Adaptive Connectivity Aware Routing Protocol for Wireless Vehicular Networks

by

Qing Yang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 9, 2011

Keywords: Network Connectivity Model, Vehicular Networks, Vehicle to Vehicle Communications, Connectivity-Aware Routing, Location Privacy Protection

Approved by

Alvin Lim, Chair, Associate Professor, Computer Science and Software Engineering
David Umphress, Associate Professor, Computer Science and Software Engineering
Xiao Qin, Associate Professor, Computer Science and Software Engineering
Wei-Shinn Ku, Assistant Professor, Computer Science and Software Engineering

Abstract

Multi-hop vehicle-to-vehicle communication is useful for supporting many vehicular applications that provide drivers with safety and convenience. Developing multi-hop communication in vehicular ad hoc networks (VANETs) is a challenging problem due to the rapidly changing topology and frequent network disconnections, which cause failure or inefficiency in traditional ad hoc routing protocols. We propose an adaptive connectivity aware routing (ACAR) protocol that addresses these problems by adaptively selecting an optimal route with the best network connectivity-quality (CQ) based on statistical and real-time density data that are gathered through an on-the-fly density collection process. The CQ metric models the joint probability that a network is connected and a packet is successfully delivered in this network. The protocol consists of two parts: 1) select an optimal route, consisting of road segments, with the best CQ, and 2) in each road segment of the chosen route, select the most efficient multi-hop path that will improve the delivery ratio and throughput. The optimal route is selected using our connectivity-quality metric that takes into account vehicles densities and traffic light periods to estimate the probability of network connectivity and data delivery ratio for transmitting packets. Our simulation results show that the proposed ACAR protocol outperforms existing VANETs routing protocols, e.g. the delivery ratio of ACAR is 19% higher than VADD, the second best protocol.

ACAR is built upon geographic routing which requires every vehicle to broadcast its location information to its neighbors, and this process will compromise user's location privacy. To address this issue, we proposed a dummy-based location privacy protection (DBLPP) protocol in VANETs. In DBLPP, routing decision is made based upon the dummy distance to destination (DOD), instead of user's true location. In this scheme, a user's true location and identification information are preserved, so the user's location privacy is protected. Simulation results show that the DBLPP provides similar network performances as other routing protocols, and achieves a higher level of

location privacy protection on vehicles in networks. This location privacy protection scheme can be easily added to other geographic routing protocols.

Acknowledgments

I am heartily thankful to my supervisor, Dr. Alvin S. Lim, for his academic guidance, generous advice, his sharp comments and support, during our many discussions. I am very appreciative of his encouragement and patience when things seemed fuzzy. Without his support, this thesis would not have been completed. It has been a privilege working with him.

I would like to thank to my committee members: Dr. David Umphress, Dr. Xiao Qin, Dr. Wei-shin Ku for their time, patience and suggestions that led to me improving this work. I owe my deepest gratitude to Dr. Kai Chang and Dr. Qin for supporting me attending conferences, to Dr. Umphress and Dr. Ku for helping me in my job hunting.

Special thanks to Dr. Prathima Agrawal for the knowledge I have gained through the Wireless Seminars and years of research we had started. I thank her for her continuous support since I came to Auburn University, her kindness, time, and collaboration. It is Dr. Prathima Agrawal who directed me to study the area of wireless vehicular networks at the time when I started my Ph.D program.

I am deeply indebted to my family, who has always supported me, and especially to my parents and my sister Liu Yang for their love, guidance and vision. I believe I have been blessed by the God for granting me such wonderful and supportive family.

Finally, special thanks to my beloved wife Tiantian Wang. It is her love and support that makes this dissertation possible.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

A-STAR	Anchor-based Street and Traffic Aware Routing
ACAR	Adaptive Connectivity Aware Routing
AODV	Ad hoc On Demand Distance Vector
ASTM	American Society for Testing And Materials
ASV	Advanced Safety Vehicle
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CAR	Connectivity Aware Routing
CBF	Contention Based Forwarding
CBF-AS	Contention Based Forwarding - Active Selection
CQ	Connectivity-Quality
CSM	Constant Speed Motion
CTF	Clear To Forward
CTS	Clear To Send
DBLPP	Dummy Based Location Privacy Protection
DOD	Distance To Destination
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communications
EDD	Expected Disconnection Degree
ETX	Expected Transmission Count
FCC	Federal Communications Commission
FER	Frame Error Rate

FTM	Fluid Traffic Motion
GeOpps	Geographical Opportunistic Routing
GPS	Global Positioning Systems
GPSR	Greedy Perimeter Stateless Routing
GSR	Geographic Source Routing
IDM	Intelligent Driver Model
IDM-IM	Intelligent Driver Model with Intersection Management
IDM-LC	Intelligent Driver Model with Lane Changing
LOS	Line Of Sight
MAC	Multiple Access Control
MANET	Mobile Ad hoc Networks
MDDV	Mobility-Centric Data Dissemination Algorithm for Vehicular Networks
MOVE	Motion Vector
MURU	Multi-hop Routing for Urban VANET
NLOS	Non Line Of Sight
NLP	Neighbor Location Prediction
OEM	Original Equipment Manufacturer
OFDM	Orthogonal Frequency-Division Multiplexing
OPERA	Opportunistic Packet Relaying Protocol
PATH	California Partners for Advanced Transit and Highways
PER	Packet Error Rate
PHY	Physical Layer
RREQ	Route Request
RTF	Request To Foward
RTS	Request To Send

SADV	Static Node Assisted Adaptive Routing
SINR	Signal to Interference plus Noise Ratio
SKVR	Scalable Knowledge Based Routing
TIGER	topologically Integrated Geographic Encoding and Referencing
TSIS-CORSIM	Traffic Software Integrated System - Corridor Simulation
V2I	Vehicle To Infrastructure
V2V	Vehicle To Vehicle
VADD	Vehicle Assisted Data Delivery
VANET	Vehicular Ad Hoc Networks
VanetMobiSim	VANET Mobility Simulator
WAVE	Wireless Access in Vehicular Environments
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Chapter 1

Introduction

Wireless communication among moving vehicles is increasingly the focus of research in both of the academic community and automobile industry, driven by the vision that exchange of information among vehicles can be exploited to improve the safety and comfort of drivers and passengers [1–4]. Some automobile manufacturers have equipped their new vehicles with global positioning systems (GPS), digital maps and even wireless interfaces, e.g. Honda-ASV3. In addition, the federal communications commission (FCC) has allocated 75MHz of spectrum in the 5.9GHz band for vehicle-vehicle and vehicle-roadside communication, called dedicated short range communications (DSRC). IEEE is also working on the IEEE 1609 family of standards for wireless access in vehicular environments (WAVE), which define an architecture and a complementary, standardized set of services and interfaces that collectively enable secure vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) wireless communications. Although IEEE 1609.3 considers the networking layer and provides an alternative for IPv6, it does not define the ad hoc routing protocol between vehicles, and has left this issue open.

Though several technical problems need to be solved before installing vehicular networks; in the near future, large scale vehicular networks will be available to provide people with more conveniences in their driving experience. For example, through such networks, people can query the price and services provided by gas stations in a certain region, or remotely control their smart-houses [5] while driving home. Drivers can even download a real-time traffic image from a traffic camera located at a certain point, or connect to access points of parking lots to inquire the number of available parking slots. These types of applications could tolerate some delay, e.g. a few minutes. If the information could be successfully retrieved from the remote server, it would be very helpful and desirable to drivers.

To realize this vision, we must first select the most appropriate architecture. There are three broad categories of network architectures: infrastructure-based, ad hoc networks and hybrid. The infrastructure-based architecture takes advantage of the roadside infrastructure or existing cellular networks. However, a big issue of such networking is the high operation cost. Moreover, the cellular networks have other drawbacks such as the limited bandwidth and symmetric channel allocation for up-link and down-link. Ad hoc networks do not need infrastructure, so the cost of building such network will be very low and it can even operate in the event of disasters. The hybrid architecture is more practical which combines these two architectures by considering vehicles as data relays between roadside base-stations [6, 7]. This architecture also requires the function of multi-hop communication between vehicles, which is the essential part of ad hoc network architecture. This work focuses on the vehicular ad hoc network (VANET) architecture with the flexible deployment and self-organizing capabilities.

Due to special characteristics of VANETs, traditional routing protocols in wireless ad hoc networks may not be suitable for vehicular communications. For example, DSR [8] and AODV [9] are not suitable for VANETs because of the large route maintenance overhead. Therefore, some variants of stateless geographic routing protocols, such as [10,11], may be better choices. However, even with geographic routing, many of the following challenges still need to be addressed:

1. Dynamic and rapidly changing topologies of vehicular networks can cause frequent communication disconnections among vehicles. As revealed in [12], the frequent network disconnection is the most important issue in designing protocols for VANETs.

2. Geographic forwarding protocols select the shortest route (minimal number of hops) that may suffer from a higher packet error rate due to the poor link quality of each hop.

3. The uneven distribution of vehicles on the roads makes route selection more complex, e.g. the shortest path in terms of geographic distance may experience more frequent network disconnections.

4. Some protocols [13,14] make use of the density information on roads to select routes but the inaccuracy of statistical data may cause routes to be incorrectly computed.

5. Because of obstacles to wireless signal by large objects, e.g. skyscrapers in cities, communication between vehicles must have line-of-sight.

To address these problems, we propose a new routing protocol called adaptive connectivity aware routing (ACAR). There are three main contributions in this work. First, based on the statistical information on the road, we propose a connectivity model that provides the probability of network connectivity on a road segment. This connectivity model also takes into account the phenomena that (red) traffic lights can block approaching vehicles and those nodes will move as a platoon in the next road segment. Second, we introduced a novel metric, connectivity-quality (CQ), which combines the network connectivity probability and data delivery ratio of packets being forwarded along a road segment. Third, as the statistical data may not be accurate, an on-the-fly information collection algorithm is developed to help ACAR adaptively select the best route.

Geographic routing provides superior scalability and thus is widely used in VANETs. However, it requires every vehicle to broadcast its location information to its neighboring nodes, and this process will compromise user's location privacy. To address this issue, we proposed a dummy-based location privacy protection (DBLPP) routing protocol, in which routing decision is made based upon the dummy distance to the destination (DOD), instead of users' true locations. In this scheme, users' true locations and identification information are preserved, so the user's location privacy is protected.

This dissertation is organized as follows: It presents the background and motivations for vehicular networks in Chapter 2 and proposes the problem statement and objectives to be achieved in Chapter 3. Chapter 4 reviews existing routing protocols for VANETs. In Chapter 5, the network connectivity in VANETs is fully investigated and an integrated connectivity model for a path that is composed of multiple road segments is proposed. Chapter 6 shows in detail how the proposed routing protocol are designed and implemented. The evaluation and analysis of the proposed protocol are explored in Chapter 7. To further investigate the location privacy protection in geographic routing, a dummy based location privacy preservation mechanism is proposed and evaluated in Chapter 8. The integration of ACAR and DBLPP is considered as our future work which is described in Chapter 9. Finally, Chapter 10 concludes this dissertation.

Chapter 2

Background and Motivations

2.1 Wireless Vehicular Ad Hoc Network

Wireless vehicular ad hoc network is a novel wireless network that emerges because of the advances in wireless technologies and the automotive industry. VANETs are spontaneously formed between moving vehicles equipped with wireless interfaces that could be of homogeneous or heterogeneous technologies.

VANETs are considered as one real-life application of mobile ad hoc network which enables communications among nearby vehicles as well as between vehicles and roadside infrastructures. Vehicles can be either private (e.g. individuals cars) or from public transportation (e.g., buses and police car).

The history of using radio and infrared communications between vehicles and infrastructures is strongly tied to the evolution of intelligent transportation systems. In 1939's World Fair, the use of communication and control techniques to make road traffic safe, efficient and environmentally friendly were first exhibited by the General Motors. From then, the interest in vehicle-to-vehicle communication continued in Japan, USA and Europe, but there were not many successful projects during this time period. Until the second half of the 1990s, many impressive projects on vehicular networks occurred because of the rapid development of wireless technology. For example, the California partners for advanced transit and highways (PATH) in 1997, the advanced safety vehicle (ASV) in 2000 in Japan, and the CarTalk and FleetNet projects investigated in Europe. The concept of VANET was dramatically impacted by the advances in wireless technology and standardization since the late 1990s.

The ”game changer” occurred when the US Federal Communication Commission allocated 75 MHz bandwidth of the 5.9 GHz band for vehicle-to-vehicle and vehicle-to-infrastructure wireless communication in 1999. The commission then established the service and license rules for DSRC Service which defines the communication service working on the 5.850-5.925 GHz band for the public safety and private applications in vehicular networks. In 2001, the ASTM International selected IEEE 802.11a as the underlying radio technology of DSRC. The pressure to make use of the assigned channels and the availability of the IEEE 802.11a technology and standard significantly increased research and development activities. In 2004, the IEEE started the work on the 802.11p amendment and wireless access in vehicular environments (WAVE) standards based on the ASTM standard. From 2004 until now, wireless communication among moving vehicles becomes the focus of researches in both of the academic research community and automobile industry.

2.2 Applications of VANETs

Vehicular network applications range from road safety oriented applications for vehicles or drivers, to entertainment and commercial applications for passengers, making use of a plethora of cooperating technologies.

The primary vision of vehicular networks includes real-time and safety applications for drivers and passengers, providing safety for the latter and giving essential tools to decide the best path along the way. These applications thus aim to minimize accidents and improve traffic conditions by providing drivers and passengers with useful information including collision warnings, road sign alarms, and in-place traffic view.

Nowadays, vehicular networks are promising in a number of useful driver- and passenger-oriented services, which include Internet connections facility exploiting an available infrastructure in an on-demand fashion, electronic tolling system, and a variety of multimedia services. Furthermore, a variety of communication networks, such as 2-3G, WLANs IEEE 802.11a/b/g, and WiMAX, can be exploited to enable new services designed for passengers apart from the safety applications, such as info-mobility and entertainment applications, which can rely on the vehicular network itself.

Regarding the discussed applications' potential, vehicular networks open new business opportunities for car manufacturers, automotive OEMs, network operators, service providers, and integrated operators in terms of infrastructure deployment as well as service provision and commercialization.

For safety-related applications, the network operator can assure the authentication of each participant through playing the role of a trusted third party that authenticates the participating nodes, or even having the role of a certification authority issuing a certificate to each participant in order to prove their authenticity later during the communication.

On the other hand, in non-safety related applications, network operators and/or service providers, besides network access and service provision, can have the role of authorizing service access and billing users for the consumed services. However, one should notice that ad hoc systems still require a certain level of penetration and necessitate high vehicle density for more reliable communication.

The investment cost for new communication infrastructure for vehicular networks will be relatively high. On the other hand, cellular communication systems offer a high coverage along roads and have a reliable authentication and security mechanism. Consequently, number of technical challenges needs to be resolved in order to help the evolution of vehicular networks for wide-scale deployment.

2.3 Characteristics of VANETs

Vehicular ad hoc networks share many common characteristics with general mobile ad hoc networks (MANET). Both VANET and MANET are self-organizing wireless ad hoc networks that are composed of mobile nodes. However, they are different in several ways. For example, vehicles can recharge their batteries frequently which usually can last much longer than batteries in regular mobile devices. However, vehicles' movements can be constrained by the road topology and traffic rules. In MANET, nodes cannot recharge their power and the energy efficiency is also critical in such networks. In addition, the nodes' movements in MANET are assumed to follow unrestricted patterns of movements. In comparison to other communication networks, vehicular networks come with unique beneficial features, as follows:

1. Unlimited transmission power: The power issue of mobile devices is usually not a significant one in VANET as that in the classical ad hoc or sensor networks. Vehicle itself can provide continuous power to computing and communication devices. Usually, the car battery can last much longer compared to those for hand-held mobile devices.

2. Higher computational capability: Vehicles can be installed with significant computing, communication, and sensing capabilities which can be even more powerful than regular desktops.

3. Predictable mobility: Unlike conventional mobile ad hoc networks in which node mobility is hard to predict, vehicles in VANETs tend to move in a predictable way that is (usually) limited to street topology. Roadway information is often available from navigation systems and map-based technologies such as GPS. Given the average number of nodes, average speed, number of lanes, the future position of a vehicle may be predicted.

However, to bring its potency to fruition, vehicular networks have to cope with some challenging characteristics including:

1. Potentially large scale: Unlike most ad hoc networks studied in the literature that usually assume a limited network size, vehicular networks can in principle extend over the entire road network and so include many participants.

2. High mobility: The environment in which vehicular networks operate is extremely dynamic, and includes extreme configurations: on highways, relative speeds of up to 300 km/h may occur, while density of nodes may be 1-2 vehicles per kilometers on less busy roads. On the other hand, in the city, relative speeds can reach up to 60 km/h and network density can be very high, especially during rush hours.

3. Partitioned network: Vehicular networks will be frequently partitioned. The dynamic nature of traffic may result in large inter-vehicle gaps in sparsely populated scenarios, and hence in several isolated clusters of nodes.

4. Network topology and connectivity: Vehicular network scenarios are very different from classic ad hoc networks. Since vehicles are moving and changing their position constantly, scenarios are very dynamic. Therefore the network topology changes frequently as the links between nodes connect and disconnect very often. Indeed, the degree to which the network is connected is highly dependent on two factors: the range of wireless links and the fraction of participant vehicles, where only a fraction of vehicles on the road could be equipped with wireless interfaces.

2.4 Technical Challenges in VANET

VANET has special behavior and characteristics, so there are several challenges for vehicular communication which greatly impact the future deployments of such networks. Many challenges need to be resolved in order to deploy vehicular networks in our real lives such as information dissemination, security and privacy, and Internet integration. Generally speaking, efficient wireless communication is an important issue, so the employed protocols and mechanisms should be robust, reliable and scalable to numerous vehicles.

VANET differs from conventional ad hoc networks by not only experiencing rapid changes in wireless links, but also having to deal with different network densities. For instance, vehicular networks in urban areas are more likely to form highly dense networks during rush hour traffic. However, vehicular networks are expected to experience frequent network disconnections in sparsely populated rural highways or during late night hours.

Moreover, VANET is expected to handle a wide range of applications ranging from safety to leisure. Consequently, routing algorithms should be efficient and cope to vehicular network characteristics and applications. Until now, most of research has focused on analyzing routing algorithms in highly dense networks with the assumption that a typical vehicular network is well-connected in nature. Actually, the penetration of vehicles with wireless communication capacity is somewhat weak. Therefore, a VANET should rely on existing infrastructure supports for wide-scale deployment. However, VANET are expected in the future to observe high penetration with lesser infrastructures, and hence it is important to consider the disconnected network problem. Network disconnection in VANET is a crucial research challenge for developing a reliable and efficient routing protocol.

Chapter 3

Problem Statement and Assumptions

Vehicular network is a dynamic mobile network in which network disconnection occurs very often, which causes most traditional routing protocols to fail in delivering packets. To address this issue, buffer is used on each vehicle so that packets can be stored when network disconnection occurs and delivered if the network re-connects. However, one drawback of using buffer is that it will cause huge network delay. Therefore, how to reduce such delay and achieve and higher network throughput is a non-trivial issue.

3.1 Problem Statement of Routing in VANETs

A vehicular ad hoc network can be modeled as a graph $G = \{V, E\}$, where V is the set of vehicles (nodes) and E is the set of edges that represent wireless links. The set E will change from time to time due to vehicles' movements. According to [15], we can define a set of time intervals Γ_E, in which intervals are numbered as $I_1, \cdots, I_q, \cdots, I_h$. Further, by construction $I_q = [t_{q-1}, t_q)$ (and $t_{q-1} < t_q$), we can define a set of time sequence. Therefore, the set Γ_E partitions the interval $[t_0, t_h)$ into h pieces. For an interval $I = [a, b)$ and $r \in R^+$, we let $I \oplus r$ denote the (shifted) interval $[a + r, b + r)$. Then, we further define a few other variables.

- $c : E \times R^+ \rightarrow R^+$, where $c_{e,t}$ is the capacity of the edge $e \in E$ at time t
- $d : E \times R^+ \rightarrow R^+$, where $d_{e,t}$ is the transmission delay of the edge e at time t
- b_v is the storage capacity of the node $v \in V$
- I^v is the set of edges whose destination node is v (incoming edges)
- O^v is the set of edges whose source node is v (outgoing edges)

Because a buffer is needed to store packet when network disconnection occurs in VANETs, we need to define the buffer capacity b_v. We also define K as a set of messages injected into the network from a source to a destination. For a certain message $k \in K$, it can be denoted as a tuple (u, v, t) where (u, v) denotes the current and next-hop nodes, t is the time instance when the message is sent from u to v. The functions $s(K)$, $d(K)$, $\omega(K)$, $m(K)$ are used to retrieve the source node, the destination node, the start time, and the number of messages in K, respectively. Moreover, the following definitions capture the states and transitions in the network.

- $N^K_{v,t}$ is the number of messages (in K) occupying the buffer at node v at time $t \in \Gamma_E$
- $X^K_{e,I}$ is the number of messages transmitted (at the tail of the edge) over edge e during $I \in \Gamma_E$
- $R^K_{e,I}$ is the number of messages received (at the destination of the edge) over edge e during $I \in \Gamma_E$
- $K^v = \{k | k \in K, d(k) = v\}$ is the set of messages whose destination node is v

The transmission variables (denoted by X) and the reception variables (denoted by R) are used together to model the transmission delay encountered in sending messages. The natural objective is to maximize the probability of message delivery. However, since a buffer is used to store messages while network disconnection occurs, the messages will eventually be delivered unless the buffer overflows. Moreover, because of packets being buffered instead of transmitted immediately, network delay is a big issue in VANETs. In this work, we focus on minimizing the delay of a message. So the objective function is to minimize the average delay, which can be realized by minimizing the sum of the delays for all messages.

$$\min \sum_{v \in V} \sum_{K \in K^v} \sum_{I_q \in \Gamma_E} (t_{q-1} - \omega(K)) \cdot (\sum_{e \in I^v} R^K_{e,I_q} - \sum_{e \in O^v} X^K_{e,I_q}) \tag{3.1}$$

The summation $\sum_{e \in I^v} R^K_{e,I_q}$ represents the number of messages in K that are coming into the node v in the interval I_q. These messages could be forwarded from other nodes or generated by the current node which is the source of the messages During the time interval I_q, a portion of messages may be transmitted to other nodes. This is accounted for by subtracting the term $\sum\limits_{e \in O^v} X^K_{e,I_q}$. The above difference is then multiplied by the delay of these messages (i.e. $t_{q-1} - \omega(K)$) to get the total delay suffered by that fraction of messages that arrived in the interval I_q at node v. Finally, we sum over all the possible intervals for all messages and all nodes. The objective function should be subject to:

$$\sum_{e \in I^v} R^K_{e,I_q} - \sum_{e \in O^v} X^K_{e,I_q} = \begin{cases} N^K_{v,t_q} - N^K_{v,t_{q-1}} + m(K) \ if \ s(K) = v, \omega(K) = t_q \\ N^K_{v,t_q} - N^K_{v,t_{q-1}} \quad otherwise \end{cases} \tag{3.2}$$

$$R^K_{e,I_q \oplus d_{e,t_{q-1}}} = X^K_{e,I_q} \tag{3.3}$$

$$N^K_{v,t_{q-1}} \leq b_v \tag{3.4}$$

$$X^K_{e,I_q} \leq c_{e,t_{q-1}} \cdot |I_q| \tag{3.5}$$

$$N^K_{v,t_0} = \begin{cases} m(K) \ if \ v = s(K), t_0 = \omega(K) \\ 0 \quad otherwise \end{cases} \tag{3.6}$$

$$N^K_{v,t_h} = \begin{cases} m(K) \ if \ v = d(K) \\ 0 \quad otherwise \end{cases} \tag{3.7}$$

Equation 3.2 gives the flow constraints, i.e. the number of messages outgoing from node v plus those staying in the buffer should be equal to the number of messages incoming to node v. During a certain time period, if node v generates a new message, this message needs to be added to the buffer too.

Equation 3.3 relates the variables X and R by stating that the traffic transmitted at the initial point of e during I_q is equal to the traffic received at the end point of e during the time interval $I_q \oplus d_{e,t_{q-1}}$ (i.e. after the edge delay).

Constraints are also needed to ensure that the number of messages that are stored at any node's buffer does not exceed the specified limit, and the data sent over a link is limited by the edge capacity over that time interval. These are captured by Equation 3.4 and Equation 3.5.

Finally, Equations 3.6 and 3.7 are the initial and the final conditions regarding the storage. Equation 3.6 says that in the beginning, only nodes that have messages to send have occupied buffers. Equation 3.7 states that at the end, only nodes that are destinations for messages have occupied buffers.

We have modeled the routing problem in VANETs as a linear programming problem which can be solved mathematically given the value of all defined variables. However, it is impossible to obtain real-time global information of a VANET such as vehicles' positions, network links, and link capacities. Therefore, an approximate routing protocol is required. Therefore, how to design a routing protocol to maximize the probability of message delivery and minimize the network delay is the major concern of this work.

3.2 Issues of Existing Solutions

The topology of VANETs has a unique characteristic – it consists of one or more sub-graphs (one sub-graph if the network is fully connected) of the road map topology. Previous researches in wireless ad hoc networks often make an unrealistic assumption on nodes mobility. For example, with the most popular Random Way Point model, nodes can freely move within a certain area with randomly chosen velocities. However, nodes in VANETs do not have the ability to roam freely without regards to obstacles and traffic regulations, i.e. road segments containing vehicles construct the network topology of a VANET.

Therefore, the problem of efficient routing of packets in VANETs can be transformed into selecting a route with the highest network throughput from the road map. A critical reason causing

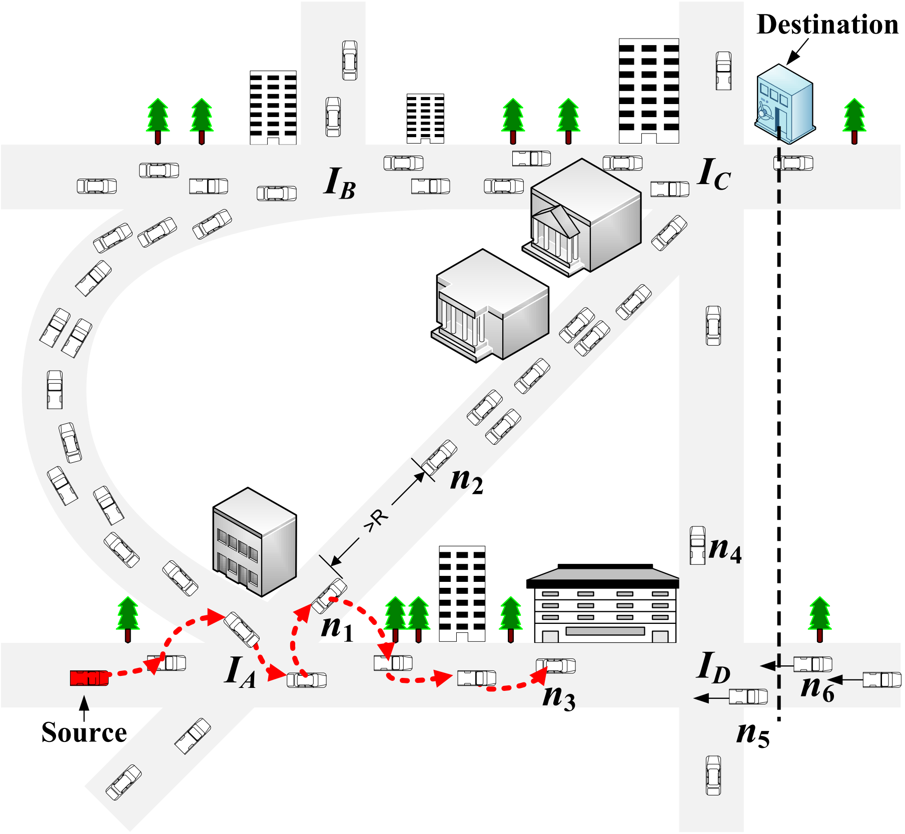


Figure 3.1: Illustration of the Issues in Existing Routing Protocols in VANETs

low network throughput is the network disconnection which occurs extremely often in VANETs. When network partition occurs, most existing routing protocols for vehicular networks will drop packets such as GPSR [10] and GSR [11]. Although VADD [13] uses carry-and-forward scheme to buffer packets if network is disconnected, the network connectivity information is not fully investigated. We note that network connectivity is the most important information for routing in VANETs, and then propose a connectivity aware routing protocol for VANETs.

Consider the network situation shown in Fig. 3.1, where the source node at the bottom left corner is trying to send packets to the destination at the top right corner. In this figure, the lengths of road segment $I_AI_BI_C$, I_AI_C, I_AI_D and I_CI_D are 1200m, 1000m, 707m and 707m, respectively. The numbers of nodes deployed on each above-mentioned road segment are 22, 9, 5 and 2, respectively. All vehicles move with the average velocity of 10m/s.

