furthermore, we have presented the extension of our analysis on other lattice structures, i.e., triangular lattice, to show the versatility of our approach.

Chapter 5

Security and Privacy Preserving in Opportunistic Routing

5.1 Overview

During the geographic routing, routing metric information is used for routing decision. However, such information can be user privacy sensitive, e.g., user location information or visiting frequency of a location. In this chapter, we particularly focus on such privacy issues. Besides, we take into consideration of the general security and privacy concerns of VANET, i.e., data confidentiality and anonymity. We propose a security framework for the general routing metric-based routing schemes, to which geographic routing in VANETs belongs.

5.2 Background and Related Work

5.2.1 Related Work

Security and privacy are always hot topics in DTN systems, i.e., VANETs [33,84,85]. [84, 86] gave comprehensive introductions of the basic assumptions, requirements, system models, adversary models, design principles and a spectrum of VANET (which is a typical DTN system) security mechanisms.

With the special focus on the anonymous routing, [30] proposed a solution to separate the routing metric from a node's true identity, so that the attackers cannot link the privacy-sensitive routing metric to a specific node. In [31], the authors utilized

an anonymous table which stores pseudonyms along with the routing metric (position data in that paper) for the routing process. [32,33,87] deployed extra servers or DNT gateway nodes or "roadside units" to manage the anonymous nodes, leading to the extra management overhead and security risk. None of such solutions provides the confidentiality of the routing metric, making it possible for attackers to spoil the user privacy. [88, 89] adapted the onion routing [90] to opportunistic networks for anonymity purpose, however, they require the encryption keys of nodes are known to the message source node, which implies a complicated key management.

In terms of the security, [85] used hierarchical identity-based cryptography to achieve authentication and key management. With a similar technique but by introducing pseudonyms, [33] achieved identity anonymity. [91–93] preserved the location privacy of the sender using trusted social contacts. Recent work [34,94] took the privacy of the "metric" information into consideration. However, these papers did not provide user identity anonymity. Besides, this work focused on a specific field, social-based DTN, where the strong social relationship (e.g., community) among nodes (e.g., cellphones) was utilized, and is not applicable to more general DTN schemes, since the social relationship among nodes is not always sufficiently strong and explicit in some DTNs, especially for mobile DTNs. Another direction of the DTN security study focuses on the detection and prevention of the attacks from the internal malicious node, e.g., black-hole [87,95–99] and Sybil attacks [100,101]. This direction is from a different perspective and thus less related to the main focus of this dissertation.

5.2.2 Security and Privacy Goals

We now introduce the general security and privacy properties that our design can provide:

- **Authentication**: Valid users must be authorized by a Certificate Authority (CA) and they can verify each other.
- **Data integrity**: A user should be able to detect the message change or damage, caused during its transmission, by either intentional or unexpected factors.
- **Data confidentiality**: The secret data is only visible to eligible users.
- **Non-repudiation**: No user can deny their past behaviors, e.g., signing, relaying a message, etc. Every node should be responsible for its behaviors.

Different from other schemes, when taking the routing metric issue into consideration, our scheme can preserve the privacy in the following two aspects:

- **Identity anonymity**: The true identity of a user should not be exposed during any networking activities, including authentication, safety beacons broadcasting, etc.
- **Users' routing metric confidentiality**: The routing metric information has been extensively utilized in opportunistic routing. The protection on such information is essential to preserve the users' privacy.

Other requirements related to the security management are revocation and traceability. Since they are less related to the routing process, we do not have them discussed in this dissertation. But with the anonymous authentication in our framework, those properties are also achievable [102]. This is because despite the anonymous authentication, according to our design, the CA can still reveal the true identity of the owner from its anonymous signature. Since the CA is trusted by all nodes, we do not consider it as a privacy breach.

5.2.3 Threats and Adversaries

On the other hand, we review the possible threats and adversaries in the routing process, which we focus on.

Threats

Threats in mobile network systems can be categorized into two types: active and passive. In active attacks, the adversaries take active actions to incur damages to the network. Typical active attacks include:

- **Message forging/cheating**: The attackers send fake messages for malicious purposes. They either cheat on their identity, e.g., Sybil attack, or simply send fake information, e.g., dishonest routing metric.
- **Message modification/dropping**: Attackers may modify or damage the messages they received and forward them to other nodes, causing disorder. Attackers may even drop the messages to conduct a black-hole attack.
- **Message relay/replay attack**: The attackers intercept the communication and maliciously delay, replay, or manipulate the network messages of other valid parties.

For passive attacks, adversaries are usually referred to as "curious but honest", which means they intend to peek at others' secret or private information but do not

conduct active actions to spoil the system. Typical passive attacks include:

- **Message eavesdropping**: Because of the openly shared medium of wireless communications, "curious" attackers can easily eavesdrop the conversation of others, causing damage to the user confidentiality and privacy.
- **Privacy digging**: With the eavesdropped information, the attackers dig up more private information of others. For example, once the attacker intercepts the routing metric (e.g., the visiting frequency to certain locations) of a node, he may learn the node's mobility patterns.

Adversaries

Adversaries can be divided into external and internal ones. External adversaries are those who are not authorized by the CA or whose certificates are revoked by the CA. With the authentication scheme proposed in the work, our framework can resist both passive and active attacks of the external adversaries, because the nodes who fail the authentication verification will be simply ignored by the authorized ones. In contrast, the internal adversaries are those who are authorized, but malicious. They can conduct attacks until they are discovered and then they will be revoked from the trusted group. In this dissertation, we consider the internal adversaries to be passive attackers, which are "curious but honest". Because the discovery and resistance of internal active adversaries can be very complicated and different, which need many other security solutions, we do not cover them in this dissertation.

5.2.4 Cryptographic Tools

Cryptographic tools are important for security scheme designs. In this section, we briefly introduce the cryptographic tools used in our framework. First of all, our anonymous authentication function is achieved by a group signature scheme. Second, the protection of the routing metric confidentiality is essentially an "Yao's millionaire problem", where homomorphic encryption is utilized as a main support of the solution. Finally, the pairing-based Sakai-Ohgishi-Kasahara (SOK) key agreement serves as the basis of the session key distribution.

Group Signature

Group signature is an efficient solution to achieve anonymity authentication. In group signature [103], network nodes are organized in groups and each group has a group manager to represent the members. The main feature of the group signature scheme is that it provides anonymous authentication to the group members. A verifier can determine whether a signer is authorized by a group without knowing or linking the true identity of the signer. Different from the other anonymity techniques, group signature reduces the workload of the public key and certificate distribution and verification operations. As an authentication scheme, group signature can satisfy other basic security requirements, such as message integrity and non-repudiation.

In this dissertation, we choose one of the group signature schemes as our anonymous authentication scheme. It is a bilinear-map based authentication scheme, which is also adopted in an enhanced version [104] of DAA (Directed Anonymous Attestation) [105]. The original DAA was adopted by the Trusted Computing Group for anonymous authentication purposes. It is essentially a group signature scheme.

Yao's Millionaire Problem

In [36], Yao first introduced a problem which is analogous to a more general problem where there are two numbers a and b and the goal is to verify the inequality $a \geq b$ without revealing the actual values of a and b. To achieve the routing metric confidentiality, it is expected that a node can compare its routing metrics with others' without knowing the values of others' routing metrics. In this dissertation, we integrate the solution proposed in [106] to our security framework, as the main idea explained below.

Let all the routing metrics be expressed in a binary form with a fixed length n. For each binary-form routing metric, two sets of its substrings can be constructed, i.e., *0-encoding* and *1-encoding*. For a binary-form routing metric $r = r_n r_{n-1} \dots r_1$, its 0-encoding set S_r^0 is defined as

$$S_r^0 = \{r_n r_{n-1} \dots r_{i+1} 1 | r_i = 0, 1 \leq r \leq n\}, \tag{5.1}$$

while its 1-encoding set S_r^1 is defined as

$$S_r^1 = \{r_n r_{n-1} \dots r_{i+1} r_i | r_i = 1, 1 \leq r \leq n\}. \tag{5.2}$$

A very important conclusion is that for two routing metric values x and y, $x > y$ if there is one common element in both S_x^1 and S_y^0 [106]. This is easy to prove. Because if $x > y$, there must be a position i so that the substring $r_n^x r_{n-1}^x \cdots r_{i+1}^x$ is the same as $r_n^y r_{n-1}^y \cdots r_{i+1}^y$, but $r_i^x = 1$ and $r_i^y = 0$. Thus with the construction of 0-encoding and 1-encoding sets described above, for S_x^1, it must contain an element $r_n^x r_{n-1}^x \cdots r_i^x$; for S_y^0, it must contain an element $r_n^y r_{n-1}^y \cdots r_{i+1}^y 1$, which is identical to $r_n^x r_{n-1}^x \cdots r_i^x$.

Homomorphic Encryption

In the implementation of the solution to Yao's millionaire problem, the homomorphism property of ElGamal encryption is utilized. Encryption schemes with homomorphism property are referred to as homomorphic encryption. The homomorphism property allows a specific type of operation, say $\otimes$, to be applied directly on two ciphertexts, e.g., $Enc(p_1)$ and $Enc(p_2)$, to obtain a result $R = Enc(p_1) \otimes Enc(p_2)$ which can be decrypted. The decryption of R is a result obtained from applying another operation, say $\odot$, on the corresponding plaintexts, which means $D(R) = p_1 \odot p_2$. The operation $\odot$ can be either multiplication or addition, corresponding to multiplication homomorphic and addition homomorphic, respectively. The homomorphism property is a desirable feature since it can operate directly on the ciphertexts, without exposing the plain texts to the parties who perform the operations.

