Connectivity-Aware Routing in Vehicular Ad Hoc Networks

by

Nizar H Alsharif

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2017

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner

Dr. Mengchu Zhou

Professor, New Jersey Institute of Technology

Supervisor

Dr. Xuemin (Sherman) Shen

Professor, University of Waterloo

Internal Member

Dr. Otman Basir

Professor, University of Waterloo

Internal Member

Dr. Liang-Liang Xie

Professor, University of Waterloo

Internal-external Member

Dr. Raouf Boutaba

Professor, University of Waterloo

Abstract

Vehicular ad hoc networks (VANETs) is a promising emerging technology that enables a wide range of appealing applications in road safety, traffic management, and passengers and driver comfort. The deployment of VANETs to enable vehicular Internet-based services and mobile data offloading is also envisioned to be a promising solution for the great demand of mobile Internet access. However, developing reliable and efficient routing protocols is one of the key challenges in VANETs due to the high vehicle mobility and frequent network topology changes. In this thesis, we highlight the routing challenges in VANETs with a focus on position-based routing (PBR), as a well-recognized routing paradigm in the vehicular environment. As the current PBR protocols do not support VANET users with connectivity information, our goal is to design an efficient routing protocol for VANETs that dynamically finds long life paths, with reduced delivery delay, and supports vehicles with instant information about connectivity to the infrastructure.

The focus of this thesis will be on predicting vehicular mobility to estimate inter-vehicle link duration in order to support routing protocols with proactive connectivity information for a better routing performance. Via three stages to meet our goal, we propose three novel routing protocols to estimate both broad and comprehensive connectivities in VANETs: iCAR, iCAR-II, and D-CAR. iCAR supports VANET users with instant broad connectivity information to surrounding road intersections, iCAR-II uses cellular network channels for comprehensive connectivity awareness to Roadside Units (RSUs), and finally D-CAR supports users with instant comprehensive connectivity information without the assistance of other networks. Detailed analysis and simulation based evaluations of our proposed protocols demonstrate the validity of using VANETs for Internet-based services and mobile data offloading in addition to the significant improvement of VANETs performance in terms of packet delivery ratio and end-to-end delay.

Acknowledgements

All praise is to Allah for giving me the ability and knowledge to accomplish this thesis.

Several people deserve sincere recognition. First and foremost, I would like to express my sincere gratitude to my advisor, Professor Xuemin (Sherman) Shen, for his encouragement, support, deep insight, and invaluable advices during my graduate studies at the University of Waterloo. Prof. Shen's commitment with high quality research is an inspiration for pursuing excellence and success not only in academic endeavors but in every other aspect of life.

I would like to thank Prof. Otman Basir, Prof. Liang-Liang Xie, and Prof. Raouf Boutaba for being my Advisory Committee. Their careful reading and comments have significantly improved the quality of this thesis. I am also very thankful to Prof. Mengchu Zhou, from New Jersey Institute of Technology, for serving as the external thesis examiner and for his insightful comments and observations. I would like also to thank Dr. David Saunders for chairing my thesis examination committee.

My appreciation goes to my colleagues and friends at the Broadband Communications Research (BBCR) Group and the Vehicular Ad Hoc Networks subgroup. It has been a privilege for me to work with so many bright people. I am sincerely grateful to Dr. Muhammad Ismail, Dr. Albert Wasef, Dr. Sandra Cspedes, Dr. Hassan Omar, Dr. Khalid Aldubaikhy, Dr. Haibo Zhou, Dr. Miao Wang, Dr. Ahmed Hamza, Dr. Essam Altubaishi, Dr. Ammar Alshamrani and Dr. Khaled Almotairi.

This dissertation could not have been completed without the company of so many friends who shared happiness and tough times with me for the past years. Among them, I thank my dear friends Omar Alrumaih, Omar and Abdullah Bin Humaid, Badr Lami, Maazen Alsabaan, Majed Alotaibi, Abdulaziz and Nawaf Almutairi, Hesham and Talal

Alharbi, Yasser and Khalid Alhazmi, Faisal and Mohammed Aldawsary, Ahmed Alqurashi, Mohammed Fekri, Hani Aldhubaib, Abdullah Alshehri, Hani Alarifi, and Ehsan Khoja.

With great appreciation, I acknowledge the financial support from my employer, Albaha University - Saudi Arabia, for granting me the scholarship and giving me the opportunity to enhance my experience and knowledge. The acknowledgment is extended also to the Saudi Cultural Bureau in Canada for their administration and supervision during my academic journey. Grateful acknowledgement is also made to the University of Waterloo for many assistantship awards and for its outstanding innovative research environment.

Forever I shall remain indebted to my family for their unconditional love, continuous support, and sincere prayers. My deepest gratitude goes to my father, Hassan Alsharif, my mother, Nora Sugati, and my siblings, Husam, Basem, Mohammed, Sahar, and Bodour. My greatest love goes to my dearly beloved wife, Amani Alfageah, and to my beautiful great kids, Jolie, Abdullah and Sarah. I would never get this far without you.

Dedication

To my dear parents,

Nora Sugati & Hassan Alsharif

Table of Contents

List of Tables **xii**

List of Figures **xiii**

List of Nomenclatures **xvi**

1 Introduction **1**

1.1 Vehicular Ad Hoc Network 1

1.2 Unicast Routing in VANETs 4

1.3 Research Motivation and Objectives 6

1.4 Summary of Research Contributions 7

1.4.1 iCAR: Junction-to-Junction Connectivity Aware Routing 8

1.4.2 iCAR-II: Cellular Network Assisted VANET Routing 9

1.4.3 D-CAR: Distributed Overlay Routing with Comprehensive Connectivity Awareness 10

1.5 Network Model 11

1.6 Thesis Outline . . . 12

2 **Background and Related Work** **14**

2.1 Routing Protocol Classification . . . 14

2.2 Vehicular Mobility . . . 19

2.3 Related Work . . . 23

2.3.1 Internet Access and Mobile Data Offloading in VANETs . . . 23

2.3.2 Mobility Prediction based Link Lifetime Estimation in VANET . . . 25

2.3.3 PBR Protocols . . . 26

3 ***i*CAR: Intersection-based Connectivity Aware Routing** **31**

3.1 *i*CAR: Protocol Description . . . 32

3.1.1 Road Segment Evaluation (RSE) . . . 33

3.1.2 Validity Period Calculation (VPC) . . . 36

3.1.3 Next-Junction Selection . . . 40

3.1.4 Next-Hop Selection . . . 40

3.2 Performance Evaluation . . . 41

3.2.1 Simulation Setup . . . 41

3.2.2 Simulation Results . . . 43

3.3 Summary . . . 46

4 *i*CAR-II: Infrastructure-based Connectivity Aware Routing 48

4.1 *i*CAR-II: Protocol Description 49

4.1.1 Beaconing and Neighbourhood Awareness 51

4.1.2 Mobility-based Link Lifetime Estimation 53

4.1.3 Broad Connectivity Evaluation 60

4.1.4 City-level Network Topology and Data Routing 63

4.2 Performance Evaluation 66

4.2.1 Simulation Setup 68

4.2.2 Simulation Results 69

4.3 Summary 82

5 D-CAR: Dynamic Connectivity-Aware Routing for Internet-based Services 83

5.1 D-CAR: Protocol Description 84

5.1.1 Framework for Link Lifetime Prediction 85

5.1.2 Framework for Network Connectivity Prediction 91

5.1.3 Data Packets Routing 95

5.2 Performance Evaluation 98

5.2.1 Simulation Setup 98

5.2.2 Simulation Results 100

5.3 Summary 109

6 Conclusion and Future Work **111**

6.1 Summary and Conclusion 111

6.2 Future Work 114

References **117**

List of Tables

3.1 Simulation Parameters for iCAR Evaluation 42

4.1 Simulation Parameters for iCAR-II Evaluation 69

5.1 Simulation Parameters for D-CAR Evaluation 100

5.2 Correlation and RMS between the NNs Model Outputs and Target ES . . 100

List of Figures

1.1 VANET Structure 2

1.2 Research Stages 8

1.3 General Network Model 11

2.1 Taxonomy of Routing Protocols in VANETs 15

3.1 RSE Control Packet (CP) Fields 34

3.2 VPC Algorithm 37

3.3 Example of VPC operation at the current CP forwarder v_m 39

3.4 iCAR Packet Delivery Ratio with PGR = 50 packets/sec 44

3.5 iCAR Packet Delivery Delay with Avg Vehicular Density = 6 $veh/lane.klm$ 45

3.6 iCAR Communication Overhead 46

4.1 Defining Driving Directions 52

4.2 An Example Case with Two Mobility Scenarios 58

4.3 Calculating RLL in Road Segment Evaluation 62

4.4 Approximate simulation map 68
4.5 Average delay for network information dissemination within a road segment using beacons 71
4.6 Probability of initiating RSE procedure for a vehicle entering the road segment 73
4.7 Average RLL in a road segment 74
4.8 Percentage of Connected Road Segments to the Core Network using *iCAR-II* 75
4.9 Average Packet Delivery Ratio ($PDR2$) using 4 RSUs and PGR = 10 packet/sec 77
4.10 Average Packet Delivery Ratio ($PDR2$) using 4 RSUs and vehicle density of 70 vehicle/lane.klm 78
4.11 Average Packet Delivery Delay (PDD) using 4 RSUs and PGR = 10 packet/sec 78
4.12 Average Packet Delivery Delay (PDD) using 4 RSUs and vehicle density of 70 vehicle/lane.klm 79
4.13 LTE Routing-Control Messages 81
4.14 Average LTE Routing Messages per Vehicle per Hour 81

5.1 The Simulated Area around University of Waterloo 99
5.2 Correlation between the NNs outputs and the Target ES with $W = 10$ *sec* 101
5.3 Percentage of Connected Road Segments to RSUs with $W = 15$ 103
5.4 Average CRT in the Network with 2 RSUs 103
5.5 Percentage of Buffered, Sent, and Received Data packets with 2 RSUs and W= 15 Seconds 105

5.6 Average Packet Delivery Ratio with 2 RSUs 105

5.7 Average Packet Delivery Delay with 1 RSU and low data traffic 107

5.8 Average Packet Delivery Delay with 2 RSUs and moderate vehicular traffic 108

List of Nomenclatures

m, n, o, q	Indices for vehicles
i, j	Indices for intersections
v_m	A vehicle with an identifier m
I_i	An intersection with an identifier i
$e_{i,j}$	A road segment between two adjacent intersections I_i and I_j
κ	An index for mobility scenario
z, ω	Indices for routing paths
$P_z(i)$	A routing path with an identifier z from intersection I_i to the core network
OBU	On-Board Unit
RSU	Roadside Units
LCs	Location Centres
CP**s**	Probe Control Packets
NN**s**	Neural Networks
NSI	Network Status Information.
LRT	The minimum predicted Link Residual Time
RLL_{R_i}	The minimum predicted road-level connectivity duration for $e_{i,j}$

$Q_{i,j}$	Road segment $e_{i,j}$ score
CRT	The minimum predicted Connectivity Residual Time
$Table_{v_m}$	The routing table for v_m
R	The transmission range for line-of-sight cases
$\hat{R}$	The transmission range for non-line-of-sight cases
Loc_{v_m}	Cartesian coordinates of v_m location
S_m	The reported average speed of v_m
u_m	The velocity vector of v_m
ES_m	The predicted future speed of v_m
LS_m	The average speed of v_m's leading vehicles
S_{max}	Averaged maximum speed
LV_m	Set of leading vehicles for v_m
W	The predicting time window
Sig_m	The turning signal status of v_m
Dir_m	A binary variable indicating the driving direction of v_m
$RSSI_m$	The RSSI value of v_m
k_m	Vehicular density in front of v_m
K_J	Traffic jam density in urban environment

f_m	A binary variable indicating if v_m is a *front* vehicle
l_m	A binary variable indicating if v_m is a *leading* vehicle
H_m	A binary variable indicating if v_m is moving towards a common intersection
$d_{m,n}$	Distance between two v_m and v_n
$d_{m,i}$	Distance between two v_m and I_i
$\mathbb{N}$	Set of neighbouring vehicles
N	Set of potential forwarders
$\mathbb{L}$,$\mathbb{F}$,$\mathbb{R}$	Sets of adjacent road segments representing *left*, *front*, and *right* road segments respectively
$\mathbb{M}_m$	Set of vehicles between two control packet forwarders
$\mathbb{L}_m$	Set of LRTs with vehicles $\in \mathbb{M}_m$
$\mathbb{C}_{m,n}$	Set of common neighbouring vehicles between v_m and v_n
$\mathbb{P}$	Set of available routing paths
$\mathbb{K}$	Set of possible mobility scenarios
P_{RSE}	Probability of initiating a road segment connectivity evaluation procedure
p_{SD}	Probability of sending probe packets to measure road-level delivery delay
$Last\_B$	Timestamp for the last mobility information update in the outgoing beacon messages

τ_{Bt}	Time period for updating mobility information in the outgoing beacon messages
$\tau_{LinkUpdate}$	Time period for updating neighbouring vehicles mobility information in routing tables
$Last\_Upd_{v_i}$	Timestamp for v_m mobility information update
ϵ_{vel}	Change in speed threshold to update a neighbouring vehicle's mobility information in routing tables
$D_{i,j}$	The experienced road-level packet delivery delay for $e_{i,j}$
$D_z(i)$	The experienced packet delivery delay for a routing path $P_z(i)$

Chapter 1

Introduction

1.1 Vehicular Ad Hoc Network

Vehicular Ad Hoc Networks (VANETs) are emerging networks that employ wireless communication technologies to enable vehicles to communicate with one another, and with other communication networks. In VANETs, each vehicle is equipped with a networking device, On-Board Unit (OBU), to enable Vehicle-to-Vehicle (V2V) communications. Similar devices, Roadside Units (RSUs), are spread along the road sides to allow Vehicle-to-Infrastructure (V2I) communications. RSUs works as gateways to infrastructure, data repositories, or packet repeaters. Figure 1.1 shows the basic structure of VANETs.

VANETs have attracted the attention of both research and industrial communities, which is reflected in the interest of governments and standardization organizations. European car manufacturers have instituted the Car-to-Car Communication Consortium (C2C-CC) [1] to improve road safety and efficiency. The U.S FCC (Federal Communication Commission) has approved a 75 MHz spectrum for vehicular networks in the 5.9 GHz

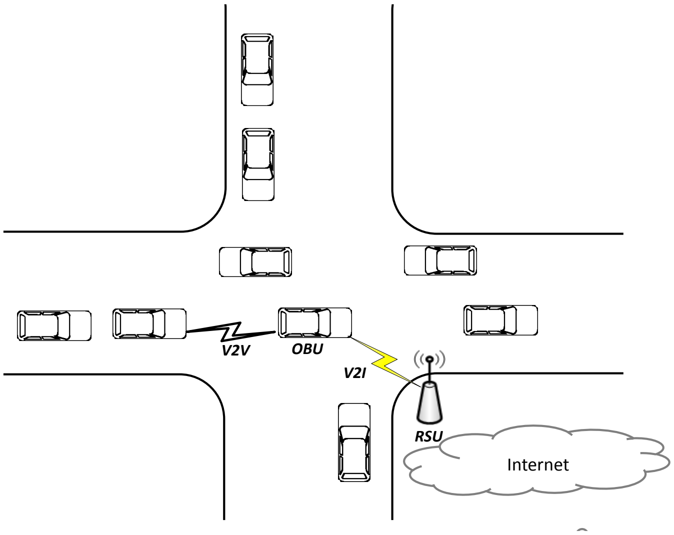

Figure 1.1: VANET Structure

band [2]. In 2008, the European Telecommunications Standards Institute (ETSI) has also allocated 30 MHz of spectrum in the same band for Intelligent Transportation Systems (ITS). In 2014, the U.S. Department of Transportation (DoT) National Highway Traffic Safety Administration (NHTSA) began taking steps to enable V2V for safety purposes [3]. The Institute of Electrical and Electronics Engineers (IEEE) also supports VANET with the IEEE 1609 family of standards for wireless access in vehicular environments (WAVE).

VANET is characterized to be decentralized and has short transmission range for its nodes. The permissible power levels of VANET give the communication signals a range of 1 km with a range of data rates between 6 and 27 Mbps [4]. VANET is a large-scale network that is frequently disconnected or partitioned, and has a highly dynamic topology, due to the high mobility of the vehicles. The network density is temporally and spatially changing. On the other hand, the mobility of VANET's nodes can be modeled and predicted because vehicle's movements are constrained by streets, and follow predictable mobility patterns. The two entities comprising VANETs, OBUs and RSUs, have sufficient computation, energy, and storage capabilities. Moreover, VANETs have hard packet delivery delay constraints, especially for safety applications.