With the GPSR protocol, packets will be forwarded through a multi-hop route. An example route is depicted as dashed lines with arrows in this figure. Because the network density on road segment I_AI_C is very low, disconnections may occur frequently. For example, node n_1 in Fig. 3.1 fails to communicate with node n_2 as they are out of communication range. In this case, GPSR enters the perimeter mode and selects nodes on road segment I_AI_D to forward packets. However, since network partitions are very common in VANETs, GPSR may face other network disconnections again. For instance, because wireless signal may be blocked by objects, e.g. skyscraper in the city, the communication between node n_3 and n_4 may be impossible due to the absence of line-of-sight. That implies the GPSR protocol may take many detours to find connected route, e.g. on segment $I_AI_BI_C$, after many perimeter mode searches. If there is no such connected route in networks, GPSR may search through the entire networks and finally fail to find a route.

To make use of road map information, the geographic source routing (GSR) protocol [11] was proposed for VANETs. With this approach, road segment I_AI_C will be selected to forward the packets. Because the assumption of connected networks does not always hold, the GSR may fail to deliver packets when network partitions occur. If the carry-and-forward scheme [16] is added into GSR, packets can finally reach the destination. However, the delay of forwarding packets on this road segment will be higher than routing packets along $I_AI_BI_C$. According to measurements in our simulations, the network connectivity probabilities of road I_AI_C and $I_AI_BI_C$ are $.29$ and $.84$, respectively. The $.29$ connectivity probability can be interpreted as the network is disconnected 71% of the time, so the network delay can be simply calculated as $.71 \times (1000/v) + .29 \times (1000/c)$ where v is the average velocity of vehicles on road I_AI_C and c is the wireless transmission speed. As $v \ll c$, the delay of forwarding packets along I_AI_C is $delay_{AC} \approx (710/v)$. Similarly, the delay of forwarding packets on $I_AI_BI_C$ is $.16 \times (1200/v) + .84 \times (1200/c) \approx (192/v)$. Therefore, routing packets along $I_AI_BI_C$ generates a much smaller delay than that of I_AI_C.

In the motion vector (MOVE) [17] protocol, the packet carrier will select the next hop that is currently or will be closest to the destination; otherwise, it will carry (buffer) the packet until a next hop is available. It provides nine rules for current node to select the next hop, and one of them states that if the current packet carrier is in AWAY state and one neighbor is in TOWARDS state, packets must be forwarded to this neighbor. For instance, as shown in Fig. 3.1, node n_5 (moving away from the destination) will forward packets to n_6 as it move toward the destination. However, if n_6 moves over the vertical dashed line, it enters the AWAY state and will forward packets back to following vehicles that are in the TOWARDS state. This situation is so-called Ping-Pong effect which will not occur if no more following vehicles become available. However, this problem becomes worse when the network density is higher.

To select a route with the minimal transmission delay, the vehicle-assisted data delivery (VADD) protocol is proposed for VANETs [13]. According to the protocol, since the network density on road I_AI_C is equal to $1/R$, the delay of forwarding packets on I_AI_C is $d_{AC} = \alpha \cdot l_{AC}$ where $l_{AC} = 1000$m is the length of road I_AI_C and α is a constant. Similarly, $d_{AB} = \alpha \cdot 1000$, so we have $d_{AB} = d_{AC}$. As stated in VADD, if the packet carrier (vehicle) at intersection I_A chooses to deliver packets on road I_AI_B, the expected packet delivery delay from the intersection I_A to the destination is:

$$D_{AB} = \frac{1}{1 - P_{AB} \cdot P_{BA}} \cdot (d_{AB} + P_{BA} \cdot d_{BA} + P_{BA} \cdot P_{AC} \cdot d_{AC} + P_{BC} \cdot d_{BC}) \qquad (3.8)$$

where P_{AB} is the probability that the packet is forwarded through I_AI_B at intersection I_A, which is smaller than 1. Since $P_{AB} \cdot P_{BA} < 1$, we have $D_{AB} > (d_{AB} + P_{BA} \cdot d_{BA} + P_{BA} \cdot P_{AC} \cdot d_{AC} + P_{BC} \cdot d_{BC})$, so $D_{AB} > d_{AB}$. On the other hand, since $D_{AC} = d_{AC} = d_{AB}$, we obtain $D_{AB} > D_{AC}$. Therefore, road I_AI_C will be chosen by VADD to forward packets as it has the smallest expected delivery delay. However, the delay of sending packets along road $I_AI_BI_C$ is actually the lowest.

In summary, to select the best route in VANETs, a proper model of the network connectivity is very important and it is determined by several factors such as network density, road length and number of lanes on roads. For a certain road segment, its network connectivity will be affected by many factors including network density, road segment length, average vehicle velocity, number of lanes and traffic light periods. Even if the probability of network connectivity of a road segment is modeled, the network connectivity of a route that consists of several road segments is still a challenging problem. We cannot simply use the product of the probabilities of all road segments on the route because these probabilities are not independent of each others. In this work, we first model the network connectivity and then propose an approach to select the optimal route that can achieve the highest network throughput.

3.3 Assumptions

As GPS and navigation systems are becoming standard equipment in vehicles, we assume every vehicle obtains its current location. We also assume vehicles are installed with a pre-loaded digital map, such as the commercial map provided by MapMechanics, which not only describes the land attribute such as road topology and traffic light period but also is accompanied by traffic statistics such as traffic density and average velocity at a certain time of the day. These digital maps with statistical data are derived from billions of GPS sampled points from vehicles on the move. Similar digital maps can also be found from the Internet, e.g. yahoo.com. We expect more accurate and detailed digital maps to be invented and equipped on vehicles in the future. We also assume the vehicles are of similar sizes and each vehicle is equipped with an 802.11 wireless interface.

Chapter 4

Related Work

There exist several routing protocols that can be applied to vehicular ad hoc networks as summarized in [18–20]. They can be grouped into two categories: 1) those that assume the networks are always connected and 2) those that focus on intermittently connected networks.

4.1 Routing Protocols in Connected Networks

Protocols in the first category are suitable for the urban rush hour scenarios, where vehicles are densely packed and locating a node for forwarding a message is typically not an issue. However, traditional ad hoc routing protocols (e.g., AODV [9] and DSR [8]) provide poor route convergence and low communication throughput because they are adversely affected by the highly dynamic nature of node mobility as shown by the results in [21].

Since GPS devices will be standard components in future vehicles, more position-based routing protocols have been proposed for VANETs [10, 11, 22–26]. Position-based approaches use geographic coordinates information or relative positions of nodes to generate an efficient route through the network. For example, the greedy perimeter stateless routing (GPSR) [10] protocol may be a good choice because it is stateless and performs well despite high mobility in VANETs. However, GPSR may encounter the problems of selecting incorrect next hops due to out-of-date neighbor's information, routing loop and too many (detour) hops as stated in [11]. In [11], packets are forwarded along the *Dijkstra* shortest path as calculated from road maps.

Similarly, in MDDV [24], the forwarding trajectory of a message is determined as the trajectory that minimizes the sum of weights on that graph between the source and a vertex in the destination region. Moreover, the authors [25] developed protocols that disseminate information to a set of target zones, rather than specific destination nodes. They utilize a propagation function

Table 4.1: Summary of Unicast Routing Protocols Assuming Connected VANETs

Characteristics	**GPSR**	**GSR**	**A-STAR**	**MDDV**	**MURU**	**CAR**
Position Based	√	√	√	√	√	√
Greedy Forwarding	√	√	√	√		√
Predictive				√	√	√
Buffering (Carry-and-forward)				√	√	√
Street Aware		√	√	√	√	√
Traffic Aware (Probabilistic)			√	√		
Traffic Aware (Real-Time)						√

whose value is minimized over the target zones. Unlike other greedy position-based unicast routing protocols, anchor-based street and traffic aware routing (A-STAR) [26] utilizes city bus routes as a strategy to find routes with a high probability for delivery.

All the above protocols omit the problem of network disconnection. The authors in [22] introduced a new metric, expected disconnection degree (EDD), to evaluate the probability that a candidate route would be broken. By broadcasting the RREQ message, the path with the smallest EDD will be selected as the route. To handle the problem of mobile end nodes (source or sink), CAR [23] adopts the idea of guards which automatically adjust the connectivity path when end nodes change their speeds and/or directions. However, it first needs to broadcast the route discovery request to the entire network to find a proper route, causing excessive networking overhead even with some optimization schemes.

In summary, all these approaches basically require networks to be fully connected; otherwise, the route discovery phase will fail, rendering the subsequent routing strategy useless. A summary of those protocols in terms of different features is listed in Table 4.1.

4.2 Routing Protocols in Intermittent Connected Networks

As concluded in [12], network partitions in VANETs are very frequent. Therefore, it is better to consider a VANET is not always connected. With this assumption another group of routing protocols are proposed in the literature [7,13,14,17,27–29]. These routing protocols can be considered as the delay tolerant protocols and the carry-and-forward [16] scheme is used when network disconnection occurs. Network disconnections occur frequently in rural highway situations and in cities at night where fewer vehicles are running, making establishing end-to-end routes impossible. Even in densely-populated urban scenarios, sparse sub-networks can also be prevalent.

To route a message from a vehicle to a roadside unit, the motion vector (MOVE) routing algorithm [17] uses knowledge of neighboring vehicles velocities and trajectories to predict which vehicle will physically travel closest to the fixed destination. Another knowledge-based scheme, scalable knowledge-based routing (SKVR) algorithm [27] utilizes the relatively predictable nature of public transport routes and schedules. The SKVR works in two levels: the top level is inter-domain routing, where a source and destination are on different bus routes, while the bottom level consists of intra-domain routing within the same bus route.

Another algorithm in the delay-tolerant network category, MaxProp [30] utilizes carry-and-forward and packet prioritization techniques to maximize message delivery in a network with limited transfer opportunities between nodes. MaxProp is implemented in a real network where it is deployed on buses, allowing each bus to communicate its location and performance information to wireless access points or other buses as they are encountered.

When network infrastructures are available at intersections, a static node assisted adaptive routing protocol (SADV) has been proposed [7] for vehicular networks. When disconnected, each static node has the capability to store a message until it can forward the message to a node traveling on the optimal path. Optimal paths are determined based on a graph abstracted from a static road map and weighted with expected path forwarding delays from a delay matrix.

Similar to other routing algorithms designed for delay-tolerant networks, the geographical opportunistic routing protocol (GeOpps) [28] uses navigational information to route packets efficiently. GeOpps assumes that each vehicle has a navigation system that provides a suggested route to a destination. Each neighbor vehicle will use a utility function built into the navigation system

Table 4.2: Summary of VANET Unicast Routing Protocols for Intermittent Connected Networks

Characteristics	**MOVE**	**MaxProp**	**SKVR**	**SADV**	**GeOpps**	**VADD**
Position Based	√	√		√	√	√
Greedy Forwarding			√	√		√
Predictive	√	√			√	√
Buffering (Carry-and-forward)	√	√	√	√	√	√
Street Aware				√	√	√
Traffic Aware (Probabilistic)			√			√
Traffic Aware (Real-Time)				√		

to calculate the amount of time required to reach the next interest point. The vehicle that can deliver the packet fastest or closest to the destination will be chosen as the next hop for the message. Those protocols either require infrastructure at intersections or vehicles following the navigation system, but these assumptions may not be true in reality.

Assuming a pure vehicular ad hoc network architecture, the VADD [13] protocol is proposed. When wireless connectivity is not available, the carry-and-forward strategy is used to transfer packets along vehicles on the fastest roads available. Since vehicles may deviate from predicted paths, the routing path should be recomputed continuously during the forwarding process. To aid in this process, VADD uses a street graph weighted with expected packet delivery delays. However, a drawback is that when the average distance between vehicles is close to the communication range, the transmission delay will be much longer than the expected one used in VADD. Unlike VADD, a delay-bounded routing protocol [29] is introduced for VANETs. The goal of this routing algorithm is to select an optimal path that not only has the least transmission cost but also meet the delay requirement given by the application. However, the delay model used in [29] still has a similar problem as VADD. Table 4.2 summarizes the differences of all above-mentioned routing protocols with the carry-and-forward mechanism.

4.3 Network Connectivity Models

Existing VANETs routing protocols omit the connectivity information in highly dynamic networks, though mobility can increase the capacity of ad hoc wireless networks [31]. Obviously, mobility is the distinguishing feature of vehicular networks, affecting the evolution of network connectivity over space and time in a unique way.

The mathematical connectivity model in ad hoc networks has been studied in [32, 33] with the assumption that nodes follow the Poisson distribution. However, node movement in VANETs can be affected by multiple factors such as the traffic lights, other vehicles in the vicinity and speed limits. Therefore, instead of using traditional mobility models, researchers proposed several mobility models for VANETs [34, 35]. Observing the clustering phenomena in a highway vehicle network, the network connectivity of highway VANET is modeled in [34] and then the opportunistic packet relaying protocol (OPERA) is proposed. This model also assumes a Poisson distribution of vehicles and do not consider VANETs in a city scenario where vehicles' distribution can be significantly affected by traffic light, number of lanes and vehicle velocity.

With the Percolation theory, a critical phase of connectivity in wireless network is investigated in [35]. The authors claim that an ad hoc network is fully connected after a certain network density is reached. However, this model can only be applied on a static network with the assumption of Poisson distributions of nodes.

The above mentioned models look at network connectivity from a macroscopic level. There are also several models that address the problem at a microscopic view such as [36–39]. In the constant speed motion (CSM) model [37], a generic vehicle i's movement is constrained on a given road topology, and its speed is set to $v_i = v_{min} + (v_{max} - v_{min}) \times \alpha$ where α is a uniformly distributed random variable in [0, 1].

The fluid traffic motion (FTM) model [36] adopts a traffic stream approach on a microscopic level. It describes speed as a monotonically decreasing function of vehicular density, forcing a lower bound on speed when the traffic congestion reaches a critical state.

Then based on the intelligent driver model (IDM) [38], IDM with intersection management (IDM-IM) and IDM with lane changing (IDM-LC) models were proposed in [39]. The IDM-IM is a flows-interaction model which adds intersection handling to the car-to-car interaction description provided by IDM; the IDM-LC further extends the flows-interaction description of IDM-IM, by adding overtaking capability to vehicles. To the best of our knowledge, the IDM-based mobility models are the most accurate ones for VANETs. A detailed analysis of those IDM-based models is described in [40], and a simulator, VanetMobiSim [41], based on these models is developed by the authors.

Although there exist some efforts to create accurate mobility models, such as the IDM with lane changing model [39], most of these models are too complicated to be used in the networking protocol design. Instead of microscopic mobility models, we look at VANETs in a macroscopic way and try to reveal the statistical property of network connectivity. In the design of the ACAR protocol, this information is used to select the route with the highest probability of connection and thus the network throughput is increased.

4.4 Location Privacy Protection in VANET

Geographic routing has been widely used in vehicular ad hoc networks (VANETs) to achieve vehicle-to-vehicle and vehicle-to-roadside communications [11, 13, 14, 22, 24, 42]. By exploiting location information, geographic ad hoc routing provides superior scalability compared to traditional reactive routing protocols. However, location security becomes an important issue in achieving high network performance [43–45]. Even though location security can be protected, location information exchange among neighbors compromise location privacy as well.

The location privacy issue in MANETs is first addressed in [46], in which the authors defined the location privacy problem, threat model and application framework. In VANETs system, vehicle's location privacy issue is addressed in [47–49].

Location privacy issue can be solved in two different ways: hiding the information of who send the data and the information of where this data come from. For the first methods, although node's location information is released, the adversary cannot link the location to a certain user, thus protecting user's location privacy [46, 47, 50, 51]. Those approaches usually require periodically changing user's ID and such schedule is initialized or maintained by a third-party trustworthy infrastructure. The potential threat of this framework is that, the infrastructure component may not always be available and itself may be subject to security or privacy problems. Moreover, changing identifiers has detrimental effects on routing efficiency and increases packet loss as shown in [52].

In the second method, packet forwarders will send out a set of dummy locations which hides the true locations. For instance, a node will send packets in a rectangle or circular area in which there exist at least other $k - 1$ nodes [53]. Thus, *k-anonymity* is achieved since an adversary can only identify a user's location with the probability of no higher than $1/k$. Unlike the k-anonymity methods, dummy-based location privacy-protection algorithms were proposed [54, 55]. In [54], the network user generates several false position data (with one that contains the true position information) sent to the service provider. Because the service provider cannot distinguish the true position data from the dummies, the user's location privacy is protected. Similarly, authors in [55] hid user's real location by sending a set of dummy positions which are deliberately generated according to either a virtual grid or circle. In the above-mentioned methods, user's true location information is fully hidden within either an area or a set of dummies, so traditional geographic routing protocols will have a big problem in making routing decision as location information is not available.

Unlike previous works, we investigate user's location privacy issue through 1) replacing user's location information by dummy distance to destination (DOD) during routing and 2) generating pseudonyms to preserve user's identification information. Despite these changes, the geographic routing protocol will still work, with a slight modification. Both identification and location information of users is preserved in our dummy based location privacy protection (DBLPP) protocols, so it can achieve a higher level of location privacy protection in VANETs.

Chapter 5

Connectivity-Quality Model in VANETs

The connectivity-quality models of road segment, intersection and route that is consists of multiple road segments and intersections are investigated in this chapter. We first propose the cell based connectivity model in Section 5.1 for vehicles moving within road segments, and the cluster based connectivity model in Section 5.2 for vehicles around intersections. Then, an integrated connectivity model and the connectivity model of route are introduced in Section 5.3 and 5.4, respectively. Considering the transmission quality of a route in a connected network, we propose a novel metric connectivity-quality (CQ) in Section 5.5 which models both network connectivity and quality information.

5.1 Cell Based Connectivity Model

We first consider the model for the one-lane case and later generalize it to multiple lanes. In the one-lane scenario, we divide the road segment equally into m cells so that each cell can contain at most one vehicle and each vehicle can occupy only one cell. The length of cell d can be set as the average length of vehicles, e.g. $5m$. It will be fairly common that a vehicle partially occupies two adjacent cells. In this case, the cell containing the majority part of this vehicle is considered occupied. Since the distance between occupied cells will be used to compute the distance between vehicles in these cells, we found that there would be an error (at most $5m$) in the distance computation. However, compared to the large wireless communication range, e.g. $250m$ in 802.11b and $1000m$ in DSRC, this error can be ignored. Therefore, the problem statement of finding probability of connectivity of networks can be formulated as follows:

Sub-problem Statement 1: If there are n vehicles (also called nodes) on a road segment, what

is the probability that the distance of any two neighboring nodes is less than the communication range $R = n_0 \cdot d$, i.e. there are no more than n_0 successive empty cells on the road.

In one-lane scenarios, the number of empty cells is always $m - n$; but in the case of multiple lanes, the number of empty cells will range from $m - n$ to $m - \lceil n/n' \rceil$ where n' is the number of lanes. For multiple lanes, each cell in the road may contain any number of nodes within $[0, n']$. So in the extreme case, if every occupied cell contains only one node, the number of empty cells is $m - n$. On the other hand, if each occupied cell has n' nodes, the number will become $m - \lceil n/n' \rceil$. For instance, suppose 5 vehicles are deployed into a road with 5 cells and 3 lanes. Let cells be ordered geographically such that cell c_0 is at the leftmost and c_4 is at the rightmost position. It may occur that 3 vehicles are located in cell c_0 and the other two in cell c_4. So the number of empty cells in this case is 3. Intuitively, if the number of empty cells k is equal or less than n_0, then the network must be connected. If $k > n_0$, the network may be connected or disconnected depending on how the empty cells are distributed.

We denote P_{dis} and $P_{con} = 1 - P_{dis}$ as the probability of network being disconnected and connected, respectively. Since it is not easy to compute P_{con}, we first calculate P_{dis}. To obtain this probability, two other probabilities are required: 1) probability $P1$ that there exist exactly k empty cells if n nodes were deployed into m cells, denoted as $P1 = P\{\mu(n,m) = k\}$, and 2) probability $P2$ that there exist more than n_0 successive empty cells given exactly k empty cells on the road segment, which is denoted as $P2 = P\{\varphi(m,k) > n_0\}$. Then the probability that the network is disconnected becomes:

$$P_{dis} = \sum_{k=max(m-n,n_0)}^{max(m-\lceil n/n' \rceil,n_0)} P\{\mu(n,m) = k\} \cdot P\{\varphi(m,k) > n_0\} \tag{5.1}$$

5.1.1 Empty-Cell Probability P1

To drive safely on roads (with one lane), a driver need to keep a certain distance from the front or rear vehicles, thus the occupancy of one cell is dependent on the adjacent cells. Considering multiple lanes cases, since traffic flows on different lanes are independent of each other, the dependency of occupied cells is broken. If vehicles move in two directions on a road, the occupied cells will be more randomly distributed. Therefore, we first assume that vehicles are uniformly deployed on roads. Then, we adjust our model to take into account the clustering (platoon) phenomena of vehicles.

Assuming uniform node distribution, we investigate the probability that there exist exactly k empty cells on the road. Suppose there are n nodes deployed on the road with m cells. Let A_i be the event that the ith cell is empty, and let $\overline{A_i}$ be the event complementary to A_i (ith cell is occupied). Then we have:

$$P\left\{\mu(n,m)=k\right\}=\sum_{1\leq i_1<\cdots<i_k\leq m} P\left\{A_{i_1}\cdots A_{i_k}\overline{A}_{j_1}\cdots\overline{A}_{j_{m-k}}\right\} \tag{5.2}$$

where $\{j_1, j_2, \cdots j_{m-k}\} = \{1, 2, \cdots, m\} - \{i_1, i_2, \cdots, i_k\}$. $P\left\{A_{i_1}\cdots A_{i_k}\overline{A}_{j_1}\cdots\overline{A}_{j_{m-k}}\right\}$ is the probability that the i_1th to i_kth cells are empty and the j_1th to $j_{m-k}th$ cells are occupied by nodes. Since every term on the right side of the above equation is the same, the total number of terms is C_m^k. Moreover, we can rewrite the term as:

$$P\left\{A_{i_1}\cdots A_{i_k}\overline{A}_{j_1}\cdots\overline{A}_{j_{m-k}}\right\} = P\left\{A_{i_1}\cdots A_{i_k}\right\}\cdot P\left\{\overline{A}_{j_1}\cdots\overline{A}_{j_{m-k}}\left|A_{i_1}\cdots A_{i_k}\right.\right\} \tag{5.3}$$

where $P\left\{A_{i_1}\cdots A_{i_k}\right\} = \frac{C^n_{(m-k)\cdot n'}}{C^n_{m\cdot n'}}$ is the probability that there exist at least k empty cells on this road, and $P\left\{\overline{A}_{j_1}\cdots\overline{A}_{j_{m-k}}\left|A_{i_1}\cdots A_{i_k}\right.\right\}$ is actually the probability of $P\left\{\mu(n, m-k)=0\right\}$. So we obtain the following recursive formula:

$$P\left\{\mu(n,m)=k\right\} = C_m^k\cdot\frac{C^n_{(m-k)\cdot n'}}{C^n_{m\cdot n'}}\cdot P\left\{\mu(n,m-k)=0\right\} \tag{5.4}$$

Notice that the probability that there exists at least one empty cell is:

$$P\left\{\mu(n,m)>0\right\} = P\left(\bigcup_{i=1}^{m} A_i\right) = \sum_i P(A_i) - \sum_{i<j} P(A_iA_j) + \sum_{i<j<h} P(A_iA_jA_h) - \cdots \tag{5.5}$$

So the probability that all cells are occupied is:

$$P\left\{\mu(n,m)=0\right\}=\sum_{l=0}^{m}C_m^l\cdot(-1)^l\frac{C_{(m-l)\cdot n'}^n}{C_{m\cdot n'}^n} \tag{5.6}$$

By substituting Equation 5.6 into 5.4, the probability that there exist exactly k empty cells can be computed.

5.1.2 Successive Empty-Cell Probability P2

$P\left\{\varphi(m,k)>n_0\right\}$ denotes the probability that there exist more than n_0 successive empty cells on the road given that there were exactly k empty cells. Since the number of occupied cells is $m-k$, we are able to formulate this problem as:

Sub-problem Statement 2: Consider throwing k items into $N=m-k+1$ bags and each bag can contain any number of items $0,1,\cdots,k$, then what is the probability that at least one bag contains at least (n_0+1) items.

Since it is hard to directly compute this probability, we first examine the case where all bags satisfy the condition:

C1: Every bag contains at most n_0 items.

We denote $Num(k,N)$ as the number of possible deployments that satisfy **C1**. Then it can be rewritten as:

$$Num(k,N)=Num(k,N-1)+Num(k-1,N-1)+Num(k-2,N-1)+\cdots+Num(n_0,N-1) \tag{5.7}$$

The proof of Equation 5.7 is stated as follows. Let us consider a certain bag, b_i, that may contain $0,1,\cdots,n_0$ items. Suppose it contains j items, then the number of deployment that satisfy **C1** is $Num(k-j,N-1)$. By summing up all the possible j, we obtain:

$$Num(k,N)=\sum_{j=0}^{k-n_0}Num(k-j,N-1) \tag{5.8}$$

Each term in the right part of Equation 5.8 can be expanded as

$$Num(k-j, N-1) = \sum_{l=0}^{k-j-n_0} Num(k-j-l, N-2) \tag{5.9}$$

After expanding each term, Equation 5.8 becomes:

$$c[0]^{N-1} \cdot Num(k, 1) + c[1]^{N-1} \cdot Num(k-1, 1) + \cdots + c[k]^{N-1} \cdot Num(0, 1) \tag{5.10}$$

where $Num(x, 1)$ refers to the number of possible deployments of putting x items into one bag.

$$Num(x, 1) = \begin{cases} 0, \ x > n_0 \ or \ x < \max\{0, k - n_0 \cdot (N-1)\} \\ 1, \ \max\{0, k - n_0 \cdot (N-1)\} \leq x \leq n_0 \end{cases} \tag{5.11}$$

This number will be 0 if $x > n_0$ or $x < k - n_0 \cdot (N-1)$, since **C1** does not hold in these cases. If $x < 0$, it means putting negative number of items into bags, so $Num(x, 1)$ is also 0; otherwise, $Num(x, 1) = 1$.

Then the number of deployments meeting **C1** will be the sum of coefficients of all terms whose value are 1, i.e.

$$\sum_{i=k-n_0}^{\min\{k,(N-1)\cdot n_0\}} c[i]^{N-1}, \ c[i]^{t+1} = \sum_{j=\max\{0,i-n_0\}}^{\min\{i,t\cdot n_0\}} c[j]^t \tag{5.12}$$

where $c[i]^1 = 1 \ (i = 0, 1, \cdots, n_0)$. Since the total number of all possible deployments is $C_{N+k-1}^k = C_m^k$, the probability $P2$ is:

$$P\{\varphi(m, k) > n_0\} = 1 - \frac{\sum\limits_{i=k-n_0}^{\min\{k,(m-k)\cdot n_0\}} c[i]^{m-k}}{C_m^k} \tag{5.13}$$

Substituting Equation 5.4 and 5.13 into Equation 5.1, we can calculate the probability of the network being disconnected or connected on a certain road, if the network density information is known.

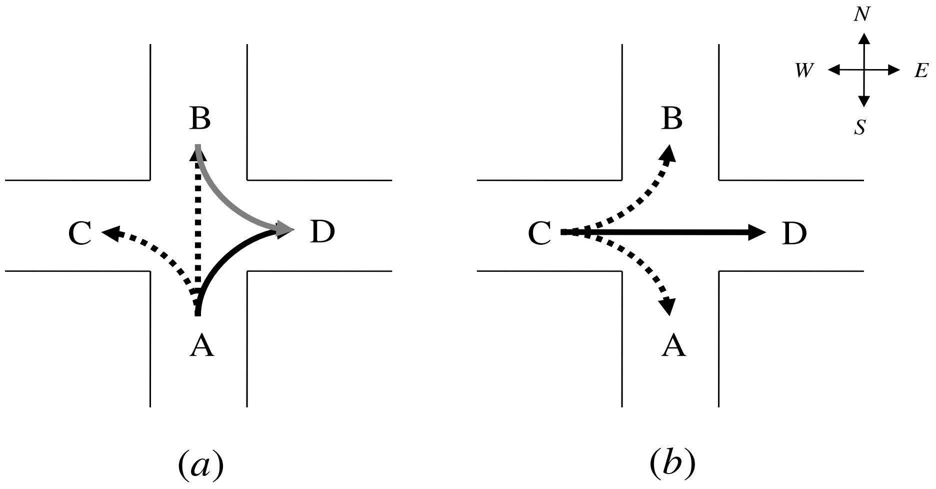


Figure 5.1: Illustration of Traffic Lights Affecting the Connectivity Model

5.2 Cluster Based Connectivity Model

Since traffic lights (red signal) can block approaching vehicles, these vehicles will form a cluster (or convoy) on the road. Therefore, the proposed connectivity model that assumes uniform node distribution needs to be modified by adjusting the network density information.