Sakai-Ohgishi-Kasahara (SOK) Key Agreement

To achieve data confidentiality, and also for efficiency consideration, the secret messages are usually encrypted by symmetric encryption schemes, such as AES, etc. Considering the ad hoc environment, it is crucial to have an efficient and light-weight key agreement scheme to manage the huge number of session keys since each pair of users should share a distinct session key. We use a key agreement scheme similar to the Sakai-Ohgishi-Kasahara (SOK) scheme [107], which has also been utilized in DTNs [33]. In the SOK key agreement, there are two groups $\mathbb{G}$ (written additively) and $\mathbb{G}_T$ (written multiplicatively) of order p (a large prime number) and an efficiently computable bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Assuming the CA possesses a master secrete key $s \in \mathbb{Z}_q$ and each user possesses an identity ID. The CA constructs each user i's secret key by calculating $d_i = sH(ID_i) \in \mathbb{G}$, where $H(\cdot)$ is a public hash function mapping an input to an element in $\mathbb{G}$. Under such a scheme, two users authorized by the same CA can non-interactively compute a shared session key with

the identity of the other participant and their own private keys. For example, for users a and b,

$$\begin{aligned} Key_{ab} &= \hat{e}(H(ID_a), d_b) = \hat{e}(d_a, H(ID_b)) \\ &= \hat{e}(H(ID_a), H(ID_b))^s. \end{aligned} \tag{5.3}$$

Dupont and Enge [108] proved that this key agreement is secure in the random oracle model under the bilinear Diffie-Hellman assumption in $\langle \mathbb{G}, \mathbb{G}_T, \hat{e} \rangle$.

5.3 Protocol Setup

In the following sections, we provide the detailed framework design towards a privacy-preserving and secure opportunistic routing in VANETs. Without the topology information and route maintenance processes, routing decision in opportunistic routing is made by exchanging and comparing the routing metrics among individuals, so the nodes that have a larger chance to deliver the message, i.e., nodes with larger routing metric values, are chosen as the relays. Under such a scenario, any one-hop routing follows the protocol flow shown in Fig. 5.1. Four main algorithms are involved in the routing, namely, Routing Request (Alg. 6), Routing Response (Alg. 8), Routing Decision (Alg. 10) and Decision Confirm (Alg. 11).

Anonymous authentication is mainly provided in Sign and Verify algorithms, i.e., Alg. 4 and Alg. 5, respectively. For security concerns, every message sent should be signed first by the sender. Any messages failing to pass the verification will automatically be dropped by the receivers. To achieve the confidentiality of the routing metrics during the routing, Alg. 7 and Alg. 9 are embedded in the Routing Request and Routing Response algorithms, respectively. Necessary processes of the routing metrics are performed by these two algorithms.

The notations of our framework are listed in Table 5.1. Our design is based on the finite-field cryptography, and the cryptographic setup is presented as follows. First, three cyclic groups are chosen: $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, of sufficiently large prime order q. Two random generators are selected such that $\mathbb{G}_1 = \langle P_1 \rangle$ and $\mathbb{G}_2 = \langle P_2 \rangle$ along with a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We write $\mathbb{G}_1$, $\mathbb{G}_2$ additively and $\mathbb{G}_T$ multiplicatively. The pairing $\hat{e}$ is a map [102] with following properties:

1. $\hat{e}$ is bilinear, which means $\hat{e}(aP_1, bP_2) = \hat{e}(P_1, P_2)^{ab}$ for any two integers a and $b \in \mathbb{Z}_q$;

Table 5.1: Notations

Notation	Explanation
$\mathbb{G}_1, \mathbb{G}_2$	Two additive cyclic groups with order q
$\mathbb{G}_T$	A multiplicative cyclic group with order q
$\mathbb{Z}_q$	A integer cyclic groups with order q
P_1, P_2	Generators for $\mathbb{G}_1$ and $\mathbb{G}_2$
$\hat{e}$	A bilinear map: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$
n_t	A timestamp
s, r	Sender and receiver node, respectively
rq, rp	Routing request and routing response, respectively
mtr	Routing metric of a node
(pk, sk)	A key pair: (public key, secret key)
$\mathcal{SL}$	Sending list, containing the chosen relays
EH, DE	Encryption and decryption with ElGamal
Eec, Dec	Encryption and decryption of a symmetric cryptosystem
n	The fixed length of the binary form of the routing metric
TB	A $2 \times n$ table of ciphertexts
CR	A list of ciphertexts with size n
S^0, S^1	0-encoding and 1-encoding sets of a binary string

2. $\hat{e}$ is non-degenerate, which means $\hat{e}(P_1, P_2) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the identity element of $\mathbb{G}_T$;

3. $\hat{e}$ is computable, i.e., there is a polynomial time algorithm for computing $\hat{e}(P, Q)$ for any $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$.

Second, two hash functions are selected, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$, mapping an arbitrary-length binary string to a number and a $\mathbb{G}_1$ element, respectively.

Third, each node has a true and secret identity $f \in \mathbb{Z}_q$. The CA, who issues the certificates, has a secret key (x, y) where $x, y \leftarrow \mathbb{Z}_q$ and a public key (X, Y) where $X = x \cdot P_2 \in \mathbb{G}_2$, $Y = y \cdot P_2 \in \mathbb{G}_2$. The CA manages the true identities of all nodes and issues a certificate to each node. The certificate is a triplet (A, B, C) where $A \leftarrow r \cdot P_1$, $B \leftarrow y \cdot A$ and $C \leftarrow (x \cdot A + fxy \cdot A)$. The number r is randomly chosen from $\mathbb{Z}_q$, so for a specific node with identity f, its certificate is not deterministic. As we can see, the certificate is constructed with the secrete key of the CA, i.e., (x, y), which is the main proof of the CA's attestation. It is also constructed with the true ID of the corresponding node, i.e., f, so that each certificate is specifically created for that specific node.

5.4 Anonymous Authentication

The signing and verification protocols are used to achieve anonymous authentication. The authentication is required for all messages, which means every message has to be signed before sent out. For every message received from other nodes, its signature needs to be verified by the receiver. In this dissertation, we use a scheme similar to DAA [102, 105], which is a group signature scheme.

Algorithm 4 Sign

procedure SIGN(Message msg)
  $a \leftarrow \mathbb{Z}_q;\ z \leftarrow \mathbb{Z}_q$
  $J \leftarrow H_2(msg);\ K = f \cdot J;\ L \leftarrow z \cdot J$
  $R \leftarrow a \cdot A;\ S \leftarrow a \cdot B;\ T \leftarrow a \cdot C;\ \tau \leftarrow \hat{e}(S, X)^z$
  $c \leftarrow H_1(R||S||T||\tau||J||K||L||n_t||msg)$
  $s \leftarrow z + c \cdot f \pmod q$
  if Version 1 **then**
    $\sigma \leftarrow (R, S, T, J, K, c, s, n_t, TTL)$
  else if Version 2 **then**
    $b \leftarrow \mathbb{Z}_q$
    $P \leftarrow b \cdot A;\ Q \leftarrow b \cdot B$
    $\sigma \leftarrow (R, S, T, J, K, c, s, P, n_t, TTL)$
  end if
return σ
end procedure

Algorithm 4 performs the signing on the message and generates a signature σ. It contains a triplet (R, S, T), which can be seen as a shuffle of the true certificate, i.e., (A, B, C), so that every message is signed with an anonymous certificate. The calculation of (J, K, L, c, s) is used to provide the proof of the connection between the certificate and the node's true identity f. n_t is a timestamp providing time information when the message is signed to resist the replay attack. Different from the schemes in [102, 105], there are two versions of Sign algorithm. Version 1 is for the normal usage and version 2 is only used when the sender wants to establish session keys with the possible relays, where a key agreement process will be executed with the help of P and Q. The details of the key agreement process will be discussed in Routing Response algorithm, i.e., Alg. 8.

Verification of the signature is described in Alg. 5. At the beginning, a few inspections are performed for a quick verification. First, the data integrity of the message is provided by checking whether $J \neq H_2(msg)$, so any corruption of the message can be detected. Second, by a quick comparison of $\hat{e}(R, Y)$ and $\hat{e}(S, P_2)$, it checks the

Algorithm 5 Verify

1: **procedure** VERIFY(Message msg, Signature σ)
2: **if** $J \neq H_2(msg)$ or $\hat{e}(R, Y) \neq \hat{e}(S, P_2)$ **then**
3: **return** Reject
4: **end if**
5: $\rho_a^\dagger \leftarrow \hat{e}(R, X); \rho_b^\dagger \leftarrow \hat{e}(S, X); \rho_c^\dagger \leftarrow \hat{e}(T, P_2)$
6: $\tau^\dagger \leftarrow (\rho_b^\dagger)^s \cdot (\rho_c^\dagger / \rho_a^\dagger)^{-c}$
7: $L^\dagger \leftarrow s \cdot J - c \cdot K$
8: **if** $c \neq H_1(R||S||T||\tau^\dagger||J||K||L^\dagger||n_t||msg)$ **then**
9: **return** Reject
10: **end if**
11: **return** Accept
12: **end procedure**

internal relationship between R, S and Y, i.e., $S = a \cdot B = ay \cdot A = y \cdot R$, so that $\hat{e}(R, Y) = \hat{e}(A, P_2)^{ay} \equiv \hat{e}(S, P_2)$.

The following verification, i.e., line 5–7, is a recovering process of τ and L. If the signature σ is correctly generated by the signer and is successfully transmitted without any corruption, τ and L should be recovered by calculating $\tau^\dagger$ and $L^\dagger$. The correctness is shown as: first,

$$L^\dagger = s \cdot J - c \cdot K = (s - cf) \cdot J \equiv L;$$

second,

$$\begin{aligned} \tau^\dagger &= (\rho_b^\dagger)^s \cdot (\rho_c^\dagger / \rho_a^\dagger)^{-c} = \hat{e}(S, X)^s \cdot \hat{e}(T, P_2)^{-c} \cdot \hat{e}(R, X)^c \\ &= \hat{e}(S, X)^s \cdot \hat{e}(P_1, P_2)^{-acxr(1+fy)+acxr} \\ &= \hat{e}(S, X)^{s+cf} = \hat{e}(S, X)^z \equiv \tau. \end{aligned}$$

If τ and L are successfully recovered and other fields, e.g., R, S, T, J, K, n_t and msg are successfully transmitted, the verifier should be able to recover c in line 8 to finish the verification.