The development of VANET is a direct response to the increasing demands of ITS services, the expectations of the automotive industry, the evolution of the Internet of Things (IoT), and the increasing demand for mobile data. Thus, VANET is designed for a wide range of applications related to safety, traffic management, and passenger comfort.

Safety applications are the main motivation for the development of VANETs. VANETs are used with the goal of spreading accurate data quickly and reliably, in order to avoid accidents and loss of life. In VANETs, vehicles help to avoid accidents through cooperation; they inform one another about their own source-of-risk behaviour, such as highway merging, and they also disseminate emergency warning messages when a hazardous status is detected, such as slippery road conditions. VANETs also improve road safety by enabling traffic lights and signs to communicate with vehicles.

In addition to safety applications, VANETs are also employed in a variety of ITS traffic management applications. Road traffic management applications focus on improving traffic flow in order to avoid traffic congestions, to reduce travel time, and to utilize the transportation infrastructure effectively. Examples include adaptive traffic lights that change according to the status of the traffic in an intersection, and direction information based on real-time traffic information.

A third type of VANETs applications can be classified as entertainment and infotainment applications. Transferring files between vehicles, accessing the Internet during trips, finding a nearby point of interest, and disseminating advertising messages about a near-by business are all examples of expected VANETs services. Recently, the deployment of

VANETs to enable vehicular Internet-based services, such as TCP-based (e.g., WWW, e-mail), FTP and P2P, and mobile data offloading is envisioned to be a promising solution for the growing demand of mobile Internet access and the anticipated mobile data explosion problem in cellular networks [5, 6].

1.2 Unicast Routing in VANETs

Designing an efficient routing protocol is required for mult-hop communication in VANETs, to deliver data packets from vehicles to RSUs, from RSUs to vehicles or from vehicles to other vehicles, when the sender and the receiver are not within the communication range of one another. Different from other networks, vehicles' high mobility and the frequent change of communication links between vehicles make the traditional topology-based routing protocols, such as AODV [7] and DSR [8], fail in VANETs as they flood the network with path finding and maintenance control messages [9]. Replacing this node-level network topology routing, vehicular communication researchers have introduced an alternative geographical location-based routing paradigm, or position-based routing (PBR), which depends on routing packets among geographical locations by arbitrary nodes, instead of routing among pre-determined nodes, in order to cope with the vehicular network environment. Studies confirm that this paradigm, PBR, outperforms topology-based routing in both urban and highway VANETs scenarios [9][10].

In PBR, packets are forwarded hop-by-hop toward the destination location. The routing decision at each intermediate forwarder is determined with respect to the position of the destination, the position of vehicles within the transmission range (neighbouring nodes), and the forwarding strategy of the protocol. Thus, each vehicle should be able to obtain its geographic location, e.g., by GPS, and share it with its one-hop neighbours. In general, PBR protocols consist of three components: 1) **Beaconing:** broadcasting a periodic message that includes the geographic location of the vehicle; 2) **Location Service:** defining a methodology that enables a source vehicle to obtain the location of a non-neighbouring destination; and 3) **Forwarding Strategy:** defining the strategy to select the next hop among neighbouring vehicles, or a next geographic anchor, toward the destination location.

Although many PBR protocols have been proposed for VANETs, as will be shown in Chapter 2, there are still some major challenges and limitations in PBR that need to be addressed. First, PBR depends on opportunistic forwarding where the existence of a communication path between the source and destination is not guaranteed, neither is the optimality of the chosen route. Only destination location and local information are available to a source vehicle prior to the start of transmission, as it is difficult for each vehicle to obtain full network connectivity information in the highly dynamic large-scale VANETs. Second, the majority of PBR protocols have not considered a realistic location service and assumed the availability of destination location in their performance evaluation. Obtaining destination location via an alternative network, such as cellular or sensor networks, can increase the communication cost, while using pure ad hoc network for location service can affect the network performance. The delay encountered by routing a location update sent by a destination, an enquiry message sent by a source, and a response message sent by a server or an agent, significantly affects the accuracy of the delivered information.

Third, most PBR protocols tend to select roads with dense vehicular traffic for a better network connectivity which causes data traffic congestion. Routing protocols should consider more factors in their forwarding strategies and path planning for better routing performance. Fourth, for Internet access and mobile data offloading, vehicles need instant information about connectivity to the core network before transmission. This information includes the existence of at least one routing path to an RSU gateway, in addition to the expected quality and duration of the connection. Since PBR protocols do not support this information, a new routing paradigm is required for Internet-based services in VANETs

1.3 Research Motivation and Objectives

From the aforementioned promising applications of VANET's multi-hop communications, VANET is envisioned to play an important role in road user safety, intelligent transportation systems (ITS), users comfort as well as addressing the expected sever problem for cellular network overload due to the ever increasing demand of mobile data. This research is motivated by the fact that designing an efficient routing protocol is still a key challenge for multi-hope communication in VANETs including Internet access and mobile data offloading. Our objective is to design a protocol for VANETs that dynamically and proactively finds long-life connected paths to the infrastructure, with reduced delivery delay, and supports vehicles with this connectivity information.

Connectivity information will assist the different applications to make their transmission decision: start data packet transmission via VANETs, reschedule the transmission, or transmit via alternative network if applicable. Supporting VANET users with instant connectivity information, such as the existence of a route (or more) to the core network, the duration of this connection, and the expected packet delivery delay via this route, will not only improve the routing performance, but also preserve the network bandwidth and improve the overall VANETs performance.

With respect to the special characteristics of VANETs, our design strategy to extract connectivity information is based on utilizing the locally available real-time mobility information, sending dedicated probe messages when needed, as well as deploying static map information, in order to predict connectivity among vehicles and to the core network. Thus, we are aiming to answer the following questions:

1. How to find the remaining link lifetime between two mobile vehicles in the city scenario?
2. How to determine whether a vehicle is connected, via multi-hop routing, to the core network or not, and in case of a valid connection, what is the remaining lifetime of that connection?
3. How to support vehicles with instant and dynamically updated connectivity information?

1.4 Summary of Research Contributions

This thesis follows three steps to address the routing challenges in VANETs, that have been highlighted in Section 1.2, and give answers to the technical questions in Section 1.3:

Step 1: Supporting VANET users with instant broad connectivity information to surrounding road intersections

Step 2: With the assistance of cellular network, supporting VANET users with instant comprehensive connectivity information to RSUs

Step 3: Supporting VANET users with instant comprehensive connectivity information without the assistance of other networks

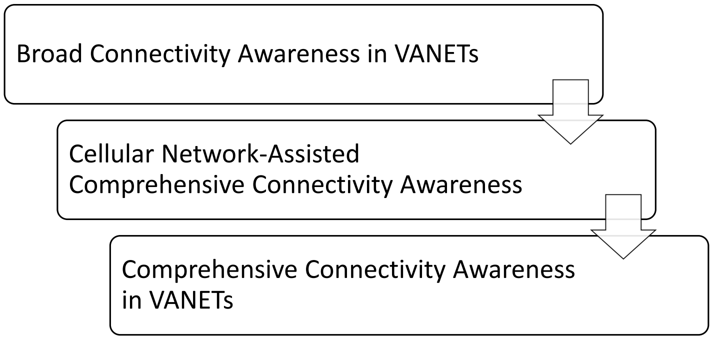

Figure 1.2: Research Stages

We define broad connectivity to be the existence of at least one path to route packets between two adjacent intersections, i.e., road-level connectivity. Comparably, we define comprehensive connectivity to be the existence of at least one path from a vehicle to a gateway RSU. Figure 1.2 describes our steps to meet the research objective in this thesis. With respect to these steps, three routing protocols have been proposed, iCAR, iCAR-II, and D-CAR. In followings, thesis contributions are summarized accordingly.

1.4.1 iCAR: Junction-to-Junction Connectivity Aware Routing

The intersection-based connectivity aware routing protocol iCAR is an improved version of the existing position-based routing protocols, and an important base for the other proposed protocols. Similar to the existing protocols, iCAR has not considered the connectivity to the core network and assumed the location service to be available. However, it supports vehicles with connectivity information to adjacent intersections and assigns scores to the connected ones for better PBR decisions. iCAR introduces the following algorithms:

- Mobility prediction based road-level connection lifetime estimation using local probe messages

- Ranking road segments for efficient next-junction selection

In iCAR, we study some key parameters in routing such as considering road-level delivery delay as a routing parameter, the dynamic updating of adjacent road segments' ranks, the selection of next packet forwarder, and the distribution of routing information.

1.4.2 *i*CAR-II: Cellular Network Assisted VANET Routing

iCAR-II is a novel infrastructure-based connectivity-aware routing protocol that deploys cellular communication for routing purposes in order to achieve comprehensive connectivity awareness for VANETs. Unlike PBR protocols, vehicles obtain instant connectivity information including routes to RSUs and start overlay source routing by the means of intersections. iCAR-II deploys distributed algorithms to obtain real-time location and mobility information in order to estimate a minimum broad connectivity lifetime and experienced packet delivery delay per road segment, and updates location centres using cellular network channels. Thus, location centers can construct a city-level dynamically updated network view, or a real-time network topology, and support inquiring senders with up-to-date connectivity information, routing paths to gateways, and destination locations. Updated comprehensive connectivity information are exchanged at intersections to proactively reach VANET users. iCAR-II includes the following contributions:

- A heuristic methodology to obtain a minimum communication link duration between each pair of communicating vehicles
- An algorithm to obtain a road-level minimum connection duration

- A distributed and dynamic routing that utilizes the introduced algorithms for efficient data routing and manages a cooperative operation between cellular networks and VANETs

1.4.3 D-CAR: Distributed Overlay Routing with Comprehensive Connectivity Awareness

D-CAR is a dynamic connectivity-aware routing protocol that supports vehicles with instant comprehensive connectivity information to the infrastructure. Unlike iCAR-II, D-CAR does not use cellular network channels. Connectivity information is carried forward and constructed from each RSU to every connected road segment. In addition to more accurate link residual time information between communicating vehicles, D-CAR enables vehicles to proactively find alternative paths, by the means of intersections, with different connection duration and expected delivery delay. D-CAR includes the following contributions:

- A neural network based short-term speed prediction module for accurate speed prediction within a given time window
- An improved mobility prediction based minimum link lifetime estimation between communicating vehicles
- A dynamic connectivity awareness module that describes the procedures to construct different paths from each RSU to every connected intersection, the remaining connection duration for each route, and the expected packet delivery delay using these routes

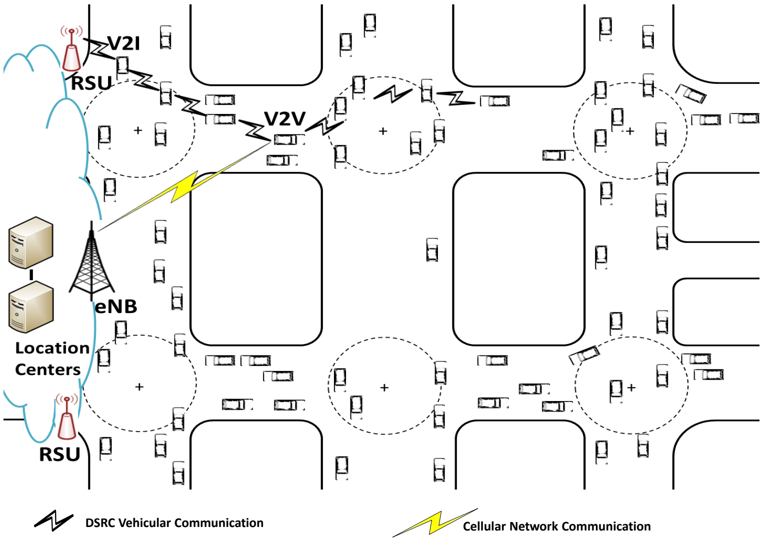

Figure 1.3: General Network Model

1.5 Network Model

The network model considers hybrid VANETs in an urban environment. VANETs consist of OBUs installed within vehicles' systems, and RSUs installed at the major city intersections. OBUs are able to obtain geographic location, mobility vectors, and turning signal status information, to share it with nearby vehicles. Periodic local sharing of driving conditions, e.g., every 100 msec, via beaconing messages is required for safety applications [11]. RSUs are VANETs gateways to the core network, i.e., Internet. Multi-hop forwarding is enabled to extend the coverage of RSUs and allow non-neighbouring vehicles to access the core network. Vehicles participate in multi-hop forwarding using their own OBUs, i.e., have sufficient inducements to forward packets belonging to other vehicles. All OBUs are synchronized and have access to identical digital maps with well-defined road segments, driving directions, and intersections.

As urban area is considered, road segments are bounded by controlled intersections and have variable length, width, and vehicles densities. The general network model is presented in Figure 1.3. In addition to the VANET, the model includes cellular networks eNBs and a set of location servers on the core network forming Location Centers (LCs). Cellular communications are considered only in our second proposed protocol, *i*CAR-II. Location centers play an important role in PBR and in our design as well. They receive a huge amount of updates, maintain updated network topology and vehicles locations, and respond to vehicles' inquiries. LCs can consider a design of distributed location servers that matches the geographically distributed nature of VANET. For example, a city-road map can be divided into a number of vicinities and each server is responsible for one or more vicinities. Adjacent vicinities can exchange their real-time road-level network topology to have a wider network view, and a proper hierarchical server architecture will enable obtaining any destination's location in the network. The details of LCs physical design such as map division and servers' management and allocation are out of our scope, and LCs will be considered as one logical unit in our system.

1.6 Thesis Outline

This thesis is organized as follows:

- Chapter 2 provides the background material and related work for this research. As the proposed protocols integrates mobility prediction and routing in the vehicular environment, this chapter covers mobility models and routing protocols in VANETs. The chapter reviews the related work in three areas: (1) Internet access and mobile data offloading in VANETs; (2) Mobility Prediction based Link Lifetime Estimation in VANET; and (3) position-based routing protocols.

- Chapter 3 presents our proposed protocol *i*CAR. It includes a description of its four components followed by the performance evaluation of the proposed protocol. The four components consisting *i*CAR are: (1) Road segment evaluation ; (2) Validity period calculation; (3) Next-junction selection; and (4) Next-hop selection.

- Chapter 4 presents our second routing protocol, *i*CAR-II. This protocol is presented with respect to its four components: (1) Beaconing and neighbourhood awareness; (2) Mobility-based link lifetime estimation; (3) Road segment connectivity estimation; and (4) City-level network topology and data packet routing. *i*CAR-II performance evaluation is followed.

- Chapter 5 introduces the third routing protocol, D-CAR. D-CAR consists of three modules: (1) Neural networks based link lifetime prediction; (2) Network connectivity prediction; and (3) Data packet routing. A performance evaluation section is presented after the details of D-CAR.

- Chapter 6 highlights the thesis findings and major results. This chapter also gives some insight on interesting and challenging directions for future research.

Chapter 2

Background and Related Work

2.1 Routing Protocol Classification

A routing protocol describes the procedure that two communicating entities, that are not in communication range of each other, use to exchange information. This includes the rules to establish a route, the strategy of forwarding data packets, the action to maintain the route, and the procedure to recover from a routing failure. In general, routing protocols are classified according to communication pattern into three main categories: unicast, multicast and broadcast [9, 10, 12]. Unicast routing is the operation of performing data communication from a single source to a single destination via a single route. In contrast, multicast is the operation of delivering the same message from a source to a group of members. If the intended members are identified by their geographic location, the routing is identified as geocast. Broadcast is the operation of disseminating the same message from

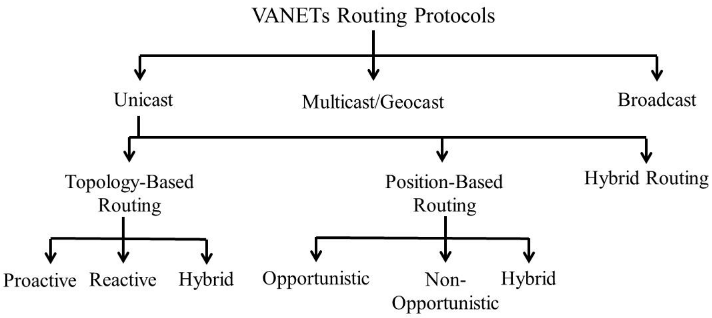

Figure 2.1: Taxonomy of Routing Protocols in VANETs

a source to all network members. Our focus in this thesis is on unicast routing protocols in VANETs.