As shown in Fig. 5.1, suppose on road segment A, there are n_A nodes moving toward the intersection. Assume the length of A is l_A, the average velocity of vehicles moving on A is v_a and time period of red traffic light is t_A. Then the expected number of vehicles stopped by every red light on road A is:

$$\bar{n}_A = \begin{cases} \frac{n_A \cdot v_A \cdot t_A}{l_A}, \ (v_A \cdot t_A) < l_A \\ n_A, \ otherwise \end{cases} \tag{5.14}$$

If $(v_A \cdot t_A) \geq l_A$, then the red signal period t_A is long enough so that all vehicles on A are blocked. When the light turns green, stopped vehicles will resume moving and those moving in the same direction will be very close to each other since usually drivers prefer to follow the traffic flow. As a result, we can assume those vehicles move as a cluster in which the networks are connected. Therefore, the number of nodes on the road needs to be modified because the clustered nodes will be considered as one node.

Since those nodes in the same cluster cannot be fitted into one cell, they may spread over several cells. For example, suppose there are $\hat{n}$ nodes in a cluster and they are uniformly distributed on each lane of a road. Then the total number of cells on this road will be reduced from m to $m - \lfloor \hat{n}/n' \rfloor \cdot (d_s/d)$, where d_s is the safety distance between vehicles, d is the length of cell and n' is the number of lanes. If nodes are uniformly deployed on each lane, $\lceil \hat{n}/n' \rceil$ will be the maximal number of nodes on each lane, and $\lfloor \hat{n}/n' \rfloor \cdot (d_s/d)$ the maximal number of cells occupied by this cluster. The safety distance between vehicles can be simply calculated by:

$$d_s = v \cdot t_r + v^2/(2b) + d \tag{5.15}$$

where v is the average velocity of vehicles, t_r is the reaction time and b is the deceleration value of comfortable braking.

Next, we investigate how to compute the number of nodes in each cluster. Suppose the numbers of nodes moving toward the intersection on each road segment A, B, C and D are n_A, n_B, n_C and n_D, respectively. Then for each vehicle on A, the probability that it moves to D will be:

$$P_{AD} = \frac{n_D}{n_B + n_C + n_D} \tag{5.16}$$

Suppose at a certain time t, there are $\bar{n}^t_A$ nodes blocked on road A, then the expected number of nodes moving from road A to D is:

$$\bar{n}^t_{AD} = \frac{\bar{n}^t_A \cdot n_D}{n_B + n_C + n_D} \tag{5.17}$$

In the same way, we can get $\bar{n}^t_{BD}$ and $\bar{n}^t_{CD}$. If the traffic light controlling the north-south traffic turns green, as shown in Fig.5.1(a), it will generate $\bar{n}^t_{AD} + \bar{n}^t_{BD}$ nodes moving as a cluster on road D. If the traffic light controlling the east-west traffic turns green, as shown in Fig.5.1(b), there will be a cluster of $\bar{n}^t_{CD}$ nodes moving on road D. During each period of traffic light, two clusters will be produced. Therefore, the number of clusters on road D is:

$$N_D = \begin{cases} \left\lceil \frac{2 \cdot l_D}{v_D \cdot T} \right\rceil, \ l_D > (T \cdot v_D) \\ 1, \ otherwise \end{cases} \tag{5.18}$$

where T is the period of traffic light at this intersection. When $l_D > (T \cdot v_D)$, it means before the first cluster moves out of road D, more clusters will be generated. Then the number of clusters is $\left\lceil \frac{2 \cdot l_D}{v_D \cdot T} \right\rceil$ that is the upper bound of the actual number of clusters on road D. Therefore, the number of nodes on road D will be reduced to:

$$\begin{aligned} & n_D - \sum_{t=1}^{N_D} (\bar{n}_{AD}^t + \bar{n}_{BD}^t + \bar{n}_{CD}^t - 2) \\ & = n_D - \sum_{t=1}^{N_D} \bar{n}_{AD}^t - \sum_{t=1}^{N_D} \bar{n}_{BD}^t - \sum_{t=1}^{N_D} \bar{n}_{CD}^t + 2N_D \\ & \approx n_D - N_D \cdot (\bar{n}_{AD} + \bar{n}_{BD} + \bar{n}_{CD} + 2) \end{aligned} \tag{5.19}$$

where $\bar{n}_{AD}$, $\bar{n}_{BD}$ and $\bar{n}_{CD}$ can be obtained from Equation 5.14 and 5.17. If there are two moving directions on road D, a similar modification needs to be done for the other direction as well. By combining this new method for determining the number of nodes with the connectivity model proposed in Section 5.1, we can compute the probability of connectivity of each road segment. By adjusting the number of clusters, the proposed connectivity model can also be used for one-way roads or roads with only one traffic light at the end.

5.3 Integrated Connectivity Model of Road Segment

We have proposed the cell-based connectivity model where nodes move on roads without clustering and the cluster-based connectivity model in which traffic lights block vehicles to form clusters around intersections. Now, we describe how to integrate those two models to compute the connectivity of road segment.

Vehicles form a cluster when they are blocked by the traffic light in an intersection. However, the cluster will exist only for a period of time. After that, these vehicles will merge into the traffic flow of roads they are moving on. In other words, vehicles deployment on a road segment changes periodically between cluster-based and cell-based modes.

Suppose there is only one cluster on a road segment, e.g. the road segment A as shown in Fig. 5.1. Nodes in this cluster are geographically labeled as $1, 2, \cdots, \bar{n}$, where node 1 is the closest one to the intersection and $\bar{n}$ is the furthest one. Therefore, the size of this cluster is $\bar{n}$. Assume these nodes will move into another road, and the density and velocity of this road are d and $\bar{v}$, respectively. We define t_i^b as the time for a node i ($i \in [1, \bar{n}]$) to move out of the cluster, i.e. after t_i^b seconds, node i will merge into the traffic flow of a road segment (e.g. D in Fig. 5.1).

To compute the time t_i^b of node i, we first investigate the one-lane one-cluster case, and then generalize it to multiple-lane multiple-cluster cases. Within one lane, a vehicle cannot accelerate freely as its movement is restricted by many factors: the distance to the preceding vehicle, velocities of the preceding vehicle and itself. This phenomena is represented by the car following model [38], in which the acceleration rate of node i at time instance t is:

$$a_i^t = \frac{dv_i^t}{dt} = a\left[1 - \left(\frac{v_i^t}{v^0}\right)^4 - \left(\frac{s_i^*}{s_i^t}\right)^2\right] \tag{5.20}$$

where v_i^t is the velocity of node i at time t, a is the maximum acceleration rate and s_i^t is the distance between node i and its preceding node. v^0 is the desired speed, which is equal to $\bar{v}$ in this case. Distance s_i^* is called *desired dynamical distance* [38] and is computed by:

$$s_i^* = s_0 + \left(v_i^t\tau + \frac{v_i^t \cdot \Delta v_i^t}{2\sqrt{ab}}\right) \tag{5.21}$$

It is a function of the minimum bumper-to-bumper distance s_0, the minimum safe time headway τ, the velocity difference with respect to front vehicle $\Delta v_i^t = (v_i^t - v_{i-1}^t)$ and the maximum acceleration and deceleration values a and b. For node 1 in the cluster, its distance to the preceding node is $s_1 = 1/d$; because in the cell-based model, vehicles are assumed to be evenly distributed on road segments. The distance node i drives from time 0 to t is $l_i^t = \int\limits_0^t \frac{1}{2} \cdot a_i^t \cdot t^2 dt$, so the value of s_i^t will be $(l_{i-1}^t - l_i^t)$.

Therefore, we obtain the time t_i^b that node i needs to reach the speed of $\bar{v}$. It is computed by solving the integral equation:

$$\int_{t=0}^{t=t_i^b} a_i^t \cdot tdt = \bar{v} \tag{5.22}$$

During time period $[t_{i-1}^b, t_i^b]$, there are only $(\bar{n} - i + 1)$ nodes remaining in the cluster. According to Section 5.2, we can compute the new number of cells and the connectivity probability during time period of $[t_{i-1}^b, t_i^b]$. Then, the overall connectivity probability of the road segment can be computed as:

$$P_{cell} \cdot \frac{T - \max\{t_i^b\}}{T} + \sum_{i=1}^{i=\bar{n}} P_{cluster}(\bar{n} - i + 1) \cdot \frac{t_i^b - t_{i-1}^b}{T} \tag{5.23}$$

where $t_0^b = 0$, $T = l/\bar{v}$ is the time a vehicle needs to move from one end to the other end of the road segment. P_{cell} is the probability of connectivity computed by the cell-based model, and $P_{cluster}(\bar{n} - i + 1)$ is the probability of connectivity obtained through the cluster-based model with a cluster of $(\bar{n} - i + 1)$ nodes. If there are N_c clusters and the size of each cluster is $\bar{n}_j, j \in [1, N_c]$, the connectivity probability of road segment is:

$$\sum_{j=1}^{j=N_c} P_{cell} \cdot \frac{T - \max\{t_i^j\}}{T} + \sum_{j=1}^{j=N_c} \sum_{i=1}^{i=\bar{n}_j} P_{cluster}(\bar{n}_j - i + 1) \cdot \frac{t_i^j - t_{i-1}^j}{T} \tag{5.24}$$

where t_i^j is the time t_i^b that node i needs to move out of the jth cluster.

In multiple lane cases, we assume clustered vehicles are evenly distributed on each lane because it is natural for drivers to change lanes if the current one is too congested. We apply the calculation of the single lane case to each lane and can compute the value of t_i^j for every $i \in [1, \bar{n}_j]$ and $j \in [1, N_c]$. Note that, the value of each t_i^j will change, and so does $(t_i^j - t_{i-1}^j)$. However, with Equation 5.24, we can compute the probability of connectivity of road segment for multiple lane and multiple cluster cases.

5.4 Connectivity Model of Route

So far, we modeled the network connectivity of a road segment based on the information of road length, number of vehicles, period of traffic light, and average velocity. In this section, we investigate the network connectivity of a route (path) that consists of multiple road segments. In other words, we will compute the probability that there exists a connected network on a certain route.

Suppose there is a route that consists of n road segments which are sequentially numbered as $1, 2, \cdots, n$. We denote the connectivity probability of each segment as P_i ($i = 1, 2, \cdots, n$). Then, the connectivity of a route will be $\prod\limits_{i=1}^{n}(P_i \times P_{ij})$ where $j = i + 1$ and $P_{ij} = 1$ when $i = n$. P_{ij} is the network connectivity of the intersection between road segment i and j.

To address the dependency issue of connectivity probabilities of adjacent road segments, we need to understand the movement of vehicles around intersections. Due to the traffic light at a intersection, vehicles may be stopped by the red signal. Therefore, the connectivity of network around the intersection will be higher than other parts of the road. In this section, we will investigate the connectivity probabilities of two types of networks. First, we look at the network with cars stopped around intersection areas by traffic lights. Second, we consider the case where no car is stopped by traffic lights. Finally, the expected network connectivity probability of an intersection is computed.

5.4.1 Vehicle's Distribution around Intersections

As shown in Fig. 5.2, when the traffic light turns to red for east-west direction, there may be several approaching cars, such as n_0 and n_3, stopped by the red signal. Therefore, the uniform distribution of vehicles on road segment C is broken. In other words, more cars are being blocked in front of the traffic light, so less vehicles will be moving on road segment C. On the other hand, because the traffic signal for road segments A and B are green, vehicles on these two roads follow uniform distribution. In this case, the network connectivity of road C is lower but the connectivity probability of road B is higher than normal. If the traffic light becomes green for

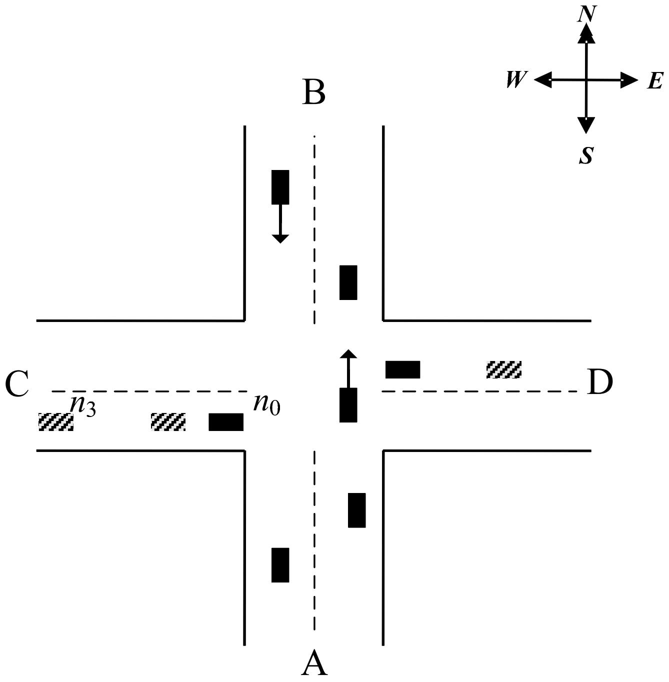


Figure 5.2: Network Connectivity around Intersections with Stopped Vehicles

east-west direction, the network connectivity of B will be lower but that of C is higher. Therefore, we note that the connectivity probabilities of two adjacent road segments are not independent due to traffic lights.

Now, we investigate the network connectivity of the intersection between road C and B. We first define $P3$ as the probability that no car is stopped by the traffic light. $P4 = 1 - P3$ denotes the probability that at least one car is stopped by the traffic light. When cars are blocked by the traffic light, there are two possibilities of their future movements: 1) stop at the intersection, or 2) move (right turn) to another road segment. Considering these two cases, we further define $P4'$ as the probability that all stopped cars move away from the intersection. This probability is usually very small because even if there is only one car stopped at the intersection, all approaching cars has to stop and stay at the intersection too. Complementary to $P4'$, we denote $P4" = P4 - P4'$ as the probability that at least one car stopped at the intersection.

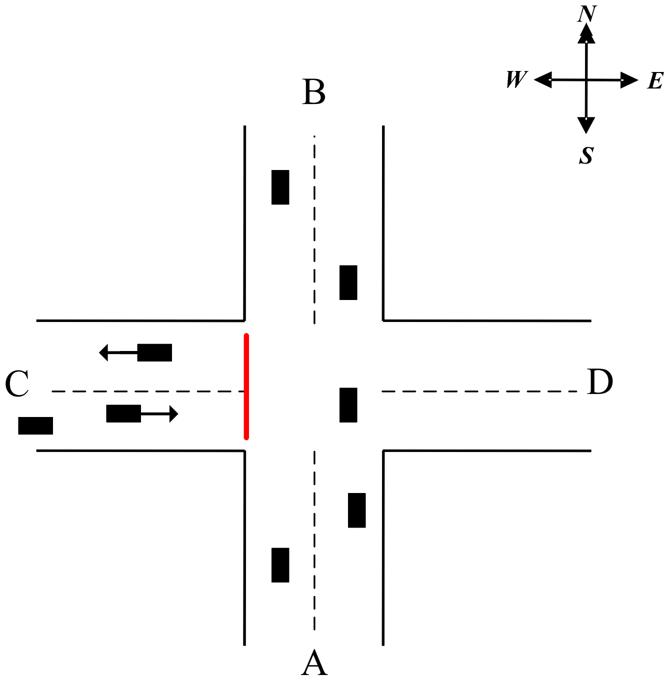


Figure 5.3: Network Connectivity around Intersections without Stopped Vehicles

5.4.2 Connectivity Probability Without Stopped Vehicles P3

Even though there is a traffic light in an intersection, it is possible that the red signal does not stop any approaching vehicles which are too far away from the traffic light. As shown in Fig. 5.3, cars are moving towards the traffic light on road C and before they reach the intersection, the signal turns to green from red. This case occurs with the probability of $P3$:

$$P3 = \frac{C^{n_B}_{m'_B}}{C^{n_B}_{m_B}}, \ \left(m'_B = m_B - \frac{v_B \times t}{l}\right) \tag{5.25}$$

where v_B, n_B and m_B are the average vehicle velocity, number of vehicles and number of cells on road segment B, respectively. l and t are the size of cell and the period of red signal. The above equation models the probability that no car is stopped by the red signal. Because the north-south traffic is not affected by the traffic light, we can consider uniform distributions of vehicles on road C and B. In this case, the network disconnects only if there is no car on road B and C around the intersection area. As shown in Fig. 5.4, we are interested in the areas of x and y on road segment

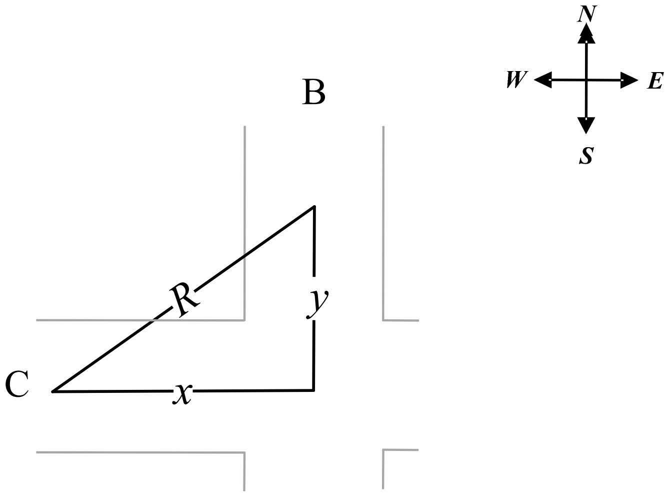


Figure 5.4: Network Disconnection around Intersections with a Uniform Node Distribution

C and B, respectively. The value of x and y must satisfy the following condition $R^2 = x^2 + y^2$ where R is the communication range.

The value of x, number of empty cells, will be ranging from 0 to $\min\{n_0, (m_C - n_C)\}$, where n_0, n_C and m_C are the communication range (in number of cells), and number of nodes and number of cells on road segment C. The value of x must be smaller or equal to n_0 because of the equation $R^2 = x^2 + y^2$. On the other hand, it has to be smaller than $(m_C - n_C)$. Otherwise, there must be at least one car being deployed in the area of x due to the pigeonhole principle. Similarly, the value of y is within $[0, \min\{n_0, (m_B - n_B)\}]$ where n_B and m_B are the number of nodes and number of cells on road segment B. If there is no car in the areas of x and y, the network disconnects around the intersection. We denote the probability of this event occurring as $\bar{P}_{BC}$ which can be computed by:

$$\bar{P}_{BC} = \frac{C^{n_C}_{(m_C - x)}}{C^{n_C}_{m_C}} \times \frac{C^{n_B}_{(m_B - y)}}{C^{n_B}_{m_B}} \tag{5.26}$$

Since the value range of x and y are known, we can easily compute expected value $E(\bar{P}_{BC})$. This value is considered as the probability of network disconnection with uniform distribution of nodes on road B and C. Therefore, the network connectivity probability in this case is:

$$P3 \times [1 - E(\bar{P}_{BC})] \tag{5.27}$$

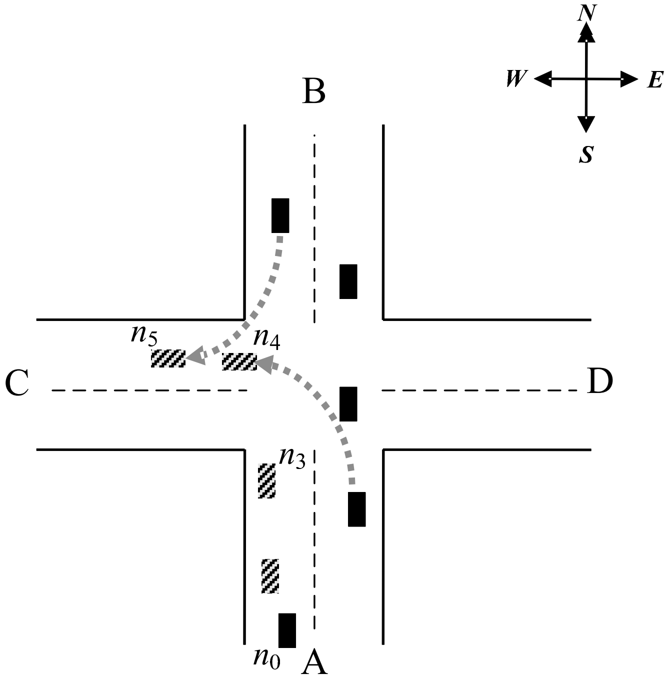


Figure 5.5: All Stopped Vehicles Move Away from Intersection

5.4.3 Connectivity Probability With Stopped Vehicles P4

Complementary to $P3$, we can compute the probability of $P4 = 1 - P3$. In this case, there is at least one car stopped in front of the intersection due to the traffic light. According to the Equation 5.14, we can compute the number of vehicles stopped on road C. We denote this value as $\bar{n}_C$. For those stopped vehicles, the probability of each one moving from road C to A can be obtained by Equation 5.16. We denote this probability as P_{CA}. We first look at the probability ($P4'$) that all stopped vehicles move away from road C to A:

$$P4' = (P_{CA})^{\bar{n}_C} \tag{5.28}$$

Since all stopped vehicles on road C move to road A, the nodes on road C follow uniform distribution. As shown in Fig. 5.5, there may be some vehicles moving to road C from other road segments, such as n_4 from A and n_5 from B. However, they will not break the uniform distribution of nodes on road C. Then, the number of nodes on C change to $n_C - \bar{n}_C + \bar{n}_{BC} + \bar{n}_{AC}$ where $\bar{n}_{BC}$ and $\bar{n}_{AC}$ denote the number of nodes moving to road C from B and A. According to the

Equation 5.26, we can compute the probability of network disconnection as $\tilde{P}_{BC}$ with the new number of nodes on road C and B. Therefore, the probability of existing connected network in this case will be:

$$P4' \times [1 - E(\tilde{P}_{BC})] \tag{5.29}$$

Finally, we look at the case where there is at least one stopped vehicle at the intersection. The probability $P4$" can be obtained by $P4 - P4'$. As we defined previously, the network connectivity of a road segment is considered as the probability that there exists a connected network from one end to the other end of the road segment. If there is a car stopped in front of the intersection, we can consider there is always a node at the eastern end of road C, which satisfies the definition of network connectivity of a road segment. In other words, the network around intersection area is always connected in this case.

Therefore, the network connectivity of the intersection between road B and C can be computed as:

$$P3 \times [1 - E(\bar{P}_{BC})] + P4' \times [1 - E(\tilde{P}_{BC})] + P4\text{"} \tag{5.30}$$

Until now, we have modeled the connectivities of networks on road segments and around intersections. Then, the connectivity probability of a path (that is composed of multiple road segments) can be computed as the product of probabilities of those road segments and adjacent intersections. For a given path starting from road segment s and ending at e, we denote i and j as two adjacent road segments, P_i as the connectivity probability of road segment i, and P_{ij} as the connectivity probability of the intersection between road segment i and j. Thus, the connectivity probability of this path can be obtained from the following equation:

$$P(s,e) = \prod_{i=s}^{e} (P_i \times P_{ij}),\ (j = i + 1) \tag{5.31}$$

where $P_{ij} = 1$ when $i = e$. The above equation can be used to compute the connectivity probability of a given route that consists of several road segments.

5.5 Connectivity-Quality of Route

For two road segments with similar network connectivity probabilities, their transmission qualities may be quite different. In other words, the proposed connectivity probability model needs to be adjusted by considering the transmission quality of a route. To meet this goal, we propose a novel metric, called connectivity-quality, which combines the information of both network connectivity and transmission quality of a route. For a route that is consists of several road segments, its CQ can be computed as $\prod (CQ_i \times CQ_{ij})$ where i and j are adjacent road segments.

5.5.1 Date Delivery Ratio of Road Segment

Considering a road segment with connected networks, we first model the packet error rate (PER) of a single hop. Then, we model the PER of a multi-hop route. Finally, the average PER of all possible routes within a road segment is used to compute the data delivery ratio of this road segment.

To model the path loss of a single hop between any two nodes, two cases need to be considered: the line-of-sight (LOS) and non-line-of-sight (NLOS) where there is at least one neighbor between these two nodes. Because of the popularity and lower price of IEEE 802.11 devices, the physical layer in VANETs (the DSRC/IEEE 802.11p PHY) will be a variation of the orthogonal frequency-division multiplexing (OFDM) based on the IEEE 802.11a standard. So the channel fading model of determining the received signal power level in the case of LOS is [56]:

$$P_r = \frac{P_t}{(4\pi)^2 \left(\frac{d}{\lambda}\right)^{\gamma}} \left[1 + \eta^2 + 2\eta \cos\left(\frac{4\pi h^2}{d\lambda}\right)\right] \tag{5.32}$$

where P_t is the transmit power, d is the distance between the transmitter and receiver, λ is the wavelength of propagating signal, η is the reflection coefficient of the ground surface, γ is the path loss factor and h is the antenna height. The model of NLOS is expressed as:

$$P_r = \begin{cases} P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 (d \leq 1m) \\ P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{d^\gamma} (d > 1m) \end{cases} \tag{5.33}$$

Taking into account the effect introduced by the cyclical prefix attached to each OFDM symbol, the signal to interference plus noise ratio (SINR) should be reduced by a factor of α:

$$SINR = \alpha \cdot 10 \log_{10} \left(\frac{P_r}{P_0 + \sum\limits_{i=1}^{N_{INT}} P_{INT}^i} \right) \tag{5.34}$$

where α is 0.8 according to [56], P_0 is the background noise, and P_{INT}^i is the interference from neighbor n_i.

Suppose on a certain road segment, as shown in Fig. 5.6, node n_a is sending packets to n_b and the distance between them is d_{ab}. From the perspective of n_b, there will be $den \times (R_{INT} - 2R - d_{ab})$ potential interfering nodes around it. In which, R and R_{INT} are the communication and interference ranges of n_b, and den is the network density of this road.

In the IEEE 802.11 protocols, before each communication the RTS/CTS (request to send/clear to send) packets need to be transmitted between sender and receiver to reduce frame collisions introduced by the hidden terminal problem. After that, during the communication between n_a and n_b, nodes within their communication ranges are not allowed to transmit packets. Thus, the potential interfering nodes must be in the area that is outside the communication ranges of n_a and n_b but inside their interference ranges. Within these areas, for a circle with a radius of R, there is at most one transmission that can interfere with the packet receptions at n_b. Therefore, there are at most $\left\lfloor \left\lceil \frac{R_{INT}-R}{2R} \right\rceil + \left\lceil \frac{R_{INT}-R-d_{ab}}{2R} \right\rceil \right\rfloor$ transmissions that interfere with node n_b simultaneously.

The receive power P_{INT}^i of each interference transmission can be computed through Equation 5.32 or 5.33 where d is the distance between n_b and the center of each segment labeled as $2R$ in Fig. 5.6. For cases with $(d_a < 2R)$ and $(d_b < 2R)$, $(3R + d_b/2)$ and $(3R + d_{ab} + d_a/2)$ are the

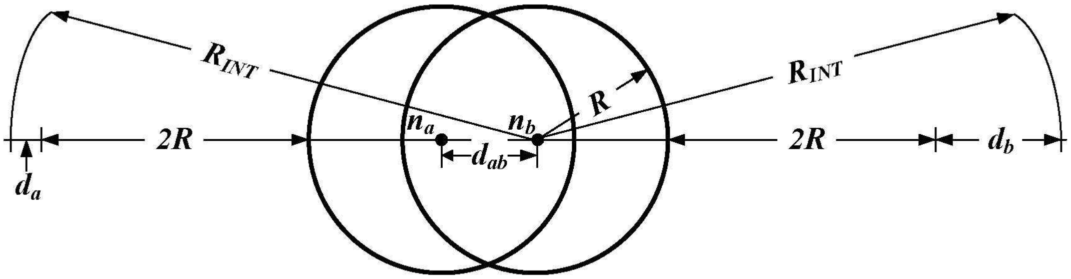


Figure 5.6: Illustration of the Number of Potential Interfering Nodes

distances of interference transmissions in d_b and d_a, respectively. If node n_b is in a nearby intersection area, there will be more potential for interfering nodes. Similarly, for roads with different network densities joined at an intersection, we can calculate the number N_{INT}.

In Equation 5.34, we use the maximum number of interfering transmissions with the communication between n_a and n_b, thus the worst case of SINR for n_b is obtained. In simulations, we found this lower bound value was very close to the real one; thus, we use it to further calculate the bit error rate and packet error rate of a single hop transmission.

Suppose the binary phase shift keying (BPSK) scheme is used to modulate the signal, the bit error rate (BER) is:

$$BER = Q\left(\sqrt{2 \cdot SINR}\right) \tag{5.35}$$

where $Q(x) = 0.5 - 0.5 \times erf(\frac{x}{\sqrt{2}})$ and $erf(\cdot)$ is the error function. Because of retransmissions in the link layer, the frame error rate (FER) can be computed as:

$$FER_{link} = 1 - \sum_{i=0}^{N} (1 - FER)FER^{i} \tag{5.36}$$

where $FER = 1 - (1 - BER)^{L}$, L is the length in bits of each frame and N is the number of retransmission times. Suppose every packet is composed of t frames, the PER is computed by:

$$PER = 1 - (1 - FER_{link})^{t} \tag{5.37}$$

Given the communication distance and number of neighbors, we can model the PER of a single hop. Therefore, if the node deployment of a network is known, it is possible to compute the PER of every hop.