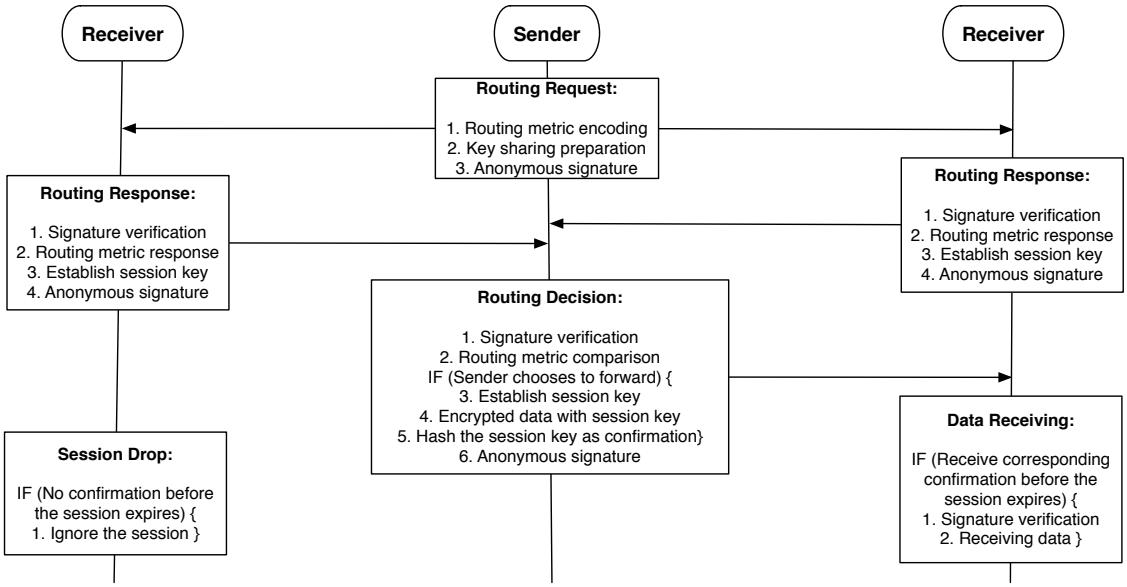


Figure 5.1: Protocol Flow.

5.5 Privacy-sensitive Secure Routing

5.5.1 Routing Protocols

The routing procedure is shown in Fig. 5.1. Before the data transmission, the sender first broadcasts a request message, i.e., rq in Alg. 6, asking other nodes for their routing metrics. Once a neighbor node receives a request, it makes a response, i.e., rp in Alg. 8, which contains its routing metric. Based on the received responses, the sender makes the routing decision in Alg. 6 and chooses those who have larger metric values as the relays. By checking the decision announcement of the sender, the receiver decides its next action, shown in Alg. 11: receiving the data if it is chosen as the relay, or ignoring the data otherwise. During the request and response process, the sender and each chosen relay also finish the key agreement process, so that they can communicate secretly for the following data transmission.

The routing procedure is straightforward, so we focus on the implementation of the two main security properties: routing metric confidentiality and key agreement.

Algorithm 6 Routing Request

1: **procedure** RoutingRequest
2: $\{pk_s, sk_s\} \leftarrow \mathbb{Z}_q$
3: $T \leftarrow Encoding(mtr_s, pk_s)$
4: $msg.data = T||pk_s$
5: $\sigma \leftarrow Sign_{v2}(msg)$
6: Keep track of $\sigma.P$ and Q
7: **return** $rq \leftarrow (msg, \sigma)$
8: **end procedure**

Algorithm 7 Encoding

1: **procedure** Encoding(mtr, pk)
2: Convert mtr to binary form $c_n c_{n-1} \dots c_1 \in \{0,1\}^n$
3: Initialize T as a $2 \times n$ table
4: **for** k from n to 1 **do**
5: $TB[c_k, k] = EH_{pk}(1)$
6: $TB[\bar{c_k}, k] = EH_{pk}(r)$ for a random r
7: **end for**
8: **return** TB
9: **end procedure**

5.5.2 Routing Metric Confidentiality

During the routing "request-response" phase, the sender inquires, obtains and compares other nodes' routing metrics. Then it chooses those who have a higher chance than itself to deliver the message to be the next relay. To keep the confidentiality of the routing metrics, we require that the sender has no access to the plaintext of the routing metrics; instead it performs the comparison without revealing the actual value of others' metric information, known as an Yao's millionaire problem. In this dissertation, we choose and integrate a solution from [106], which is based on the homomorphic encryption, to our framework. Recall the homomorphism property mentioned in Section 5.2. To be specific, the multiplicative homomorphism of ElGamal encryption system, denote as $EH(\cdot)$, is utilized, i.e., line 9 in Alg. 9, so that

$$EH(x_1) \otimes EH(x_2) = EH(x_1 \cdot x_2).$$

The ciphertext of the ElGamal encryption is a pair of values, say (a, b), and the operation $\otimes$ is defined as $EH(x_1) \otimes EH(x_2) = (a_1, b_1) \otimes (a_2, b_2) = (a_1 \cdot a_2, b_1 \cdot b_2)$.

The main idea of the solution to the Yao's millionaire problem is described in Section 5.2, i.e., by checking whether there is a common element in the sender's 1-

Algorithm 8 Routing Response

1: **procedure** RoutingResponse(Request rq)
2: **if** Verify(rq) fail **then**
3: **return** Ignore
4: **end if**
5: $CR \leftarrow CodingResponse(mtr_r, rq.T, rq.pk_s)$
6: $msg.data = CR$
7: $\sigma \leftarrow Sign_{v2}(msg)$
8: Keep track of $\sigma.P$ and Q
9: $Key \leftarrow \hat{e}(rq.\sigma.P, Q)$
10: **return** $rp \leftarrow (msg, \sigma)$
11: **end procedure**

Algorithm 9 Coding Response

1: **procedure** CodingResponse(mtr, TB, pk)
2: Convert mtr to binary form $c_n c_{n-1} \dots c_1 \in \{0,1\}^n$
3: **for** k from n to 1 **do**
4: **if** $c_k == 0$ **then**
5: Add binary string $c_n c_{n-1} \dots c_{k-1} 1$ into set S^0
6: **end if**
7: **end for**
8: **for** each $t = t_n t_{n-1} \dots t_k$ in S^0 **do**
9: $c_t = TB[t_n, n] \otimes TB[t_{n-1}, n-1] \otimes \dots \otimes TB[t_i, i]$
10: Add c_t to set CR
11: **end for**
12: **for** k from 1 to $n - |S^0|$ **do**
13: Add $EH_{pk}(r)$ to set CR for a random r
14: **end for**
15: **return** CR
16: **end procedure**

encoding set S_s^1 and the receiver's 0-encoding set S_r^0, the sender can determine whether its routing metric mtr_s is larger than that of the receiver mtr_r. The implementation details are described as follows. During the routing request phase, the sender performs Encoding algorithm, i.e., Alg. 7, using its own routing metric mtr_s to construct a $2 \times n$ table TB, i.e., line 3–7. Essentially, TB integrates the S_s^1 of the sender metric in an anonymous way, since each element in the table is in a ciphertext form. TB is then included in the routing request message rq and broadcast to the potential relay nodes.

Once received the request, the receiver performs Routing Response algorithm, i.e., Alg. 8, to make a response to the request. The Coding Response algorithm, i.e., Alg. 9, is called at this point. The algorithm first derives the 0-encoding set S_r^0

Algorithm 10 Routing Decision

```
1: procedure RoutingDecision(Response {rp_1, rp_2, ···})
2:     for Any rp_i in {rp_1, rp_2, ···} do
3:         if Verify(rp_i) fail then
4:             return Ignore
5:         end if
6:         CR ← rp_i.msg.data
7:         for Any t in set CR do
8:             k = DE_{sk_s}(t)
9:             if k == 1 then
10:                Go to line 2 and try another rp
11:            end if
12:        end for
13:        Key_i ← ê(rp_i.σ.P, Q)
14:        Add (rp_i, Key_i) to SL
15:    end for
16:    for Any (rp, Key) in SL do
17:        msg.data ← Enc_{Key}(Message)
18:        Send announcement anc ← H(Key)
19:        σ ← Sign_{v1}(msg)
20:        Send data ← (msg, σ)
21:    end for
22: end procedure
```

of receiver's mtr_r, i.e., line 3–7. Then along with the table TB from the sender, it generates CR, i.e., line 8–11, where each element c_t is the result of applying $\otimes$ on the ciphertexts in TB following the rules defined by the element of S_r^0, i.e., t and S_r^0. So S_r^0 is integrated in CR. Because of the homomorphism of the ElGamal encryption EH, each c_t is essentially a ciphertext encrypted by EH. Up to line 11, the size of CR is determined by the number of elements in S_r^0, i.e., $|S_r^0|$. Extra $n - |S^0|$ random ciphertexts are padded into CR for security consideration, i.e., line 12–14. Details will be provided in the security analysis.

Then, the receiver sends CR back to the sender. In Alg. 10, for each CR received by the the sender, it decrypts the ciphertexts contained in each CR_i from receiver i. If there is a result equal to 1, that means there is a common element between S_s^1 and $S_{r_i}^0$ and the sender's metric is larger than that of the receiver i. Thus the sender will not choose i as its next relay. This conclusion owes to the ingenious constructions of TB and CR. However, if all ciphertexts in CR_i are not decrypted to 1, it means the metric of the receiver i is larger than that of the sender and receiver i can be chosen as the next relay. The effectiveness and security of such a solution is further

Algorithm 11 Decision Confirmation

```
1: procedure DecisionConfirmation(Announcement {anc})
2:     if H(Key) == anc then
3:         Receive data
4:         if Verify(data) fail then
5:             return Reject
6:         else
7:             Message ← Dec_Key(data)
8:             return Accept
9:         end if
10:    else
11:        return Ignore
12:    end if
13: end procedure
```

explained later.

5.5.3 Key Agreement

During the routing request and response processes, the sender and each of the chosen relays establish a unique secret session key, with which the data can be encrypted so the pair-wise confidentiality can be achieved. The second version of the *Sign* algorithm is used for the key agreement purpose. Assume b_s and b_r are two random numbers generated by the sender and receiver respectively. In the second version of Alg. 4, when sending a request, the sender calculates $P_s = b_s \cdot A_s$, $Q_s = b_s \cdot B_s$ and broadcasts P_s. In Alg. 8, when a receiver receives P_s, it first generates $P_r = b_r \cdot A_r$ and $Q_r = b_r \cdot B_r$, and then obtains a session key $Key_{rs} = \hat{e}(P_s, Q_r) = \hat{e}(b_s \cdot A_s, b_r \cdot B_r) = \hat{e}(A_s, B_r)^{b_s b_r} = \hat{e}(P_1, P_1)^{r_s b_s r_r y b_r}$. Note that this session key is only valid when the receiver is chosen by the sender as a relay. The receiver includes its P_r in its response rp to the sender. According to the responses received, the sender chooses the proper receiver as the relay and establishes the session key $Key_{rs} = \hat{e}(P_r, Q_s) = \hat{e}(b_r \cdot A_r, b_s \cdot B_s) = \hat{e}(A_r, B_s)^{b_r b_s} = \hat{e}(P_1, P_1)^{r_r b_r r_s y b_s}$.