Figure 2.1 shows a general classification of VANETs routing protocols. Many protocols have been proposed for different applications and scenarios. In the following, we describe some important features and attributes that a routing protocol can characterized by and classified accordingly. In literature, two common routing paradigms are used for multi-hop wireless routing, the traditional topology-based routing and the position-based routing [9, 10, 13]. Topology-based protocols require full path information where every communication entity maintains a routing table. Path information is acquired either pro-actively or on demand (reactive routing). On the other hand, PBR protocols require only the *location* information of the transmitting node, its neighbouring nodes and the destination node. According to the data type and delivery requirements, PBR protocols can be further divided into opportunistic, non-opportunistic, and hybrid routing protocols. Opportunistic routing is designed for delay tolerant networks (DTNs) which consider intermittent connectivity, while non-opportunistic routing considers the existence of at least one path and is designed for dense networks. Hybrid protocols are designed for partial network connectivity. Below, we highlight the differences between these strategies and describe some other important features and attributes that a routing protocol can be characterized by and classified accordingly.

Proactive and Reactive Routing

Most traditional topology-based routing protocols are proactive (table-driven), reactive (on-demand) or hybrid. Topology-based protocols use information for existing links in the network to determine the route. In proactive routing, nodes maintain a routing table to all other reachable nodes (destinations). Constructing and maintaining the table requires constant broadcast of control packets. In VANETs, proactive protocols (e.g., FSR [14]) use significant amounts of the available bandwidth to keep available lookup table but provide low latency due to the absence of route discovery or destination locating procedures. In contrast, reactive routing finds a path between two entities only when needed, and maintains routes in use only. Typically, reactive routing protocols (e.g., AODV [7], DSR [8]) use a route discovery procedure to find a path between source and destination before starting data packets transmission. Query packets are flooded into the network to find the best path to a certain destination. Reactive protocols define the way to control this flooding and to maintain the link between the end entities. Hybrid topology-based routing protocols (e.g., ZRP [15]) maintain available neighbourhood routing information in a proactive manner and use the discovery phase of reactive routing as needed. Many research works [9, 10, 13] show that topology-based routing does not perform well in VANETs and has a scaling problem.

Opportunistic and Non-opportunistic Routing

PBR makes the routing decision based on the geographic position information of nodes. PBR is more robust and promising in VANETs as the links state information exchange and maintinance of existing links information are not required. PBR protocols can be classified into opportunistic and non-opportunistic routing protocols. Opportunistic routing protocols (e.g., VADD [16]) consider VANET as a Delay Tolerant Network (DTN) where the link existence between source and destination is not guaranteed and the vehicles depend on their physical movement to deliver packets. Vehicles store, carry and forward packets to a closer vehicle to the destination, or a vehicle that has a better opportunity to carry packets to that destination. A recent study [17] has considered using buses and taxis to disseminate data in VANETs using external storage units at intersections working as "*drop boxes*". On the other hand, non-opportunistic routing assumes the existence of a path between source and destination (e.g., GPSR [18]). Thus, when a packet reaches a vehicle with no neighbour closer to the destination than the vehicle itself according to the forwarding strategy, the forwarding strategy is considered to have failed and a recovery strategy is required to deal with this failure. This failure is called local maximum as the forwarding strategy has made the maximum local progress for the current vehicle. Hybrid routing protocols apply a combination of opportunistic and non-opportunistic routing; for example, using opportunistic routing as a recovery strategy for a non-opportunistic routing protocol.

Anchor-based and Node-based Routing

VANETs routing protocols can be classified into anchor-based routing and node-based routing, which are also called overlay and non-overlay routing. In node-based routing, the routing protocol operates at the node-level and the routing decision is taken by individual nodes (e.g., GPSR [18]). On the other hand, an anchor-based routing protocol (e.g., A-STAR [19]) operates on some particular anchors overlaid on the top of the network. Anchors can be geographic locations that have high importance in the routing decision such as road intersections. Thus, the design of anchor-based routing protocol considers routing at anchors and routing between them. Routing between anchors is usually the simple greedy routing, where the next-hop is the closest node to the next intersection, while routing at anchors considers a variety of forwarding strategies and next anchor selection parameters.

Source Routing and Distributed Routing

In source routing, the path between source and destination that packets should traverse is determined by the source node. The source appends the path information to the packet header by means of a set of node IDs (e.g., AODV) or geographic anchors (e.g. GSR [20]). On the other hand, distributed routing protocols take the routing decision at each node or anchor. A hybrid routing protocol is also possible by considering source routing with flexibility to update the path on-the-fly (e.g., DSR [8]).

Offline Information Based Routing and Real-time Information Based Routing

In PBR, packets are forwarded to neighbours that are closest to their final destinations or have a better chance to deliver them. Recent protocols consider a higher level of view by taking into consideration road maps and junction information. The selection of the road segment that a packet should traverse or the next anchor depends on the forwarding strategy of the protocol. PBR protocols make their decision of selecting the next hop or next anchor based on a variety of parameters and information. Based on the information required to select the forwarding path, advanced PBR protocols can be further classified into two categories: offline information-based routing protocols and real-time information-based routing protocols.

Offline information-based routing protocols utilize static information, such as city maps, road width, and bus routes, or statistical information, such as average traffic density at certain time for each road, in order to assign weight for different network's edges and select the best routing path accordingly. Protocols with the assumption of availability and accuracy of such information (e.g., VADD [16]) outperform the ordinary PBR protocols.

On the other hand, recent PBR routing protocols use real-time traffic information to dynamically route packets toward a destination via paths having better momentary conditions (e.g. GyTAR [21]). Obtaining real-time traffic information is a challenge for this type of protocol; however, real-time traffic information-based routing protocols can outperform statistical information-based routing protocols especially when the variance of the statistical information is high. For example, in the case of car accidents or road closures due to constructions, real-time traffic information helps the protocol adapt to the current road condition and maintain its routing performance.

2.2 Vehicular Mobility

The unique characteristics of vehicular mobility influence the complexity of VANET studies. The high speed of vehicles, the constrained mobility patterns, the temporal and spatial variation in vehicles densities, and the clustering of vehicles at intersections are examples of these characteristics. Vehicles movement is restricted by roads, traffic rules, speed limits, and sometimes, the movement of surrounding vehicles. In addition to traffic engineering fields, these phenomena have been studied by technology developers to capture a level of realism in simulating vehicles movement for better validation of new technologies. Thus, a large variety of mobility models have been proposed for different purposes and needs.

A mobility model is a systematic description of a node's movement; how it changes its speed, acceleration, and mobility direction over time. In literature, vehicular mobility models can be classified, according to the level of details of the interaction between vehicles and the required/provided information, into three classes: macroscopic, mesoscopic, and microscopic [22]. Macroscopic models considers gross quantities, such as vehicular density and average speed of vehicles, and deal with vehicular traffic according to fluid dynamics. On the other hand, microscopic models consider individual vehicles mobility and pay attention to the driver behaviour and the interaction between vehicles [23]. The level of details in the mesoscopic models is located in the middle between macroscopic and microscopic models. For example, in mesoscopic models, individual vehciles are considered and charectrized independentally and identically [22].

As the previous classification seems to be very broad, the available mobility models have been categorized differently in literature [22, 23, 24]. Mobility models vary in defining parameters related to city maps, vehicular traffic generation, trip sources and destinations, trip trajectories, vehicle categories, human driving behaviours, intersection management, and more. Models have been designed for one or more of these attributes, and larger projects include comprehensive models and different engines for optional model selection. Following is a list of the main categories for developing mobility models for vehicular mobility:

Synthetic Models: The most well-know category which considers developing mobility models based on mathematical models to reflect realistic vehicular physical movement.

Survey-based Models: Where models are generated using real data statistics by designing a generic mobility model that is able to reproduce the observed behaviour.

Trace-based Models: In which generic mobility models are extracted from movement traces. This type of model generating becomes more common as it is faster, less complicated than synthetic models, and many projects started to make trace data available.

Traffic Simulator-based Models: Some commercial companies and research teams have developed realistic traffic simulators using sets of complicated synthetic models. These simulators, such as SUMO [25] and VISSIM [26], have been verified by real traces and survey data and showed the ability to simulate urban microscopic vehicular mobility.

In addition, synthetic models can be further classified into five classes [27]: *Stochastic Models* which include models with pure random movement, *Traffic Stream Models* which consider fluid hydrodynamics for vehicular mobility, *Car Following Models* which consider the effects of the vehicles ahead on the driver's behaviour, Queuing Models roads and vehicles as FIFO queues and entries respectively, and *Behavioural Models* which consider a set of behavioural rules, such as social influences, to determine the vehicle's movement. Below is a briefly illustrate a basic car following model.

Car Following Mobility

In car following mobility (CFM), the behaviour of the vehicle movement is related to the vehicle, or a group of vehicles, ahead. The fundamental basic rule is to keep a safe distance ahead. CFM models fall in the microscopic category where the details of individual vehicle mobility is considered. In CFM models, the vehicle location, velocity and acceleration are functions of different inputs, stimulating its mobility pattern, such as the distance to the front vehicle and the current speed of both vehicles. Other inputs in different models increases the level of realism considered, such as the driver's attitude and reaction time and the characteristics of the vehicles under consideration. CFM models often describes rules for lane changing. These models describe vehicle movement in multi-lane highways or independent road segments; however, models becomes more complicated in simulations where stop/priority signs and traffic lights are present.

While most CFM models are time-continuous defined by ordinary differential equations of kinematics, the discrete time framework of Cellular Automaton (CA) is also used in simulations. In CA, the road segment is divided into cells, where each cell is occupied by at most one vehicle. The mobility model then describes the rules of determining the existence and the velocity of a vehicle in a certain cell based on the previous status of the vehicle and the status of the surrounding cells. For example, the following simple algorithm [24] determines the updated speed of a vehicle i after Δt time units, $S_i(t+\Delta t)$, in a highway lane using CFM based on CA:

Step 1: **If** $S_i(t) < S_{max}$ **then** $S_i(t+\Delta t) = S_i(t) + 1$
Step 2: **If** $S_i(t+\Delta t) \geq C_j - C_i$ **then** $S_i(t+\Delta t) = C_j - C_i - 1$
Step 3: **If** $S_i(t+\Delta t) \geq 1$ **then** with probability $\rho :\ S_i(t+\Delta t) = S_i(t+\Delta t) - 1$

In the first step, a default acceleration is applied by increasing the speed one unit, every time unit, until reaching a maximum speed S_{max}. The second step accounts for breaking when reaching a leading vehicle j, where C_i and C_j denote the cells occupied by vehicle i and vehicle j respectively. The third step includes the randomness of the driver's behaviour. After determining the speed of vehicle i, its location is updated to be $C_i + S_i(t+\Delta t)$.

2.3 Related Work

The problem of VANET routing for Internet packets and mobile data offloading is a recent research trend, and few studies have considered its various challenges. On the other hand, mobility prediction-based connectivity-aware routing in VANETs is a renewed research area that has been investigated by different researchers. Therefore, related studies can be divided into three parts: 1) studies that consider Internet access and data offloading in VANETs, 2) studies that deploy mobility-based link lifetime prediction to improve routing in VANETs, and 3) studies that consider analytical methods for efficient PBR routing in VANETs.

2.3.1 Internet Access and Mobile Data Offloading in VANETs

The idea of *drive-thru* Internet, where moving vehicles obtain low-cost Internet access from roadside access points, was introduced by Ott and Kutscher in [28]. After that, several studies have considered Internet access and mobile data offloading using VANETs [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 5]. The feasibility and throughput of one-hop V2I Internet access are studied in [29, 30, 31, 32]. Then, cooperative download from an access point on a highway is proposed in [33] to show the feasibility of maximizing the portion of downloaded data from the Internet via multi-hop cooperation. In [34, 35], different strategies to optimize RSUs placement are proposed to improve the performance of multi-hop Internet access in VANETs. In [36], a survey of Internet access routing protocols in the vehicular communication environment is provided. In [5, 37, 38, 39], the Internet access in VANETs is studied from the perspective of cellular data offloading.

Authors in [29] and [32] analytically investigated the troughput of one-hop drive-thru Internet. In [29], the throughput of V2I Internet access is studied with reference to the impact of road density, vehicle speed, service penetration rate, and RSUs transmission range. This throughput is further studied in [32] with an optimal access control to boost it. The throughput is enhanced by selecting an optimal transmission region within an RSU's coverage for the coordinated medium sharing of all vehicles. In addition, the MAC DCF is also considered in [30] and [31]. In [30], Tom Luan *et al.* studied the effect of vehicle's velocity on the drive-thru Internet and, accordingly proposed different DCF models to enhance its performance. Similarly, Miao Wang *et al.* studied the effect of neighbouring vehicles' density on the one-hop drive-thru Internet and proposed a density-adaptive MAC protocol for better Internet access performance.

Enabling multi-hop Internet access via RSUs has been considered for the highway scenarios in [33] and for the urban scenarios in [34] and [35]. In [33], closer vehicles to RSUs are selected to be forwarders as they can achieve faster downloads via I2V; then, the downloaded packets are forwarded to their final destinations via V2V communication. The proposed algorithm has shown to provide a maximum download and minimum delay for cooperative downloading. Both [34] and [35] have analytically studied the problem of RSU placement in VANETs, where the objective is to deploy the minimum number of RSUs while meeting certain QoS requirements. In [34], the maximum distance that an RSU can cover for delay-tolerant data packets and real-time data packets are studied differently and the planning has been done accordingly with respect to the data packet delivery delay as a QoS constraint. On the other hand, Hassan Omar *et al.* have shown in [35] the feasibility of multi-hop Internet access via RSUs placement strategy considering the probability that a vehicle can find a network path to a gateway, which is based on the traffic conditions in the deployment region.

The potential of VANETs for cellular traffic offloading is studied in [5, 37, 38]. A survey of the general mobile data offloading techcniques was provided in [5], while the challenges, research issues and possible sollutions related to the effectiveness of data offloading in the vehicular environment are discussed in [37]. In [38], the authors show that 100% of mobile data flows can be offloaded via multi-hop VANETs with the availability of link and connectivity information.

2.3.2 Mobility Prediction based Link Lifetime Estimation in VANET

The utilization of mobility prediction for long-lived routes was established early for mobile ad hoc networks (MANETs), such as in [40] and [41], where position information was used for reliable routing. In VANETs, the mobility patterns have unique characteristics, and the estimation of a link lifetime or a connection residual time based on vehicular mobility prediction becomes a new challenge. In literature, deterministic methods, such as in [42, 43, 44, 45, 46] and stochastic methods, such as those in [47, 48, 49] have been proposed to estimate link lifetime between two vehicles, or path lifetime between a source and a destination. The estimated link duration information have been deployed to construct routing paths in few protocols such as in [43] and [45].

Deterministic mobility prediction based link residual time estimation methods either utilize the position and velocity vectors information of nearby vehicles or consider cross layer parameters for mobility prediction. In [42], [43], and [46], information related to position, speed, and driving direction are used to calculate the time required for two communicating vehicles to move out of each other's communication range. Driving direction is either estimated using the velocity angle of a moving vehicle or by applying its position to a digital map. In [44] and [45], vehicle's mobility information is assumed to be unknown. Alternatively, a series of received signal strength indicator values, or signal to noise ratio, collected from each neighbouring vehicle are used to predict the residual link life time.

The collected link quality indicators form a time series for each nearby vehicle, and the remaining time before the link quality drops below a certain threshold is estimated.

On the other hand, link duration has been studied analytically in [47, 48, 49]. Key mobility parameters, such as the distribution of relative velocity, are considered to determine the expected link lifetime. In [47], the distribution of the signal-to-noise ratio is used in order to predict the probability that a link is broken in a certain time. Cellular Automata (CA) concept is used in [48] to provide an analytical framework to study key connectivity parameters such as link duration, connectivity duration, and re-healing time. The distribution of relative velocity is used in [49] to predict the relative velocity and estimate the link residual time. In addition to relative velocity, authors of [49] have considered traffic lights and turning vehicles as the main causes of link breakage.

Utilizing link residual time awareness, few studies have considered end-to-end connectivity and constructing long lifetime routes such as in [42, 45, 47, 50]. In [42], vehicles are grouped according to their driving directions and paths are constructed among vehicles from the same group with longer link residual time for more stable routes. The link duration estimation method proposed in [45] has been evaluated using a modified Dijkstra's algorithm. In [47], a reactive protocol is used to find a node-based path from a source to destination using link duration between vehicles as weights. In [50], the link duration estimation proposed in [43] has been combined with GPSR [18] for a better greedy routing performance.