Next, we discuss how to model the PER of a certain road segment (denoted as PER_{rs}). On a certain road segment, suppose there is a route $route_j$ that is composed of h hops with PER at every hop of PER_l ($l = 1, 2, \cdots, h$), then the PER of forwarding packets along this route $route_j$ can be computed as:

$$PER_{route_j} = 1 - \prod_{l=1}^{h} (1 - PER_l) \tag{5.38}$$

This equation is valid only if the PER is independent from one hop to the next; but due to the wireless communication environment there could be interference which violates this assumption. However, in this work, we use this equation as the first-order approximation of the PER of forwarding packets on a certain route.

Since different routes (that are composed of different hops) give different PERs, we consider a routing algorithm that minimizes PER, so the problem is to determine the minimal expected PER. If there are n nodes and k' empty cells on the road, for a certain distribution of these empty cells, the minimal PER of this road segment is denoted as $\min\{PER_{route_j}\}$. To compute this value, we need to know how the nodes (and empty cells) are deployed in the network. However, it is impossible to obtain such information because vehicles are always moving. To address this issue, we average these minimal PERs and obtain the PER of this road segment as $PER^i_{k'} = E[\min\{PER_{route_j}\}]$.

This value can be easily determined because we can compute the PER of every route. Therefore, the expected value of PER_{rs} can be calculated as:

$$E\left[PER^i_{k'}\right] = E_k\left[E\left[PER^i_{k'}|k' = k\right]\right] \tag{5.39}$$

which can further be rewritten as:

$$PER_{rs} = \sum_{k=m-n}^{m-\lceil n/n' \rceil} \sum_{i=1}^{C^k_m} \frac{1}{C^k_m} \cdot PER^i_k \cdot P\left\{\mu(n, m) = k\right\} \tag{5.40}$$

where m and n' are the number of cells and number of lanes on this road segment, respectively. Thus we use $D_{rs} = 1 - PER_{rs}$ to model the data delivery ratio (transmission quality) of a certain road segment.

5.5.2 Connectivity-Quality Metric

Data delivery ratio and packet error rate (PER) are usually used to evaluate the transmission quality of a route in networks. When we use these two metrics, we always assume that the network is fully connected because otherwise the delivery ratio will be zero. Therefore, delivery ratio is considered a useful metric with the condition that networks are connected.

In the previous section, we model the PER and data delivery ratio of a road segment. However, that data delivery ratio D_{rs} is actually the probability $P(D|C)$ where the event D means a packet is successfully delivered and C denotes the event of network being connected. Therefore, $P(D|C)$ gives the probability of a packet being successfully delivered with the condition that networks are connected. If we multiply $P(D|C)$ by the network connectivity probability $P(C)$, we will have the following equation:

$$P(D, C) = P(D|C) \times P(C) \tag{5.41}$$

In other words, $P(D, C)$ gives the joint probability that a packet is successfully delivered in a connected network. If we apply this probability to a road segment, it will become the connectivity-quality (CQ) metric which will be introduced later.

According to our connectivity model, the larger the network density, the higher the network connectivity probability will be. However, higher densities can cause larger interferences (more nodes in interference ranges), and thus reduce the packet delivery ratio. On the other hand, it is possible that a road segment has a low network connectivity probability. However, if the network on it becomes connected, the delivery ratio may be very high (due to low interferences). Therefore, both network connectivity and data delivery ratio are important in selecting routes.

If we investigate the probability $P(D, C)$, it contains two probabilities $P(D|C)$ and $P(C)$. For a certain road segment, these two probabilities can be re-written as D_{rs} and P_{rs}, respectively.

Therefore, we define a novel metric, connectivity-quality (CQ), in this way:

$$CQ_{rs} = D_{rs} \times P_{rs} \tag{5.42}$$

We can interpret the Equation 5.42 as a weighted connectivity probability of a road segment. The weight is the data delivery ratio of this road segment (with connected networks). The CQ metric is not only useful for VANETs but any other intermittent-connected networks because it models both network transmission quality and network connectivity in a mobile network with frequent network disconnections.

As in the computation of CQ_{rs}, it is easy to compute the CQ of networks around an intersection. The network connectivity of an intersection has been discussed previously. To compute the data delivery ratio of networks around an intersection area, we can still use the PER models proposed in Section 5.5.1. The only difference is that there will be only single hop communication around intersection areas, so computing CQ for an intersection should be easier than that for a road segment.

For a given path starting from road segment s and ending at e, we denote i and j as two adjacent road segments, CQ_i as the CQ of road segment i, and CQ_{ij} as the CQ of the intersection between road segment i and j. Then, the CQ value of the entire path is obtained from the following equation:

$$CQ(s,e) = \prod_{i=s}^{e} \left(CQ_i \times CQ_{ij} \right), \ (j = i + 1) \tag{5.43}$$

where $CQ_{ij} = 1$ when $i = e$. As it will be shown in Chapter 7, since the ACAR protocol chooses routes with the highest connectivity-qualities, the data delivery ratio and network throughput of ACAR are drastically increased compared to other protocols.

Chapter 6

Adaptive Connectivity Aware Routing Algorithm

The ACAR protocol includes two essential elements: 1) correctly selecting an optimal route that consists of road segments with the best connectivity-quality, and 2) efficiently forwarding packets hop-by-hop through each road segment in the selected route. To eliminate the impact of inaccurate statistical density data, we developed an adaptive route selection algorithm that collects real-time density information on-the-fly while forwarding packets. In each road segment in the selected route, the next hop is selected using a metric that minimizes the packet error rate (PER) of the entire route based on measured PERs at each node. In addition, carry-and-forward [16] mechanism is adopted to handle frequent network partitions in VANETs.

6.1 Selection of Route with the Highest Connectivity-Quality

According to the proposed connectivity model and CQ metric, a node can compute a route with the best connectivity-quality. We consider this as the optimal route which will be used to forward packets. Required information includes network densities, road segment lengths, average velocities, number of lanes and traffic light periods which are provided in pre-installed digital maps. Therefore, every packet forwarder (vehicle) can locally compute and find the optimal route to deliver packets.

Based on the classic *Dijkstra* algorithm, we propose an algorithm to find the optimal route with the best connectivity-quality. As shown in Algorithm 1, the inputs of the FIND() function include: the road topology map G, the source s and destination d. In the map G, vertices are intersections and edge are road segments between intersections. Given the location of source and destination nodes, the output of the FIND() function is a sequence of intersections that are used to construct the final route.

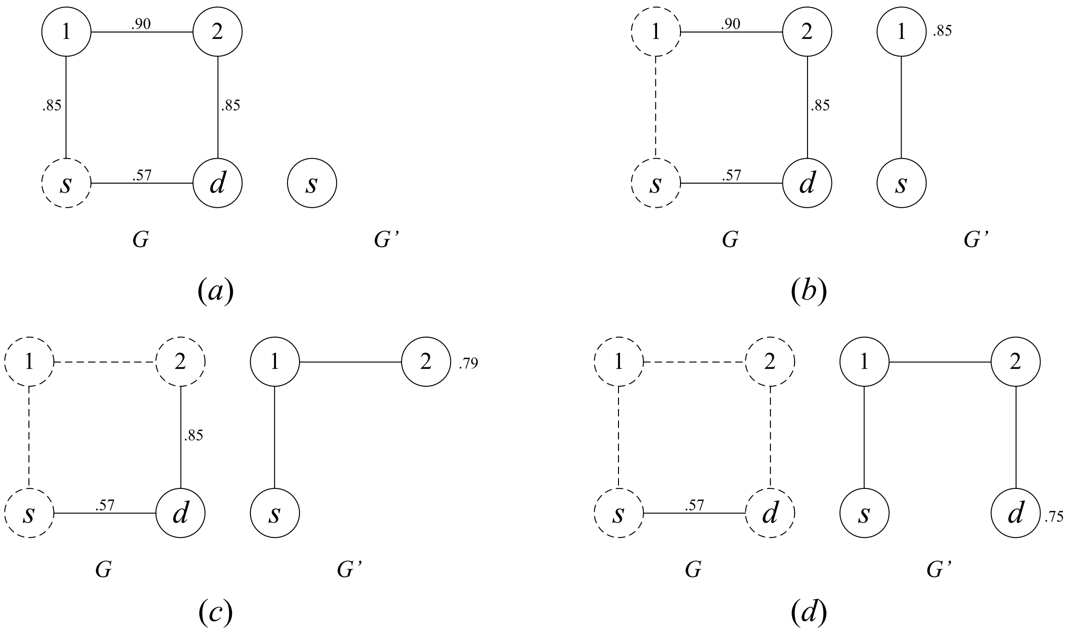
Algorithm 1 FIND (G, s, d)

```
1:  Add $s$ and $d$ as vertices into graph $G$
2:  $G' \leftarrow s$
3:  $G \leftarrow G - s$
4:  while $G$ is not empty do
5:      $m \leftarrow 0$
6:      for each vertex $u \in G'$ do
7:          for every $u$'s neighbor $v \in G$ do
8:              if $max\{CQ(s, v)\} > m$ then
9:                  $m \leftarrow CQ(s, v)$
10:                 $v' \leftarrow v$
11:                 $u' \leftarrow u$
12:             end if
13:         end for
14:     end for
15:     $pre[v'] \leftarrow u'$
16:     if $v' = d$ then
17:         return $pre[]$
18:     end if
19:     $G' \leftarrow G' + v'$
20:     $G \leftarrow G - v'$
21: end while
```

For each vertex $v \in G$, if it is on the optimal route, its parent node (also on the route) is stored in $pre[v]$. If v is not on the route, its $pre[v] = NULL$. Therefore, from the destination d, we can trace backward to the source s and construct the route. The graph G', which is a tree that saves the optimal path from the source to the destination. For every node $u \in G'$, we will check all its neighbors v in graph G. Therefore, lines $8 - 12$ will find a new node $v \in G$ which is the neighbor of $u \in G'$ where the following property holds.

Property 1: If a new node v is added to G', the connectivity-quality from s to v is the largest compared with any other remaining nodes in G.

In line 8, $max\{CQ(s, v)\}$ denotes the highest connectivity-quality of a route from s to v in graph G'. It is possible there are more than one path from s to v in graph G', so we need to compute every $CQ(s, v)$ and select the path with the highest CQ. To obtain each $CQ(s, v)$, we need to use equations in Section 5.5. Based on the above description, we have the second property:



(*a*) (*b*) (*c*) (*d*)

Figure 6.1: Illustration of Route Selection Algorithm

Property 2: For every node $v \in G$, only the path from s to v with the largest connectivity-quality will be added into G'.

Due to these two properties, we could easily proof the proposed algorithm satisfies the property of optimality. Now, we will use an example to illustrate how this algorithm works. As shown in Fig. 6.1(a). The lengths of road segments r_{s1}, r_{12}, r_{2d} and r_{sd} are $1000m$, $800m$, $1000m$ and $800m$, respectively. There are 20, 16, 20 and 8 nodes on road segments r_{s1}, r_{12}, r_{2d} and r_{sd}, respectively. Then, we can compute the CQs of r_{s1}, r_{12}, r_{2d} and r_{sd} as .85, .90, .85 and .57, respectively. According to our FIND() algorithm, node s is first moved to G'. Then, as shown in Fig. 6.1(b), node 1 will be added to G' as it provides a higher connectivity-quality than that of node d. Since the connectivity-quality $CQ_{s12} = .78$, node 2 is moved to G' as shown in Fig. 6.1(c). Finally, as shown in Fig. 6.1(d), node d was added to G' with the connectivity-quality $CQ_{s12d} = .69$ which is still larger than $CQ_{sd} = .57$. Therefore, the route r_{s12d} will be considered as the optimal route for forwarding packets.

After receiving a packet, a node calculates the optimal route and selects the next hop which is closer to the next intersection. For the example shown in Fig. 6.1, vehicles on road segments r_{s1} or r_{12} can compute the same route r_{s12d} to forward packets.

When packets are routed around an intersection, the chosen next hop will be the one which is closest to the next intersection along the optimal route. Routing policies such as location first, direction first and hybrid probes in [13] can be adopted in ACAR to further improve its performance, which are considered as our future works.

6.2 Velocity Compensated Neighbor Location Prediction

In geographic routing, every vehicle periodically broadcasts (beacons) its current location information to its neighbors. However, since the broadcast period cannot be too small, the neighbor's information may be out-of-date and thus affects the next hop selection in geographic routings.

To address the issue of out-of-date neighbors, many neighbor location prediction (NLP) algorithm are proposed [11, 21, 57, 58]. The basic idea is that, before selecting the next hop, a node needs to predict all its neighbors' locations based on their position and velocity information broadcasted in the last time interval. In the ACAR protocol, we simply adopt the NLP scheme used in [58].

After predicting neighbors' locations, node i selects the next hop only through those still within the communication range. Although the NLP algorithm used in ACAR is very simple, it does help improve the network performance as shown in our simulation results. In some cases, a node cannot find another neighbor to forward packets (in the event of network partition), then these packets will be saved into its buffer and carried [16] with the vehicle as it moves towards a next node to which it can forward the packet again.

6.3 Adaptive Route Selection

If the density information on each road segment is correct, the optimal route will be the one with the highest connectivity-quality. However, in reality, there may be some errors in the statistical density data. For example, suppose on road A there are 100 nodes (on average) in the afternoon, then it is possible that the network density between 2:00pm-4:00pm is 50 and from 4:00pm to 6:00pm is 150.

One possible solution to this problem is to flood the entire network to collect the real-time density information. However, even with directional and efficient flooding, this approach could still cause too much broadcast overhead. Therefore, we propose an adaptive path selection approach that collects real-time density data when packets are being forwarded into the network.

ACAR first computes a route based on statistical density data from the pre-loaded map. It then puts the route information into packet headers and transmits packets along this selected route. While the packets are being forwarded to the destination, network densities of all road segments along this path are collected simultaneously. This process, called on-the-fly density collection, is described in the next section. After a pre-defined number of on-the-fly density collections (e.g. 10), the density information on road segments in the route can be obtained at the destination. If the error rates of some road segments density exceed the threshold, e.g. 30%, the sink node sends an acknowledge message to notify the source about the updated density of that road. Next, the source node re-computes a new route based on the recently received and more accurate density data. Eventually, the selected route will converge to an optimal route.

6.4 On-The-Fly Density Collection

As stated above, the on-the-fly density collection process is done while data packets are being forwarded. Before transmitting data packets, every forwarder adds into the packets its local density information, which is obtained through collecting beaconing messages. Then, the total density of a road segment can be obtained at the end of it. When packets reach the destination, the density data for all road segments along the path are collected.

As shown in Fig. 6.2, the data packet is composed of two parts: packet header and data payload. At the beginning of data payload, there are some reserved fields (bytes) for on-the-fly density collections. The first byte records how many road segments (e.g. N_r) on selected route,

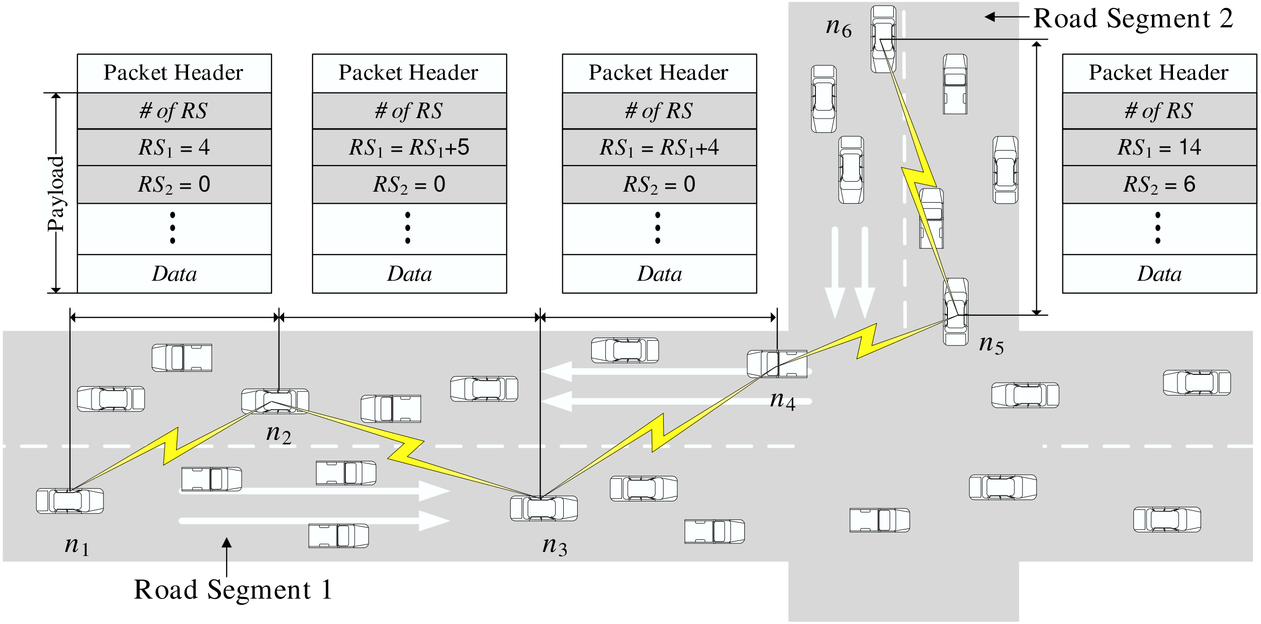


Figure 6.2: On-the-fly Density Collection Mechanism

and subsequent N_r bytes record the density information of every road segment on the route. The initial values of these fields are 0. Since the source node is able to compute entire route based on historical density data from digital maps, it is easy to get the number of road segments on the route.

We now state how a packet forwarder collects its local density information and updates the corresponding byte in data packets. Since every node periodically beacons its location, velocity and id to its neighbors, a node obtains the number of its one-hop neighbors. In addition, with the neighbor's location information, a node can determine whether a neighbor is in front of or behind it. For example, node n_2 in Fig. 6.2 infers that 4 nodes (including n_1) are in front of it and 5 nodes behind. Suppose node n_1 is the current packet forwarder which is at the beginning of road segment 1, and n_2 is the next hop. Before n_1 sends data packets, it adds the number of nodes between itself and n_2 (including itself) to the field RS_1 and forwards them to n_2. Then, n_2 follows the same strategy and sends packets to n_3. Node n_3 modifies RS_1 again by add its collected local density information, and sends out packets. Finally, packets reach the end of this road segment at node n_4.

Node n_4 needs to decide if its next hop is still on the same road segment. If so, it continues the procedure as node n_3 did. Otherwise, it adds 1 to RS_1 because itself is also on the road segment 1, and forward packets to its next hop, e.g. n_5 in Fig. 6.2. Consequently, node n_5 adds 6 to RS_2 and forward packets to n_6. In the same way, when packets reach the destination, density of every road segment on the route is collected.

After the on-the-fly density collection, the destination node needs to notify the source if there are significant discrepancies between statistical and real-time density data. If so, source node recalculates routes with newly collected density information; otherwise, the same route will be used for delivering future packets.

6.5 Next Hop Selection

On each road segment in the selected route, packets may be forwarded through multiple hops from the beginning to the end of the road segment. The next hop will be selected using a metric that minimizes the PER of route on each road segment. The PER of a link between two nodes can be calculated by counting the number of successfully delivered packets and dropped ones. This is calculated during the beaconing period and thus does not incur additional network overhead.

The original geographic routing protocols [10, 11] choose the farthest node as the next hop, since this selection can minimize the total number of hops to the destination. However, the link quality to the farthest node is usually weak because PER increases as the transmission distance increases. However, selecting next hop with a shorter distance will increase the number of hops. As proven in [59], the data delivery ratio will decrease as the hop number increases. So there is a trade-off between shorter transmission distance and smaller number of hops.

To address this issue, every node needs to measure the packet error rate of all its neighbors. Suppose on a road segment there are two neighboring node n_a and n_b, and they periodically send their locations to each other. By counting the number of packets successfully delivered and dropped, the expected transmission count (ETX) can be calculated using the approach in [60]. Then the PER from n_a to n_b is obtained as:

$$PER_{ab} = 1 - \frac{1}{ETX_{ab}} \tag{6.1}$$

where ETX_{ab} is the expected transmission count from node n_a to n_b. In the same way, PER_{ba} can be computed. Since the route is already known (stored in the packet header), node n_a then computes the remaining distance (denoted as Dis) from itself to the next intersection. Suppose the distance between node n_a and n_b is d, then the PER of the remaining route on this road segment can be estimated by:

$$PER = 1 - (1 - PER_{ab})^{[\frac{Dis}{d}]} \tag{6.2}$$

We assume different parts of the same road segment have the similar communication environment, thus the distance between nodes will be the dominant factor that affects the data delivery ratio. So among its neighbors, node n_a selects the one that minimizes the PER of the remaining path as the next hop. The same next hop selection will be done on all following road segments aiming to achieve the highest data delivery ratio along the whole route. However, due to frequent network partitions in VANETs, a data forwarder may have no neighbors in the forward direction. In these cases, we adopt the carry and forward scheme [16] that buffers packets and waits until there exists an available next hop. Then the packet will be fetched from the buffer and forwarded again.

Chapter 7

Simulations and Results

7.1 Mobility of Nodes

Since modeling of complex vehicle movement is important for accurately evaluating protocols, we generated the movement of nodes using VanetMobiSim [41] whose mobility patterns have been validated against TSIS-CORSIM, a well-known and validated traffic generator. The VanetMobiSim features new realistic automotive motion models at both macroscopic and microscopic levels, and also supports traffic lights, lane changes and speed regulations.

We compared the network connectivity model with data collected through VanetMobiSim simulations for a set of parameters: length of road segment, average vehicle velocity and traffic light period. Specifically, as shown in Fig.7.1, there are 7 road segments (each is $1000m$) in the map, the average velocity of vehicles is $10m/s$ and the traffic light period is 120 seconds. Those small squares denote vehicles moving on the road, the number besides them are the node IDs. Our goal is to collect the network connectivity and density information on the middle road segment ending with two traffic lights. The simulation time is 2000 seconds and we check every second if the network is connected. The number of times that networks are connected is denoted as t_c and the probability of network connectivity can be calculated as $t_c/2000$. Similarly, the average network density can be collected, though it may not be an integer. We repeated the same scenario 10 times with 10 different random seeds to achieve a high confidence level. As shown in Fig.7.2-Fig.7.7, with different road lengths, velocities and traffic light periods, the connectivity model matches the value obtained from VanetMobiSim very well (confidence level is 95%).

In the above simulations, there is only one road segment containing two lanes in each driving direction. We also verified the connectivity model in the cases of more lanes (e.g. 3-5 lanes), one traffic light at the end of a road segment and routes that consist of multiple road segments. The

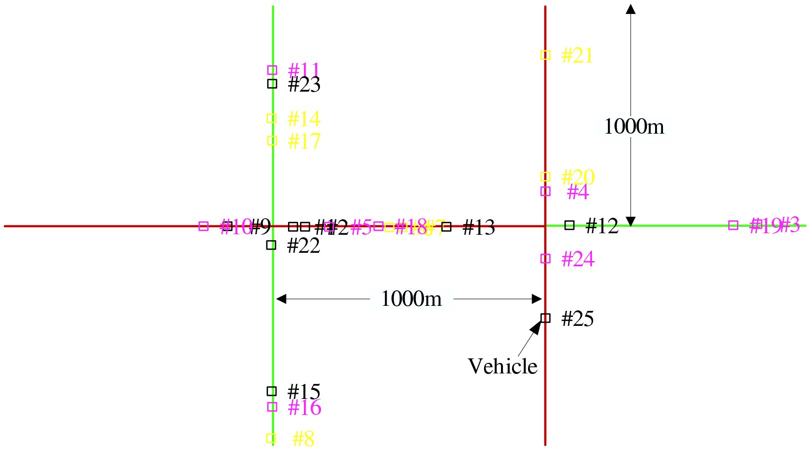


Figure 7.1: A VanetMobiSim Snapshot of Connectivity Model Validations

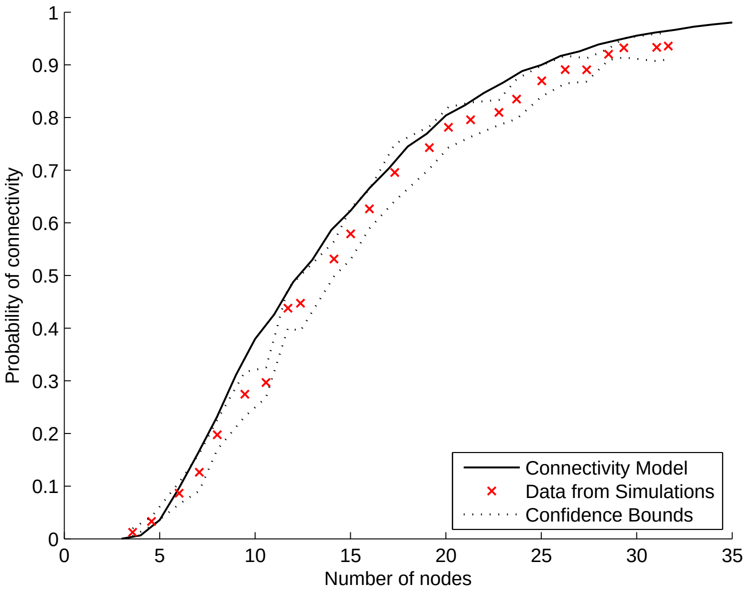


Figure 7.2: Validation of Connectivity Model with Road = 1000m, Velocity = 5m/s and Traffic light = 120s

results showed our connectivity model matched the simulation results very well. However, due to space limitation those results are omitted in this work.

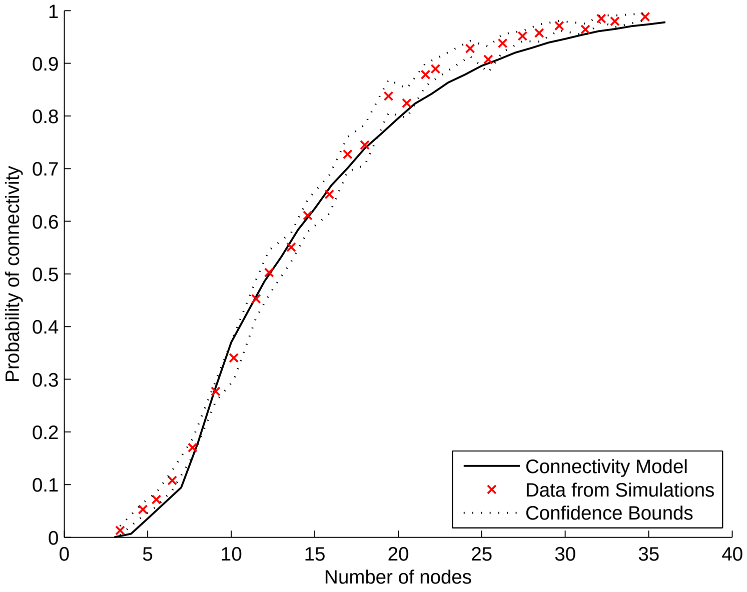


Figure 7.3: Validation of Connectivity Model with Road = 1000m, Velocity = 7.5m/s and Traffic Light = 60s

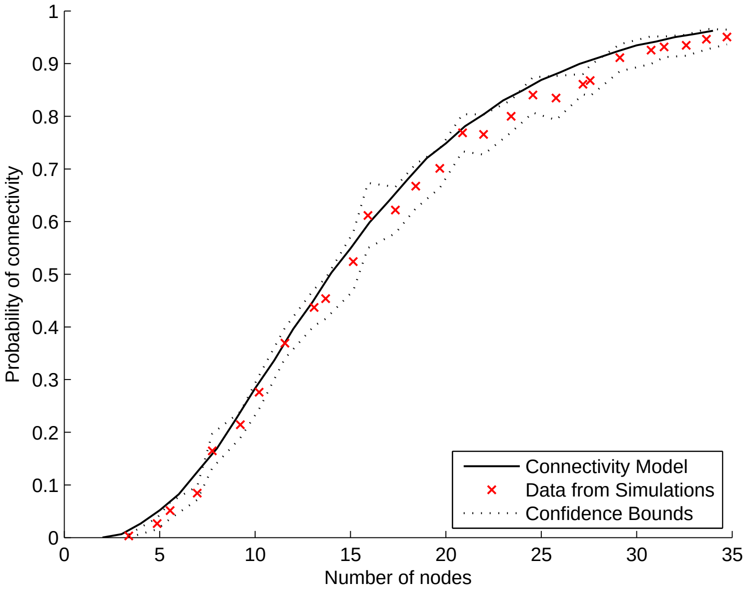


Figure 7.4: Validation of Connectivity Model with Road = 1000m, Velocity = 7.5m/s and Traffic Light = 120s

7.2 Digital Maps

We used two maps in simulations to show the high performance of ACAR, and how different network densities and vehicles velocities affect this protocol.