5.6 Security and Performance Evaluation

5.6.1 Security Analysis

Security of the Signature

The security of the signature is guaranteed by the hardness of the **LRSW Assumption** [109]: Suppose that a *Setup*(1^k) algorithm generates a multiplicative group $\mathbb{G}$ with a generator g and an order q, where k is a parameter related the security level. There exist $X, Y \in \mathbb{G}$, $X = g^x$ and $Y = g^y$. Let $O_{X,Y}(\cdot)$ be an oracle that, with an input value $m \in \mathbb{Z}_q$, outputs a triplet $A = (a, a^y, a^{x+mxy})$ for a randomly chosen $a \in \mathbb{G}$. Then for all probabilistic polynomial-time adversaries $\mathcal{A}$, $v(k)$, which is defined as follows, is a negligible function:

$$
\begin{aligned}
&Pr[(q, \mathbb{G}, g) \leftarrow Setup(1^k); x \leftarrow \mathbb{Z}_q; y \leftarrow \mathbb{Z}_q; \\
&X = g^x; Y = g^y; (m, a, b, c) \leftarrow \mathcal{A}^{O_{X,Y}}(q, \mathbb{G}, g) : \\
&a \in \mathbb{G} \wedge b = a^y \wedge c = a^{x+mxy}] = v(k).
\end{aligned}
$$

That means given the group setup $(q, \mathbb{G}, g)$ and the system public key (X, Y), it is impossible for a polynomial-time adversary to construct a triplet (a, a^y, a^{x+mxy}) without knowing the secret key (x, y), where a and m are random numbers. This assumption guarantees the effectiveness of our authentication scheme. Only the CA who possesses the secret key x and y can construct valid certificates to users. Without knowing the secret key, a malicious node cannot forge a valid certificate $(A, B = y \cdot A, C = x \cdot A + fxy \cdot A)$, where the group in our scheme, i.e., $\mathbb{G}_1$, is written additively but isomorphic to the multiplicative form in the assumption above.

Security of the Key Agreement Process

The security of the key agreement is guaranteed by the **BDH (Bilinear Diffie-Hellman) Assumption** [110]. Suppose that $\mathbb{G}_1$ is an additive group with a generator g, $\mathbb{G}_2$ is a multiplicative group and $\hat{e}$ is a bilinear map of $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ as described in Section 5.2. Let $P \in \mathbb{G}_1$, $a, b, c \leftarrow \mathbb{Z}_q$, then $a \cdot P, b \cdot P, c \cdot P \in \mathbb{G}_1$. Let $O_{a \cdot P, b \cdot P, c \cdot P}(\cdot)$ be an oracle that outputs $r = \hat{e}(P, P)^{abc} \in \mathbb{G}_2$. Then for all probabilistic polynomial-time

adversaries $\mathcal{A}$, $v(k)$ defined as follows is a negligible function:

$$Pr[(q, \mathbb{G}_1, g, \mathbb{G}_2, \hat{e}) \leftarrow Setup(1^k); a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q;$$
$$c \leftarrow \mathbb{Z}_q; P \leftarrow \mathbb{G}_1; r \leftarrow \mathcal{A}^{O_{a\cdot P, b\cdot P, c\cdot P}}(q, \mathbb{G}_1, g, \mathbb{G}_2, \hat{e}) :$$
$$r = \hat{e}(P, P)^{abc}] = v(k)$$

In the key agreement process mentioned, the session key established by any two nodes a and b can be expressed as $\hat{e}(A_s, B_r)^{b_s b_r} = \hat{e}(A_r, B_s)^{b_s b_r} = \hat{e}(P_1, P_1)^{b_a r_a b_b r_b y}$, where random numbers r_a, r_b are introduced in the certificate construction process and random numbers b_a, b_b are introduced in the second version of the Sign algorithm, i.e., Alg. 4. During the wireless transmission, an adversary can easily eavesdrop $P_a = b_a r_a \cdot P_1$ and $P_b = b_b r_b \cdot P_1$. Assume it also knows the system public key $Y = y \cdot P_1$. Because of the hardness of the BDH assumption, the adversary is not able to recover the session key $\hat{e}(P_1, P_1)^{b_a r_a b_b r_b y}$ with P_a, P_b and Y known.

Security of the Routing Metric Comparison

The correctness has been explained before. As the way how S^0 and S^1 are constructed, the confidentiality of the routing metric lies on the confidentiality of these two sets. The 1-encoding set S_s^1 of the sender's routing metric is embedded into the table TB which is broadcast during the routing request phase. From the adversary point of view, the table TB does not reveal any information about S_s^1 because this table contains only ciphertexts encrypted by the sender's public key and thus only the sender can decrypt them. According to Alg. 7, although all $T[x_i, i] = E(1)$ where $1 \leq i \leq n$, these $E(1)$s are different from each other because ElGamal encryption is probabilistic. Because of the security of ElGamal encryption, it is also not feasible for the adversaries or the receiver to distinguish the $E(1)$ and $E(r)$ where r is a random number. Thus, the secrecy of the sender's routing metric is preserved.

For each receiver, the 0-encoding set of the routing metric S_r^0 is embedded in its CR list, which is sent back to the sender in the routing response phase. During the calculation of CR, the multiplicative homomorphism of ElGamal encryption is applied and each element in the 0-encoding set corresponds to an element in the CR. However, the size of the 0-encoding set, i.e., $|S^0|$, is determined by the number of 0s in the binary-form of the receiver's routing metric (according to its definition in Section 5.2) and can be smaller than n. Thus in order to conceal the value of

$|S^0|$, extra $n - |S^0|$ random encryptions are padded into CR, so that the size of CR is always n. Even with the CR eavesdropped, the adversary cannot obtain any information of the receiver's routing metric since it contains n ciphertexts. Because of the homomorphism operations and the padding, even the sender will not be able to obtain extra information of mtr_r except for the comparison result between mtr_s and mtr_r. So the secrecy of the receiver's routing metric is preserved.

5.6.2 Efficiency Analysis

Computation Overhead

Since all messages are signed before being sent out and verified after being received, the signing and verification processes introduce some computation overhead. According to the existing implementation results from [111], the most expensive operations are the scalar multiplication in $\mathbb{G}_1$, exponentiation in $\mathbb{G}_T$ and pairing evaluation. In comparison, the overhead of the hash functions and arithmetic operations in $\mathbb{Z}_q$ is very small. Because of the bilinear property of the mapping $\hat{e}$, we can transform some exponentiations in $\mathbb{G}_T$ into scalar multiplications in $\mathbb{G}_1$ for a faster implementation. For example, to calculate $\hat{e}(S, X)^x$ in the Sign algorithm, we can first compute $x \cdot S$ and then obtain the value of $\hat{e}(S, X)^x$ by computing $\hat{e}(x \cdot S, X)$. This trick also applies to the Verify algorithm, i.e., $(\rho_b^\dagger)^s = \hat{e}(s \cdot S, X)$, $(\rho_c^\dagger / \rho_a^\dagger)^{-c} = \hat{e}(-c \cdot T, P_2) \cdot \hat{e}(c \cdot R, X)$. If we let $n \cdot \mathbb{G}_1$ denote n scalar multiplications in $\mathbb{G}_1$, and $m \cdot P$ denote m pairing operations. Then by applying the above trick, we can obtain the following computation overhead for signing: Sign v1, $6 \cdot \mathbb{G} + 1 \cdot P$; Sign v2, $8 \cdot \mathbb{G} + 1 \cdot P$; Verify, $5 \cdot \mathbb{G} + 5 \cdot P$. Here, we evaluate these operations with the implementation results from [111] obtained on a Pentium IV 3.0 GHz machine. To achieve an 80-bit security level, approximately the same level as a standard 1024-bit RSA signature, a 512-bit prime number q and a group $\mathbb{G}_1$ where each element is 160-bit long are chosen. The experiment results show that the average time required for a scalar multiplication in $\mathbb{G}_1$ and a pairing operation is 3.08 ms and 2.91 ms, respectively. So the computation overheads for signing and verification are 21.39 ms (Sign v1), 27.55 ms (Sign v2) and 29.95 ms, respectively.

The secret routing metric comparison also introduces extra computation. In the Routing Request phase, i.e., Alg. 6, the sender encrypts n 1s and n random numbers to fill the table TB with size $2 \times n$. Because these encryptions are ciphertexts of either 1 or random numbers, they can be pre-calculated and will not introduce extra

computation overhead in real time. Once receiving TB, the receiver calculates CR, which contains n ciphertexts. Among these ciphertexts, $n - |S_r^0|$ of them are the results of random number encryptions, which can be pre-calculated and $|S_r^0|$ of them are calculated by applying arithmetic multiplication on the elements in the table TB, whose computation overhead can be neglected. After the sender receives a CR, it decrypts the elements in the list. If one of the elements is decrypted to 1, the rest of the elements in the CR are ignored. Only when all the elements are decrypted to values which are not 1, the corresponding node will be chosen as the relay. With n elements in the CR, a sender performs decryption at most n times. Because each ElGamal decryption takes approximately 0.54 ms[1], for each receiver whose CR is received by the sender, the sender will spend at most $n \cdot 0.54 = 3.78$ ms to make a decision when we consider $n = 7$.

When the sender chooses a relay, they establish a session key. For each node, it performs two scalar multiplications and one pairing operation and thus the overhead introduced is $2 \cdot \mathbb{G} + 1 \cdot P$ with roughly 9.07 ms on the benchmark platform. Note that the two scalar multiplications have also been counted in the second version of the Sign algorithm.