2.3.3 PBR Protocols

It has been shown eailer in this thesis that PBR paradigm is more suitable in the vehicular context than the traditional topology-based routing. One of the fundamental protocols that deploys PBR for mobile environment is GPSR [18]. GPSR uses *Greedy forwarding* where packets are forwarded to nodes that are closer to the destination. When this strategy fails, GPSR uses *Perimeter forwarding* as a recovery strategy, where packets are forwarded around the perimeter of the failing region. In addition to the geographic location required by GPSR, other protocols, such as [20, 19, 51, 21], consider the availability of further network information for better routing performance. GSR [20] is an overlay routing that uses digital maps information and deploys source routing, where the shortest path, by the means of intersections, is attached to each packet. A-STAR [19] is another overlay source routing protocol that uses a statistically rated map for street-traffic aware routing. On the other hand, TIGeR [51] and GyTAR [21] are distributed routing protocols which deploy real-time vehicular traffic information for intersection-based traffic aware routing, where routing decisions are made at intersections based on local vehicular traffic information obtained from each road.

The functionality of GPSR, GSR and GyTAR routing protocols are described below as examples of PRR protocols that have been widely used as performance benchmarks for new protocols evaluation. GPSR represents the family of distributed node-based routing protocols. GSR, on the other hand, represents the family of source anchor-based routing. GyTAR is a distributed anchor-based routing protocols. While GSR uses offline map information, GPSR and GyTAR use real-time information, where GPSR deploys one-hope neighbouring vehicles position information and GyTAR deploy road-level traffic information. Thus, these protocols covers the different important aspects in routing as presented in section 2.1.

GPSR

GPSR (Greedy Perimeter Stateless Routing) [18] is a well-known PBR protocol developed originally for mobile ad hoc networks (MANETs). GPSR uses position information of one-hop neighbours exchanged in beacons to make greedy forwarding toward the destination position. GPSR requires one-hop topology information to make a local forwarding decision in addition to the destination location. GPSR greedy forwarding strategy defines the next forwarder as the progressively closest immediate neighbour to the final destination. When this strategy fails, i.e., there is no neighbour closer to the destination than the current node, GPSR uses a recovery strategy by routing around the perimeter of the failing region. As many other PBR protocols, GPSR does not specify a location service to obtain the destination position.

As a PBR routing protocol, GPSR performs well in scenarios with highly dynamic topology, such as in VANETs, as it does not require full path finding or maintaining operations. However, greedy routing in the VANETs context causes multiple local minimum events where GPSR recovers by forwarding in perimeter mode, in which a packet traverses successively closer faces of a planar subgraph of the connected VANET, until reaching a node that is closer to the destination than the position that the perimeter mode started at, where greedy forwarding is resumed. This causes a major increase in the number of intermediate forwarders and, accordingly, the end-to-end packet delivery delay.

GSR

In order to address the node-level routing challenge in the highly dynamic topology of VANETs, GSR [20] uses source PBR. By utilizing map information and planning the routes by the means of consecutive junctions, GSR overcome the problem of traversing high intermediate forwarders presented in GPSR. In GSR, the route is calculated using Dijkstra's algorithm to find the shortest path in the graph between a source and a destination. The graph is the city road map with bidirectional edges representing roads, and graph-nodes representing road intersections. Edges in GSR are not rated and only the location of the source and destination locations are required in addition to the map information. Each data packet has the full route included in it's header fields. Intermediate forwarders use greedy routing to select the next-forwarder in order to deliver packets independently to the next-junction indicated in their routes.

Although GSR is using a shortest path algorithm, the connectivity of these paths are not ensured. GSR does not use statistical or real-time traffic information to rate the map, while planning the path, which affects its performance. In dense networks and limited data traffic streams, GSR performs well and shows low delivery latency. However, in light-traffic areas, GSR fails to discover connected routes and shows low packet delivery ratios. Moreover, as GSR applies static routing, it can easily cause data traffic congestions on some road segments.

GyTAR

The improved greedy traffic-aware routing protocol, GyTAR [21], is an intersection-based routing protocol that uses real-time traffic information to dynamically select path intersections. In GyTAR, road maps are represented as junctions and road segments. Each segment is divided into a number of equal-size cells. Considering cells centres as anchors, particular vehicles leaving a road generate cell density packets (CDP) and forward them to the other end (intersection) through the road's anchors in order to collect vehicle density information. At the other end, another group of vehicles calculates the average and the variation of vehicle density per cell and disseminates the results in the intersection. Packets are forwarded from an intersection to another where the next intersection is selected based on the vehicle density information and the curve metric distance between the adjacent intersections and the final destination.

GyTAR uses an improved greedy forwarding between intersections where senders estimate the current location of their neighbours, before selecting the next forwarder, using velocity vectors information exchanged in beacons. This enhancement in the greedy forwarding strategy is to avoid selecting a forwarder that has already left the sender's transmission range or became no longer the most progressive next hop due to its mobility during the inter-beacon interval. As a recovery strategy, GyTAR considers carry-and-forward techniques to overcome the local maximum problem of greedy routing.

GyTAR is a heuristic routing approach that utilizes map information and local vehicle traffic information within the neighbouring intersections to improve routing performance. It performs better than static information based protocols such as GSR and A-STAR. GyTAR suffers from the local vehicle traffic awareness problem. The forwarding decision is taken at each intersection considering traffic density to the adjacent intersections only. In some cases, this limited vision causes packets to be bounced between two intersections or forwarded via unoptimised roads causing higher delivery delay. Also, selecting dense roads for data forwarding and path planning causes data traffic congestions and high queuing delay which degrades the network performance.

Chapter 3

*i*CAR: Intersection-based Connectivity Aware Routing

In this chapter, we propose an intersection-based connectivity aware routing protocol (*i*CAR), which combines static map and real-time traffic information, in order to improve VANET performance in city scenarios. *i*CAR calculates an adaptive lower bound of broad connectivity lifetime, which enables better routing decisions based on guaranteed connectivity information to the adjacent intersections, with a minimized cost of communication overhead. For each road, *i*CAR takes into consideration both vehicular density and average communication delay. Thus, roads with high data volume and high vehicular density have a low preference to be selected as forwarding paths, in order to avoid an increased average transmission delay. As a result, a fair distribution of packets is achieved across the network, and the overall network performance can be improved.

Similar to other PBR protocols, *i*CAR-II assumes the availability of destination location and does not consider comprehensive connectivity awareness. In the following, we describe iCAR in Section 3.1 in terms of its four components: a) Road Segment Evaluation (RSE), b) Validity Period Calculation (VPC), c) Next-junction Selection, and d) Next-hop Selection. Next, we present a simulation-based evaluation and discussion of the results in Section 3.2. The chapter is summarized in Section 3.3.

3.1 iCAR: Protocol Description

iCAR combines local real-time road condition information and static road-topology information extracted from digital maps. Real-time information is locally and dynamically calculated at each road, by sending out a probe control packet (CP) to discover connectivity and collect vehicular traffic information while traversing the road segment. CPs are probabilistically generated at each intersection to maintain updated connectivity information. Scores are assigned to each road segment, based on the volume of vehicular traffic in that road and the delay experienced by the associated CP. After that, the scores are disseminated locally in beacon packets exchanged by vehicles at the intersections. The beacons also include the validity period of each score.

Two routing strategies are employed: next-junction selection and next-hop selection. Packets are forwarded from junction-to-junction based on the next-junction selection strategy, and forwarded hop-by-hop within roads based on the next-hop selection strategy. Accordingly, we describe iCAR by its four components as follows:

3.1.1 Road Segment Evaluation (RSE)

RSE is a heuristic distributed approach aimed at evaluating the broad connectivity of road segments, as well as their suitability to be selected in packets routing paths. It also maintains a global parameter that enables the fair and accurate distribution of packets. RSE procedure is carried out by a vehicle v_m entering to a road segment $e_{i,j}$. v_m triggers the RSE with probability P_{RSE}, where P_{RSE} is a function of the road segment conditions and the remaining lifetime of the road score $Q_{i,j}$. When RSE is triggered, v_m transmits a unicast discovery packet (CP) to the center of the next road intersection. CP is forwarded hop-by-hop according to the next-hop selection strategy. Figure 3.1 shows the lightweight packet format of CP. Upon reception of CP, each forwarder (including v_m) accumulates in the field N_{total} the number of vehicles located between itself and the vehicle chosen as the next forwarder. The origination time and the number of hops h are also recorded in CP. The forwarder runs Validity Period Calculation (VPC) algorithm (described in Section 3.1.2) and updates the lifetime field if it has a shorter estimated link lifetime, before sending the packet to the next hop.

When CP reaches the next intersection, the closest vehicle to the center of the intersection, say v_n, is responsible of generating the updated score $Q_{i,j}$. v_n then announces the score across the intersection, and sends it back to the location where the RSE procedure was triggered. $Q_{i,j}$ is calculated by v_n as follows:

$$Q_{i,j} = \alpha_1 \cdot \min\Big(1, \frac{N_{avg}}{N_{con}}\Big) + \alpha_2 \cdot \Big(\frac{T}{t_{avg}}\Big) + \alpha_3 \cdot \Big(\frac{h_{min}}{h}\Big), \tag{3.1}$$

where N_{avg} is the average number of vehicles per one hop transmission distance, N_{con} is a constant representing the average number of vehicles per one hop transmission distance, based on statistics of city scenarios, T is the minimum one-hop transmission delay (i.e.,

Originator: *Vehicle_ID*	**ToG:** *Timestamp*
From: *Junction_ID*	**To:** *Junction_ID*
N_{Total}: *Total Number of Vehicles*	
Lifetime: *Minimum Link Lifetime*	
h**:** *Total number of intermediate hops*	

Figure 3.1: RSE Control Packet (CP) Fields

the delay of transmitting a similar packet with no buffering delay and perfect channel conditions), t_{avg} is the average per hop transmission delay of the CP, h_{min} is the minimum number of hops required to traverse the road segment, h is the number of hops actually traversed from v_m to v_n, and α_1, α_2 and α_3 are weighting factors for the vehicular density, the one-hop transmission delay, and the number of intermediate forwarders, respectively.

The delivery of CP at the next intersection indicates the instantaneous connectivity of the road. The information stored in CP helps the vehicle at the target intersection to assign a road score with a validity period (or lifetime) for such a score. As shown in Equation 3.1, the effect of the vehicular density on the score is upper-bounded by α_1, and N_{avg} is calculated as follows:

$$N_{avg} = \frac{N_{total}}{h}. \tag{3.2}$$

The average delay per hop indicates the delay due to both queuing in the forwarders' buffers and retransmissions. t_{avg} is calculated as follows:

$$t_{avg} = \frac{(t_2 - t_1)}{h}, \tag{3.3}$$

where t_2 and t_1 are the reception time of CP at the target intersection and its originating time, respectively.

Vehicles with variable dimensions may work as obstacles for transmission, and may reduce the effective transmission range in their vicinity [52]. A large number of obstructing vehicles result in shorter effective transmission ranges, and hence, a higher number of intermediate transmissions. *i*CAR reduces the score for road segments with relatively high number of intermediate forwarders, as shown in Equation 3.1. The minimum number of forwarders, h_{min}, is calculated as follows:

$$h_{min} = \lceil l/R \rceil, \tag{3.4}$$

where l is road segment length and R is transmission range.

When v_m triggers the RSE procedure, it sets a timer T_{max} and waits for reception of the returning $Q_{i,j}$ or another CP coming from the other side. If v_m does not receive such information before the timer expires, then v_m sets the score to zero. If a forwarder does not find a next-hop during the forwarding of CP, it sends the CP back to the originator with an indication of road disconnection. $Q_{i,j}$ is also set to zero in such a case. The $Q_{i,j}$ is announced across the intersection and a random validity period (RBP), which works as a backoff period, is set to prevent multiple CP transmissions.

The probability P_{RSE} that v_m triggers the RSE procedure when entering the road segment is designed in a way that the score, $Q_{i,j}$, is refreshed when it has a long validity period, and to allow re-computing the value before the current validity expires. Since *i*CAR considers not only the road segment connectivity, but also the packet delivery delay at the moment of $Q_{i,j}$ calculation, the renewing of $Q_{i,j}$ before the expiration time is beneficial. In Equation 3.5, we present one way to calculate P_{RSE}, where t_{rem} is the remaining validity period and C is a constant. To ensure the renewing of $Q_{i,j}$ before the validity expires,

C is related to the expected time required to traverse the particular road segment when performing the RSE procedure.

$$P_{RSE} = \begin{cases} e^{-\frac{t_{rem}-C}{2}}, & t_{rem} \geq C \\ 1 & t_{rem} < C \end{cases} \tag{3.5}$$

3.1.2 Validity Period Calculation (VPC)

The goal of VPC is to define a lower bound for the connectivity lifetime at a given road segment. In other words, it aims at predicting the time at which a communication disconnection may occur between two adjacent intersections. By using local information stored by the CP forwarders in the routing table, iCAR performs the VPC algorithm described in Figure 3.2. Once VPC is executed, it is possible to assign a validity period for each score associated with a successful CP delivery.

In Figure 3.2, each CP forwarder estimates the time required for the first link breakage in the area between itself and the destination junction of CP that falls within its transmission range. This zone is called the area of interest (AoI) of the forwarder, as illustrated in Figure 3.3. A link breakage in the AoI is detected at the time when less than one node is present in the AoI. In order to perform this detection, the forwarder employs local information, e.g., positions, velocities, and directions of neighbouring vehicles. VPC divides the vehicles within AoI into two clusters. The cluster of vehicles moving in the same direction in which CP is being forwarded, called the positive cluster (PC), and the cluster of vehicles moving in the opposite direction, called the negative cluster, or opposite cluster (OC).

The vehicle at the tail of each cluster is identified, so that the tail vehicle of the PC is referred as v_{lp}, and the tail of the OC is referred as v_{lo}. According to Figure 3.2, each CP forwarder calculates the link lifetime in its AoI based on one of the following cases:

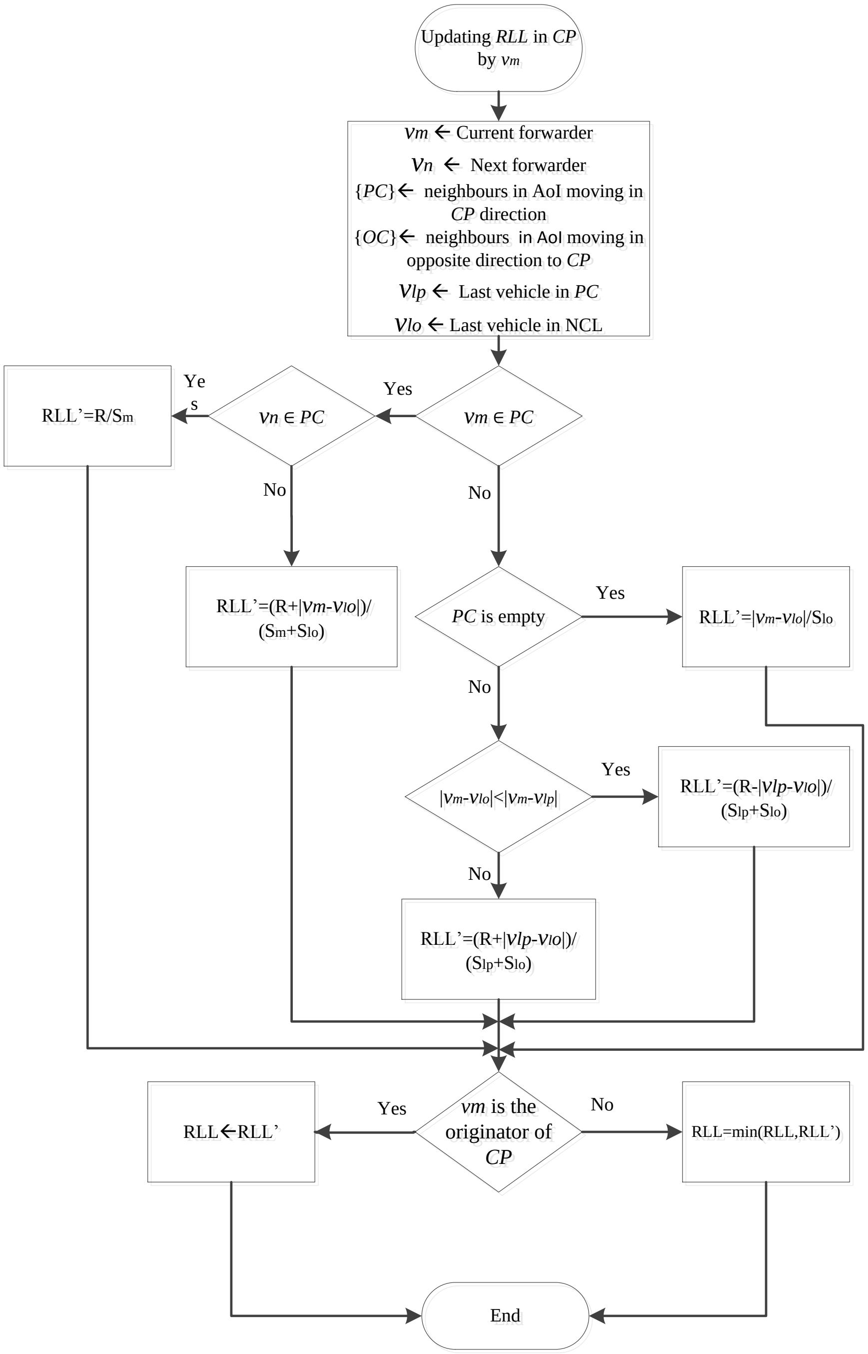

Figure 3.2: VPC Algorithm

a. Current forwarder, v_m, and next forwarder, v_n are both in PC: In this case, the first link breakage is predicted to happen at the time when v_m leaves the zone previously defined by AoI.