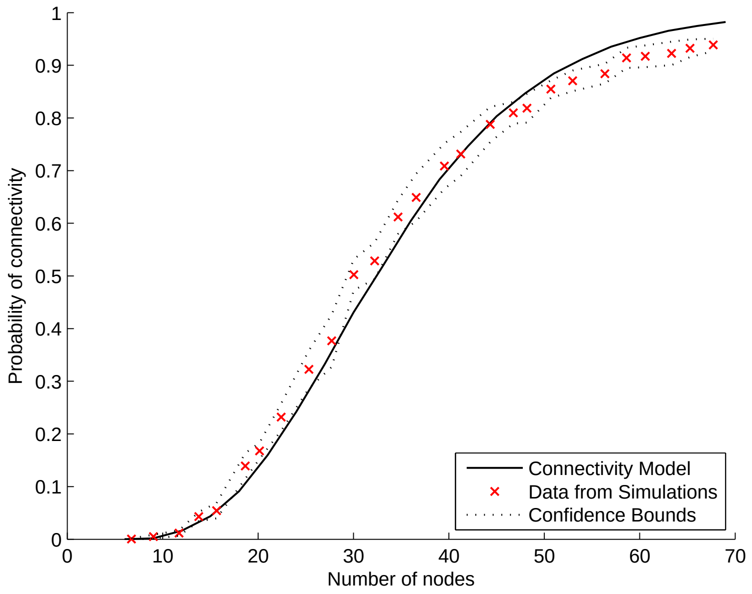


Figure 7.5: Validation of Connectivity Model with Road = 1800m, Velocity = 10m/s and Traffic Light = 60s

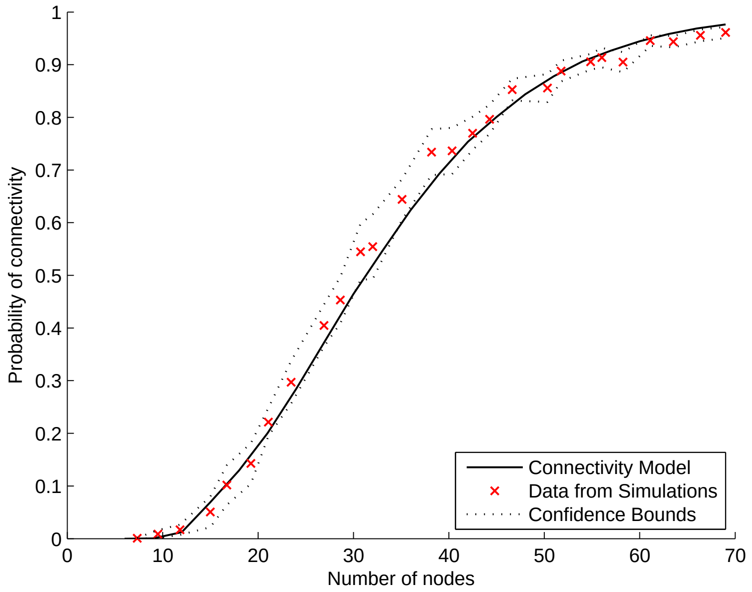


Figure 7.6: Validation of Connectivity Model with Road = 1800m, Velocity = 7.5m/s and Traffic Light = 60s

One map is illustrated in Fig 3.1, which contains 5 major road segments: I_AI_B, I_AI_C, I_AI_D, I_BI_C and I_CI_D. The length of each road segment and number of nodes deployed on them are the

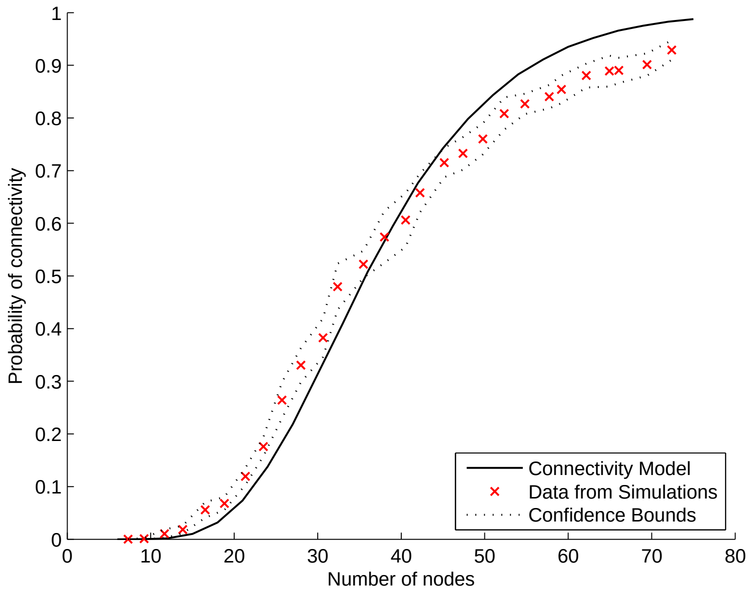


Figure 7.7: Validation of Connectivity Model with Road = 1800m, Velocity = 10m/s and Traffic Light = 120s

same as we described in Chapter 3. This map is used in Scenario I in which we evaluate the basic network performance of ACAR such as: data delivery ratio, end-to-end delay and throughput.

In the Scenario II, we load a real map topology from the from the topologically integrated geographic encoding and referencing (TIGER) database, which is used by the United States census bureau to describe land attributes of U.S.. Within a $1000m \times 1000m$ area, street layout is from a city in Tennessee, centered at latitude 35162102 and longitude -84877562, has 15 intersections and 38 road segments as shown in Fig.7.8. To evaluate how different network densities affect network performance, we double the network density of several roads which are marked with bold black lines in Fig.7.8. We also adjust the velocity of vehicles in the network and evaluate how network mobility affects the routing performance.

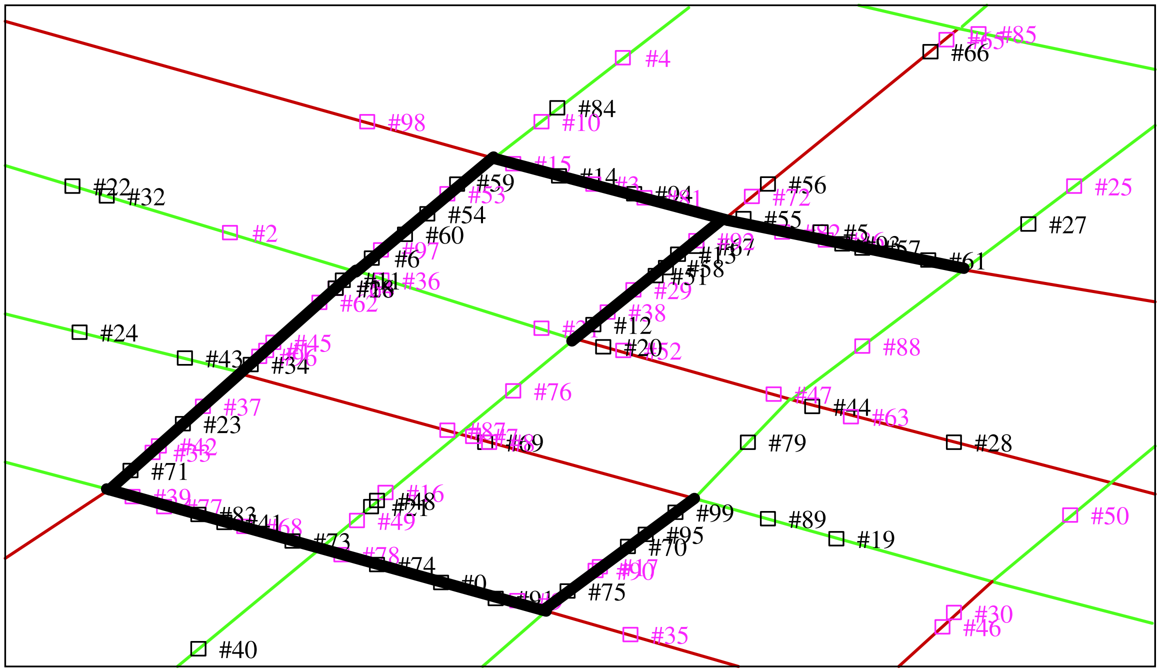


Figure 7.8: Street Layout of the Area Centered at $(35162102, -84877562)$ in the Tennessee State

7.3 Networking Simulation

We simulated the ACAR protocol in NS2 (ns-2.33) and compared it with VADD [13], MOVE [17], GPSR* and GSR*. The original GPSR [10] and GSR [11] simply drop packets when network disconnections occur, so we add carry-and-forward schemes in them and named them as GPSR* and GSR*, respectively. To make fair comparisons between ACAR and other trajectory based routing protocols, we also implemented the neighbor location predication scheme on VADD and GSR*.

Because the proper PHY/MAC modules for vehicular communications are still under development and not available for NS2, we adopt the channel fading model proposed in [56] and IEEE 802.11a as the MAC/PHY protocol. Since we are interested in network performances of different protocols, we omit the exact simulation of lower layers but consider it in our future work when IEEE standards for vehicular communication are finalized. Details of simulation parameters are listed in Table 7.1.

We first simulated the Scenario I in which the basic network performances are evaluated such as the data delivery ratio, end-to-end delay and network throughput. Then, we simulated

Table 7.1: Simulation Set-Up Parameters for ACAR

Parameter	**Value**
Number of lanes	2 lanes per direction
Number of nodes	40-200
Velocity	10-90 miles/hour
Period of traffic lights	60 seconds
Communication range	250 m
Beacon interval	1.0 second
Buffer size	64 KB
Packet size	512 Bytes

the Scenario II with different network densities and vehicles' velocities and evaluated the network performances and network overheads of the proposed protocol. In each scenario, different data sending rates (1 to 10 pkts/s) were used. A source node is randomly selected to communicate with a fixed destination. Given a real-time location service, ACAR works well if the destination is mobile. However, we considered a fixed destination to model applications described in Chapter 1. The simulation time is 2000 seconds and each scenario is repeated 20 times to achieve results with a high level of confidence.

7.4 Data Delivery Ratio

Data delivery ratio is the number of received packets at the destination divided by the total number of packets sent into networks. As shown in Fig. 7.9, ACAR achieves the highest data delivery ratio (above 90%). This is because ACAR forwards packets along route on road $I_AI_BI_C$ with the highest connectivity-quality.

As shown in Fig. 7.9, GPSR* and GPSR give the second and third highest data delivery ratios, respectively. When network partitions occur, GPSR and GPSR* utilize perimeter mode searches to find routes, so packets may finally delivered on road $I_AI_BI_C$ which has the highest connectivity-quality. However, GPSR* only successfully delivered about half of packets compared to the performance of ACAR. This is because, after packets are forwarded on road I_AI_C or I_AI_D,

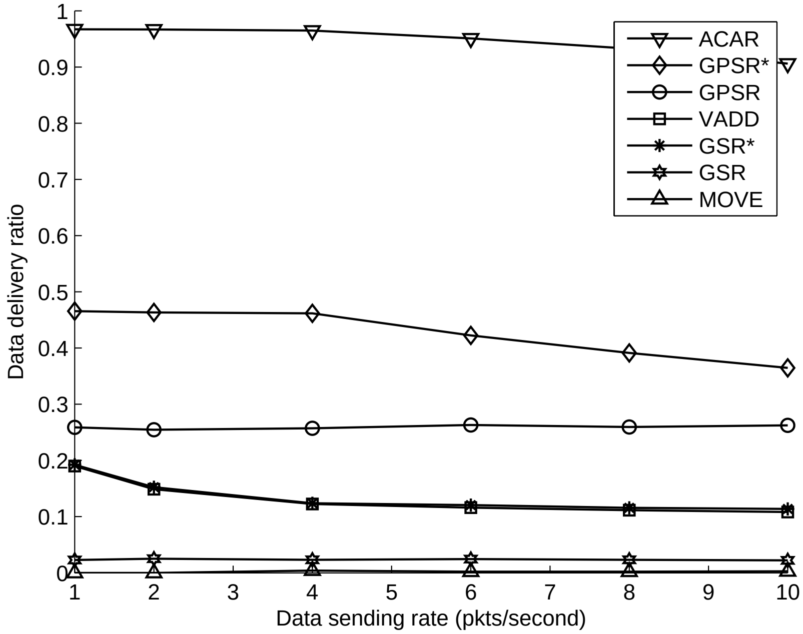


Figure 7.9: Data Delivery Ratio in the Scenario I

it is possible that there are no connected links back to road I_AI_B. So these packets are buffered and carried by nodes moving on road I_AI_C or I_AI_D. On the other hand, wireless transmission qualities of these two roads are very bad, so the data delivery ratio of GPSR* forwarding packets along them is very low. Since we implemented the carry-and-forward scheme on GSPR*, it delivered $10 - 20$% more packets than GPSR. So we conclude that the carry-and-forward scheme is very helpful for routing protocols in VANET to achieve high data delivery ratios.

GSR* selects road I_AI_C to forward packets, as it is the geographic shortest path to the destination. According to the connectivity model in VADD, path I_AI_C provides the shortest delivery delay, so it is chosen to route packets. However, the connectivity probability of this road is just $.29$, and the wireless transmission quality is even lower. Therefore, the overall data delivery ratio of packets being routed on this road is very low. Since GSR* and VADD choose the same path for routing, they generate very similar data delivery ratio results.

The original GSR protocol gives a lower data delivery ratio (only $.02$), compared to the extended version GSR*. This is because on GSR*, we implemented NLP and carry-and-forward mechanisms. The NLP scheme can help nodes to correctly select the next hop and the carry-and-forward scheme can avoid packet loss due to network partitions. The data delivery ratio of GSR* is about $5 - 10$ times that of GSR. Therefore, we conclude the NLP mechanism is also necessary for VANET routing protocols to achieve high data delivery ratios.

MOVE protocol delivered the least number of packets in our simulations. In MOVE, there are seven forwarding rules being used to select the next hop. If none of neighbors satisfies these forwarding rules, packets will be carried by current node. So packets are more likely to be buffered and carried by vehicles instead of being greedily sent out. As we will describe later, these packets may be dropped due to packets expiration, weak wireless links to next hops and buffer overflows. The number of packet loss due to these reasons is very high for MOVE, so it gives the lowest data delivery ratio compared to others.

7.5 Reasons of Packet Loss

There are mainly three reasons of packet loss for all VANET protocols: packets expired, weak wireless links and buffer overflow. We measured the number of lost packets due to each reason, and then find the major cause of packet loss for each protocol.

7.5.1 Expired Packets

Since we cannot run simulations an infinite number of times, when simulations are terminated, there might be some packets, called expired packets, still in buffers and these packets will be dropped due to their huge delays. As shown in Fig. 7.10, the fraction of expired packets of MOVE is almost 5-6 times that of the others. However, ACAR, VADD, GSR* and GPSR* have the similar number of expired packets. The reason is that, in ACAR, VADD, GSR* and GPSR* protocols, packets are greedily forwarded to the next hop; but in MOVE, if the neighbor satisfying none of the forwarding rules (totally 7 rules), packets will be carried by the current node. Therefore, packets will be more likely buffered in MOVE than the others. However, due to the small number of expired packets, we conclude that packet expiration is not the dominant reason for packet loss.

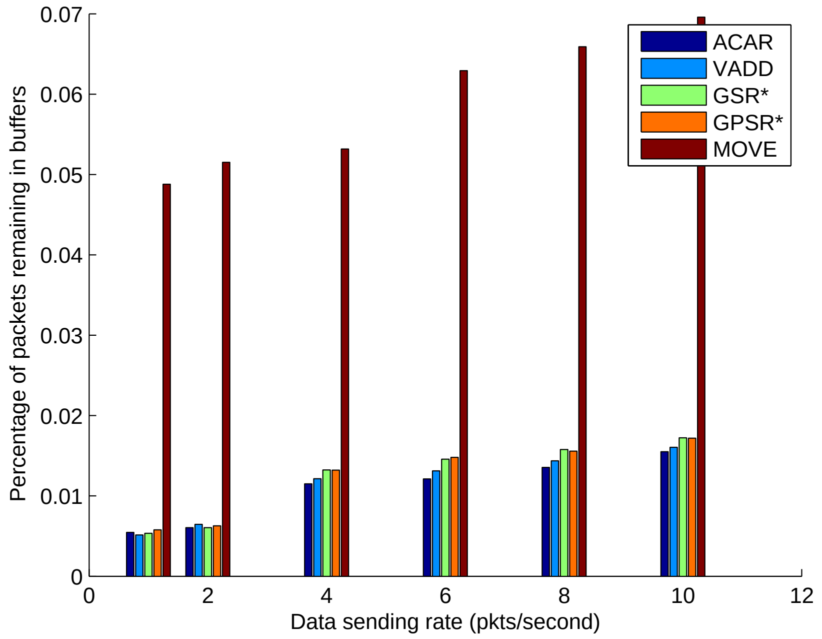


Figure 7.10: Fraction of Packets Still in Buffers After Terminating Simulations

7.5.2 Wireless Transmission Loss

Packet loss can also be caused by weak wireless links to next-hop nodes, e.g. the next hop is too far away or even out of the communication range of current packet forwarder. As shown in Fig. 7.11, the number of this type of packet loss is much higher than that of expired packets. In Fig. 7.11, we note the original GSR has about 95% packets dropped due to this reason. GSR chose nodes on road I_AI_C to forward packets, but the probability of network connectivity on this road was so low that most packets were dropped because there were no available next hops. The original GPSR also suffers this problem because not all packets can be routed along road $I_AI_BI_C$, i.e. some packets were dropped on road I_AI_C or I_AI_D before they were forwarded back to $I_AI_BI_C$ through perimeter searches. However, GPSR* can reduce this kind of packet loss. Because if there is no available next hop, packets are not simply dropped but buffered and sent until another next hop occurs. Since GPSR* does not have the NLP mechanism, most packets dropping in GPSR* is caused by the problem of out-of-date neighbors.

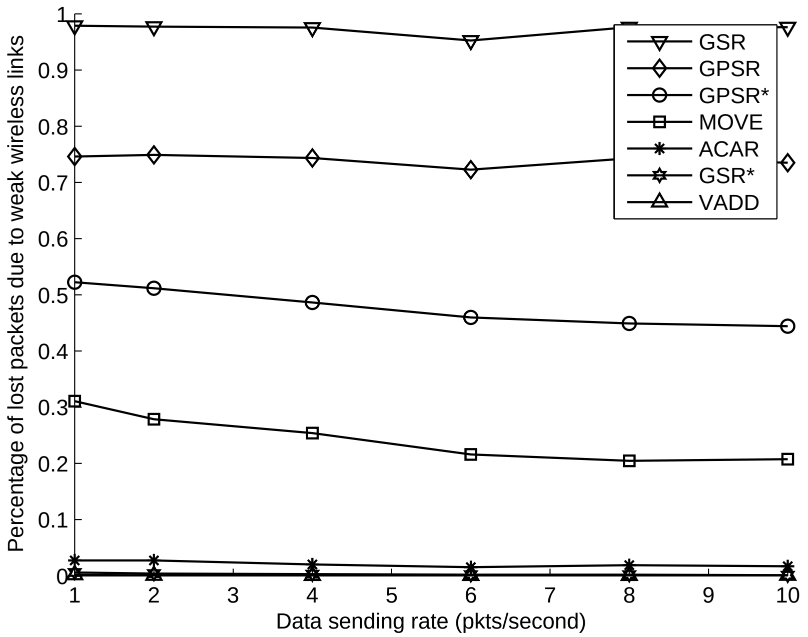


Figure 7.11: Fraction of Dropped Packets Due to Weak Wireless Links

MOVE gives a fewer packet losses in this case because most packets are buffered instead of being greedily sent out. Since NLP mechanism is implemented on both VADD and GSR*, they have fewer packets dropped for this reason. For ACAR, besides NLP, it will carefully select every next hop; therefore, compared with others, it gives the lowest packet loss due to weak wireless links.

In summary, we can conclude that weak wireless link is the major reason of packet loss for GSR, GPSR and GPSR*.

7.5.3 Buffer Overflow

Another reason of packet loss in networks is: the buffer may be overflowed so that all incoming packets have to be dumbed as there is no more space for them. Fig. 7.12 presents the percentage of lost packets due to this problem. As shown in the figure, VADD and GSR dropped more than 70% packets due to this reason. Therefore, if the size of buffer is large enough so that all packets can be buffered and carried by vehicles, VADD and GSR can give a similar data delivery

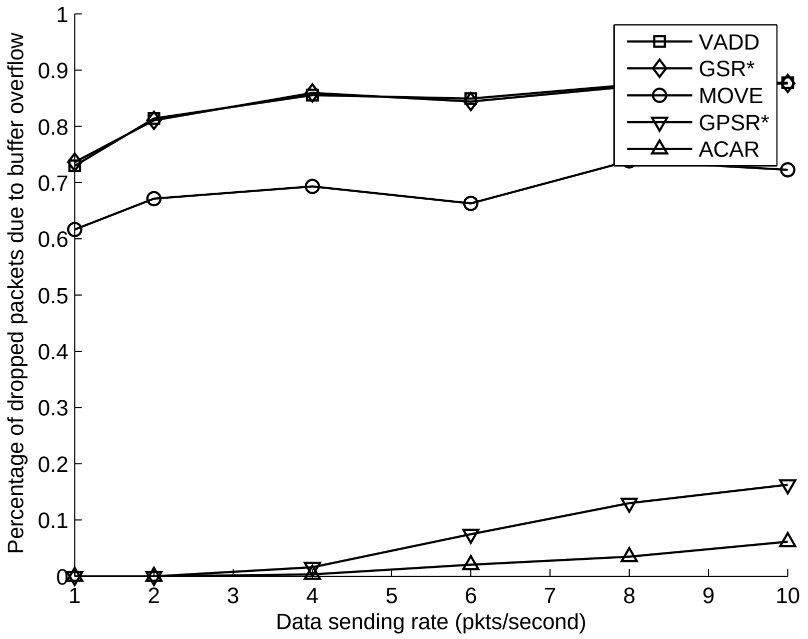


Figure 7.12: Fraction of Dropped Packets Due to Buffer Overflow

ratio as that of ACAR. In other words, ACAR has a lower requirement of the capacity of buffer on vehicles to achieve a high data delivery ratio. As we mentioned before, because of the 7 rules of selecting next hop, most packets will be buffered by MOVE. So we can see from Fig. 7.12, more than 60% packets were dropped in MOVE because it has already buffered too many packets and had no space in buffer for those packets.

Comparing with the packet loss caused by weak wireless link, buffer overflow problem is not a significant issue for GPSR*; but is a significant one for ACAR. Therefore, we conclude buffer overflow is the major reason of packet loss for GSR*, VADD, MOVE and ACAR.

7.6 End-to-End Delay

The end-to-end delay is defined as the average time taken for a packet to be transmitted across networks from source to destination. As shown in Fig. 7.13, MOVE gives the largest end-to-end delay, which is mainly because of the long time vehicles carry packets. There are 7 forwarding rules in MOVE which determine if packets are transmitted from current node to the next hop.

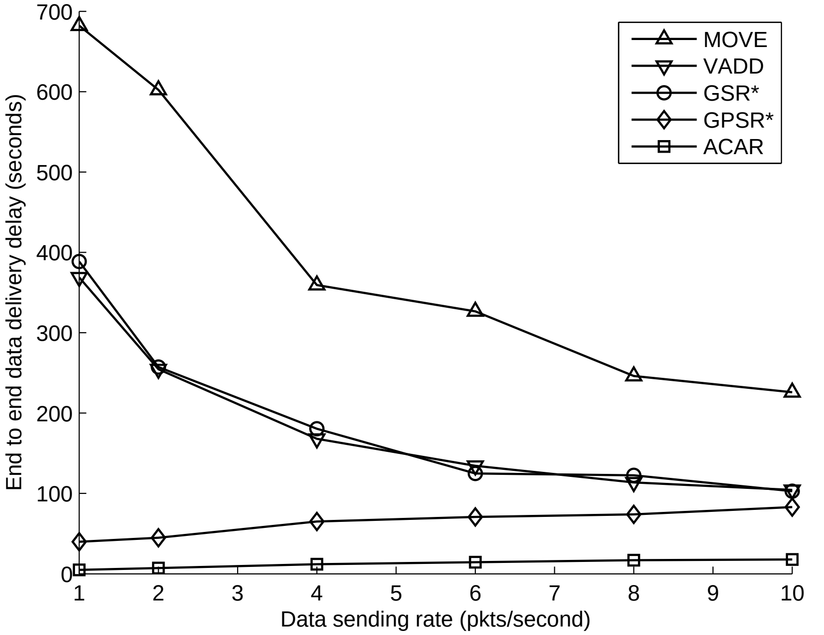


Figure 7.13: End-to-end Network Delay in the Scenario I

Even though a neighbor is closer to the destination, it may not satisfy the forwarding rules and thus cannot relay packets. Therefore, more packets will be put into the buffers and that results in a larger delivery delay since the velocity of vehicles is much lower than the wireless transmission speed.

VADD and GSR* give a similar end-to-end delay because they select the same road segment I_AI_C (in Fig. 3.1) to forward packets. However, the probability of network connectivity on this road is very low; thus, most packets are buffered as there is no next hop available. Since the velocity of vehicles is much slower than the speed of wireless transmission, VADD and GSR* generate a larger delay compared to ACAR and GPSR* which use connected routes on $I_AI_BI_C$ to forward packets.

An interesting observation is that when the data sending rate increases from 1 to 10 pkts/s, the end-to-end delay of MOVE decreases from about 700 to 260 seconds, and VADD or GSR* decreases from about 400 to 100 seconds. The reason of this huge reduction is: when the data sending rate is increased, more packets will be forwarded without being buffered. For example, suppose the network on $I_A I_C$ is disconnected during $[0.0, .5)$ seconds and connected within $[.5, 1.0]$ seconds. When the sending rate is 1 pkt/s, the first packet will be buffered. However, if the sending rate is 10 pkts/s, the last 5 packets are delivered without being buffered. Therefore, the average end-to-end delay of 10 pkts/s sending rate will be lower than that of 1 pkt/s case.

ACAR gives the lowest end-to-end delay, since packets are forwarded along the path $I_A I_B I_C$. There are 22 nodes on this road segment, so the probability its network connectivity is very high $(.84)$. That also means most packets are delivered to the destination without being buffered, and thus ACAR saves the total time of delivering packets from the source to destination. GPSR* generates a higher end-to-end delay than that of ACAR because some packets are forwarded to road $I_A I_C$ and $I_A I_D$, then they are detoured to road $I_A I_B I_C$ by perimeter searches. So the longer delay of GPSR* is caused by the longer route. However, it still gives a lower delay compared to VADD, GSR* and MOVE.

Unlike VADD, GSR* and MOVE, the end-to-end delay of ACAR and GPSR* increases as the data sending rate increases, due to two reasons. Firstly, when more packets are injected into the networks, the probability of packet collision is larger and thus the transmission delay increases. Secondly, higher network traffic increases the queuing time on each forwarder (vehicle) and also the end-to-end delay.

7.7 Network Throughput

We compared the throughput of MOVE, GPSR*, GSR*, VADD and ACAR in the network shown in Fig. 3.1. Results in Fig. 7.14 show that ACAR outperforms all the other protocols, i.e. it achieves the highest network throughput of 84 kb/s. This value is about three times that of GPSR* which is second best protocol. Since packets are forwarded along routes with the highest connectivity-qualities in ACAR, link quality per hop is higher than that of others. Therefore, the data delivery ratio and end-to-end delay can be improved. We also note the shapes of GPSR and GPSR* results are very similar to that of ACAR because both GPSR and GPSR* delivered most packets along the route on $I_A I_B I_C$, which is the route chosen by ACAR too.

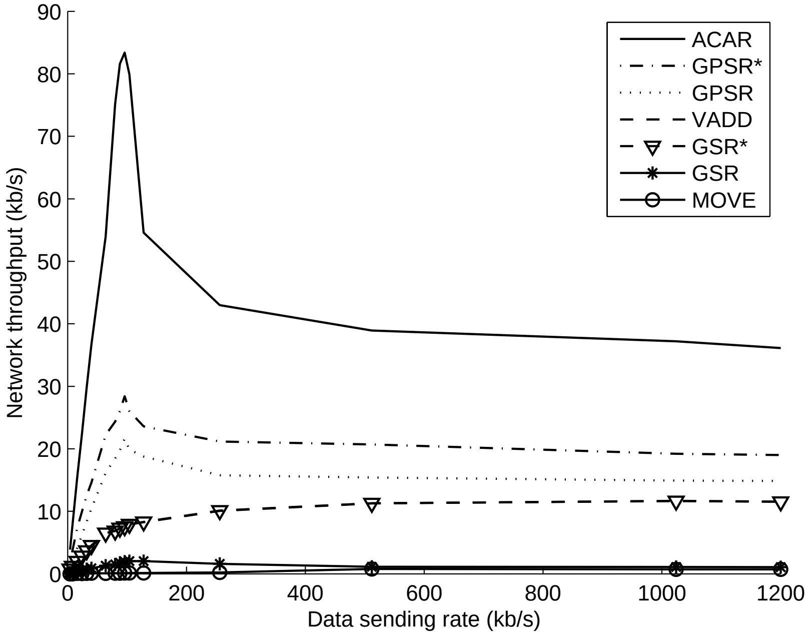


Figure 7.14: Network Throughput in the Scenario I

VADD and GSR* give the similar throughput as the data sending rate increases because they all chosen routes on the same road segment (I_AI_C) to deliver packets. Since the probability of connectivity of road I_AI_C is low, the throughput of VADD and GSR* is lower than that of ACAR, GPSR* and GPSR.