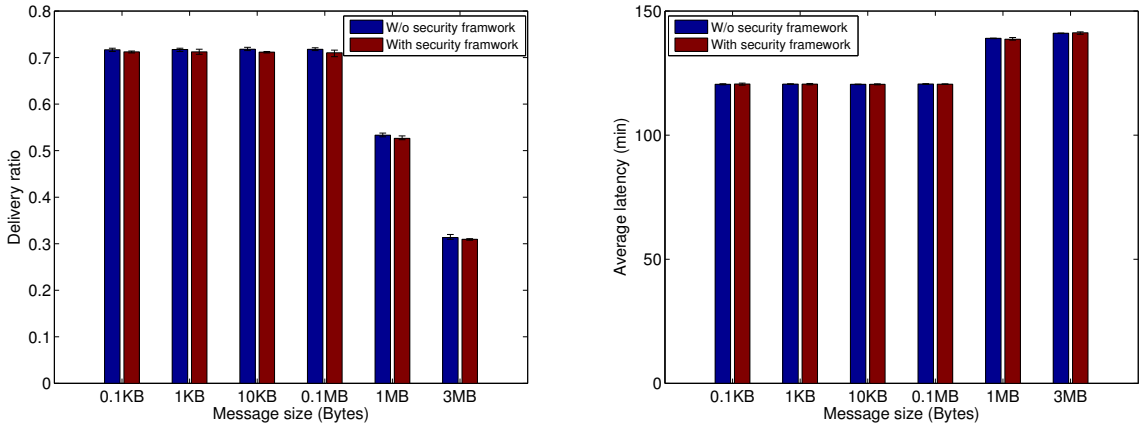
Communication Overhead

To achieve an 80-bit security level, we choose a prime q that is 512-bit long, i.e., $|q| = 512$ bits and groups with element length of 160-bit long, i.e., $|\mathbb{G}| = 160$ bits. Because the signature needs to be included in each broadcast message, the communication overhead of the authentication is determined by the signature size, which is approximately $5|\mathbb{G}| + 4|q| = 2.848$ Kb for version 1 and $6|\mathbb{G}| + 4|q| = 3.008$ Kb for version 2. If we consider that n_T and TTL do not require as many as 512 bits, the overhead can be even smaller.

For the routing metric comparison, the sender needs to broadcast its TB table along with its public key in the routing request phase, which has size $2n|\mathbb{G}| + |q|$. Let $n = 7$, then the communication overhead in the routing request message is around 2.752 Kb. In the routing response phase, each receiver sends back the CR with size $n|\mathbb{G}| = 1.12$ Kb.

For the key agreement, the only communication overhead is the transmission of

[1]The decryption of an ElGamal ciphertexts takes 1 exponentiation operation in $\mathbb{G}_T$ and 1 arithmetic multiplication. Because 1 exponentiation operation in $\mathbb{G}_T$ takes 0.54 ms [111] and arithmetic multiplications can be neglect, one decryption takes 0.54 ms.



(a) Delivery Ratio (b) Average Latency

Figure 5.2: Network Performance Comparison.

P_s from the sender to receiver and P_r from a receiver to the sender, with the size of $|\mathbb{G}| = 0.16$ Kb each. Again, this has already been counted in the overhead of the second version of the signature.

5.6.3 Simulation Evaluation

Simulation Study: Case One

We use the simulator theOne [65] for the simulation, where the real-world traces of 500 nodes (vehicles) are imported to simulate the node mobility. Message sources are randomly selected from all nodes and 40 regions in the city are randomly selected as destinations. According to IEEE 802.11p, we set the transmission rate and range to 6 Mbps and 250 m [112], respectively. Each vehicle carries a buffer of 50 MB. The total simulation time is 6 hours while the message TTL is 5 hours. To evaluate the security scheme, we hard-coded the overheads into the simulator. The message generation interval is set to 100 s. The message size varies from 0.1 KB, 1 KB, 10 KB, 0.1 MB, 1 MB to 3 MB.

Figure 5.2(a) and (b) show the results of the two most important network performance indicators, i.e, delivery ratio and average latency, respectively. The delivery ratio decreases with the increase of the message size. With a larger message size, the storage competition of the buffer at each node is severer, leading to a lower delivery ratio. But when the message size is much smaller than the buffer size, say

from 0.1 KB to 0.1 MB, the message size impact is not apparent under the given message generation intervals. On the other hand, the average latency increases. This is mostly because of the message retransmissions, when the message size increases and the buffer competition becomes fierce.

We are more interested in the impact of our security framework on the original routing scheme. Results of the two scenarios, with and without the proposed security framework, are plotted in Fig. 5.2(a) and (b). Both figures show the same trend, i.e., the network performance with the proposed security framework is almost the same as that of the original routing scheme, which means although we have added security features to the routing, the network performance is not significantly affected by the extra overhead introduced. According to the trace study about the contact duration distribution, the average pair-wise vehicle contact duration time is 63.7 s for the transmission range 250 m. However, all the security operations, e.g., signing, verification and routing decision, are in the millisecond order. On the other hand, the extra communication overhead introduced by the framework is in the Kbits order, much smaller than the message size of 0.1 MB, 1 MB and 3 MB. For very small messages, i.e., those with size of 0.1 KB, 1 KB or 10 KB, even with the extra overhead from the framework, their total size is still too small to make any noticeable impact on the networking system.

Another reason for the unnoticeable impact is that the current network setting (e.g., network traffic load) has not pushed the network to its capacity limit. When the network is not saturated, the small security overhead is not easy to be observed. In order to push the network to its capacity limit, we change the network settings a little bit and conduct the simulation study case two as follows.

Simulation Study: Case Two

The previous simulation results do not show an obvious impact of the security framework overhead. In fact, in a non-saturated network, the framework impact is hard to be detected since both the computation and the communication overheads are relatively small. In order to better understand the impact of the security framework and make such impacts obvious, we have to push the network towards its capacity limit. This can be achieved by applying the following approaches: 1), increasing the message generation rate to increase the total network traffic workload; 2), densifying the node contacts to accelerate the traffic transfers. Given the difficulty of apply-

ing the two approaches in the previous simulator, i.e., theOne (mainly due to the limitation of the simulator and traces), we conduct an abstract scenario to model the opportunistic message forwarding process, with higher message generation rates and denser node contacts. We have to mention that, although the network setting is changed, the simulation still reflects the essential of the VANET geographic routing, i.e., the opportunistic contact and routing. The change is only made to help us better understand the impact of the security framework under the extreme condition.

The new simulation is conducted using network simulator, OMNeT++, which provides a finer granularity and better flexibility for simulation settings. The security framework is designed for general opportunistic-based routing schemes, to which the geographic routing in VANETs belongs. In VANETs, MANETs or DTNs, a message forwarding only happens when nodes encounter with each other opportunistically. In the simulation, we use opportunistic links between nodes to model the opportunistic nodes contacts, so that at any time instance, a link between two neighbor nodes can be either active (i.e., nodes encounter with each other) or inactive (i.e., no encounter happens).

In terms of routing, whenever a message carrier has an active link to a neighbor (i.e., encountering with the neighbor node), it forwards the messages only when the neighbor is "closer" (i.e., routing metrics depending on different routing algorithms) to the destinations. In our simulation, we use the hop distance to the destination as the routing metric. In this sense, we are simulating the routing metric-based opportunistic routing.

The simulation is conducted on 30 nodes with a random topology and each node generates messages following a Poisson process. In different simulation settings, the mean value of the message general interval varies among 0.01s, 0.1s and 1s, which leads to much higher message generation rates. To achieve denser node contacts, we set the inter-contact time of any pair of neighboring nodes is 10 times the complete transmission time of one message (including the message transmission and security overhead). Such inter-contact time is much smaller than that of the original trace, indicating a much frequent node contact. We let the message size vary among 0.1 KB, 1 KB, 10 KB, 0.1 MB and 1 MB. The message source and destination are randomly chosen among all nodes. We assume each node carries a buffer with size 30 MB (smaller buffer size also makes it easier to reach to network capacity). The total simulation time for each parameter setting is 500 s.

We compare the performance results of different scenarios, i.e., with or without

security framework, with different message sizes, and with different message generation intervals. Similar to Chapter 3, we investigate the results with four performance metrics, i.e., delivery ratio, average latency, overhead ratio and average hop count, as shown in Fig. 5.3.

From Fig. 5.3(a), we see that with the increase of the message size, the delivery ratio decreases. This is mainly due to the limited buffer size. With a larger message size, the storage competition of the buffer at each node is severer, leading to a lower delivery ratio. However, when the message size is much smaller than the buffer size, say 0.1 KB or 1 KB, the message size impact is not very apparent with given message generation intervals. We can also observe that, with a shorter message generation interval, the delivery ratio is lower. This is because the smaller the message generation interval, the more messages are generated, leading to a severer buffer competition. However, if the message size is too small, say 0.1 KB, compared with the buffer size, the impact of the traffic intensity is less apparent. These results also echo the ones of the simulation study of case one, i.e., Fig. 5.2 (a).

In terms of the security framework, its impact is more apparent when the message size equals to 1 KB and the message generation interval equals to 0.01 s. With the security framework, the size of the signature is at most 3.008 Kb (i.e., 0.376 KB). For message size 10 KB, 0.1 MB and 1 MB, the overhead is too small to make an apparent impact. The security framework is supposed to have a great impact on the messages with small sizes, i.e., 0.1 KB and 1 KB. However, because messages with size 0.1 KB are too small, even with the signature overhead, the size 0.476 KB is still too small to make obvious performance difference. The difference is shown with the messages with size 1 KB though. The effect is also particularly obvious when the message generation interval is short (i.e., 0.01 s), indicating the traffic intensity is high.

Figure 5.3 (b) shows some interesting results for the average latency. As we can see, for each message generation interval, the result (the bars with the same color) fluctuates. For most cases (except the ones whose message generation interval is 1 s), the delay first increases then decreases with the increase of the message size. This is a complex result of two factors: the message size and the buffer size. When the message is very small, all transmissions are smooth without much buffer competition, leading to a high delivery ratio and short delay. However, with the increase of the message size, the buffer competition is fierce, leading to the decrease of the delivery ratio and the increase of the delay, mostly because of the message retransmissions. As

the message size keeps increasing, the buffer resource becomes too limited to support the majority of the message transmissions, leading to a very low message delivery ratio. In such cases, the successfully transmitted messages are usually those whose sources are close to their destinations. That explains the short average latency. This can also be shown with the small average hop count for larger message size cases shown in Fig. 5.3 (d). The cases with interval 1 s only show the increase phase. This also reminds us of the similar results from the simulation study case one in Fig 5.2 (b), i.e., the delay trend only shows the increase phase when the interval is large (100 s in simulation study case one). The impact of the security overhead is also more obvious when message size equals 1 KB with interval 0.01 s.