$$R\hat{L}L = \frac{R}{S_m} \tag{3.6}$$

b. The current forwarder v_m is in PC and the next forwarder v_n is in OC: A disconnection may happen when v_m and v_{lo} move out of each other's transmission range.

$$R\hat{L}L = \frac{R + |v_m - v_{lo}|}{S_m + S_{lo}} \tag{3.7}$$

c. Current forwarder v_m and next forwarder v_n are both in OC, and PC is an empty set: The disconnection may occur when v_{lo} leaves the AoI.

$$R\hat{L}L = \frac{|v_m - v_{lo}|}{S_{lo}} \tag{3.8}$$

d. The current forwarder v_m is in OC, and PC is not an empty set: When v_{lo} and v_{lp} are approaching each other, the minimum estimated link lifetime is the time for these vehicle to reach and then move away from each other's transmission range.

$$R\hat{L}L = \frac{R + |v_{lp} - v_{lo}|}{S_{lp} + S_{lo}} \tag{3.9}$$

On the other hand, if v_{lo} and v_{lp} are already moving away from each other, the estimated link lifetime is the time required for them to be out of each other's transmission range.

$$R\hat{L}L = \frac{R - |v_{lp} - v_{lo}|}{S_{lp} + S_{lo}} \tag{3.10}$$

As R is larger than the road width, and vehicles whithing the same road segment have parallel mobility, we neglect the effect of vehicles located in multiple lanes assuming that they move in one dimension. In Equations 3.6 to 3.10, $|v_m - v_n|$ denotes the absolute

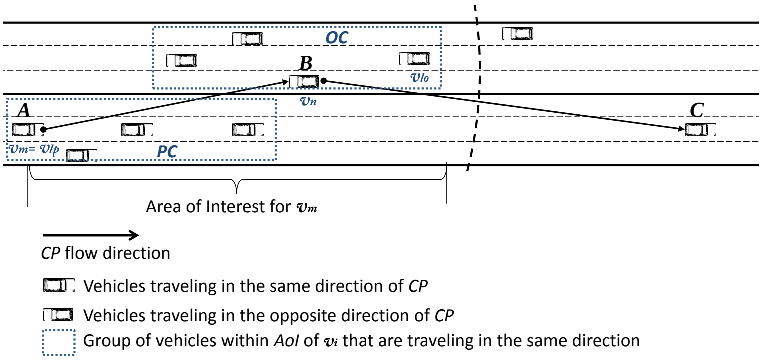

Figure 3.3: Example of VPC operation at the current CP forwarder v_m

distance between vehicle v_m and v_n, S_m represents the reported speed of a vehicle v_m, and $R\hat{L}L$ is a temporary value to calculate the road link lifetime, or broad connectivity for a certain road. The road link lifetime, RLL, is updated by each CP forwarder with respect to the calculated value $R\hat{L}L$ and the previous RLL value registered at CP, RLL^*:

$$RLL = min\{R\hat{L}L, RLL^*\} \tag{3.11}$$

The calculated lifetime is upper-bounded by the time required by the forwarder v_m, to drive for R meters in the same direction that CP is being forwarded, i.e., $t_{max} = R/S_m$, where R is the forwarder transmission radius and S_m is the speed of v_m (for simplicity we assume, in this protocol only, that neighbouring vehicles moving in the same road segment and with the same mobility direction are moving with the same speed). The final lifetime for the entire road segment would be the minimum lifetime of all the lifetimes calculated by each forwarder. This value is updated and recorded in CP before being forwarded at each hop.

3.1.3 Next-Junction Selection

When a data packet reaches an intersection, the next junction is selected from the set $\mathbb{I}$ of adjacent intersections based on each intersection's score, the geographic location of the intersection, and the packet's final destination location. The routing header of the packet is then updated accordingly. The next junction is selected to be the one with the highest q score according to the following formula:

$$q(I_j) = \beta_1 \cdot (1 - \frac{D_j}{D_i}) + \beta_2 \cdot Q_{i,j}, \quad \forall\ I_j \in \mathbb{I} \tag{3.12}$$

The first component in Equation 3.12 is the progression toward the destination, where D_j denotes the driving distance from the adjacent junction j to the destination, and D_i denotes the driving distance from the current junction i to the destination. The second component is the road segment score for the road between i and j. β_1 and β_2 are weighting factors for each component.

In this way, iCAR adopts a distributed anchor-based routing where data packets are routed from intersection to intersection based on real-time road condition information. Roads scores are updated periodically and dynamically via the RSE procedure, and exchanged via beacon messages.

3.1.4 Next-Hop Selection

iCAR employs a greedy-based next-hop selection to choose the next forwarder for a packet being transmitted between two junctions. The location of neighbouring vehicles is known by means of the beacon packets; however, vehicles may move out of each other's transmission range during an inter-beacon interval, which in turn causes wrong routing decisions and retransmissions. This problem can be avoided by predicting the existence of available forwarders based on the last reports about neighbours' positions and speeds [21]. Moreover, beacon packets may include RSSI information about neighbours, which reflect the status of signal quality and potential interference. In addition to beacons, RSSI information can be refreshed by RTS, CTS, and other data packets. iCAR selects the next-hop from the set of neighbours that are predicted to be within the communication range of the current forwarder, and that has a strong RSSI. If the algorithm fails to find a forwarder with such a strategy, the recovery strategy store-carry-and-forward is employed instead.

3.2 Performance Evaluation

In this section, we present a simulation-based evaluation of iCAR. iCAR is compared with the implementations of GPSR [18] and GyTAR [21]. GPSR is a basic PBR protocol commonly employed for performance benchmarks. GyTAR is a recent PBR protocol and one of the most closely related protocols to iCAR.

3.2.1 Simulation Setup

We have implemented a simulation for VANETs in MATLAB. The environment includes a digital city map with a grid area of 7000m×7000m and bidirectional roads. Roads vary in terms of the number of lanes: bidirectional lanes with lower vehicular traffic to represent residential areas, and roads with two to four lanes per direction to represent main connecting city roads. A total of 165 intersections with 45 controlled intersections have been included, and two different average vehicular densities (6 and 12 vehicles/lane.Km) are employed to represent low and high vehicular traffic volumes.

Table 3.1: Simulation Parameters for iCAR Evaluation

Parameter	**Value**	**Parameter**	**Value**
α_1	0.333	β_1	0.5
α_2	0.333	β_2	0.5
α_3	0.333	Simulation Duration	30 sec
N_{con}	6 $Vehicles$	Inter-beacon period	500 $mesec$
T	0.3 $msec$	R	250 m
T_{max}	$2msec$ x h_{min}	Routing Protocols	GPSR, GyTAR, iCAR
C	2 x T_{max}	Packet Size	512 $byte$
RBP	1-5 sec	Transmission Rate	12 $Mbps$
$RSSI_{thresh}$	0.6 x $RSSI_{max}$	Packet lifetime	500 $msec$

The system and simulation parameters for the operation of iCAR are described in Table 3.1. GPSR and GyTAR parameters are set according to [18] and [21], respectively. Nodes implement a FIFO packet queue, such as the access categories (AC) queues designed for WAVE's MAC layer [13], to buffer packets pending for transmission. A free space model with urban area path loss exponent is deployed to estimate the RSSI [53]. Besides attenuation, we marked 5% of the vehicles as obstructing vehicles, and P_{LOS} is calculated according to the model presented in [13]. The path loss exponent is then chosen to be LOS or non–LOS, depending on the P_{LOS} value [53]. 10% of the vehicles in each simulation run are selected to be packet sources with random destinations. Each simulation scenario is repeated five to eight times.

3.2.2 Simulation Results

The performance metrics used to compare and evaluate the proposed protocol are: packet delivery ratio (PDR), packet delivery delay (PDD), and routing overhead. The simulation results and discussion are presented as follows.

Packet Delivery Ratio

The PDR is the average ratio of packets received to packets sent. Figure 3.4 shows that iCAR outperforms both GyTAR and GPSR. iCAR and GyTAR, which are anchor-based, have significantly higher PDR than GPSR, due in part to the prediction of the existence of neighbours before transmitting packets. iCAR and GyTAR rely on the existence of vehicular traffic in order to consider a road in packets routing path. GPSR instead, frequently resorts to the recovery strategy, which results in a larger number of hops traversed and the dropping of packets before they reach their final destination.

iCAR achieves more than 15% increase of PDR comparing to GyTAR in both low and high vehicular density scenarios. This is mainly because iCAR deploys a deterministic algorithm to trigger the RSE procedure. Thus, it is expected for iCAR to always have deterministic connectivity information of the adjacent roads. On the other hand, GyTAR triggers the road connectivity evaluation procedure only when one of the *cell leaders* reaches the center of an adjacent intersection. Therefore, GyTAR's PDR is affected by traffic lights and controlled intersections: since vehicles are clustered at the road end-points during red lights, road disconnection occurs before the procedure to re-calculate the road connectivity score is triggered. In addition, the greedy routing and the convergence of packets on certain roads that have high vehicular traffic, as well as the buffering of packets during store-carry-and-forward, cause GyTAR to have multiple transmission failures, retransmissions, and

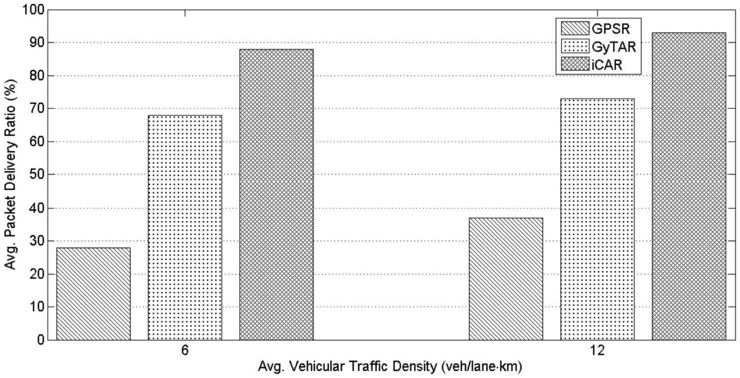

Figure 3.4: iCAR Packet Delivery Ratio with PGR = 50 packets/sec

high delivery delay, which eventually leads to packet dropping.

Packet Delivery Delay

The packet delivery delay refers to the average end-to-end delay to deliver data packets from the source nodes to packets final destinations. Figure 3.5 illustrates the average end-to-end packet delivery delay obtained from simulations by employing different packet generation rates for the low vehicular traffic density scenario. iCAR shows to have the lowest packet delivery delay among the compared protocols. Unlike GyTAR, which considers the large volume of vehicular traffic at a certain road as a positive condition, iCAR takes into consideration the actual delay required to traverse that road. Thus, alternative connected roads with less vehicular traffic and less experienced delay are considered for packets delivery. Moreover, iCAR's RSE procedure deterministically guarantees the connectivity of the road for a minimum period of time, which helps forwarders at intersections to make effective routing decisions. In this way, iCAR minimizes the use of store-carry-and-forward strategy. On the other hand, packets forwarded with GyTAR are frequently delayed when

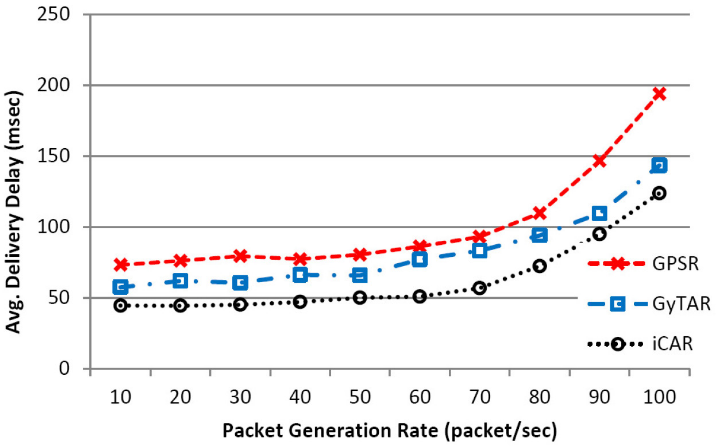

Figure 3.5: *i*CAR Packet Delivery Delay with Avg Vehicular Density = 6 $veh/lane.klm$

employing the store-carry-and-forward strategy.

Routing Overhead

In general, PBR protocols have less communication overhead than traditional reactive routing protocols, because they do not employ route discovery and maintenance control messages for every flow of packets. On the other hand, beacon packets, which are required by safety applications, are the main communication overhead for PBR protocols. GyTAR and *i*CAR introduce additional overhead when discovery packets are used to collect vehicular information along road segments. However, the frequency for generating such packets is much lower than the beaconing frequency, and the unicast nature of these discovery packets makes the introduced overhead almost negligible when compared to overhead caused by beacon packets.

Figure 3.6 shows the average control packets sent per second on each road. The results indicate that the average beaconing overhead is the same for the different routing protocols;

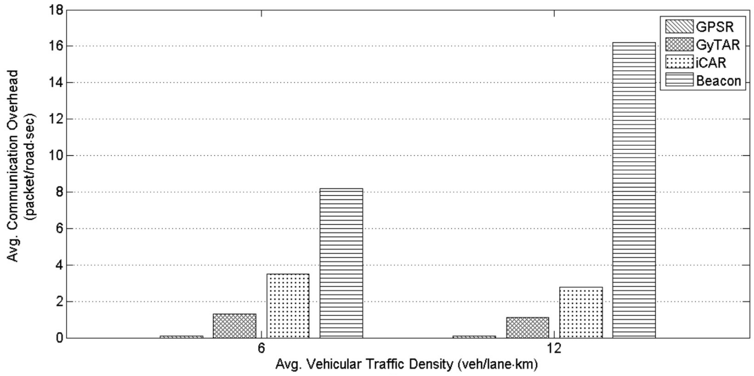

Figure 3.6: iCAR Communication Overhead

however, iCAR has a higher average of discovery packets sent compared with GyTAR, which indicates that our protocol triggers more frequently the road segment evaluation. Nonetheless, it is observed that with a higher vehicular density, the number of discovery packets is noticeably reduced. This is because iCAR relates the RSE calls with the score validity period, as shown in Equation 3.5. In both cases, the number of discovery packets is small and almost negligible.

3.3 Summary

In this chapter, we have proposed iCAR, a position-based routing protocol that improves the VANETs routing performance in dense city scenarios, by adjusting the next-junction selection procedure based on real-time traffic and delay information for each road and with a deterministic connectivity lifetime estimation. Simulation results have demonstrated that iCAR outperforms other position-based routing protocols, such as GPSR and GyTAR, in terms of higher packet delivery ratio and reduced packet delivery delay, with a negligible communication overhead.

In this chapter, only broad connectivity has been considered. Also, the reported vehicle's speed has been used for future link break prediction. Using the reported speed values in urban scenarios is not very efficient as vehicles change their speed frequently. In the next chapter, comprehensive connectivity, and an improved mobility prediction-based residual link lifetime estimation are considered, as well as more performance evaluation scenarios and results.