An interesting observation of VADD and GSR* is that their throughput increase when the data sending rate increases from 1 kb/s to 500 kb/s and become stable after that. This is quite different from ACAR and GPSR*, whose throughput decrease after reaching the peak values. As mentioned previously, when the data sending rate increases, the chance of packets being delivered increases and so does the network throughput. However, if the data sending rate is so high that buffer overflows occur on nodes, the larger data sending rate is not helpful for network throughput. At this point, every node will periodically send out one packet from its buffer when a next hop is available. Since the time interval for periodic buffer checking is a fixed value, the network throughput becomes stable in this situation.

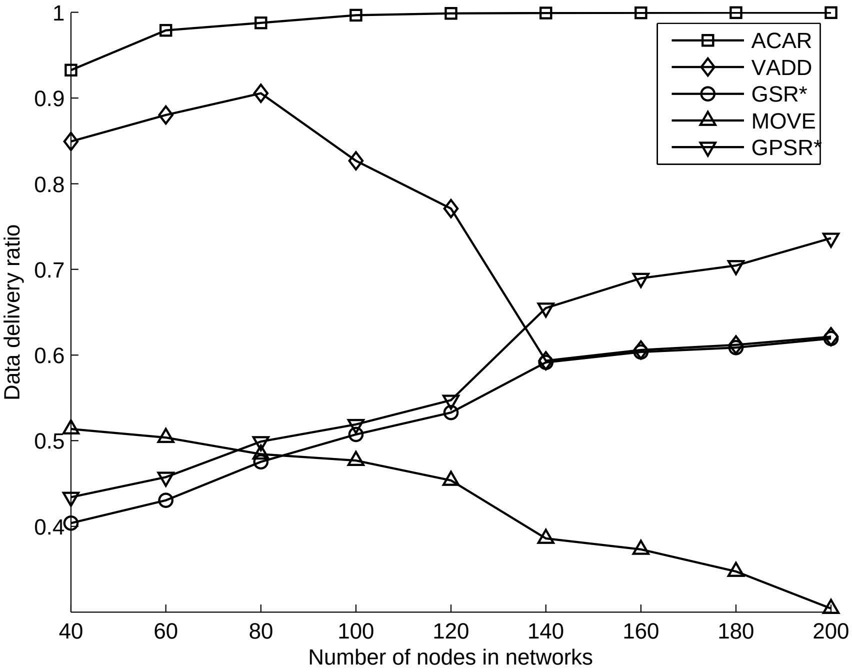


Figure 7.15: Data Delivery Ratio vs Number of Nodes in the Scenario II

GSR gives the second lowest throughput performance because packets will be simply dropped when it faces network disconnections which is very common on road I_AI_C. MOVE will choose nodes on I_AI_D and I_CI_D to forward packets, so the overall network throughput will be very low due to low network connectivity and longer delivery path.

7.8 Impact of Network Density

To evaluate the network performance of ACAR in a more general case, we simulated networks in the second map with data from the U.S. TIGER database. We evaluate how different network densities affect the network performance, in terms of data delivery ratio and end-to-end delay. In reality, vehicles are not evenly distributed on roads, so we manually deploy more vehicles (70% of the total number) on certain roads which are highlighted by bold lines in Fig. 7.8, and fewer nodes (30%) on the others. The total number of nodes in networks varies from 40 to 200.

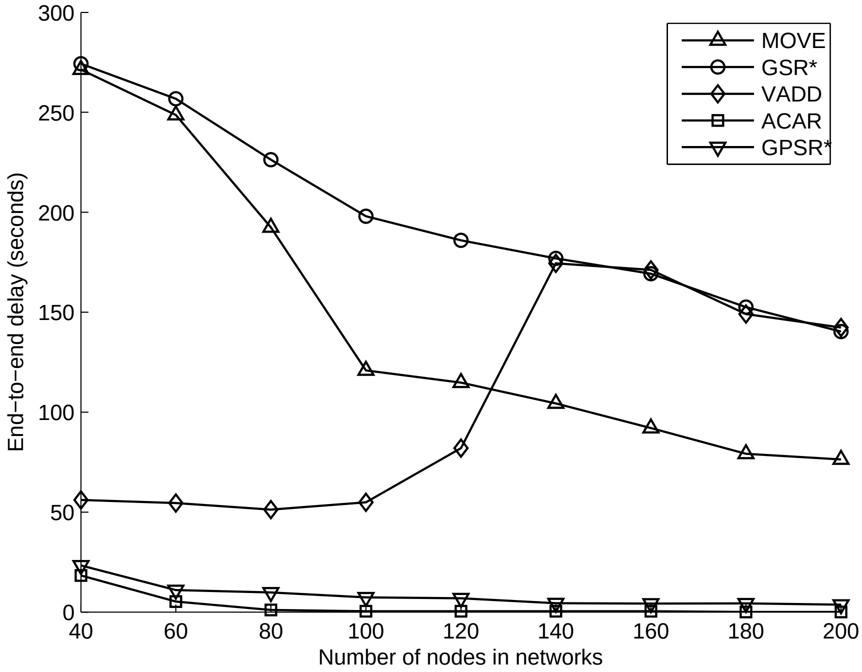


Figure 7.16: End-to-end Delay vs Number of Nodes in the Scenario II

Since vehicles can only move on roads instead of the entire simulation area, we define *network density* as the ratio between number of nodes and the total length of all road segments. The total length of roads in the map is 7878m, so the network density varies from $1/197$ to $1/40$ nodes per meter.

7.8.1 Data Delivery Ratio with Different Network Densities

As shown in Fig. 7.15, except for VADD and MOVE, all protocols deliver more packets as the network density increases. This is because when the number of nodes increases, the expected network connectivity probability increases too and so does the data delivery ratio. From Fig. 7.15, we note that when the network density is low, 40 to 120 nodes, GPSR* and GSR* give similar data delivery ratios. That means the perimeter search in GPSR* cannot drastically improve the data delivery ratio when network density is low, but it does help to reduce the end-to-end delay as shown in Fig. 7.16. However, when the number of nodes is larger than 120, the network connectivity probability increases. Then, it is more likely for GPSR* to find a connected path instead of forwarding packets on the geographic shortest path. Therefore, it delivered more packets than GSR* which still forwards packets on the geographic shortest road segments.

In the MOVE protocol, the larger the network density, the higher the probability of Ping-Pong situation occurring (as described in Chapter 3). So the delivery ratio of MOVE is reduced. For VADD, its data delivery ratio increases when network density is low ($40 - 80$ nodes), decreases when network density is medium ($80 - 140$ nodes), and slightly increases when network density is large ($140 - 200$ nodes). When the network density is low, VADD considers no connected network on most road segments, and will forward packets along the path with higher probability of connectivity, e.g. roads marked by bold lines in Fig. 7.8. Thus, the delivery ratio will increase when more nodes are deployed in networks. However, when the network density becomes larger, VADD may find there are some connected networks on other roads which are closer to the destination. Then, VADD will forward packets along those roads. Due to the limitation of connectivity model in VADD, probabilities of connectivity of these roads are actually very low. So the data delivery ratio decreases until these roads are really connected (140 nodes). After that, the data delivery ratio of VADD is similar to that of GSR* because both of them will choose the geographic shortest path to forward packets. The delivery ratio slightly increases when more nodes are deployed on the shortest path.

7.8.2 Network Delay with Different Network Densities

The end-to-end delay of all protocols, except for VADD, drops when network density increases. This is because network connectivity probability increases when more nodes are deployed in networks. As shown in Fig. 7.16, ACAR and GPSR* give the lowest and second lowest end-to-end delay, respectively. Since ACAR forwards packets along routes with the highest connectivity-qualities, the number of buffered packets (during network disconnections) is less than that of GPSR*, resulting in a lower delay. On the other hand, when network disconnections occur, GPSR* in the perimeter mode can search for another connected path (e.g. the path used by ACAR), so it also generates a small delay compared to others.

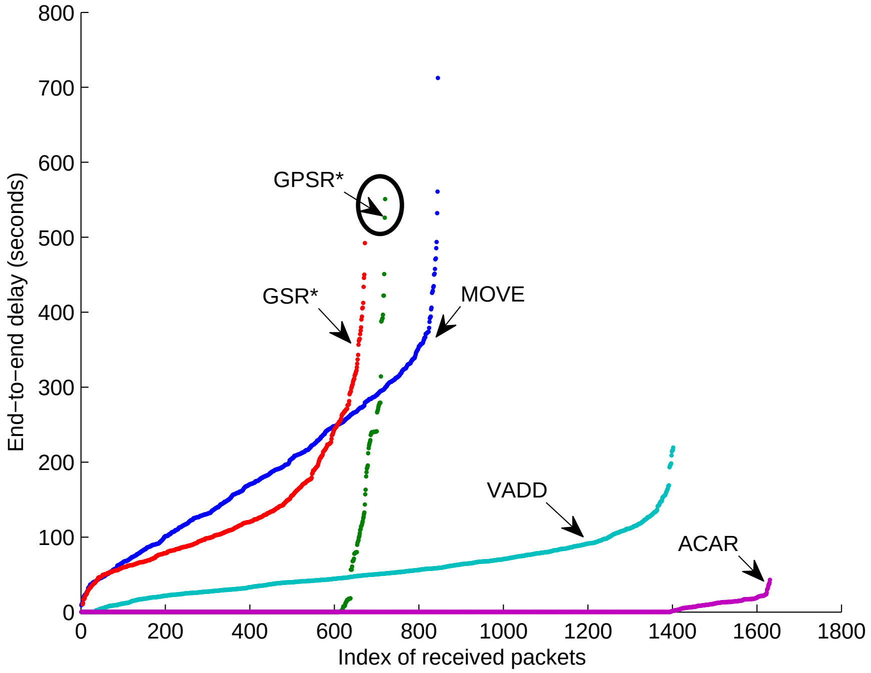


Figure 7.17: Delay Distribution of Received Packets with 40 Nodes in the Networks

GSR* only forwards packets along pre-defined routes, i.e. the geographic shortest path, so it has no opportunity to find a better connected path as GPSR* does. Therefore, it gives a much higher end-to-end delay. As mentioned above, in MOVE, nodes will carry more packets in their buffers and this will reduce the data delivery rate. Thus, MOVE gives a very high end-to-end delay.

An interesting observation of VADD's end-to-end delay is: it decreases from 40 to 80 nodes, and increases from 80 to 140 nodes, and decreases again from 140 to 200 nodes. The reason is similar to that of the delivery ratio results: with the connectivity model used in VADD, some disconnected paths are considered connected and selected as routes to forward packets. Along those frequently-disconnected paths, packets are frequently buffered so the average end-to-end delay of VADD is higher than GPSR* and ACAR.

7.8.3 Delay Distributions of Different Protocols

As the end-to-end delays of ACAR and GPSR* are very similar, we further investigate the delay distribution of delivered packets. For example, when there are 40 nodes in networks, the delay

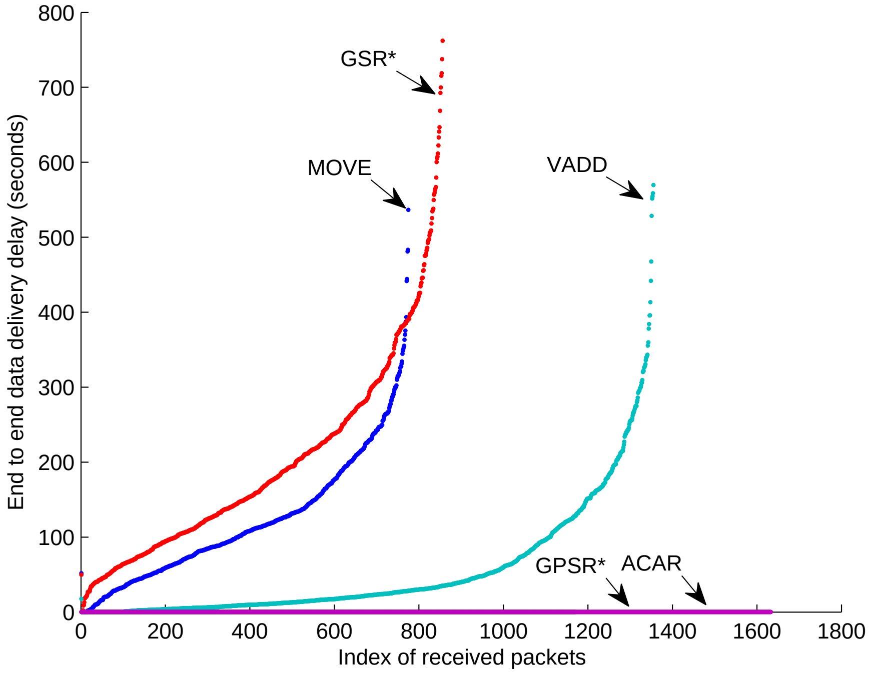


Figure 7.18: Delay Distribution of Received Packets with 100 Nodes in the Networks

distribution of received packets for all protocols is shown in Fig. 7.17. The x-axis denotes indices of the received packets, and y-axis for end-to-end delays which are measured in seconds. We order received packets by their end-to-end delays. Dots denote the end-to-end delays of corresponding packets.

As we can see, ACAR delivers most packets with smaller delays; while in GPSR*, some delivered packets have very large delays (a few hundred seconds). In addition, although GPSR* and GSR* deliver similar numbers of packets, GPSR* definitely routes packets along faster but longer paths than those used by GSR*. Some packets (1st to 600th) in GPSR* are delivered successfully along connected paths, while others (after 600th) are buffered and carried by vehicles. Since paths selected by GPSR* are longer, delays of some packets circled in Fig. 7.17 are even larger than those of GSR*. However, this situation changed when the network density is increased to 100 nodes, as shown in Figure 7.18.

With larger network density, GPSR* can deliver more packets with large delay to the destination. However, no matter what the network density is, given a certain delay value, ACAR delivered

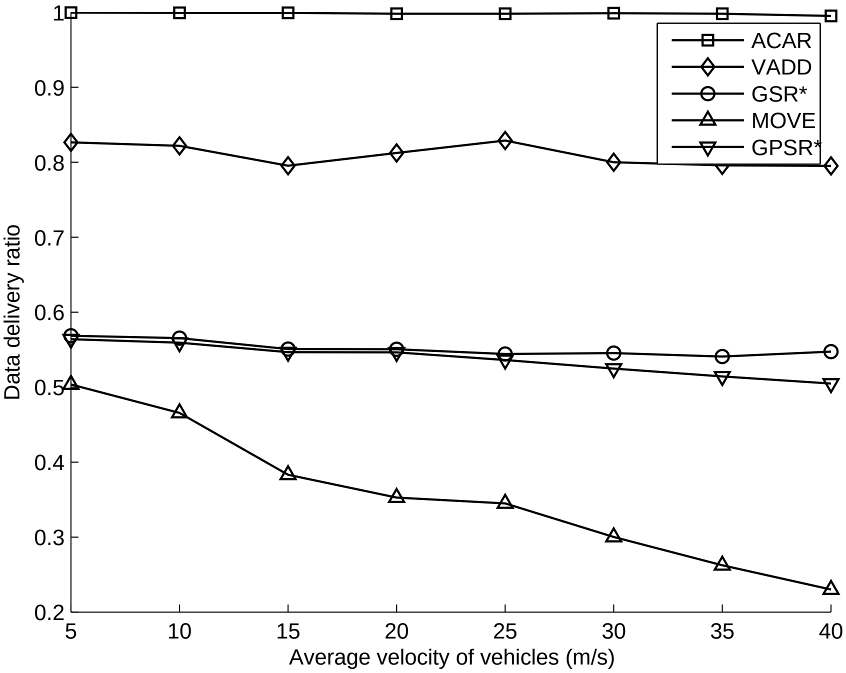


Figure 7.19: Data Delivery Ratio vs Different Velocities with 100 Nodes in the Networks

more packets than any of the others. In summary, we conclude that ACAR not only gives the lowest average delay but also delivers more packets with smaller delays compared to other protocols.

7.9 Impact of Velocity

Since mobility of nodes may affect the performance of protocols, we simulated networks with 100 nodes moving with different velocities. As shown in Fig. 7.19, when networks become more dynamic, the data delivery ratio decreases for all protocols. However, ACAR is only slightly affected (reduced by 1%) by the change of node mobility. This is because higher velocity does not affect our connectivity model but only the choice of each next hop. In fact, the larger the velocity, the lower the accuracy of predicting neighbors positions.

Since we implemented NLP on VADD and GSR*, their data delivery ratios drop more slowly than GPSR* and MOVE. For GPSR*, as no NLP algorithm is available, it may select next hops which are already out of the communication range due to the high speed of its neighbors. The

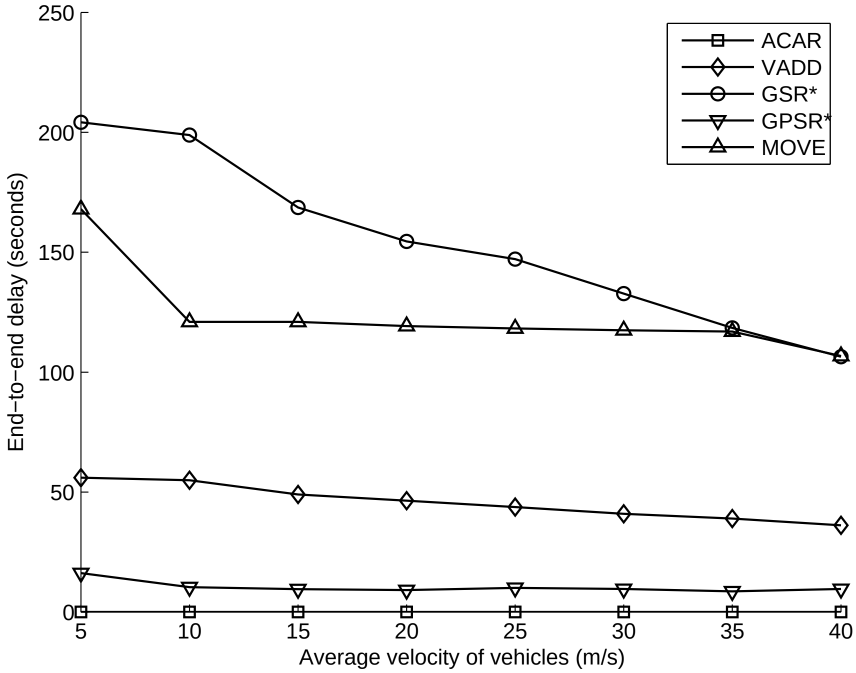


Figure 7.20: End-to-end Delay vs Different Velocities with 100 Nodes in the Networks

situation is even worse for MOVE protocol. Unlike other protocols in which packets are routed on either geographic shortest paths or high connectivity paths, MOVE forwards packets to nodes moving towards the destination. However, this node may move away from the destination a few seconds later. If no next hop is available, which is very common for MOVE, current forwarder (carrying packets) will move away from the destination very fast and so extends the total routing path. The longer the route, the higher is the chance of wrongly selecting next hops. So the delivery ratio of MOVE drops very fast when the velocity increases.

As shown in Fig. 7.20, the end-to-end delay of ACAR is very low because ACAR forwards packets on routes with the highest connectivity-qualities. So the delay of ACAR is mainly composed of wireless transmission and protocol queuing delays, which are very small. Since GPSR* utilizes the perimeter mode to find connected paths, its delay is also very low. However, end-to-end delays of VADD, GSR* and MOVE are much higher and drop when the velocity increases. Because VADD, GSR* and MOVE have no mechanisms for selecting connected paths, their delays

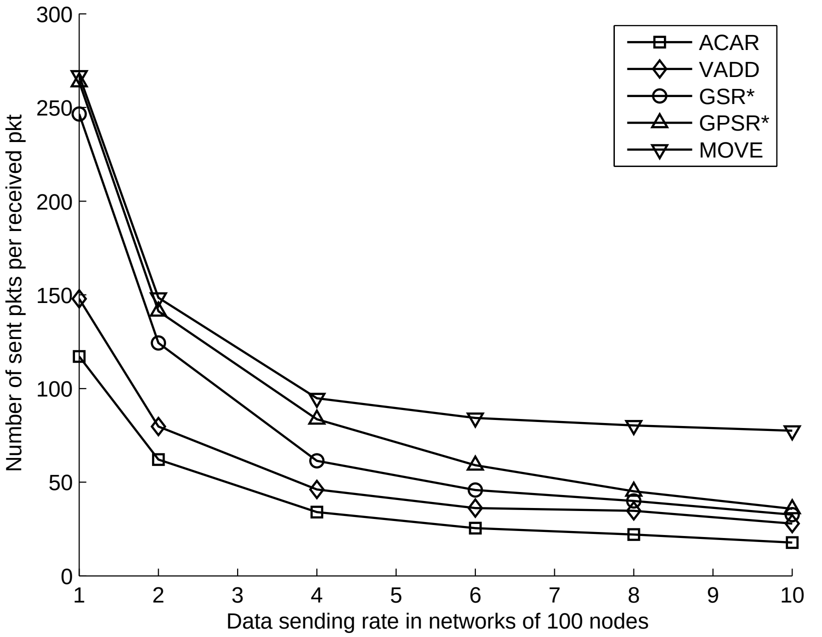


Figure 7.21: Number of Packets Sent in Networks Per Delivered Packet with 100 Nodes in the Networks

are higher due to more packets being buffered. In addition, when the velocity increases, packet carriers can move faster to the destination and thus decrease the average end-to-end delay.

7.10 Networking Overhead

Networking overhead is defined as the number of packets sent into networks for every delivered packet. In other words, it is the ratio between the number of sent packets (beacon and data messages) and the number of received packets. As every node sends beacon messages periodically, e.g. every one second, this kind of packets make up the majority of networking overhead. When the data sending rate increases, more packets will be delivered to the destination, so the overall networking overhead decreases. The total number of sent packets for all protocols are similar, so the networking overhead of ACAR is the lowest as it delivers more packets than others (Fig. 7.21).

In ACAR, there is an on-the-fly density collection scheme which will increase the size of every forwarded packet. So we further evaluate the networking overhead by investigating the total size of packets sent into networks per delivered packet. As the periodic beacon scheme is the same

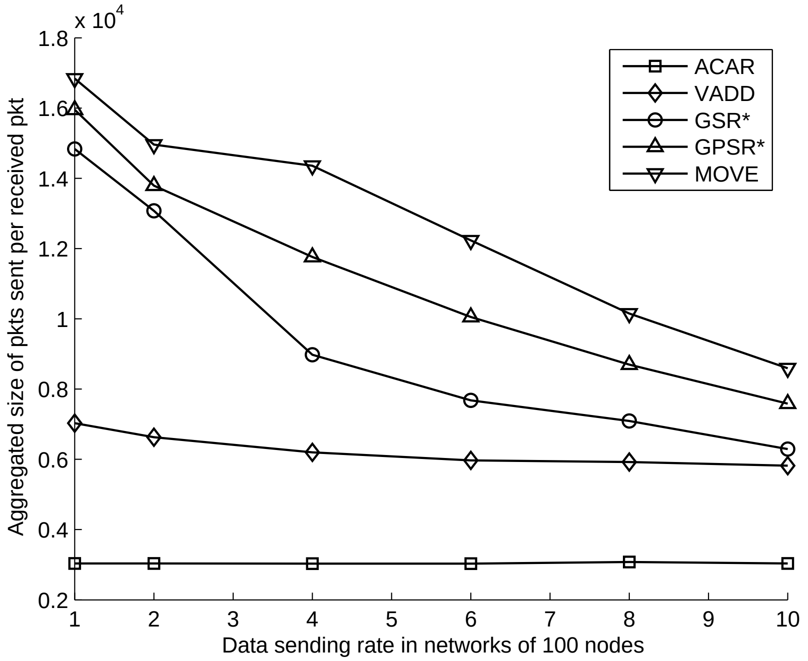


Figure 7.22: Size of Packets Sent in Networks Per Delivered Packet with 100 Nodes in the Networks

for all protocols, we only consider the size of data packets here. As shown in Fig. 7.22, ACAR gives the lowest networking overhead in terms of the average size of data messages per delivered packet. The major reason is that ACAR delivers more packets than others, so reduces the overall networking overhead.

Chapter 8

Location Privacy Protection in VANET

In previous sections, we proposed and discussed in detail the adaptive connectivity aware routing protocol which is built upon the fundamental geographic routing. However, geographic routing requires every vehicle to broadcast its location information to its neighboring nodes, and this process will compromise user's location privacy. Existing solutions to this problem can be categorized into two groups: 1) hiding user's location or 2) preserving user's identification information in routing protocols, which drastically reduce network performances. To address this issue, we proposed a dummy-based location privacy protection (DBLPP) routing protocol, in which routing decision is made based upon the dummy distance to the destination (DOD), instead of users' true locations. In this scheme, users' true locations and identification information are preserved, so the user's location privacy is protected. Our protocol is compared to existing solutions by simulations, results show that the DBLPP provides similar network performances as other routing protocols, and achieves a higher level of location privacy protection on vehicles in networks.

8.1 Introduction of Location Privacy Issues in VANET

To support geographic routing in VANETs, most routing protocols require location information of vehicles being periodically exchanged among one-hop neighbors. This broadcasting process will release a user's location and identification information to its neighbors in which malicious nodes may exist. For example, by passively overhearing these beacon messages, an adversary can easily identify the locations visited by a certain vehicle and then breach the privacy of this user.

Location privacy protection of geographic routing can be defined as: without losing the benefits of geographic routing, a routing protocol should not reveal any user's current and historical locations to unauthorized nodes. The unauthorized nodes may be malicious infrastructures (e.g WiFi access points), laptops with wireless interfaces, or vehicles moving on roads.

There are several previous works on protecting user's location privacy. They can be categorized into two groups: preservation of node identity (ID) [47, 50, 51] and geographic location [53–55]. If a user's identity is hidden, even though the adversary eavesdrops this user's locations, it cannot link those locations to the user [46]. However, it is almost impossible to completely eliminate a vehicle's identity because ID information is critical for routing, security and billing purposes. Therefore, randomly-changing pseudonyms are used on vehicles to replace the permanent ID in VANETs [47,50,51]. However, when pseudonyms are applied on vehicles, the network performance can be drastically affected as it was shown in [52].

Another approach is hiding vehicle's location information [53–55] so that the adversary can only detect an area (where a node is located) but not the node's true location. Since location information is the foundation of geographic routing protocols [10, 11, 13, 42], geographic routing will fail if such information is not provided.

To avoid periodically broadcasting location information, another type of geographic routing – contention based forwarding (CBF) [61] is proposed. In CBF, only nodes participating in routing reveal their locations. Therefore, the location privacies of other nodes are preserved because they keep silent in the routing process. To further preserve the location privacies of nodes involved in routing, we propose a dummy-based location privacy protection (DBLPP) protocol. Unlike sending its real locations in CBF, a packet forwarder (vehicle) first sends a dummy distance to destination (DOD) to its neighbors. After receiving the dummy DOD information, receivers compete to each other and the one which is the closest to the destination will be elected as the next hop. Packets are then sent to this node, and it will route the packets just as its last-hop node did. The dummy DOD has to be carefully chosen so that not only the adversary is not able to infer the forwarder's true location, but also the geographic routing goals are achieved.

To measure how well a protocol protects user's location privacy, we proposed a novel entropy based location privacy protection metric. This metric models the entropy (in the unit of bits) required for the adversary to attach user's location privacy. The network performance of DBLPP protocol is compared with existing solutions by simulations. Results show that DBLPP can provide similar network performances as those of other protocols. In addition, with the new privacy protection metric, we also discovery DBLPP achieves a higher level of location privacy protection comparing to others.

8.2 Threat Model and Problem Statement

8.2.1 Threat Model in VANET

Geographic routing in VANETs can significantly facilitate the tracking of vehicles. Because location information is shared among neighbors in geographic routing protocols such as [10,61,62], attackers can easily eavesdrops on vehicle's and location. This may cause the leakage of a driver's privacy, e.g. a patient at an AIDS testing clinic might not want his or her movements (or even evidence of a visit) revealed to others.

The adversary can be external, which installs its own wireless receivers along roads and passively eavesdrops on vehicle's communication messages. As stated in [51, 63], by exploiting already deployed 802.11a/b/g infrastructures, it is possible to build a global adversary which eavesdrops on the entire networks.

The attackers can also be internal, which utilize devices that are legitimate members in VANETs. Such type of malicious nodes may passively collect data transmitted among neighboring nodes by the pre-installed IEEE 802.11 receiver. In our work, we assume both internal and external malicious nodes exist in the networks.