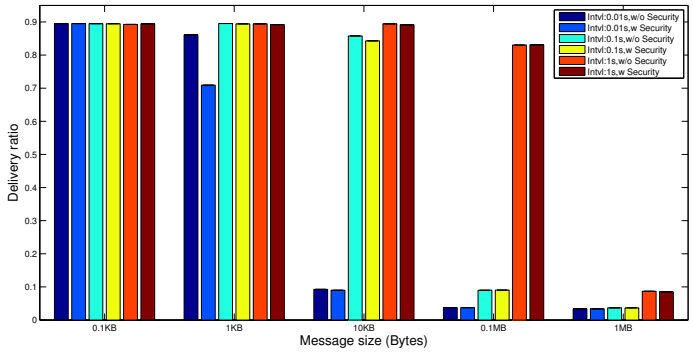
Figure 5.3 (c) shows the performance results for the overhead ratio, as defined in Section 3.5.3. With the same message generation interval, when the message size increases, the buffer competition becomes fierce, leading to a larger amount of message retransmission, i.e., increase of the overhead. The security framework increases the original message size, leading to a larger overhead, especially for messages with original size 1 KB and 10 KB. For the same message size, with a larger message generation interval, fewer messages are generated in the network, leading to a lower resource competition and less overhead.

Figure 5.3 (d) shows the performance results of the average hop count. As mentioned, for the same message generation interval, with the increase of the message size, the delivery ratio decreases. And the delivered messages are those whose sources and destinations are close. This explains the decrease of the average hop count. The security framework increases the original message size, leading to a smaller average hop count. For the same message size, with a larger message generation interval, fewer messages are generated in the network, leading to the lower resource competition and higher delivery ratio; besides, messages have a larger chance to be transmitted farther away from the sources with a larger average hop count.

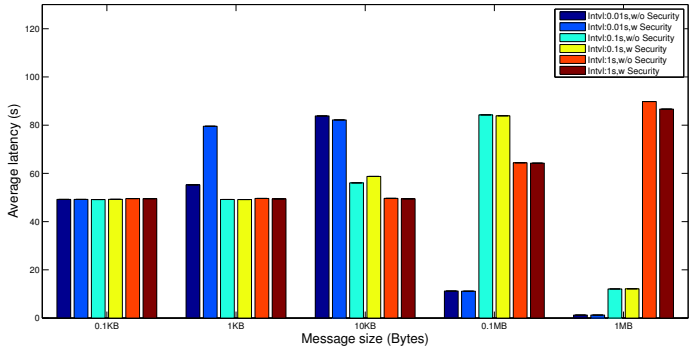
5.7 Conclusions

Opportunistic routing is widely employed in many mobile networks, e.g., DTNs, VANETs and mobile sensor networks, etc, as well as in the proposed geocast scheme in this dissertation. Considering that nodes' local and private information (i.e., the routing metric) is extensively utilized in opportunistic routing, in this chapter, we focused on its security and privacy concerns, and proposed a framework for oppor-

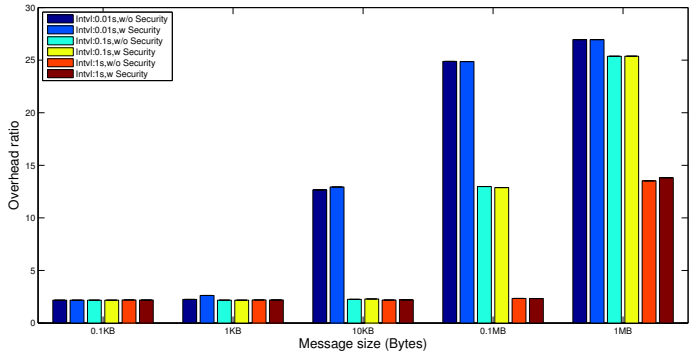
tunistic routing, providing various security and privacy preserving properties. A comprehensive evaluation was conducted to show the security and network performance impact of the proposed framework.



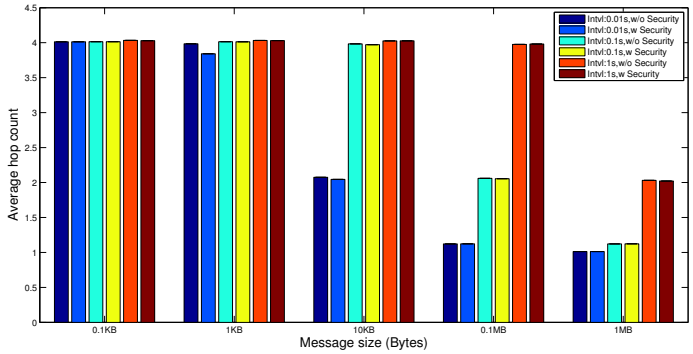
(a) Delivery Ratio



(b) Average Latency



(c) Overhead Ratio



(d) Average Hop Count

Figure 5.3: Network Performance Comparison.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we have investigated three important and fundamental aspects in geocast of VANETs: 1) A trace-driven mobility-contact-aware geographic routing scheme (GeoMobCon) for large-scale urban VANETs geocast. Our scheme employs the vehicle mobility information at different levels (i.e., macroscopic and microscopic mobilities) and vehicle contact history information, both of which are extracted from the real-world GPS traces of vehicles in Shanghai, China. Our scheme is featured by its scalability and adaptivity to large-scale urban VANETs. 2) Thorough analysis of network connectivity during the message broadcast phase. For simplicity and generality purposes, we model the urban VANET as a square grid lattice. The proposed algorithm gives the exact analytical solution to the directed connectivity problem on square lattices and can quickly determine the network connectivity without lengthy simulations. We also explore the obtained analytical expressions and analyze the impact of the bond probability, and the lattice size and ratio on network connectivity. It is also shown that the algorithm can be applied to other lattice structures to be more general. 3) A secure and privacy-preserving framework for opportunistic routings, which can achieve following properties: confidentiality of the routing metric, anonymous authentication and efficient key agreement. A comprehensive evaluation is conducted, including security analysis and performance evaluation with both cryptographic implementation specifications and event-driven simulations.

6.2 Future Work

The future research plans beyond this dissertation, revolve around the data-driven geographic routing design and the applications of the routing and connectivity analysis. Related security and privacy issues are also of our interest. We believe that various applications will appear and attract more attention.

6.2.1 Data-driven Geographic Routing

Real-world data analysis can always provide valuable insights of the reality. Therefore, one of our future research directions will be continuously on data analysis and knowledge discovery. Our current analysis mostly focuses on the large (i.e., city-wide) scale mobility, which means the movement between different regions or districts. However, mobility of a smaller scale, e.g., the movement within a district or between different roads, is not fully investigated. We believe that with a smaller granularity data analysis, we can discover more insights about the vehicle on-road movement and the contact behaviors with nearby vehicles. Besides, in Chapter 2, we mentioned the transition residence time which vehicle spends in a region. However, this is not fully considered in the current geographic routing design. Following the routing scheme description in Chapter 3, how to utilize the transition residence time to improve our design is an interesting problem. One possible solution is to take the transition residence time into consideration when constructing the optimal routing path. It is also of great interest to investigate some practical applications of the geographic routing scheme. Some examples can be location-based services and data collection applications.

6.2.2 Connectivity Analysis in Dynamic Networks

Our approach still encounters an exponential complexity (which is much better than the PIE principle that has the combinatorial factor on the exponent) of connectivity expressions. Dynamic programming approaches can be used to leverage the known connectivity of smaller components, but when the analytical expressions are reassembled, it will lead to extremely high-order polynomials. Fortunately, in most engineering problems, one dimension is often of limited size while the other dimension can grow, which keeps the exponential complexity manageable. Also depending on the needed precision, polynomial truncation can be used to limit the length and complexity of these expressions.

Although it is our goal to shed new light on the directed percolation problem, since the polynomial grows quickly with the increase of the lattice size, so far it is not possible for us to obtain the exact expressions for arbitrarily large size lattices. However, the polynomial expressions in terms of p for small m and n are readily available [83], and we are intended to explore more properties based on these analytical expressions. The future work will include how to verify the convergence behaviors mentioned in the literature using our exact results and exploring the math properties and coefficient patterns of the polynomials, which can reveal more insights of the connectivity.

The proposed algorithm can also be applied to other types of networks, as long as there exit probabilistic or weighted edges or nodes, i.e., dynamic networks. Social network can be considered as one of such networks, where we can define each person as a network node and the social relationship strength as the weight of the edges. Therefore, our analysis can be used to evaluate the relationship strength between a pair of indirectly connected persons.

6.2.3 Security and Privacy Preserving of VANETs Routing

Although with the proposed security framework, we are able to provide fundamental security and privacy features, there are still other security requirements can be integrated into the current framework. One of the features is revocation. However, in a distributed system like VANET which lacks a central control, efficient revocation is still very challenging.

Our current scheme is only effective with the defence against the external malicious users, who do not possess valid certificates from the certificate authority. Another direction can be the protection against internal malicious users, who possess valid certificates but cheat on purpose. Although it is a different problem, it is also a typical problem in reality. That means new techniques and security tools need to be invented and identified to deal with it. Related topics include position validation in MANETs and identity management, etc.

Besides, because the protection of the routing metric information is modeled as Yao's millionaire problem, it will be of our interest to explore other solutions to this problem in order to further simplify the routing and communication process.

Bibliography

[1] Elmar Schoch, Frank Kargl, Michael Weber, and Tim Leinmuller. Communication patterns in VANETs. *Communications Magazine, IEEE*, 46(11):119–125, 2008.

[2] IEEE guide for Wireless Access in Vehicular Environments (WAVE) - architecture. *IEEE Std. 1609.0-2013*, pages 1–78, 2013.

[3] José Santa, Antonio F Gómez-Skarmeta, and Marc Sánchez-Artigas. Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks. *Computer Communications*, 31(12):2850–2861, 2008.