Chapter 4

iCAR-II: Infrastructure-based Connectivity Aware Routing

In this chapter, we present a novel infrastructure-based connectivity-aware routing protocol, iCAR-II. This protocol deploys distributed algorithms to obtain real-time location and mobility information in order to estimate a minimum broad connectivity lifetime and experienced packet delivery delay per road segment, and updates location centres using cellular network channels. Thus, location centres can construct a city-level dynamically updated network view, or a real-time network topology, and support inquiring senders with up-to-date connectivity information, routing paths to gateways, and destination locations. With this comprehensive connectivity-awareness, iCAR-II significantly improves VANET performance and enables efficient mobile data offloading via RSUs.

In iCAR-II, vehicles frequently update LCs with their locations and local network status as described in Section 4.1. These updates are sent to LCs either via LTE channels or

RSUs. Vehicles also periodically broadcast their locations, mobility vectors, and network status information (NSI) to their one-hop neighbours. LCs maintain tables of vehicles locations; a vehicle updates its location periodically or whenever it enters a new road segment. Moreover, LCs construct a dynamic network topology consisting of road segments weighted by experienced packet delivery delay. Whenever a source vehicle has packets to transmit via the infrastructure, it chooses either to send via VANET or LTE, based on the available network connectivity information. If VANET disconnection is reported, the source either selects LTE mobile data or reschedules the transmission. If such information is not available, a source transmits an inquiry message to LCs via LTE to obtain network status along with the best route.

This chapter is organized as follows. Section 4.1 describes the proposed routing scheme in terms of its four components: a) Beaconing and Neighbourhood Awareness, b) Mobility-based Link Lifetime Estimation, c) Broad Connectivity Evaluation, d) City-level Network Topology and Data Routing. Analysis and simulation-based performance evaluation are followed in Section 4.2. The chapter is summarized in Section 4.3.

4.1 *i*CAR-II: Protocol Description

The Infrastructure-based Connectivity Aware Routing protocol,*i*CAR-II, is a PBR routing scheme designed for multi-hop vehicular infotainment applications and Internet-based services as well as mobile data offloading. The principal of *i*CAR-II scheme is to support vehicles with instant information about VANET connectivity to infrastructure. Vehicular applications can, accordingly, decide to use VANET or LTE channels to access the core network. In order to achieve this principal, *i*CAR-II considers a number of algorithms and procedures run by vehicles' OBUs and LCs:

1. Beaconing and neighbourhood awareness
2. Mobility-based link lifetime estimation between each pair of neighbouring vehicles
3. Broad connectivity estimation (Road-level Connectivity)
4. City-level network topology construction and data routing

Vehicles are required, for safety purposes, to periodically report road and driving conditions to nearby vehicles [54, 11]. This is achievable by VANETs' one-hop broadcast *beaconing* messages, which also include vehicles locations and mobility information. Using beacon information, vehicles estimate local connectivity lifetime with one-hop neighbouring vehicles and achieve local neighborhood connectivity awareness. Beacons also help to exchange Network Status Information (NSI) which include comprehensive connectivity status to infrastructure, route to an RSU, and expected expiry time for that route. It will be shown later that routes in iCAR-II are represented by intersection IDs, and accordingly, routes to infrastructure are different at different roads. Thus, NSIs are exchanged locally within road segments while vehicles at intersections might receive NSIs from different roads.

As in iCAR protocol described in the previous chapter, when a vehicle, v_m, enters a road segment, $e_{i,j}$, it initiates, with a probability P_{RSE}, a measurement procedure called Road Segment Connectivity Evaluation (RSE) by sending a unicast control packet (CP) transverses the road segment to the other end, collecting some connectivity information from forwarders' routing tables. When failing to reach the destination intersection, CP is dropped due to a local network disconnection, and a random backoff time is set in NSI. Otherwise, a vehicle at the other end reports the minimum expected connectivity lifetime of $e_{i,j}$ and the experienced delivery delay of CP to LCs via LTE channels. The response that includes one or more routes to RSUs as well as a route lifetime is attached in a beaconing message and broadcasted to vehicles on $e_{i,j}$.

When a vehicular application or mobile data user needs to access the core network, or a non-neighbouring vehicle, via iCAR-II, it either finds a valid route in NSI or sends an inquiry message to location centers via LTE. LCs locate the target destination, run a shortest-path algorithm, (e.g., Dijkstra) on part of the graph that includes both the source and destination, and sends back an NSI message to the source. LCs includes a number of RSUs in the source vehicle vicinity in its search in order to select the best route and suggest alternative routes. Upon receiving NSI with a valid route, the source starts the low cost VANETs communication for the specified period of time, and refreshes path information before the expiry time of the current path if needed. In following, we describe the different stages of iCAR-II in more details.

4.1.1 Beaconing and Neighbourhood Awareness

Every vehicle is required to broadcast road conditions periodically to its one-hop neighbours, to enable several safety applications. These messages are used by PBR in VANETs as beacons to support the awareness of a vehicle's existence, location, and communication channel status within the communication range. In addition to road and driving conditions, vehicles in iCAR-II are required to include some essential information to enable its functionality. Information includes: 1) vehicle identifier (v_{ID}), 2) vehicle location coordinates (Loc_{vID}), 3) average driving speed (S_{vID}) for the last m seconds, 4) driving direction (Dir_{vID}), 5) turning signal status (Sig_{vID}), and 6) the predicted effective speed ES_{vID} which is a function of S_{vID} and average speed of leading vehicles (LS_{vID}) as will be shown in the next section. Leading vehicles are the group of neighbouring vehicles located in front of a transmitting vehicle, moving in the same road segment and direction, and having the same turning signal status. Leading vehicle average speed is easily calculated using

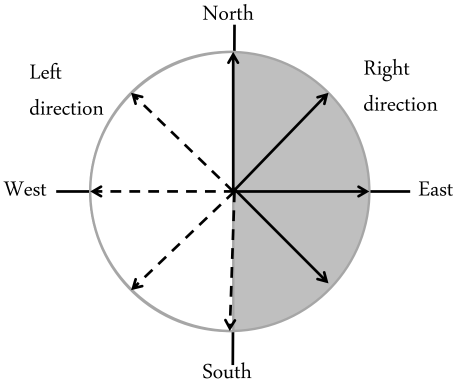

Figure 4.1: Defining Driving Directions

information from the vehicle's routing table, $Table_{v_{ID}}$.

Each road segment, $e_{i,j}$, is bounded by two intersections, I_i and I_j, and has two possible opposite directions. A vehicle is considered to be moving in a left direction if it is heading any direction from north/south to west, and considered to be moving in a right direction if it is heading any direction from north/south to east, as shown in Figure 4.1. Turning signal variable (Sig_m) for a vehicle v_m can take one of three values representing two signalling directions, *Right* and *Left*, and an *Idle* status.

The routing table is a table that is maintained by each vehicle to store neighbouring vehicles' information. In addition to routing information reported in beacons, $Table_{v_m}$ includes fields to track received signal strength indication ($RSSI_{v_{ID}}$), timestamps of last recorded entries and row update ($Last\_Upd_{v_{ID}}$), the estimation of minimum communication link lifetime (LRT_{vID}), a binary variable l_{vID} to indicate if the vehicle belongs to the leading vehicles group, and another binary variable f_{vID} indicating if v_{ID} is located in front of v_m at the updating moment regardless of its mobility direction and turning signal.

$Table_{v_m}$ is maintained by: adding new row information when receiving a beacon from a newly arrived vehicle to the communication range, updating row information when a beacon message is received from a neighbouring vehicle, deleting a row information from the table when no beacon is received from a current neighbour for a certain period of time τ_{delete_row}, and updating l_{vID} and f_{vID} values with periods of time τ_{l_update} and τ_{f_update} respectively. Row entries for an individual neighbour v_{ID} are updated periodically upon receiving a beacon message from v_{ID} with an acceptable $RSSI$ and a period of $\tau_{LinkUpdate}$. τ_{delete_row}, τ_{l_update}, τ_{f_update} and $\tau_{LinkUpdate}$ are much larger than the inter-beacon interval in order to reduce $Table_{v_m}$ maintenance operations. In addition, a neighbouring vehicle's information is updated if the difference between the reported predicted speed in the received beacon, ES^*_{vID}, and the recorded predicted speed in $Table_{v_m}$ exceeds a certain speed threshold ϵ_{vel}, or if the remaining time before the expiry of LRT_{vID} is less than ϵ_{LRT} as described in Algorithm 1.

Similarly, the routing information for a vehicle v_m is updated in the outgoing beacons periodically with respect to the timestamp of the last update, $Last\_B$, and a threshold value τ_{Bt} to control the frequency of updating this information. Routing information in outgoing beacons are also updated upon detecting a change in NSI.

4.1.2 Mobility-based Link Lifetime Estimation

Finding the minimum link lifetime, or the predicted link residual time (LRT), between two vehicles based on their mobility information exchanged in beacons is an imperative component within iCAR-II. Based on mobility prediction, LRT is defined as the expected remaining time duration for two communicating vehicles to stay within the communication range of each other before the first possible link breakage occurs due to their mobility, i.e., before the distance between them is predicted to exceed R meters due to a possible mobility

Algorithm 1 Beaconing

```
if $v_m$ is Sending a Beacon then
    if $|Current\_time - Last\_B| \geq \tau_{Bt} ||$ New NSI has been received then
        Obtain $S_m$, $LS_m$, $Sig_m$, $Dir_m$
        Update routing information in the Beacon
    else Reuse routing information
    end if
    Prepare a Beacon message with $v_m$'s ID, road/driving status, routing information, timestamp
    Send the Beacon message for broadcasting
end if
if Receiving a Beacon from $v_n$ then
    if $RSSI_n \geq RSSI_{thresh}$ then
        Extract $v_n$'s ID
        if $v_n \notin Table_{v_m}$ then
            Add $v_n$, Find $LRT_n$, and Complete $v_n$ entries in $Table_{v_m}$
        else Set $Last\_velocity = ES_n$ (table value)
            Set $Crrnt\_velocity = ES_n^*$ (beacon value)
            if $|Crrnt\_velocity - Last\_velocity| \geq \epsilon_{vel} \; || \; Current\_time - Last\_Upd_n \geq \tau_{LinkUpdate} \; || LRT_n \leq \epsilon_{LRT}$ then
                Find $LRT_n$ using recent information
                Update $v_n$ entries in $Table_{v_m}$
            end if
        end if
    end if
end if
```

scenario. Many vehicular mobility models can be applied in order to predict LRT, e.g., Car Following Models [23]. In this paper, we consider a unique prediction model that takes into consideration the actual requirements for iCAR-II as a routing protocol, as well as the information available at, or derived from, beacons and routing tables. The LRT-prediction model considers the following factors:

1. Relative Location: Which includes the relative distance, $d_{m,n}$, between two communicating vehicles, v_m and v_n, in addition to the road segments that v_m and v_n belong to. v_m and v_n can either belong to the same road segment, $e_{i,j}$, or to two adjacent road segments, $e_{i,j}$ and $e_{j,k}$. Two adjacent roads have a common intersection, and accordingly, $e_{j,k}$ can be described to be to the right, in front, or to the left of $e_{i,j}$. Thus, at each road, the set of adjacent road segments can be divided into three subsets, $\mathbb{R}$, $\mathbb{F}$, and $\mathbb{L}$, according to the orientation of v_m and the common intersection, regardless of driving direction.

2. Vehicles Speed: The *Predicted Effective Speed ES* is introduced in order to mitigate the effect of the frequent change in a vehicle's speed and acceleration in the city environment driving pattern. The predicted speed for a vehicle v_m, ES_m, is a function of both the vehicle's average speed in the last m seconds, S_m, and the average speed of its leading vehicles, LS_m, and also depends on the density k_m of leading vehicles in front of it as follows:

$$d_L = \begin{cases} d_I, & \text{if } R > d_I \\ R & \text{else} \end{cases} \tag{4.1}$$

$$k_m = \frac{1}{d_L} \cdot \sum_{q=1}^{N} l_q \tag{4.2}$$

$$LS_m = \begin{cases} \frac{1}{\sum\limits_{q=1}^{N} l_q} \cdot \sum\limits_{q=1}^{N} S_q \cdot l_q, & \text{if } \sum\limits_{q=1}^{N} l_q > 0 \\ 0 & \text{else} \end{cases} \tag{4.3}$$

$$ES_m = \begin{cases} (1 - \frac{k_m}{k_J}) \cdot S_m + \frac{k_m}{k_J} \cdot LS_m, & \text{if } k_m < k_J \\ LS_m & \text{else} \end{cases} \tag{4.4}$$

where d_I is the distance between the vehicle and the next intersection based on its mobility direction, d_L is the distance that leading vehicles occupy, k_m is the leading vehicles traffic density ($vehicle/m$), and k_J is the traffic jam density ($vehicle/m$).

3. Driving Direction: Each vehicle is aware, by the means of beacons, of the driving direction of itself and its neighbouring vehicles within the same road segment, i.e., either the same or the opposite driving direction. For neighbouring vehicles belong to adjacent road segments, and with respect to their driving direction and common intersection, I_j, the binary variable H_{vID} is defined as follows:

$$H_{v_{ID}} = \begin{cases} 1, & \text{if } v_{ID} \text{ is heading to } I_j \\ 0 & \text{else} \end{cases} \tag{4.5}$$

   H_n information of each neighbouring vehicle, v_n, that belongs to a different road segment can be maintained in the vehicle's routing table. In addition, turning signal information, Sig_n, gives another key indication for prospective driving direction.

Between two neighbouring vehicles v_m and v_n, each combination of the previous variables (i.e., $e_{j,k}$, H_m, H_n, Sig_m, Sig_n, f_n) defines a unique *Case*. Accordingly, *i*CAR-II mobility prediction model defines 144 possible cases. Each *case* is studied to predict one or more potential mobility *Scenarios* between the communicating vehicles. Then, each *scenario* is further studied to derive a corresponding equation to obtain LRT. First, the different scenarios are defined according to the following rules and assumptions:

1. For a vehicle v_m, neighbouring vehicles within the same road segment($e_{i,j} = e_{j,k}$), and those belonging to a *front* road segment ($e_{j,k} \in \mathbb{F}$) are considered to be moving in one dimension; on the other hand, neighbouring vehicles belonging to a *right* or *left* road segment ($e_{j,k} \in \{\mathbb{R} \bigcup \mathbb{L}\}$) are considered to be moving in a perpendicular direction to v_m.

2. A neighbouring vehicle, v_n, within the same road segment that has an idle turning signal maintains its predicted speed ES_n and reported mobility direction for the prediction period.

3. Three mobility scenarios are studied for each vehicle, v_n, that has an active turning signal: moving in the same driving direction with the speed of ES_n, stopping at the reported location (waiting to make a turn), and making an instant change of direction according to Sig_n and moving at the *Averaged Maximum Speed* S_{max}. S_{max} is a constant that considers an initial speed of 0 m/s and a maximum acceleration, until reaching a maximum speed, for a total travel distance of R meters.

4. When the communicating vehicles v_m and v_n belong to different road segments, and v_m is moving towards the common intersection, two additional scenarios are considered: instant stopping of v_m (due to a red traffic light) and proceeding of v_m at the speed of S_{max}.

5. In scenarios where vehicles move in perpendicular directions or where instant change of vehicle's driving direction is considered, a reduced effective transmission range $\hat{R}$ is used to represent a non-line-of-sight communication environment.

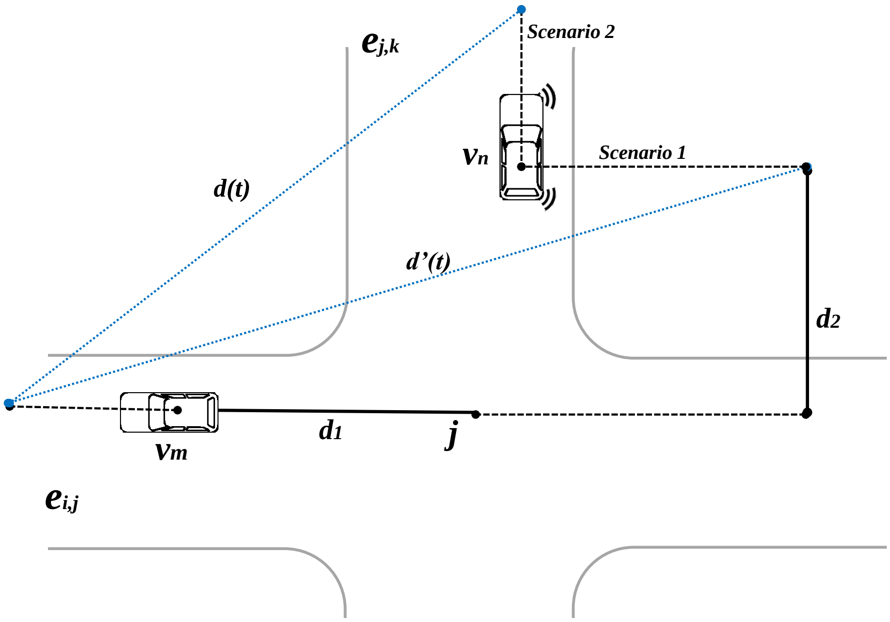

Figure 4.2: An Example Case with Two Mobility Scenarios

6. When there are more than one mobility scenarios for a certain case, only the scenario/scenarios that can cause earlier communication disconnection is/are considered. If the first disconnection depends on the actual values of the case variables in more than one scenario, equations from the different scenarios are considered, and LRT takes the minimum result. The predicted LRT might be obtained from a different scenario than the actual one, or from a misinterpreted turning signal, i.e., an active turning signal for a lane change only. This can result only in a shorter LRT, and is corrected via the frequent LRT updates as shown in Algorithm 1 to maintain valid LRT information.