8.2.2 Greedy Forwarding Model

A vehicular ad hoc networking can be represented as a directed graph $G = (V, E)$, where V is the set of nodes (vehicles) and E is the set of wireless links, such that packets can be sent from node i to j for all $(i, j) \in E$. Node i and j are called the origin and destination of link (i, j), respectively,. The origin and destination of a link $l \in E$ are denoted $o(l)$ and $d(l)$, respectively. We assume $o(l) \neq d(l)$ for $\forall l \in E$. For each node i, we assume it has the same communication range R as others. If node i is located at (x_i, y_i), we define its communication area as A_i which is a circle centered at i with radius of R.

As shown in Fig. 8.1, suppose the current packet forwarder is node i, its neighbor set N_i can be represented as follows.

$$N_i = \{k : d(l) = k \wedge o(l) = i, \forall l \in E\} \tag{8.1}$$

If we track all the outgoing links $l \in E$ from node i, i.e. $o(l) = i$, we can find a set of nodes k on the other sides of these links. These node denoted by k are actually the neighbors of node i.

The purpose of greedy geographic routing is to select the neighbor which is the closest to the destination as the next hop. Therefore, the next hop should be:

$$\underset{\forall k \in N_i}{\arg} \max P(i, k, d) = \left\{ 0, \frac{dis(i, d) - dis(k, d)}{R} \right\} \tag{8.2}$$

where function $dis(k, d)$ provides the distance between node k and the destination d. As VANET is a dynamic mobile network, the graph G may be different from time to time. However, for a certain time instance t, we consider it as a static graph, i.e., $G = G_t$ at time instance t.

8.2.3 Problem Statement of Location Privacy Protection in Greedy Forwarding

Greedy forwarding is widely used in geographic routing protocols, such as GPSR and CBF. In GPSR, every node in networks periodically beacons its ID and location to its neighbors and thus cause a significant privacy issue. However, in CBF, only those nodes participating in routing share their location information. The problem we are solving is to check if the ID and location information are necessary to achieve greedy forwarding. If not, we investigate how to preserve those information so that the network performance is not significantly affected.

In CBF, node i first broadcasts a request message to its neighbors and waits for replies. When a neighbor $k \in N_i$ receives this packet, it sets up a timer with the interval of $T(1 - P(i, k, d))$ where T is the maximum one-hop forwarding delay. Let A_d denote the circle centered at the destination with radius of $dis(i, d)$, then nodes within the area of ($A_i - A_d$) do not set up timers since they are farther to the destination compared to the node i. Then, the node which is the closest to the destination, e.g. j, will first time out and reply to the sender i because of the shortest timer interval. This reply also serves as a suppression message to other neighbors. That means nodes in $\{k : o(l) = j \wedge d(l) = k, \forall k \in (A_i \cap A_d)\}$ will cancel their timers.

From the above analysis, location information seems to be critical for geographic routing. If there is a global adversary or some passive malicious nodes in networks, the driver's location privacy cannot be protected. Therefore, the challenging problem we exploited is to develop techniques that let a user benefits from location-based geographic routing, at the same time, retains its location privacy. The proposed protocol must satisfy the following conditions:

- User's location information should be protected
- Greedy forwarding should be achieved, i.e., every selected next hop must be the one which is closest to the destination
- Not too much overhead is added to existing routing protocols
- Network performance should be similar as that of original ones

8.3 Details of DBLPP Protocol

In DBLPP, before a data packet is transmitted, a packet forwarder first sends a request to forward (RTF) message. In such RTF message, the sender will provide a dummy DOD, instead of its real location. Then, the elected next hop sends pseudonyms instead of its true ID in the clear to forward (CTF) message to reply the sender. Finally, the data packet will be transmitted from the sender to the next hop. This process is different from the active selection in the contention based forwarding with active selection (so-called CBF-AS) because the DBLPP does not release forwarder's location or next hop's identification.

8.3.1 Control Messages Exchange in DBLPP

In this section, we will describe how a next hop is chosen in DBLPP. As shown in Fig. 8.1, suppose the current packet forwarder is node i which received a packet with the sequence number of $seq\#$ from node m. This packet will be delivered to the destination located at (x_d, y_d). In the figure, we note node j should be the next hop as it is closer to the destination comparing to other neighbors. Before forwarding this data packet, node i first broadcasts a RTF message including the $seq\#$ and (x_d, y_d). To illustrate the basic idea of greedy forwarding in DBLPP, we currently do not use the dummy DOD which will be introduced later.

When a neighbor (e.g. node k) receives this RTF, it first checks if the packet was received before by comparing the sequence number in the RFT to those of buffered packets. If there is a cache hit, node k will simply drop the RTF because it received this RTF before. If the RTF is a brand new message, node k saves this RTF into its buffer and then sets up a timer with the runtime of:

$$t(r_k) = f(1 - 1/r_k) \tag{8.3}$$

where r_k is the DOD of node k. According to the above equation, the runtime of timer on each node is proportional to its DOD. Therefore, the one which is the closest to the destination will first time out.

As shown in Fig. 8.1, node j first times out and sends the CTF message including the $seq\#$ and a set of pseudonyms denoted as $< ID_l >$. These l pseudonyms in $< ID_l >$ are randomly chosen from a set of L pseudonyms which are pre-installed in each vehicle. Given a relatively large

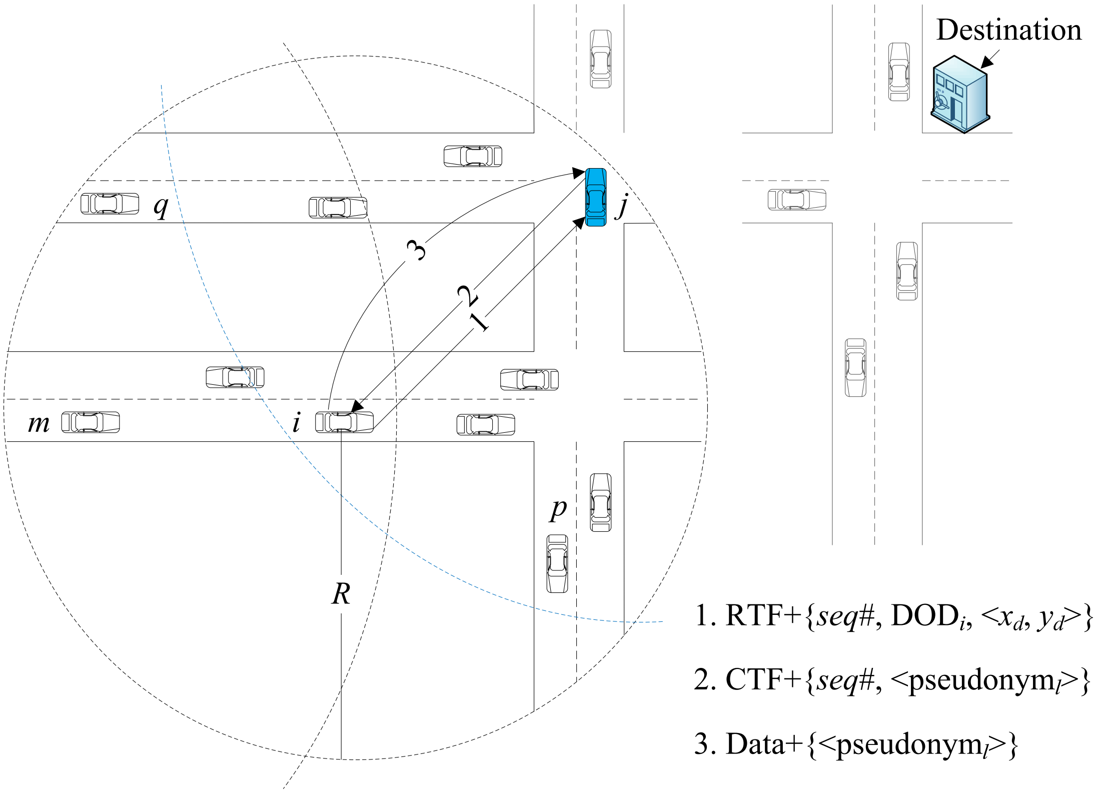

Destination
q
j
3
2
1
m
i
p
R
1. RTF+{seq#, DOD$_i$, <x_d, y_d>}
2. CTF+{seq#, <pseudonym$_l$>}
3. Data+{<pseudonym$_l$>}

Figure 8.1: Dummy Based RTF/CTF Exchange Among Vehicles

value of L and l, the probability of choosing the same pseudonym in two different CTF messages will be extremely low.

This CTF message also serves as a suppression message for all j's neighbors. When these nodes (neighbors of node j and i) receive this CTF message, they will immediately cancel their timers because there is a better next hop selected. Since the CTF from j can only suppress its neighbors, those nodes which are neighbors of node i but not node j may send duplicated CTF messages. That means multiple CTFs may be received at the sender i.

After receiving the first CTF sent from j, node i immediately sends the data packet including the pseudonyms $< ID_l >$ of previously received CTF from node j. If node i sends the data packet before the second CTF is received, all its neighbors are suppressed by the data message. If a duplicated CTF message is received before the data packet is sent, node i simply omits this CTF because a better next hop is already chosen.

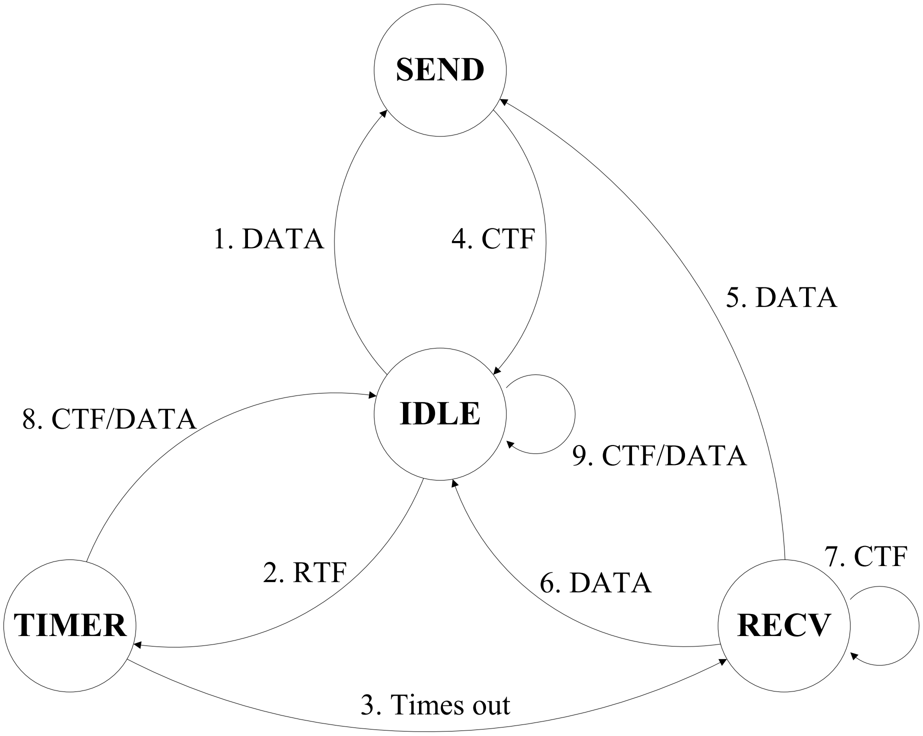


Figure 8.2: State Machine Of Nodes in DBLPP

When a neighbor of node i receives the data packet, if it did not send any CTF message, it drops the data message. Otherwise, it checks whether the $< ID_l >$ in the data packet are the same as what it sent out previously. If so, the packet are delivered; otherwise, the packet is dropped.

By exchanging one pair of RTF and CTF messages, geographic greedy forwarding is achieved between node i and j. The whole process of such control message exchange will be better explained by using state machine transition diagram. As shown in Fig. 8.2, when the system starts, all nodes are in the IDLE mode. Depending on what type of message is received, a node will change its mode as follows.

1. A node changes its mode from IDLE to SEND only if it intends to send data packets to another node (destination) in the networks. To successfully send out a data packet, this node (sender) first sends a RTF message to its neighbors.

2. When the sender's neighbors receive this RTF message, they come into the TIMER mode in which timers are set up according to the neighbors' DODs.

3. The node which is the closest to the destination will first time out and fire the CTF message. That means it is elected as the next hop of the sender. After it sends out CTF, it goes into the RECV mode which indicates it is ready to receive data packets.

4. When the sender receives this CTF message, it unicasts data to the next hop and returns to IDLE mode.

5. If the next hop is not the destination, it will come to the SEND mode to keep forwarding the data packet.

6. If it is the destination, it delivers the message and comes to IDLE.

7. It is possible that the next hop receives a duplicated CTF from another node as we will discuss in Section 8.3.2. It will simply dump this duplicated CTF message.

8. Other neighbors of the sender also receive the RTF message and set up timers as shown in step 2. When they receive a CTF or data packet before their timers expire, they will cancel the timers because there is a better next hop selected. In those cases, they return to IDLE from TIMER mode.

9. When a node is in IDLE mode and receives a CTF or data packet, it will ignore those messages and keep in IDLE mode. Because it was not involved in any network activities, it needs to be silent.

8.3.2 Duplicated Responses and Location Privacy Protection

In the previous section, we notice that there may be multiple duplicated CTFs in the networks. These duplicated CTFs are useless for forwarding data packets but harmful for network performance. On the other, from the length of a timer, the adversary can easily infer the DODs of corresponding nodes. Then, it can comprise the user's location privacy. Therefore, it is important to set up a proper timer so that the location privacies of receivers and senders are preserved and the number of duplication CTFs is minimized. In this section, we will investigate how to set up a timer to achieve those two goals.

According to Equation 8.3, when the timer of a node expires depends only on the node's DOD. Therefore, a simple way to set up a timer of a node with DOD as r will be:

$$t(r) = T \cdot (1 - 1/r) \tag{8.4}$$

where T is the maximal one-hop forwarding delay. If timers are set up in this way, duplicated responses may be generated by receivers in a certain area, called duplication area. As shown in Fig. 8.3, suppose the best next hop is located at r_1 away from the destination. Assume there is another node with the DOD of r such that $t(r) - t(r_1) < \delta$, where δ is the minimal time interval required for successful suppression. Then, this node will send a duplication response before the CTF from the best suited node can successfully suppress its timer. Obviously, the larger the width of the duplication area, the more duplicated responses will be generated.

Following Equation 8.4, the value of T needs to be very large to achieve a reasonable small duplication area. For example, according to Equation 8.4, the timer's interval on the best suited node should be $t(r_1) = T \cdot (1 - 1/r_1)$. Duplicated messages can be generated from another node with the DOD of r satisfying the following condition:

$$t(r_1) < t(r) < t(r_1) + \delta = T(1 - \frac{1}{r_1}) + \delta \tag{8.5}$$

To avoid generating duplicated messages, the node needs to set up a timer with runtime greater than:

$$t(r) = T\left(1 - \frac{1}{r_1}\right) + \delta = T\left(1 - \frac{1}{\frac{r_1 T}{T - \delta r_1}}\right) \tag{8.6}$$

The width of duplication area can be computed as:

$$\frac{r_1 T}{T - \delta r_1} - r_1 = \frac{\delta r_1^2}{T - \delta r_1} \tag{8.7}$$

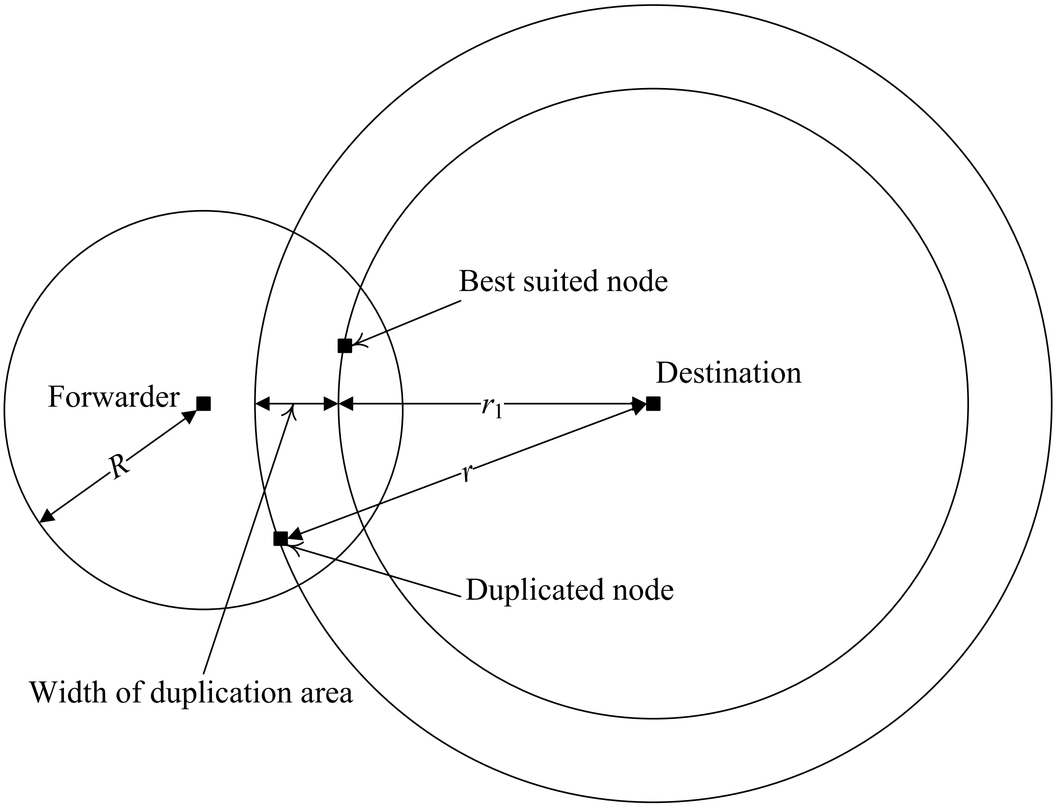


Figure 8.3: Duplicated Responses and Duplication Area

Since δ is a fixed value, we can rewrite the maximal one-hop forwarding delay T as $k \cdot \delta$. To achieve an acceptable duplication area with the width of Δ, the following equation must hold:

$$\frac{\delta r_1^2}{T - \delta r_1} = \frac{r_1^2}{k - r_1} < \Delta \tag{8.8}$$

In other words, the delay T will be:

$$T = \frac{r_1^2 + r_1 \Delta}{\Delta} \cdot \delta \tag{8.9}$$

From the above equation, we find that the longer the DOD of a node, the larger the value of T will be. However, the value of T must be very small to avoid large networking delays. To address this issue, we modify the Equation 8.4 by considering the dummy DOD information obtained from the last-hop node. Suppose a node (current packet forwarder) sends a RTF packet, all receivers will set timers according to the following equation:

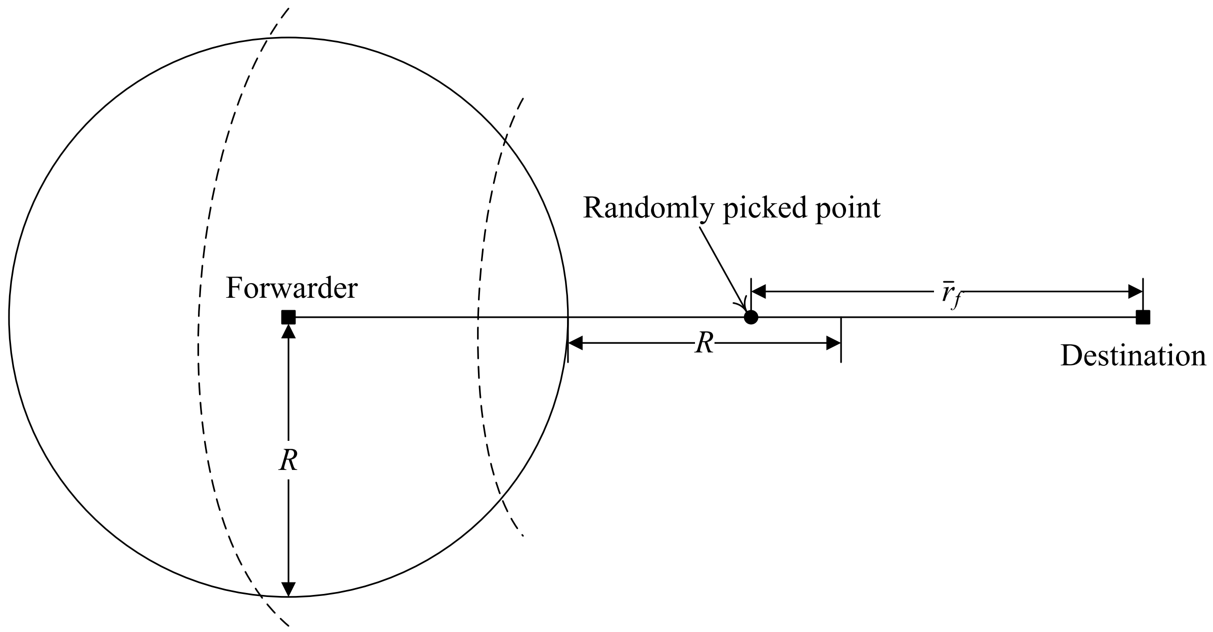


Figure 8.4: Dummy DOD Selection On Packet Forwarder

$$t(r) = T \cdot \left(\frac{r - \bar{r}_f}{3R} \right) \tag{8.10}$$

where r is the DOD of a receiver and $\bar{r}_f$ is the dummy DOD information from the received RTF. The value of $\bar{r}_f$ is randomly chosen, so the real location of last hop node is preserved. The selection of dummy DOD $\bar{r}_f$ is shown in Fig. 8.4. We first randomly choose a point on the line between current packet forwarder and the destination. This point must be R away but within $2R$ from the packet forwarder, where R is the wireless communication range. The value of $\bar{r}_f$ can be computed as:

$$\bar{r}_f = r_f - (1 + \rho) \cdot R \tag{8.11}$$

where r_f is the real DOD of the forwarder, and ρ is a real number randomly chosen from $(0, 1)$. Since the value of ρ ranges from 0 to 1, the forwarder's real location is hidden within a range of R. That means the difference $|r_f - \bar{r}_f|$ between the real and dummy DODs is within $[R, 2R]$. By using the random variable ρ, the forwarder's location is protected.

We also know that $r - \bar{r}_f$ is equal to $r - r_f + (1+\rho) \cdot R$. Since the forwarder and the receiver are neighbors, the value of $r - r_f$ must be within $[-R, R]$. Therefore, the value of $r - \bar{r}_f$ is within $[0, 3R]$. According to Equation 8.10, the runtime of the timer on this node will be within $[0, T]$. Therefore, the value of T is independent upon a node's DOD and it does not need to be very large to reduce the number of duplicated CTFs.

With these equations, we calculate the width of duplication area as $3R \cdot \delta / T$. If $T = k \cdot \delta$, to achieve an acceptable duplication area, we must have:

$$T = 3R \cdot \Delta \cdot \delta \tag{8.12}$$

We note that the one-hop maximal delay in the above equation is a fixed value. This delay is much smaller than what is computed from Equation 8.9, and is acceptable in VANETs.

Since dummy DODs and pseudonyms can preserve the locations and identifications of vehicles, the DBLPP provides a higher level of location privacy protection on vehicles in VANETs. Meanwhile, with the active selection of every next hop, geographic routing is achieved in networks as well. In addition, the number of duplication responses and average one-hop delay are reduced in DBLPP comparing to the original CBF-AS.

8.4 Entropy Based Privacy Protection Measure

To evaluate a location privacy protection scheme, we need to measure the hardness of an adversary node attacking a user's location privacy. Therefore, we propose an entropy based location privacy protection metric.

As we described in previous sections, the location privacy of vehicles in VANETs includes two types of information: node's identification and location. Previous works on location privacy protection either focus on the preservation of node's location or node identification. We claimed that preservations of both ID and location are necessary because it is possible for the adversary node to detect the node ID or location. For example, the IEEE 802.11 MAC address or IP address

$$\begin{array}{c|cccccc|}
 & L_0 & L_1 & \cdots & L_i & \cdots & L_n \\
\hline
I_0 & 1 & 1 & \cdots & 1 & \cdots & 0 \\
I_1 & 1 & 1 & \cdots & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
I_j & 1 & 1 & \cdots & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
I_n & 1 & 1 & \cdots & 1 & \cdots & 0 \\
\hline
\end{array}$$

Figure 8.5: Matrix Recording Possible Node Identifications And Locations

of a vehicle can be easily captured by wireless network sniffer software. On the other hand, there are many available wireless localization technologies which can be utilized by the adversary node to track user's locations.

In our work, we assume it is impossible or costly expensive for the adversary detecting both node's ID and location. We first model the probability of the adversary node being able to predict a user's ID and location information in VANETs. Because this probability is usually very small, it is difficult to distinguish the difference between different privacy location protection schemes. Therefore, we use entropy to model the overall hardness of location privacy attack on the adversary node.

We define two matrices and denote them as $IM(s,p)$ and $LM(s,p)$, respectively. Each matrix contains two dimensions: node IDs and locations. As shown in Fig. 8.5, the dimension denoted as $I = I_1, I_2, \cdots, I_n$ records all nodes IDs in networks. The other dimension denoted as $L = L_1, L_2, \cdots, L_n$ is used to record node's locations information. The initial value of $IM(s,p)$ and $LM(s,p)$ is 0. It will be updated by adding 1 if the adversary node receives a message indicating node I_i is probably located at L_j.

At a certain time instance t, there will be several communications occurring between nodes. Let's look at the communication link between node i and j, as shown in Fig. 8.1. Node i first sends out a RTF including its dummy DOD to its neighbors. Then, node j sends a CTF containing pseudonyms to node i. Finally, data packets are delivered from i to j with the pseudonyms previously received from j.

In the first step, node i releases its rough location to the networks by providing a dummy DOD. Since the dummy DOD is randomly chosen, there may be other nearby nodes providing the same DOD and those nodes can be presented as:

$$\chi_i = \{k : dis(k,d) > [dis(i,d) - R] \wedge dis(k,d) < [dis(i,d) + R]\} \qquad (8.13)$$

where $dis(k,d)$ and $dis(i,d)$ are the DODs of node k and i, respectively. In this case, the adversary node updates $LM(s,p)$ to $LM(s,p)+1$ where $p = L_k(k \in \chi_i)$ because any node $k \in \chi_i$ can give the same dummy DOD. Therefore, from this dummy DOD of node i, the adversary node can only predict this message was from χ_i but not sure which node in the set. For example, suppose node n_1 and n_2 are in the set of χ_i. The adversary node only knows there is a message from location L_1, L_2 or L_i, but has no idea about who sent this RTF. Therefore, as shown in Fig. 8.5, the adversary node updates $LM(s,p) = LM(s,p)+1$ where $s = I_1, I_2, \cdots, I_n, p = L_1, L_2, L_i$. Similarly, if the CBF protocol is used, we need to update $LM(s,p) = LM(s,p)+1$ where $s = I_1, I_2, \cdots, I_n, p = L_i$.

For the second step, node j sends a set of randomly chosen pseudonyms in its CTF message. Because every node can send the same pseudonyms, the adversary node leans nothing about sender's ID information from this message. In this case, it updates the matrix by changing $IM(s,p)$ to $IM(s,p)+1$ for all $s = I_1, I_2, \cdots, I_n$ and $p = L_1, L_2, \cdots, L_n$.

For the third step, since there is no more new information (identification or location) revealed in packets, we simply omit this step in the privacy protection measurement.

If we look at the CBF-AS protocol, a packet forwarder i sends out RTF along with its location. The next hop j sends CTF along with its ID to the previous packet sender. In the first step, we set $LM(s,p) = LM(s,p)+1$, where $p = L_i$, for every $s = I_1, I_2, \cdots, I_n$. In this case, the adversary node only needs to predict from which node this message is sent. In the second step, we set $IM(s,p) = IM(s,p) + 1$, where $s = I_j$, for every $p = L_1, L_2, \cdots, L_n$. This is because the adversary node only needs to predict where node j is.

For a matrix M (either IM or LM), the value of $M(s,p)$ records the number of times that node s probably appear at location p. The entries of M are proportional to the joint probabilities, which we obtain by normalization:

$$P(s = I_0, p = L_0) = \frac{M(I_0, L_0)}{\sum\limits_{s,p} M(s,p)} \tag{8.14}$$

This equation models the probability of the adversary node being able to predict that node I_0 is at location L_0. For example, if the adversary node receives a RTF from node I_0, the probability of the adversary node being able to predict node I_0 is located at L_0 will be $1/n$. If the adversary node receives one CTF, the pseudonyms provide nothing useful about node identifications. Therefore, the probability will be $1/n^2$ because this message can be sent from any node at any location.