[4] Fay Hui and Prasant Mohapatra. Experimental characterization of multi-hop communications in vehicular ad-hoc network. In *VANET*, pages 85–86. ACM, 2005.

[5] Yanyan Zhuang, Jianping Pan, Yuanqian Luo, and Lin Cai. Time and location-critical emergency message dissemination for vehicular ad-hoc networks. *Selected Areas in Communications, IEEE Journal on*, 29(1):187–196, 2011.

[6] Young-Bae Ko and NF Vaidya. GeoTORA: A protocol for geocasting in mobile ad-hoc networks. In *ICNP*, pages 240–250. IEEE, 2000.

[7] Young-Bae Ko and Nitin H Vaidya. Flooding-based geocasting protocols for mobile ad-hoc networks. *Mobile Networks and Applications*, 7(6):471–480, 2002.

[8] Wen-Hwa Liao, Yu-Chee Tseng, Kuo-Lun Lo, and Jang-Ping Sheu. GeoGRID: A geocasting protocol for mobile ad-hoc networks based on grid. *J. Internet Tech.*, 2000.

[9] Tomasz Imieliński and Julio C Navas. GPS-based geographic addressing, routing, and resource discovery. *Communications of the ACM*, 42(4):86–92, 1999.

[10] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *Networking, IEEE/ACM Transactions on*, 16(1):77–90, 2008.

[11] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3):19–20, 2003.

[12] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM*, volume 6, pages 1–11. IEEE, 2006.

[13] Lei Zhang, Maryam Ahmadi, Jianping Pan, and Le Chang. Metropolitan-scale taxicab mobility modeling. In *Globecom*, pages 5404–5409. IEEE, 2012.

[14] Lei Zhang, Boyang Yu, and Jianping Pan. GeoMob: A mobility-aware geocast scheme in metropolitans via taxicabs and buses. In *INFOCOM*, pages 1779–1787. IEEE, 2014.

[15] Madhav Desai and D Manjunath. On the connectivity in finite ad-hoc networks. *Communications Letters, IEEE*, 6(10):437–439, 2002.

[16] Olivier Dousse, Patrick Thiran, and Martin Hasler. Connectivity in ad-hoc and hybrid networks. In *INFOCOM*, volume 2, pages 1079–1088. IEEE, 2002.

[17] Hiroki Nishiyama, Thuan Ngo, Nirwan Ansari, and Nei Kato. On minimizing the impact of mobility on topology control in mobile ad-hoc networks. *Wireless Communications, IEEE Transactions on*, 11(3):1158–1166, 2012.

[18] Shibo He, Jiming Chen, and Youxian Sun. Coverage and connectivity in duty-cycled wireless sensor networks for event monitoring. *Parallel and Distributed Systems, IEEE Transactions on*, 23(3):475–482, 2012.

[19] Kenji Miyao, Hidehisa Nakayama, Nirwan Ansari, and Nei Kato. LTRT: An efficient and reliable topology control algorithm for ad-hoc networks. *Wireless Communications, IEEE Transactions on*, 8(12):6050–6058, 2009.

[20] Arakaparampil M Mathai. *An introduction to geometrical probability: distributional aspects with applications*, volume 1. CRC Press, 1999.

[21] François Baccelli and Bartlomiej Blaszczyszyn. *Stochastic geometry and wireless networks: volume 1: Theory*, volume 1. Now Publishers Inc, 2009.

[22] Christian Kurrer and Klaus Schulten. Dependence of percolation thresholds on lattice connectivity. *Physical Review E*, 48(1):614, 1993.

[23] Grimmett Geoffrey. Percolation, 1999.

[24] Lung-Chi Chen and Fa Yueh Wu. Directed percolation in two dimensions: An exact solution. *arXiv preprint cond-mat/0511296*, 2005.

[25] Yanyan Zhuang, Jianping Pan, and Lin Cai. A probabilistic model for message propagation in two-dimensional vehicular ad-hoc networks. In *VANET*, pages 31–40. ACM, 2010.

[26] Lei Zhang, Lin Cai, and Jianping Pan. Connectivity in two-dimensional lattice networks. In *INFOCOM*, pages 2814–2822. IEEE, 2013.

[27] Lei Zhang, Lin Cai, Jianping Pan, and Fei Tong. A new approach to the directed connectivity in two-dimensional lattice networks. *Mobile Computing, IEEE Transactions on*, 13(11):2458 – 2472, 2014.

[28] Minming Ni, Lei Zhang, Jianping Pan, Lin Cai, Humphrey Rutagemwa, Li Li, and Tianming Wei. Connectivity in mobile tactical networks. In *Globecom*, pages 4575–4580. IEEE, 2014.

[29] Ning Lu, Tom H Luan, Miao Wang, Xuemin Shen, and Fan Bai. Capacity and delay analysis for social-proximity urban vehicular networks. In *INFOCOM*, pages 1476–1484. IEEE, 2012.

[30] Fraser Cadger, Kevin Curran, Jose Santos, and Sandra Moffett. A survey of geographical routing in wireless ad-hoc networks. *Communications Surveys & Tutorials, IEEE*, 15(2):621–653, 2013.

[31] Zhou Zhi and Yow Kin Choong. Anonymizing geographic ad-hoc routing for preserving location privacy. In *ICDCS Workshops*, pages 646–651. IEEE, 2005.

[32] Sk Md Mizanur Rahman, Masahiro Mambo, Atsuo Inomata, and Eiji Okamoto. An anonymous on-demand position-based routing in mobile ad-hoc networks. In *SAINT*, pages 300–306. IEEE, 2006.

[33] Aniket Kate, Gregory M Zaverucha, and Urs Hengartner. Anonymity and security in delay tolerant networks. In *SecureComm workshop*, pages 504–513. IEEE, 2007.

[34] Linke Guo, Chi Zhang, Hao Yue, and Yuguang Fang. A privacy-preserving social-assisted mobile content dissemination scheme in DTNs. In *INFOCOM*, pages 2301–2309. IEEE, 2013.

[35] Lei Zhang, Jun Song, and Jianping Pan. Towards privacy-preserving and secure opportunistic routings in VANETs. In *SECON*, pages 627–635. IEEE, 2014.

[36] Andrew C Yao. Protocols for secure computations. In *FOCS*, pages 160–164. IEEE, 1982.

[37] I Cameron, Jeffrey R Kenworthy, and Tom J Lyons. Understanding and predicting private motorised urban mobility. *Transportation research part D: Transport and environment*, 8(4):267–283, 2003.

[38] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *INFOCOM*, volume 6, pages 1–13. IEEE, 2006.

[39] Gregory G Finn. Routing and addressing problems in large metropolitan-scale internetworks. ISI research report. 1987.

[40] Ivan Stojmenovic, Anand Prakash Ruhil, and DK Lobiyal. Voronoi diagram and convex hull based geocasting and routing in wireless networks. *Wireless communications and mobile computing*, 6(2):247–258, 2006.

[41] Ram Shringar Raw and Sanjoy Das. Performance analysis of P-GEDIR protocol for vehicular ad-hoc network in urban traffic environments. *Wireless personal communications*, 68(1):65–78, 2013.

[42] C Li, C Zhao, L Zhu, H Lin, and J Li. Geographic routing protocol for vehicular ad-hoc networks in city scenarios: a proposal and analysis. *International Journal of Communication Systems*, 27(12):4126–4143, 2014.

[43] Sotirios Tsiachris, Georgios Koltsidas, and Fotini-Niovi Pavlidou. Junction-based geographic routing algorithm for vehicular ad-hoc networks. *Wireless personal communications*, 71(2):955–973, 2013.

[44] Ilias Leontiadis and Cecilia Mascolo. GeOpps: Geographical opportunistic routing for vehicular networks. In *WoWMoM*, pages 1–6. IEEE, 2007.

[45] Pei-Chun Cheng, Kevin C Lee, Mario Gerla, and Jérôme Härri. GeoDTN + Nav: Geographic DTN routing with navigator prediction for urban vehicular environments. *Mobile Networks and Applications*, 15(1):61–82, 2010.

[46] Qing Yang, Alvin Lim, Shuang Li, Jian Fang, and Prathima Agrawal. ACAR: Adaptive connectivity aware routing for vehicular ad-hoc networks in city scenarios. *Mobile Networks and Applications*, 15(1):36–60, 2010.

[47] Omprakash Kaiwartya, Sushil Kumar, and Reena Kasana. Traffic light based time stable geocast (T-TSG) routing for urban VANETs. In *IC3*, pages 113–117. IEEE, 2013.

[48] Omprakash Kaiwartya, Sushil Kumar, DK Lobiyal, Abdul Hanan Abdullah, and Ahmed Nazar Hassan. Performance improvement in geographic routing for vehicular ad-hoc networks. *Sensors*, 14(12):22342–22371, 2014.

[49] Carolina Tripp-Barba, Luis Urquiza-Aguiar, Mónica Aguilar Igartua, David Rebollo-Monedero, Luis J de la Cruz Llopis, Ahmad Mohamad Mezher, and José Alfonso Aguilar-Calderón. A multimetric, map-aware routing protocol for VANETs in urban areas. *Sensors*, 14(2):2199–2224, 2014.

[50] Si-Ho Cha, Keun-Wang Lee, and Hyun-Seob Cho. Grid-based predictive geographical routing for inter-vehicle communication in urban areas. *International Journal of Distributed Sensor Networks*, 2012.

[51] Marco Di Felice, Luca Bedogni, and Luciano Bononi. Group communication on highways: An evaluation study of geocast protocols and applications. *Ad Hoc Networks*, 11(3):818–832, 2013.

[52] Yuh-Shyan Chen and Yun-Wei Lin. A mobicast routing protocol with carry-and-forward in vehicular ad hoc networks. *International Journal of Communication Systems*, 27(10):1416–1440, 2014.

[53] Smitha Shivshankar and Abbas Jamalipour. Spatio-temporal multicast grouping for content-based routing in vehicular networks: A distributed approach. *Journal of Network and Computer Applications*, 39:93–103, 2014.

[54] Amin Vahdat, David Becker, et al. Epidemic routing for partially connected ad-hoc networks. Technical Report CS-200006, Duke University, 2000.