7. The prospective mobility scenario is predicted for a short period of time to insure the validity of the given mobility information; thus, LRT is upper-bounded by R/S_{max}.

The aforementioned rules determine one or more mobility scenarios for each case under consideration. Each scenario, κ is associated with a predicted link lifetime, t, between the communicating vehicles. To obtain $LRT(\kappa)$, a corresponding equation to each scenario is derived as follows:

1. A diagram for the potential mobility scenario is created; a case example for mobility in two dimensions is presented in Figure 4.2 with two potential mobility scenarios.

2. According to the aforementioned rules and a certain scenario under consideration, the different variables of the scenario are determined, e.g., using R, $\hat{R}$, S_{max}, ES etc.

3. For mobility in one dimension, simple Kinematic equations are used to find $LRT(\kappa)$. For example, for a scenario of two vehicles moving towards each other with predicted speeds ES_m and ES_n, and with an initial distance $d_{m,n}$ between them, we would have:

$$LRT(\kappa) = (R + d_{m,n})/(ES_m + ES_n) \tag{4.6}$$

4. For mobility in two dimensions, the Parametric equations for the predicted trajectory of each vehicle are defined with respect to the parameter t, i.e., defining $x_m(t)$, $y_m(t)$, $x_n(t)$, and $y_n(t)$ as functions in time. Then, the Pythagorean theorem is used to find the predicted change in distance between the communicating vehicles $d_{m,n}(t)$:

$$d(t) = \sqrt{(x_m(t) - x_n(t))^2 + (y_m(t) - y_n(t))^2} \tag{4.7}$$

   By substituting $\hat{R}$ for $d(t)$ and solving for t to find the required link lifetime, we obtain an equation associated with the mobility scenario to predict $LRT(\kappa)$. For example, considering *Scenario 1* in Figure 4.2, the variables under consideration are $\hat{R}$, d_1, d_2, ES_m, and S_{max}. The parametric equations for this scenario are: $x_m(t) = -d_1 - ES_m t$, $y_m(t) = 0$, $x_n(t) = S_{max}t$, and $y_n(t) = d_2$. By applying the Pythagorean theorem:

$$d_{m,n}(t) = \sqrt{(-d_1 - ES_m t - S_{max} t)^2 + (-d_2)^2} \tag{4.8}$$

Replacing $d_{m,n}(t)$ by $\hat{R}$ and solving for t in the case that $\hat{R} \geq d$:

$$LRT(Scenario\ 1) = \begin{cases} \frac{-d_1 + \sqrt{\hat{R}^2 - d_2^2}}{S_{max} + ES_{vi}} & \hat{R} \geq d \\ 0 & \hat{R} < d \end{cases} \tag{4.9}$$

Similarly, the different scenarios have been studied for the different cases and a set of equations have been determined. When a vehicle v_m needs to update the value LRT_n in $Table_{v_m}$ upon receiving a beaconing message from v_n, v_m determines the mobility case based on the available information and calculates the predicted link lifetime. When more than one scenarios are considered, the minimum value of $LRT(\kappa)$ is maintained. Then, the minimum link lifetime between v_m and v_n, LRT_n is updated in $Table_{v_m}$:

$$LRT_n = min\{LRT(\kappa), \frac{R}{S_{max}}\} \tag{4.10}$$

LRT_n is updated frequently at $Table_{v_m}$ with respect to three criteria, as shown in Algorithm 1: 1) periodically with a period of $\tau_{LinkUpdate}$, 2) if a major change in v_n's predicted speed has been detected, and 3) if v_m is receiving beacon messages, with acceptable $RSSI_n$, after, or close to, the expiry time of the expected LRT_n.

4.1.3 Broad Connectivity Evaluation

Broad Connectivity Evaluation, or Road Segment Evaluation (RSE), in iCAR-II is a heuristic procedure dynamically initiated by some vehicles to sense the different parts of the network and update the network status information NSI. NSI includes road segment connectivity to infrastructure (RSU) status, the best route to infrastructure, the expected packet delivery delay via that route and the expiry time of it. NSI is shared locally within a road segment and exchanged via beacon messages. In RSE, a light-weight control packet CP traverses the road segment via relaying forwarders and collects connectivity and link lifetime information at each intermediate forwarder. When reaching the target intersection I_j, a vehicle v_n at I_j reports the connectivity status of $e_{i,j}$ and its predicted minimum link lifetime $RLL_{e_{i,j}}$ to LCs via LTE, and obtains an updated NSI accordingly.

RSE procedure is initiated, with a probability P_{RSE}, by a CP originator v_m entering a road segment $e_{i,j}$ towards an intersection I_j. The principle of RSE is predicting a minimum link lifetime per road segment based on link lifetime information, LRTs, between individual vehicles available at their routing tables. RSE divides $e_{i,j}$ into smaller vicinities, or areas of interest (AoIs), between CP forwarders as shown in Figure 4.3. While passing CP, each forwarder finds the maximum link lifetime between itself and the previous forwarder, directly or via one-hop relay vehicle. The originator and each forwarder, v_{lf}, attaches in CP the set $\mathbb{L}_{v_{lf}}$ which includes LRTs values for all neighbours in its AoI along with their identifiers' set $\mathbb{M}_{v_{lf}}$. A receiver forwarder, v_{cf}, extracts the set of common neighbours $\mathbb{C}_{v_{lf},v_{cf}}$ and finds the maximum possible link between itself and the last forwarder, $RLL_{v_{lf},v_{cf}}$, as indicated below:

$$\hat{l}_{max} = \max_{v_q \in \mathbb{C}} (min\{LRT_{v_{lf},v_q}, LRT_{v_{cf},v_q}\} \tag{4.11}$$

$$RLL_{v_{lf},v_{cf}} = max\{\hat{l}_{max}, LRT_{v_{lf},v_{cf}}\} \tag{4.12}$$

Intermediate forwarders update $\mathbb{M}$ and $\mathbb{L}$ while keeping only the minimum value of $RLL_{e_{i,j}}$. The last CP receiver, say v_n, which is the closest to I_j, reports the total delivery delay of CP, $D_{e_{i,j}}$, along with $RLL_{e_{i,j}}$ to LCs. LCs update the network graph, find the route(s) to RSU(s), and send NSI back to v_n. Then, v_n unicasts the updated NSI to CP

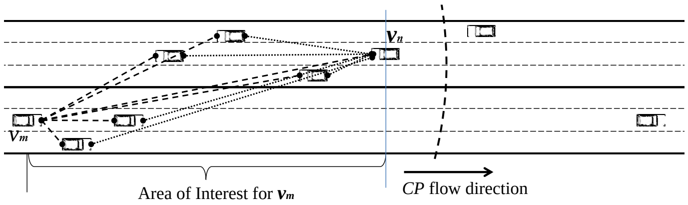

Figure 4.3: Calculating RLL in Road Segment Evaluation

originator and broadcasts it via its beacons. Every vehicle within $e_{i,j}$ updates NSI and includes it in its beacons.

As greedy routing without store-carry-forward is used to deliver CP, reaching I_j indicates local network connectivity at $e_{i,j}$ for a period of time registered in CP. The delivery delay of CP also gives an indication of packet delivery delay in the road as it experiences similar transmission and queuing delay in addition to interference and fading conditions in $e_{i,j}$. For a disconnected road segment, CP is dropped when a forwarder, or an originator, v_m fails to find a next forwarder. v_m creates an NSI indicating disconnectivity with a small random validity period, which works as a back-off time to prevent multiple RSE calls by vehicles entering $e_{i,j}$.

When v_m enters $e_{i,j}$, it is expected to receive an NSI from its neighbours, which includes the expiry time of NSI. To ensure the availability of a valid NSI, P_{RSE} is designed to be a function of the remaining validity time of NSI, t_{rem}, and $e_{i,j}$ length $|e_{i,j}|$. When v_m does not receive any valid NSI, it also initiates the RSE procedure. Equations 4.13 and 4.14 present one way to design P_{RSE} [55]:

$$P_{RSE} = \begin{cases} e^{-\frac{t_{rem}-C}{2}}, & t_{rem} \geq C \\ 1 & t_{rem} < C \end{cases} \tag{4.13}$$

$$C = 2 \cdot t_{max} \cdot \lceil |e_{i,j}|/R \rceil + \epsilon \tag{4.14}$$

where t_{max} and ϵ are constants representing the maximum acceptable delay per forwarder, including average transmission delay and queuing delay, and the expected time to obtain NSI from LC, respectively.

4.1.4 City-level Network Topology and Data Routing

The frequent distributed calls of the RSE procedure and the associated connectivity and delay information sent to LCs, enable LCs to draw a real-time network graph providing a city-level network topology awareness, where the graph consists of vertices, representing road intersections, and weighted edges, representing road segments where each edge is weighted by the experienced delay. As LCs receive RSE update messages for only connected roads, the graph represents only real-time network view of the map, and edges with expired validity lifetime can be removed. With a known set of RSUs locations in a city, each road segment has a subset of nearby RSUs; thus, after receiving an RSE update message related to a certain road $e_{i,j}$, a shortest path algorithm, e.g., Dijkstra, is run on the subgraph of the network that has the road segment $e_{i,j}$ and the subset of nearby RSUs to find the best route to the core network. LCs send back a response message to the sender, which has an NSI. Then, the sender broadcasts the NSI in $e_{i,j}$ via beacons, which enables connectivity awareness to all vehicles in the vicinity of $e_{i,j}$. NSI includes the path, by the means of intersections, the path's lifetime, which is the minimum $RLL_{e_{i,j}}$ among road segments constructing the path, and the expected delivery delay, which is the summation of experienced delivery delay for road segments constructing the path.

According to the direction of data forwarding, either towards RSU or a destination vehicle, data routing can be described as uplink routing or downlink routing. For uplink routing, vehicles that have data to send find connectivity and expected delay information available in NSI. According to this information, vehicles either use VANETs, LTE, or reschedule transmission for better VANETs conditions. In the case of a connected network, the path from a source road segment to a destination RSU is predetermined by the means of consecutive intersections. Thus, iCAR-II deploys source PBR where the path is attached to the header of each packet, which reduces cost, delay, and overhead of multiple route enquiries via LTE. In case a packet has reached a disconnected road, a forwarder can encapsulate the packet and forward it via a new path using a more recent NSI available at its road segment, if any, otherwise the packet is dropped. Disconnection can occur due to an unexpected delivery delay beyond the path lifetime, or an unexpected local network disconnection in the routing path during its lifetime. On the other hand, vehicle's location, an associated RSU, and the path from RSU to the vehicle, by the means of intersections, are determined by LCs in the downlink routing case. Data packets are forwarded from the core network to the RSU, and VANETs data routing takes place from RSU to the destination vehicle using source PBR.

For routing within roads, iCAR-II uses a greedy-based next-hop selection method. Algorithm 2 shows a light-weight next-hop selection procedure to filter one-hop neighbours based on their location and the latest received $RSSI$. The location filter in the forwarding process aims to maximize the progress towards the target intersection. Such greedy forwarding protocol selects next-forwarders that are farther from a sender, which are more likely to leave the communication range causing transmission interruption, or have bad signal quality. Thus, $RSSI$ filter excludes neighbours with $RSSI$ below a certain threshold.

Algorithm 2 Next-hop Selection

```
Require: $Table_{v_{cf}}$, $ei,j$,$ej,k$, $I_{Crnt\_Target}$, $I_{Nxt\_Target}$, $RSSI_{thresh}$
for $n$= 1 to $|\mathbb{N}|$ do                    (check all neighbours)
    if $ei,j == ej,k || I_{Nxt\_Target} \in \{I_j, I_k\}$ then
        if ($v_{cf}$ moving towards $I_{Crnt\_Target}$ & $f_n == 1$) $\|$ ($v_{cf}$ moving away from $I_{Crnt\_Target}$ & $f_n == 0$) $\| I_{Nxt\_Target} \in \{I_j, I_k\}$ then ($v_n$ makes forwarding progress)
            if $RSSI_n \geq RSSI_{thresh}$ then
                $\mathbf{N} = \mathbf{N} \cup v_n$    ($v_n$ is a potential forwarder)
            end if
        end if
    end if
end for
if $\mathbf{N} \neq \phi$ then
    Find $v_{nf}$ $s.t.$ $d_{v_{nf},I_{Nxt\_Target}} = max\{d_{v_n,I_{Nxt\_Target}} \forall v_z \in \mathbf{N}\}$
else
    Next-forwarder is not found  (packet will be dropped)
end if
```

A vehicle's mobility direction is not considered in order to maximize the number of potential forwarders, taking into consideration that vehicle's mobility can be negligible compared to data transmission speed, and the distance between vehicles are updated frequently on routing tables.

While forwarding data packets, a next forwarder v_{nf} is chosen only from the current road segment, or the road segment connecting to the next target intersection I_{Nxt_Target} in the packet's path. When a packet reaches the last road segment in its path, each forwarder v_{cf} looks up the packet's destination in its routing table. In Algorithm 2, ei, j and ej, k represent the road segments that the current forwarder v_{cf} and the potential next-forwarder v_{nf} belong to, respectively.

4.2 Performance Evaluation

This section presents the evaluation of iCAR-II. First, the individual components of iCAR-II are considered in brief analysis and discussion, namely:.

- Beaconing and neighbourhood awareness
- Node-Level Link Lifetime
- Road-Level Connection Lifetime
- City-Level Network Connectivity

Then, the overall performance is evaluated using a special MATLAB-based simulation program developed to evaluate VANETs routing protocols performance. In addition to iCAR-II, we considered three other VANETs routing protocols, which have been slightly modified in order to have a fair comparison with *i*CAR-II, i.e., having the same infrastructure resources. These protocols are: GPSR [18], GSR [20], and GyTAR [21]. These protocols are modified to use LTE channels to report vehicles location periodically and acquire the location of the destination, or the closest RSU, from LCs.

The performance evaluation of the routing protocols has considered variable network density, packet generation rate, and number of deployed RSUs. The performance metrics are:

- Packet Delivery Ratio (PDR): we define two forms of packet delivery ratio to show the ability of a routing protocol to successfully transfer data from a source to a destination on an end-to-end basis, with respect to protocols under consideration, 1) PDR1: number of successfully received data packets by destinations per number of sent data packets per sources, and 2) PDR2: number of successfully received data packets by destinations per the total number of data packets sent, or ready to send, at sources.
- Average Packet Delivery Delay (PDD): This metric shows the latency of data packet delivery introduced by each routing protocol and defined as the average end-to-end delivery delay of all successfully delivered data packets.
- Average Routing Overhead: This metric shows the extra communication overhead required by routing protocols. Two types of routing overhead can be defined, 1) average LTE routing messages, e.g., location updates and enquiry messages, per second, and 2) average unicast routing control packets which is the average of extra unicast packets sent by vehicles to maintain the routing protocol per second per road segment.

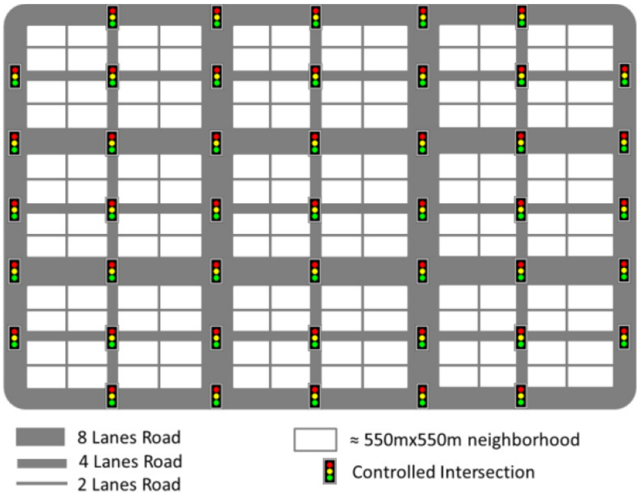

Figure 4.4: Approximate simulation map

4.2.1 Simulation Setup

Road grid has been implemented in MATLAB to represent 7000 m × 7000 m area of bidirectional roads. Roads vary in terms of length, width, and vehicles density to represent major roads and residential areas in the city. Each road segment has a predefined maximum speed. Figure 4.4 shows approximate map that represents the roads grid which has a total of 165 intersections, with 45 of them being traffic-light controlled. The open-source microscopic vehicular traffic generator SUMO [25] is used to generate vehicles movement files. SUMO uses car-following model and the input of our grid map including the number of lanes and speed limit of each road segment.