If the adversary node can spoof node's ID, the conditional probability of the node I_0 being located at L_0 will be:

$$P(p = L_0 | s = I_0) = \frac{M(I_0, L_0)}{\sum\limits_{p} M(I_0, p)} \tag{8.15}$$

Therefore, the Shannon's entropy required by the adversary node to correctly predict that node I_0 is located at L_0 will be:

$$H_{I_0}^{L} = \sum_{p} P(p|s = I_0) \cdot \log \frac{1}{P(p|s = I_0)} \tag{8.16}$$

Similarly, if the adversary node can detect a node's location, the conditional probability that at location L_0, the node must be I_0 is:

$$P(s = I_0 | p = L_0) = \frac{M(I_0, L_0)}{\sum\limits_{s} M(s, L_0)} \tag{8.17}$$

So we can compute the entropy of predicting that at location L_0, the node must be I_0:

$$H^I_{L_0} = \sum_s P(s|p = L_0) \cdot \log \frac{1}{P(s|p = L_0)} \tag{8.18}$$

If the adversary node can localize node's locations easily, the uncertainty of predicting IDs of all nodes will be cumulative entropy:

$$E_I = \sum_s H^I_p, p = L_1, L_2, \cdots, L_n \tag{8.19}$$

where we assume the network events (e.g. sending RTF message) are independent to each other. Therefore, cumulative entropy models the hardness of the adversary node to predict all nodes IDs. If these events are dependent to each other, we can use the average entropy to model the hardness of predicting only one node's ID. This average entropy can be computed as $\bar{E}_I = E_I/n$.

If the adversary node can detect node's IDs easily, the uncertainty of predicting locations of all nodes can be modeled as cumulative entropy:

$$E_L = \sum_s H^L_s, s = I_1, I_2, \cdots, I_n \tag{8.20}$$

where we also assume network events are independent to each other. Similarly, when these events are dependent to each other, the average entropy $\bar{E}_L = E_L/n$ can be used.

Suppose the costs of enabling identification spoofing and localization at the adversary node are c_I and c_L, respectively. Then, we obtain the balanced cumulative entropy as:

$$H(M) = w_I \cdot E_L + w_L \cdot E_I \tag{8.21}$$

where $w_I = c_I/(c_I + c_L)$ and $w_L = c_L/(c_I + c_L)$. The two matrices IM and LM record events of sending RTF and CTF messages, respectively. Therefore, the cumulative entropy required by the adversary node will be $H = H(IM) + H(LM)$. $H(IM)$ is the entropy of predicting a node's

Table 8.1: Simulation Set-Up Parameters for DBLPP

Parameter	**Value**
Number of lanes	2 lanes per direction
Number of nodes	100
Communication range	250 m
Max. one-hop delay T	0.1 ms
Size of pseudonyms pool	1000
Number of pseudonyms in CTF	5

locations if the ID information is given. The second $H(LM)$ is the entropy of predicting a node's ID if location data are given.

From the Equation 8.21, we note that the higher the cumulative entropy, the harder it will be for the adversary attacking user's location privacy. In the same way, we obtain the average entropy as:

$$\bar{H}(M) = w_I \cdot \bar{E}_L + w_L \cdot \bar{E}_I \tag{8.22}$$

Accumulate or average entropy will be used to measure how well a location privacy protection scheme works. We will use them in our simulations to quantify the location privacy protection measurement of DBLPP and other methods.

8.5 Simulations of DBLPP

We implement the DBLPP protocol in ns-2.29 and compare its network performance to other two geographic routing protocols: GPSR and CBF-AS. To evaluate the location privacy protection in DBLPP, we implement the periodic changing-pseudonym scheme which is widely used in previous works [47, 50, 51]. Therefore, by extending the GPSR and CBF-AS, we have another two protocols with the periodic changing-pseudonym scheme: CBF-AS-ID and GPSR-ID. Details of the simulation setup parameters are listed in Table 8.1. The movement of vehicles in the networks is generated by VanetMobiSim [41].

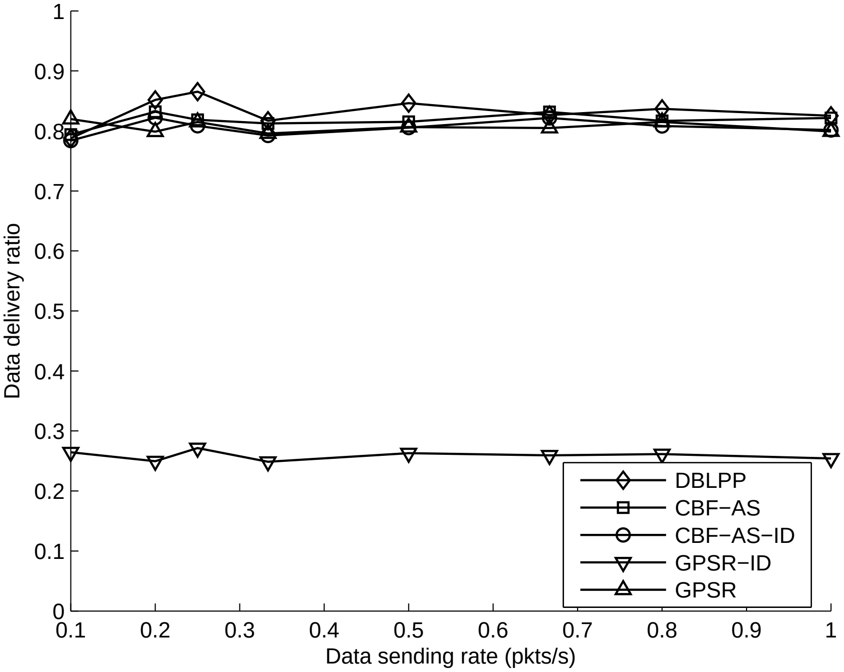


Figure 8.6: Data Delivery Ratio Vs Data Sending Rate for DBLPP

8.5.1 Data Delivery Ratio

Data delivery ratio is defined as the number of received packets at the destination divided by the number of sent packets from the source. As shown in Fig. 8.6, DBLPP, CBF-AS, CBF-AS-ID and GPSR achieve similar data delivery ratios.

GPSR-ID gives the lowest data delivery ratio because chosen next hops often change their IDs so that it cannot receive packets which supposed to be delivered to them. In GPSR, every node selects the next hop based on the stored neighbor's location information. Since neighbor's location information is updated periodically, it is possible that out-of-date neighbors exist in one's neighbor list. In this case, packets may be dropped because the selected next hop may be out of communication range.

In DBLPP, the next hop will be elected through competitions and only the winner response to the packet forwarder. Packets will then be immediately sent to this elected next hop. So the chance

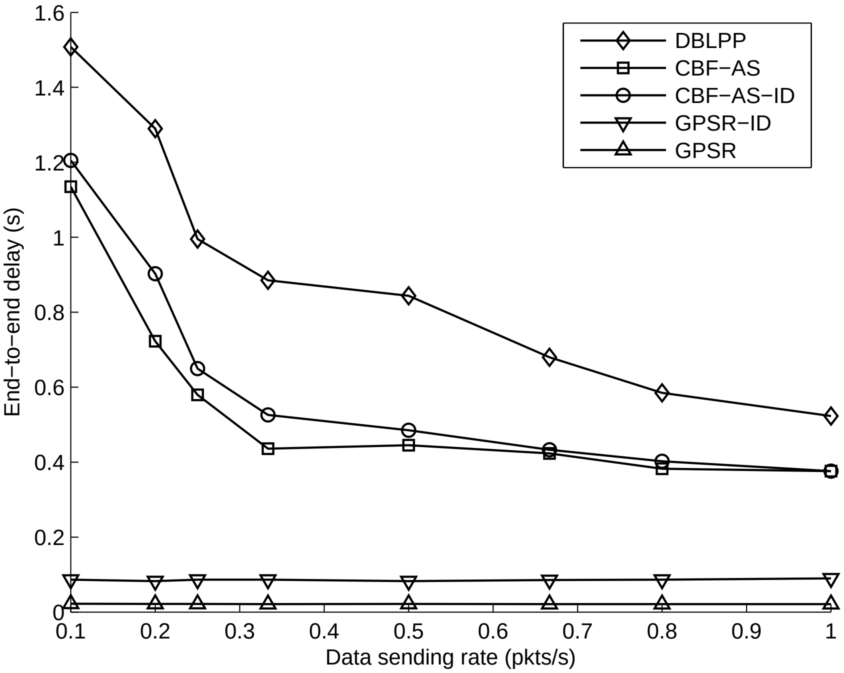


Figure 8.7: End-to-end Delay Vs Data Sending Rate for DBLPP

of forwarding packets to an out-of-date neighbor in DBLPP is very low. This is why DBLPP delivers more packets than GPSR.

Because the contention based forwarding scheme is used in DBLPP and CBF-AS, the data delivery ratios of these two protocols are similar. Although periodic changing ID is applied on CBF-AS-ID, its data delivery ratio is slightly worse than those of DBLPP and CBF-AS. This is because after a next hop sends its ID in a CTF message, the sender immediately delivers data packets, the time difference between those two events is too small to allow the next hop change its ID.

8.5.2 End-to-end Delay

The end-to-end delay is defined as the average time taken for a packet being transmitted from source to destination in the networks. As shown in Fig. 8.7, GPSR and GPSR-ID provide smaller end-to-end delays compared to other protocols because GPSR and GPSR-ID do not need to set up timer to select next hops. However, in DBLPP, CBF-AS and CBF-AS-ID, timers are used in every

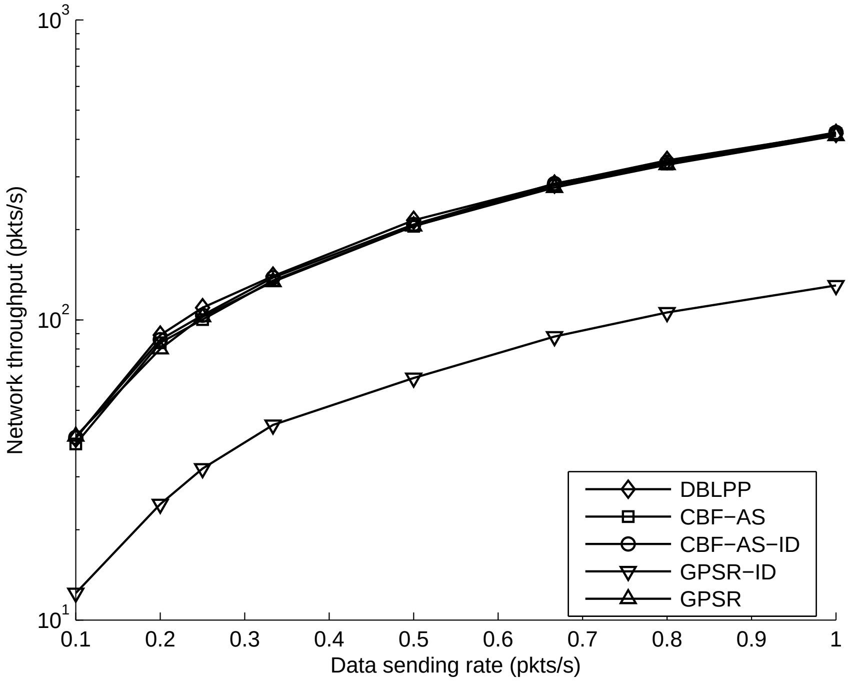


Figure 8.8: Network Throughput Vs Data Sending Rate for DBLPP

next hop selection. Therefore, the end-to-end delays of CBF-AS, CBF-AS-ID and DBLPP become large. However, with the same contention based forwarding scheme, DBLPP generates a larger end-to-end delay comparing to CBF-AS and CBF-AS-ID. The reason is that DBLPP generates more duplicated responses which cause networks become more congested and thus the end-to-end delay increases. This delay can be further reduced by using a smaller maximal-runtime of timers, which will be our future work.

Since frequent network disconnections occur in VANETs, carry-and-forward based geographic routing protocols [13, 14, 42] are widely used in VANETs. Comparing to the huge delay caused by carry-and-forward scheme, these generated by DBLPP can be ignored.

8.5.3 Network Throughput

Network throughput is defined as the number of packet delivered to the destination per second. As shown in Fig. 8.8, besides GPSR-ID, all protocols give similar network throughput which

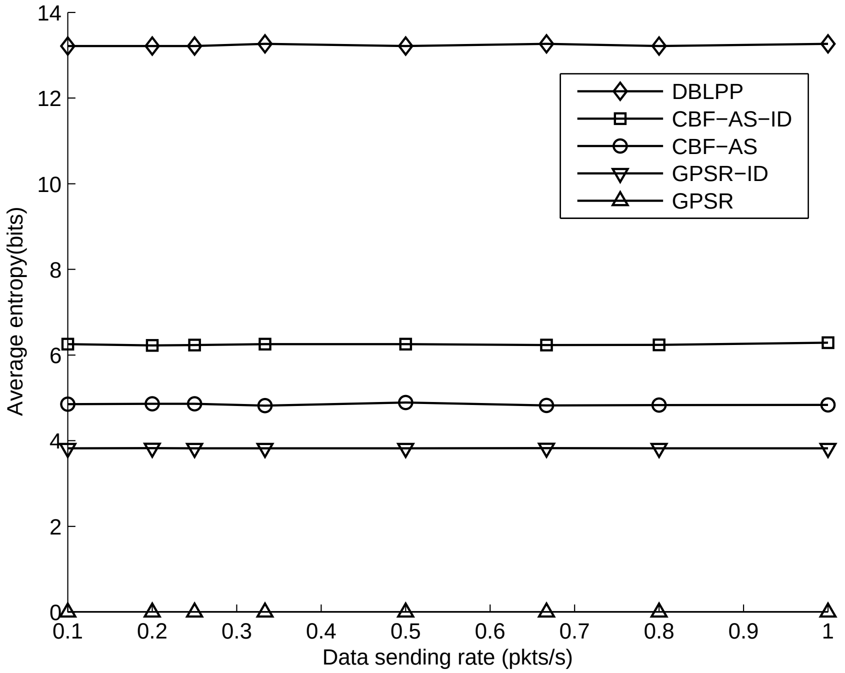


Figure 8.9: Average Entropy of Location Privacy Protection

increases as the data sending rate increases. Because the link quality of every hop in DBLPP is better than that of GPSR, DBLPP achieves a slightly larger network throughput than GPSR.

8.5.4 Location Privacy Protection

Entropy was first introduced in *Information Theory* to quantify the uncertainty of a system. In our work, the higher the privacy entropy value is, the more difficult it will be for attackers predicting user's location. In the simulations, we tracked all communication events (RTF, CTF and data packets) and computed the probability of predicting the location and identification information of a node involved in routing. Based on the definition of entropy, we then calculate the average entropy required for the adversary to predict a user's location and identification.

As shown in Fig. 8.9, in order to attack a vehicle's location privacy, more bits are required in DBLPP compared to others. In GPSR, every node periodically beacons its location and ID to neighbors, so the entropy of computing every vehicle's location is zero. In CBF-AS, every packet forwarder sends its location (not ID) in RTF messages to its neighbors. When the self-elected next hop sends a CTF message, its ID (not location) is put into the packet. Because either ID or location information is protected in CBF-AS, it provides a higher entropy value. In DBLPP, dummy DODs and pseudonyms are used, so it requires more bits for the adversary to attack even one node's location. Although the CBF-AS-ID and GPSR-ID can provide a certain degree of location privacy protection, they are not as good as DBLPP. It is because DBLPP preserves both identification and location information while the random changing-pseudonym scheme only protects user's identification data. In summary, the location privacy protection in DBLPP is much better than others.

Chapter 9

Future Work

In the previous chapters, we proposed and evaluated the ACAR and DBLPP protocols for efficient and privacy-protection communications in VANETs. However, it is not straightforward to integrate those two protocols. Although ACAR is built upon regular greedy geographic routing, it uses a unique method to select every next hop in the routing process. There are basically three differences between ACAR and DBLPP in forwarding packets in networks. First, ACAR requires every node to broadcast periodically its location and ID information in the network. However, to preserve user's location privacy, such broadcasting procedure is not needed in DBLPP. Second, DBLPP selects a next hop based on how much distance advance a neighbor can provides, i.e. the next hop must gives the maximal distance advance. While when ACAR selects a next hop, the maximal distance advance is not the determining factor. It also considers the EXT information which models a link's quality. Third, ACAR is a trajectory based routing protocol which means packets are forwarded along a computed path (that is composed of several road segments). The DBLPP is a location based protocol, so it does not consider any road topology information in its routing process.

To successfully integrate ACAR and DBLPP, those three differences has to be considered. For the first difference, as we discussed in previous chapters, broadcasting location and ID information is not necessary for geographic routing. For the second difference, since a node obtains and maintains EXT information from its neighbor's beacon message, broadcasting seems necessary for ACAR. There are two possible solutions for this issue: 1) adding broadcasting to DBLPP, and 2) modeling link quality without broadcasting. If a broadcasting scheme is added to DBLPP, only pseudonyms are sent in beacon messages to preserver user's location privacy. In this way, a node can easily record the qualities of links to its neighbors without revealing its own true ID. On the other hand, ETX is only one metric modeling link's quality, many other metrics may be applied as well. In this case, broadcasting is not essential for ACAR's next hop selection. The third difference is related to the second one, if ETX is replaced by other metrics, a new next hop selection algorithm is needed for ACAR. Because DBLPP uses contention based forwarding, this new algorithm must be able to properly set up timers so that the next hop (with the best link quality and distance advance) will first time out and then is elected from other neighbors.

Current active selection of next hop in DBLPP generates too much network overhead. To reduce such overhead, the DBLPP protocol can also be implemented in the RTS/CTS exchange of 802.11 protocols. Besides the regular data in RTS and CTS packets, we will add a few more information e.g. packet sequence number, destination location, pseudonyms. Such modification can be easily implemented in current MAC protocol stack. Therefore, the new RTS/CTS design can be programmed as a software library which is integrated to current 802.11 protocol stacks. Moreover, as greedy geographic forwarding is widely used in date communication for mobile devices, such as smart phones, PDAs and iPhones, the DBLPP protocol can be also applied to pervasive computing to achieve a high level protection of user's location privacy.

Chapter 10

Conclusion

We have presented a protocol for adaptively selecting routes based on statistical and real-time network information to avoid the influence of inaccurate statistical data. This protocol uses a novel model of network connectivity, which combines the cell-based and cluster-based connectivity models to capture the probabilistic property of network connectivity on road segments. The connectivity model considers the uniform (cell-based) and clustered (cluster-based) movements of vehicles, and provides a scheme to combine those two phenomena and computes network connectivity. Although the model requires historical data (e.g. road length, network density and traffic light period) from digital maps, connectivity information can be computed by every vehicle in a distributed manner.

Because the selected path provides the best connectivity-quality, ACAR achieves a higher data delivery ratio and lower end-to-end delay compared to other protocols. Moreover, since the route length can be calculated before forwarding packets, every next hop is selected by minimizing the packet error rate of the entire path. Our simulation results show that ACAR is much more suitable for VANET than other protocols because of its higher data delivery ratio, throughput and lower networking delay. In addition, it works very well even when the statistical data of road density is not accurate.

Since computations are performed on each vehicle, there is no additional network overhead in ACAR compared to other protocols. Every vehicle in the network only maintains its one-hop neighbors' information, so ACAR is a stateless routing protocol. Because every packet forwarder computes the best route and selects next hops individually, the implementation of ACAR algorithm is distributed and scalable. In summary, due to the smaller network overhead, stateless and distributed features, ACAR is a practical and efficient routing protocol for VANETs.

We also designed and implemented a dummy-based location privacy protection mechanism on geographic routing, which can be easily added to greedy geographic routing protocols. Location information exchange among vehicles is required by all kinds of geographic routing protocols. However, the proposed DBLPP does not need vehicles to exchange their true locations but only dummy DODs. In addition, elected next hops respond to forwarders with a group of pseudonyms, so the ID of a next hop is hidden as well. Simulation results show that DBLPP not only protects user's location privacy but also achieves similar network performances as other protocols.

Bibliography

[1] Jedrzej Rybicki, Björn Scheuermann, Wolfgang Kiess, Christian Lochert, Pezhman Fallahi, and Martin Mauve. Challenge: peers on wheels - a road to new traffic information systems. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 215–221, 2007.

[2] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Cabernet: Vehicular content delivery using WiFi. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 199–210, 2008.

[3] Aruna Balasubramanian, Ratul Mahajan, Arun Venkataramani, Brian Neil Levine, and John Zahorjan. Interactive wifi connectivity for moving vehicles. *SIGCOMM Computer Communication Review*, 38(4):427–438, 2008.

[4] Seung-Hoon Lee, Uichin Lee, Kang-Won Lee, and M. Gerla. Content distribution in VANETs using network coding: The effect of disk I/O and processing O/H. In *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 117–125, 2008.

[5] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: a programmable pervasive space. *Computer*, 38(3):50–60, 2005.

[6] Yang Zhang, Jing Zhao, and Guohong Cao. On scheduling vehicle-roadside data access. In *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, pages 9–18, 2007.

[7] Yong Ding, Chen Wang, and Li Xiao. A static-node assisted adaptive routing protocol in vehicular networks. In *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, pages 59–68, 2007.

[8] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353. 1996.

[9] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[10] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[11] Christian Lochert, Martin Mauve, Holger Füssler, and Hannes Hartenstein. Geographic routing in city scenarios. *Mobile Computing and Communications Review*, 9(1):69–72, 2005.

[12] N. Wisitpongphan, Fan Bai, P. Mudalige, and O.K. Tonguz. On the routing problem in disconnected vehicular ad-hoc networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications*, pages 2291–2295, 2007.

[13] J. Zhao and G. Cao. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications*, pages 1–12, 2006.

[14] Qing Yang, Alvin Lim, and Prathima Agrawal. Connectivity aware routing in vehicular networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 2218–2223, 2008.

[15] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 145–158, 2004.

[16] Zong Da Chen, H.T. Kung, and Dario Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 247–250, 2001.

[17] J. LeBrun, Chen-Nee Chuah, D. Ghosal, and M. Zhang. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *Proceedings of the IEEE 61st Vehicular Technology Conference*, pages 2289–2293, 2005.

[18] James Bernsen and D. Manivannan. Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification. *Pervasive and Mobile Computing*, 5(1):1–18, 2009.

[19] Yao H. Ho, Ai H. Ho, and Kien A. Hua. Routing protocols for inter-vehicular networks: A comparative study in high-mobility and large obstacles environments. *Computer Communications*, 31(12):2767–2780, 2007.

[20] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2(2):12–22, 2007.

[21] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 108–119, 2006.

[22] Zhaomin Mo, Hao Zhu, Kia Makki, and N. Pissinou. MURU: A multi-hop routing protocol for urban vehicular ad hoc networks. In *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems*, pages 1–8, 2006.

[23] V. Naumov and T.R. Gross. Connectivity-aware routing (CAR) in vehicular ad-hoc networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications*, pages 1919–1927, 2007.

[24] Hao Wu, Richard Fujimoto, Randall Guensler, and Michael Hunter. MDDV: A mobility-centric data dissemination algorithm for vehicular networks. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 47–56, 2004.

[25] Paolo Costa, Davide Frey, Matteo Migliavacca, and Luca Mottola. Towards lightweight information dissemination in inter-vehicular networks. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, pages 20–29, 2006.

[26] Genping Liu, Busung Lee, Boonchong Seet, Chuanheng Foh, and Keokkee Lee. A routing strategy for metropolis vehicular communications. In *Proceedings of The International Conference on Information*, pages 533–542, 2004.

[27] Shabbir Ahmed and Salil S. Kanere. SKVR: Scalable knowledge-based routing architecture for public transport networks. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, pages 92–93, 2006.

[28] Ilias Leontiadis and Cecilia Mascolo. GeOpps: Geographical opportunistic routing for vehicular networks. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6, 2007.

[29] Antonios Skordylis and Niki Trigoni. Delay-bounded routing in vehicular ad-hoc networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 341–350, 2008.

[30] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications*, pages 1–11, 2006.

[31] M. Grossglauser and D.N.C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.

[32] P. Santi and D.M. Blough. The critical transmitting range for connectivity in sparse wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):25–39, 2003.

[33] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings of the 21st IEEE International Conference on Computer Communications*, pages 1079–1088, 2002.

[34] M. Abuelela, S. Olariu, and I. Stojmenovic. OPERA: Opportunistic packet relaying in disconnected vehicular ad hoc networks. In *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 285–294, 2008.

[35] Bo Gu and Xiaoyan Hong. Critical phase of connectivity in wireless network expansion. In *Proceedings of the IEEE Global Telecommunications Conference*, pages 1–5, 2010.

[36] I. Seskar, S.V. Maric, J. Holtzman, and J. Wasserman. Rate of location area updates in cellular systems. In *Proceedings of the 42nd IEEE Vehicular Technology Conference*, pages 694–697, 1992.

[37] I. Stepanov, J. Hahner, C. Becker, Jing Tian, and K. Rothermel. A meta-model and framework for user mobility in mobile networks. In *Proceedings of the 11th IEEE International Conference on Networks*, pages 231–238, 2003.

[38] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000.

[39] M. Fiore, J. Harri, F. Filali, and C. Bonnet. Vehicular mobility simulation for VANETs. In *Proceedings of the 40th Annual Simulation Symposium*, pages 301–309, 2007.

[40] Marco Fiore and Jérôme Härri. The networking shape of vehicular mobility. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 261–272, 2008.

[41] J. Härri, F. Filali, C. Bonnet, and Marco Fiore. VanetMobiSim: Generating realistic mobility patterns for vanets. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, pages 96–97, 2006.

[42] 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.

[43] Ziwei Ren, Wenfan Li, and Qing Yang. Location verification for vanets routing. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 141–146, 2009.

[44] Ziwei Ren, Wenfan Li, Qing Yang, Shaoen Wu, and Lei Chen. Location security in geographic ad hoc routing for vanets. In *Proceedings of the International Conference on Ultra Modern Telecommunications Workshops*, pages 1–6, 2009.

[45] Qing Yang, Alvin Lim, and Prathima Agrawal. GPSFR: GPS-free routing protocol for vehicular networks with directional antennas. *International Journal of Wireless & Mobile Networks*, 1(2):64–78, 2009.

[46] A.R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[47] K. Sampigethaya, Mingyan Li, Leping Huang, and R. Poovendran. AMOEBA: Robust location privacy scheme for VANET. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, 2007.

[48] Florian Dötzer. Privacy issues in vehicular ad hoc networks. In *Proceedings of the Workshop on Privacy Enhancing Technologies*, pages 197–209, 2005.

[49] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Pin-Han Ho, and Xuemin Shen. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In *Proceedings of the 27th IEEE Conference on Computer Communications*, pages 1229–1237, 2008.

[50] Krishna Sampigethaya, Leping Huang, Mingyan Li, Radha Poovendran, Kanta Matsuura, and Kaoru Sezaki. CARAVAN: Providing location privacy for VANET. In *Proceedings of the 3rd Workshop on Embedded Security in Cars (ESCAR)*, 2005.

[51] Alastair R. Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 127–132, 2004.

[52] Elmar Schoch, Frank Kargl, Tim Leinmuller, Stefan Schlott, and Panagiotis Papadimitratos. Impact of Pseudonym Changes on Geographic Routing in VANETs. In *Proceedings of the 3rd European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, pages 43–57, 2006.

[53] Latanya Sweeney. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[54] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of location privacy using dummies for location-based services. In *Proceedings of the 21st International Conference on Data Engineering Workshops*, pages 1248–1248, 2005.

[55] Hua Lu, Christian Sndergaard Jensen, and Man Lung Yiu. PAD: Privacy-area aware, dummy-based location privacy in mobile services. In *Proceedings of the 7th International ACM Workshop on Data Engineering for Wireless and Mobile Access*, pages 16–23, 2008.

[56] Yunpeng Zang, Lothar Stibor, Georgios Orfanos, Shumin Guo, and Hans-Juergen Reumerman. An error model for inter-vehicle communications in highway scenarios at 5.9GHz. In *Proceedings of the 2nd ACM International Workshop on Performance evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 49–56, 2005.

[57] Ivan Stojmenovic, Mark Russell, and Bosko Vukojevic. Depth first search and location based localized routing and qos routing in wireless networks. In *Proceedings of the 2000 International Conference on Parallel Processing*, pages 173–185, 2000.

[58] Dongjin Son, A. Helmy, and B. Krishnamachari. The effect of mobility-induced location errors on geographic routing in ad hoc networks: analysis and improvement using mobility prediction. 3(3):233–245, 2004.

[59] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, Mar 2000.

[60] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[61] Holger Fler, Jrg Widmer, Michael Ksemann, Martin Mauve, and Hannes Hartenstein. Contention-based forwarding for mobile ad hoc networks. *Ad Hoc Networks*, 1(4):351–369, 2003.

[62] Qing Yang, Alvin Lim, Shuang Li, Jian Fang, and Prathima Agrawal. ACAR: Adaptive connectivity aware routing protocol for vehicular ad hoc networks. In *Proceedings of 17th International Conference on Computer Communications and Networks*, pages 1–6, 2008.

[63] K. Mehta, Donggang Liu, and M. Wright. Location privacy in sensor networks against a global eavesdropper. In *Proceedings of the IEEE International Conference on Network Protocols*, pages 314–323, 2007.