[55] Matthias Grossglauser and David Tse. Mobility increases the capacity of ad-hoc wireless networks. In *INFOCOM*, volume 3, pages 1360–1369. IEEE, 2001.

[56] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *PerCom Workshops*, pages 79–85. IEEE, 2007.

[57] Hongzi Zhu, Shan Chang, Minglu Li, Kshirasagar Naik, and Sherman Shen. Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks. In *INFOCOM*, pages 2192–2200. IEEE, 2011.

[58] Hongzi Zhu, Mianxiong Dong, Shan Chang, Yanmin Zhu, Minglu Li, and Xuemin Shen. ZOOM: Scaling the mobility for fast opportunistic forwarding in vehicular networks. In *INFOCOM*, pages 2832–2840. IEEE, 2013.

[59] Hideya Ochiai and Hiroshi Esaki. Mobility entropy and message routing in community-structured delay tolerant networks. In *AINTEC*, pages 93–102. ACM, 2008.

[60] Baolin Sun, Chao Gui, Hua Chen, and Yue Zeng. An entropy-based stability QoS routing with priority scheduler in MANET using fuzzy controllers. In *Fuzzy Systems and Knowledge Discovery*, pages 735–738. Springer, 2006.

[61] Jérémie Leguay, Timur Friedman, and Vania Conan. DTN routing in a mobility pattern space. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 276–283. ACM, 2005.

[62] Radhika Ranjan Roy. *Handbook of mobile ad-hoc networks for mobility models*. Springer Science & Business Media, 2010.

[63] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic routing in intermittently connected networks. In *SAPIR*, pages 239–254. Springer, 2004.

[64] Robert J Hall. An improved geocast for mobile ad-hoc networks. *Mobile Computing, IEEE Transactions on*, 10(2):254–266, 2011.

[65] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *SIMUTools*, page 55. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[66] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. Enhancing IEEE 802.11 p/WAVE to provide infotainment applications in VANETs. *Ad Hoc Networks*, 10(2):253–269, 2012.

[67] Hannes Hartenstein and Kenneth P Laberteaux. A tutorial survey on vehicular ad-hoc networks. *Communications Magazine, IEEE*, 46(6):164–171, 2008.

[68] Lung-Chi Chen. Asymptotic behavior for a version of directed percolation on a square lattice. *Physica A: Statistical Mechanics and its Applications*, 390(3):419–426, 2011.

[69] Haiyan Cai, Xiaohua Jia, and Mo Sha. Critical sensor density for partial connectivity in large area wireless sensor networks. *ACM Transactions on Sensor Networks*, 7(4):35, 2011.

[70] Satish Ukkusuri and Lili Du. Geometric connectivity of vehicular ad-hoc networks: Analytical characterization. *Transportation Research Part C: Emerging Technologies*, 16(5):615–634, 2008.

[71] Shigeo Shioda, Junko Harada, Yuta Watanabe, Tomoaki Goi, Hiraku Okada, and Kenichi Mase. Fundamental characteristics of connectivity in vehicular ad-hoc networks. In *PIMRC*, pages 1–6. IEEE, 2008.

[72] Jie Gao and Leonidas Guibas. Geometric algorithms for sensor networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):27–51, 2012.

[73] Peng-Jun Wan, Khaled M Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad-hoc networks. In *INFOCOM*, volume 3, pages 1597–1604. IEEE, 2002.

[74] Yanyan Zhuang and Jianping Pan. Random distances associated with equilateral triangles. *arXiv preprint arXiv:1207.1511*, 2012.

[75] Yanyan Zhuang and Jianping Pan. Random distances associated with rhombuses. *arXiv preprint arXiv:1106.1257*, 2011.

[76] Yanyan Zhuang and Jianping Pan. Random distances associated with hexagons. *arXiv preprint arXiv:1106.2200*, 2011.

[77] Martin Haenggi, Jeffrey G Andrews, François Baccelli, Olivier Dousse, and Massimo Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *Selected Areas in Communications, IEEE Journal on*, 27(7):1029–1046, 2009.

[78] Ingmar Glauche, Wolfram Krause, Rudolf Sollacher, and Martin Greiner. Continuum percolation of wireless ad-hoc communication networks. *Physica A: Statistical Mechanics and its Applications*, 325(3):577–600, 2003.

[79] Paolo Santi. Topology control in wireless ad-hoc and sensor networks. *ACM computing surveys (CSUR)*, 37(2):164–194, 2005.

[80] Dianjie Lu, Xiaoxia Huang, Pan Li, and Jianping Fan. Connectivity of large-scale cognitive radio ad-hoc networks. In *INFOCOM*, pages 1260–1268. IEEE, 2012.

[81] Nawaporn Wisitpongphan, Fan Bai, Priyantha Mudalige, Varsha Sadekar, and Ozan Tonguz. Routing in sparse vehicular ad-hoc wireless networks. *Selected Areas in Communications, IEEE Journal on*, 25(8):1538–1556, 2007.

[82] Eytan Domany and Wolfgang Kinzel. Directed percolation in two dimensions: numerical analysis and an exact solution. *Physical Review Letters*, 47(1):5, 1981.

[83] Lei Zhang. Square lattice network directed connectivity calculator. `http://grp.pan.uvic.ca/~leiz/lattice_poly.html`. Online; accessed 19-July-2014.

[84] Panagiotis Papadimitratos, Levente Buttyan, Tamás Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and J-P Hubaux. Secure vehicular communication systems: design and architecture. *Communications Magazine, IEEE*, 46(11):100–109, 2008.

[85] Rabin Patra, Sonesh Surana, and Sergiu Nedevschi. Hierarchical identity based cryptography for end-to-end security in DTNs. In *ICCP*, pages 223–230. IEEE, 2008.

[86] Panagiotis Papadimitratos, Virgil Gligor, and Jean-Pierre Hubaux. Securing vehicular communications - assumptions, requirements, and principles. In *Workshop on Embedded Security in Cars (ESCAR)*, volume 2006, 2006.

[87] Rongxing Lu, Xiaodong Lin, and Xuemin Shen. SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In *INFOCOM*, pages 1–9. IEEE, 2010.

[88] Zhengyi Le, Gauri Vakde, and Matthew Wright. PEON: privacy-enhanced opportunistic networks with applications in assistive environments. In *PETRA*, page 76. ACM, 2009.

[89] Cong Shi, Xiapu Luo, Patrick Traynor, Mostafa H Ammar, and Ellen W Zegura. Arden: Anonymous networking in delay tolerant networks. *Ad Hoc Networks*, 10(6):918–930, 2012.

[90] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, 1999.

[91] Sameh Zakhary and Milena Radenkovic. Utilizing social links for location privacy in opportunistic delay-tolerant networks. In *ICC*, pages 1059–1063. IEEE, 2012.

[92] Sameh Zakhary, Milena Radenkovic, and Abderrahim Benslimane. The quest for location-privacy in opportunistic mobile social networks. In *IWCMC*, pages 667–673. IEEE, 2013.

[93] Sameh Zakhary, Milena Radenkovic, and Abderrahim Benslimane. Efficient location privacy-aware forwarding in opportunistic mobile networks. *Vehicular Technology, IEEE Transactions on*, 63(2):893–906, 2014.

[94] Omar Hasan, Jingwei Miao, Sonia Ben Mokhtar, and Lionel Brunie. A privacy preserving prediction-based routing protocol for mobile delay-tolerant networks. In *AINA*, pages 546–553. IEEE, 2013.

[95] Feng Li, Jie Wu, and Anand Srinivasan. Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets. In *INFOCOM*, pages 2428–2436. IEEE, 2009.

[96] M Chuah, Peng Yang, and Jianbin Han. A ferry-based intrusion detection scheme for sparsely connected ad hoc networks. In *MobiQuitous*, pages 1–8. IEEE, 2007.

[97] Md Yusuf Sarwar Uddin, Ahmed Khurshid, Hee Dong Jung, Carl Gunter, Matthew Caesar, and Tarek Abdelzaher. Making DTNs robust against spoofing attacks with localized countermeasures. In *SECON*, pages 332–340. IEEE, 2011.

[98] Aysha Al-Hinai, Haibo Zhang, Yawen Chen, and Yidong Li. TB-SnW: Trust-based Spray-and-Wait routing for delay-tolerant networks. *The Journal of Supercomputing*, 69(2):593–609, 2014.

[99] Sameh Zakhary and Milena Radenkovic. Erasure coding with replication to defend against malicious attacks in DTN. In *WiMob*, pages 357–364. IEEE, 2011.

[100] Sacha Trifunovic, Maciej Kurant, Karin Anna Hummel, and Franck Legendre. Preventing spam in opportunistic networks. *Computer Communications*, 41:31–42, 2014.

[101] Sacha Trifunovic, Franck Legendre, and Carlos Anastasiades. Social trust in opportunistic networks. In *INFOCOM Workshops*, pages 1–6. IEEE, 2010.

[102] Liqun Chen, Siaw-Lynn Ng, and Guilin Wang. Threshold anonymous announcement in VANETs. *Selected Areas in Communications, IEEE Journal on*, 29(3):605–615, 2011.

[103] David Chaum and Eugène Van Heyst. Group signatures. In *Advances in Cryptology–EUROCRYPT 1991*, pages 257–265. Springer, 1991.

[104] Ernie Brickell, Liqun Chen, and Jiangtao Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. *International Journal of Information Security*, 8(5):315–330, 2009.

[105] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *CCS*, pages 132–145. ACM, 2004.

[106] Hsiao-Ying Lin and Wen-Guey Tzeng. An efficient solution to the millionaires problem based on homomorphic encryption. In *ACNS*, pages 456–466. Springer, 2005.

[107] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *SCIS*, pages 135–148. IEICE, 2000.

[108] Régis Dupont and Andreas Enge. Provably secure non-interactive key distribution based on pairings. *Discrete Applied Mathematics*, 154(2):270–276, 2006.

[109] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology–CRYPTO 2004*, pages 56–72. Springer, 2004.

[110] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology–CRYPTO 2001*, pages 213–229. Springer, 2001.

[111] Michael Scott. Implementing cryptographic pairings. *Lecture Notes in Computer Science*, 4575:177, 2007.

[112] Panos Papadimitratos, A La Fortelle, Knut Evenssen, Roberto Brignolo, and Stefano Cosenza. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95, 2009.