For wireless consideration, a simple DCF MAC is applied for MAC contention, a FIFO packet queue, such as the AC queues design for WAVE's MAC layer [13], is implemented for packet buffering, and a free space model with urban area path loss exponent [52][56] is deployed for $RSSI$ estimation. Source vehicles are randomly selected, where source

Table 4.1: Simulation Parameters for iCAR-II Evaluation

Parameter	Value	Parameter	Value
Scenario Duration	40 s	$RSSI_{thrrdh}$	0.6x$RSSI_{max}$
Scenario Repetition	5-12 times		
R	250 m	τ_{Bt},$\tau_{deletrow}$, $\tau_{lupdate}$,τ_{fupdt}, $\tau_{LinkUpdate}$, m	3 s
$\hat{R}$	150 m	ϵ_{vel}	7 m/s
Packet Size	512 $Byte$	S_{max}	15 m/s
Transmission Rate	12 $Mbps$	K_J	115 $veh/lane.klm$
Packet Lifetime	1500 ms	C	1.5 s

vehicles are always 10% of the total number of vehicles for the different vehicles density scenarios. Each source vehicles continuously sends data packets to the core network via RSUs, where packets are routed independently. LTE channels are assumed to have ideal communication and represented by a fixed delay of 200 $msec$ for one-way communication. Fetching information from LCs is also represented by a fixed delay of 500 $msec$. Each simulation scenario has been repeated several times for accurate results. Table 5.1 presents the different simulation parameters used in this evaluation.

4.2.2 Simulation Results

Beaconing and Neighbourhood Awareness

Beaconing is one of the main components in any PBR protocol. Beaconing rate can be either fixed or dynamic with respect to speed, vehicle density, or other parameters. Since we have considered a fixed beaconing rate in this study, any other beaconing scheme is still valid as long as it enables iCAR-II to predict LRTs and exchange NSIs. iCAR-II considers two strategies to reduce the computation and communication overhead that can occur to update LRTs and share NSIs. First, instead of updating LRT value for each neighbouring vehicle upon receiving its beacon, iCAR-II reduces the number of LRT updates for each neighbour per second by the factor of $1/\tau_{LinkUpdate}$. Simulation shows that the effect of updating LRT values on the iCAR-II performance is negligible if the $\tau_{LinkUpdate}$ is less than 3 *sec*. Second, iCAR-II uses beacons to share NSIs in order to preserve the bandwidth.

Figure 4.5 shows the delay required to deliver NSI to all vehicles within a road segment of 1 klm length and 4 lanes width. It is shown that NSI distribution time is generally decreased by the increased vehicular density and/or beaconing rate. In light and moderate traffic densities, increasing vehicular density or beaconing rate significantly decrease the delay to deliver NSI. However, in dense areas, and with high static beaconing rate, the delivery of beaconing packets is delayed due to packets' collisions, or rescheduling, causing slightly delayed NSI delivery. Thus, NSI delivery delay in such situations highly depends on the performance of the deployed MAC protocol. In general, results in Figure 4.5 show acceptable delay taking into consideration that Equations 4.13 and 4.14 preserve time for NSI delivery.

Node-Level Minimum Link Lifetime

The LRT finding procedure predicts the worst possible case scenario for future movement of two neighbouring vehicles based on their mobility vectors, distance between them, distance to a common intersection, and their turn-signal status, in order to assign a lower-bound of link lifetime between them. LRT is frequently updated, while vehicles are exchanging beacons, every $\tau_{LinkUpdate}$. Simulation results show that this procedure succeeds in putting a lower-bound of link lifetime in all cases. In other words, it successfully predicts link breakage before it happens. However, in real-life situations, other cases can occur causing

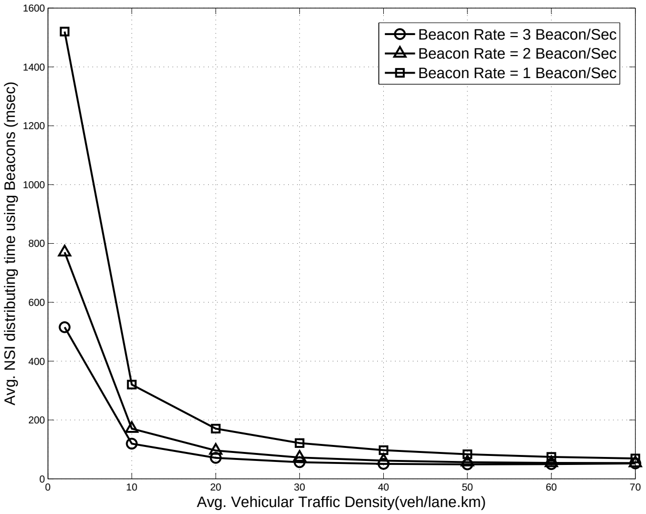

Figure 4.5: Average delay for network information dissemination within a road segment using beacons

link breakage within the predicted minimum lifetime. For example, a parking vehicle on the side of the road can have a high LRT value; however, it is more likely to turn-off its OBU causing a communication termination. Such situations represent a small percentage and can be ignored.

Equations 4.1 to 4.4 present the expected speed ES of vehicles based on its average speed and the average speed of its leading group. Differentiating between leading vehicles based on their turn-signal status makes ES more accurate. The mean percentage error of ES prediction, with $\tau_{LinkUpdate}$ defined in Table 5.1, is 4.3%. However, with large $\tau_{LinkUpdate}$ value, this mean increases significantly, considering urban scenario with controlled intersections where drivers change their speed frequently due to traffic lights status. To avoid the effect of such error in ES estimation of iCAR-II performance, Algorithm 1 calls LRT procedure when a major change in speed is detected.

Road-Level Minimum Connection Lifetime

Minimum road connectivity lifetime algorithm is a heuristic algorithm that uses the link lifetime information available at nodes' routing tables to assign a minimum road-level link lifetime (RLL) to each road segment. First, one possible routing path is considered to check instantaneous connectivity, then one-hop relay between each pair of consecutive forwarders in the path is considered to predict future connectivity. Among each pair, the maximum predicted link lifetime is selected, and among the selected set, the minimum link lifetime is considered to be the RLL. Intuitively, the road segment has at least one connected path from end-to-end during RLL second. More than one initial path can be considered, and more than one-hop possible intermediate relay can be calculated, which increase the predicted RLL. However, this increase comes at the cost of communication overhead to share more than one-hop neighbouring LRT information, and calculation overhead to find all possible future links among those vehicles. However, iCAR-II considers only the previously calculated one-hop LRT, available at vehicles' routing tables, along with dynamic updating procedure using P_{RSE} in Equation 4.13. P_{RSE} is able to maintain valid RLL while the road segment has a connected path. It takes into consideration the time required to generate RLL and obtain and distribute NSI, before the expiry time of the current one.

RLL is limited by an upper-bound of R/S_{max} in order to avoid the effect of long term LRT prediction using ES. As LRT values are frequently adjusted using $\tau_{LinkUpdate}$, the LRT values that construct RLL can be inaccurate on the long-run. For example, a path of CP among stopping vehicles for a red-light traffic signal might have a large expected link lifetime, e.g., infinity. However, a disconnection is possible after vehicles move due to traffic light status change. Thus, RLL is upper-bounded by time required to travel R

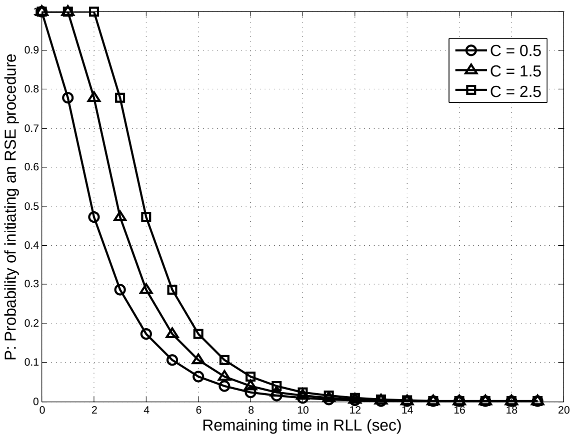

Figure 4.6: Probability of initiating RSE procedure for a vehicle entering the road segment

meter with an averaged maximum speed S_{max}.

Figure 4.6 shows the probability of initiating an RSE call by a vehicle entering the road segment as a function of RLL's remaining time considering different values of C. C is a design parameter related to road length, transmission range, maximum acceptable transmission time per one-hop, and the time required to obtain NSI from LCs. Figure 4.7 shows the average reported RLL values with respect to different vehicles density. It shows that even with low vehicles density, roads can maintain connected paths for a considerable duration of time, and RSE procedure enables source vehicles to instantaneously utilize these paths. Moreover, the results show that the average RLL is directly proportional to vehicles density within road segment. This can be related to the decrease in average vehicles speed in the high density scenario as well as the availability of more intermediate nodes between each pair of CP forwarders. RLL is also inversely proportional to $\tau_{LinkUpdate}$ as with large $\tau_{LinkUpdate}$ values, the remaining time of LRTs decreases before updates, and RLL decreases accordingly.

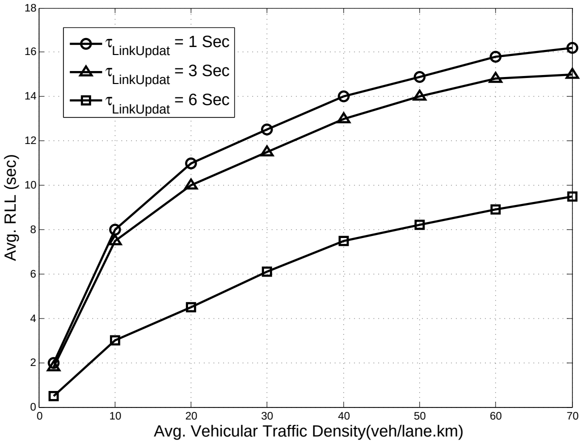

Figure 4.7: Average RLL in a road segment

City-Level Network Connectivity

iCAR-II is a proactive protocol that enables vehicles to have immediate global network condition information by making NSIs available at vehicles' beacons. Vehicles, via NSI, can know about the road connectivity to the core network, the route to an RSU, the expected delivery delay, and the expiry time of that route. Routes are dynamically updated on LCs by probabilistically initiating RSE procedures among different network edges. LCs maintain updated network values as Equations 4.13 and 4.14 insure that.

Figure 4.8 shows the percentage of connected road segments to the infrastructure with respect to different network node densities and number of deployed RSUs. It can be seen that iCAR-II can construct connected networks even with low deployment of RSUs. This can be related to the global view of connected road segments at LCs. With respect to road segments length, number of lanes per road segment, and the transmission range under consideration, it is shown that the number of connected road segments to the core network is increasing rapidly with the increase in vehicular density in the light traffic densities test

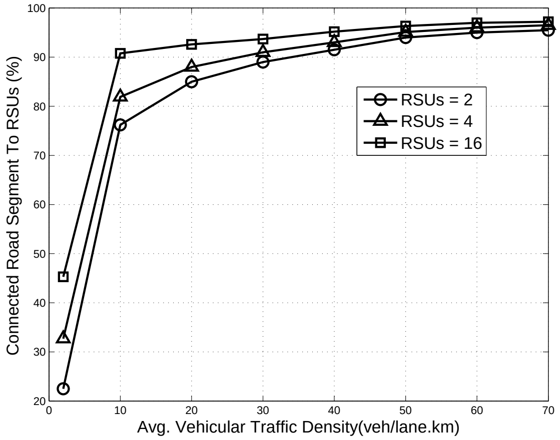

Figure 4.8: Percentage of Connected Road Segments to the Core Network using *iCAR-II*

points (less than 10 $vehicles/lane.km$) as the network connectivity becomes more sensitive to vehicular densities in this range. With higher vehicular densities, the increase becomes slower as most main roads are already connected to RSUs and only few roads are joining the network when increasing the number of vehicles.

Packet Delivery Ratio

As packets are transmitted by source vehicles using iCAR-II only when connected path is detected, the ratio of delivered data packets to the sent packets (PDR1) is expected to be high regardless of network node density. With PGR = 10 packets/sec and the deployment of 4 RSUs, simulation shows that PDR1 always exceed 97%. Data packets that have not been delivered during the lifetime of the path might be dropped due to an expected network disconnection. Also, packets that have not been delivered during their lifetime due to delivery delay are dropped. Notice that iCAR-II conserves network bandwidth by buffering data packets when VANETs is not connected to the core network.

In order to have a valid performance comparison between iCAR-II and GPSR, GSR, and GyTAR, which do not require prior determination of path existence before transmission, we define PDR2 to be the ratio of packets successfully delivered to packets that are ready to be sent. Figures 4.9 and 4.10 show that iCAR-II still has a significantly higher PDR2 than the other PBR protocols. Figure 4.9 shows that increasing vehicles density, with a low data packet traffic in the network, improves packet delivery ratio, as VANETs become more connected. With low vehicles densities, GPSR, GSR, and GyTAR show a very low PDR2 as they blindly route data packets through an intermitted network, while iCAR-II has a noticeably high PDR2 due to its connectivity awareness feature. The curve trend of iCAR-II is analogous to the network connectivity curve in Figure 4.8. Data packets might be routed along paths that are not the shortest curvemetric routes yet connected. It is observed that GSR performs better than GPSR only in high vehicular density, when VANETs are connected, as GSR does not consider vehicular traffic in the routing decision. In high vehicular densities, GPSR suffers from higher routes length compared to other protocols as it does not use map information or anchor routing. In such cases, GPSR packets reach their expiry time before delivery.

Figure 4.10 shows PDR2 of the different protocols with respect to a variable PGR and a high vehicular density. Results show that iCAR-II maintains high performance even with an increasing PGR. As iCAR-II considers delivery delay per road segment in its route constructing level, new paths are dynamically suggested by LCs to maintain network performance. On the other hand, GSR does not consider dynamic routing while GPSR and GyTAR considers only local connectivity and distance to destination, which result in routing convergence to dense roads, which causes data traffic congestions and high queuing delay. Delay is associated with PDR as delayed packets can reach their expiry time before delivery and be dropped. The slight dropping in iCAR-II PDR2 shown in Figure 4.10 with

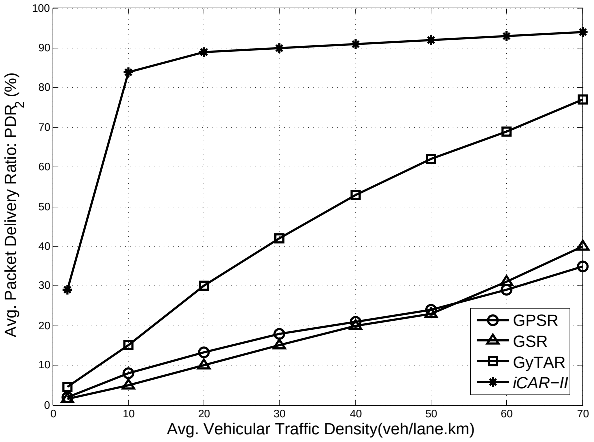

Figure 4.9: Average Packet Delivery Ratio ($PDR2$) using 4 RSUs and PGR = 10 packet/sec

high PGR is due to reaching the communication capacity of RSUs.

Packet Delivery Delay

As iCAR-II considers packet experienced delivery delay of CPs a major metric in route calculation, average packet delivery delay (PDD) using iCAR-II is expected to be low. Simulation results show that iCAR-II significantly reduces PDD compared to other routing protocols as shown in Figures 5.7 and 5.8. With low vehicle densities, iCAR-II selects connected paths even with long trajectories to achieve higher PDR with the cost of slightly high PDD. PDD of GSR is analogous to that of iCAR-II in the case of light data traffic as packets are routed along predetermined paths. However, these paths are either connected to the core network or the packets are dropped, causing very low GSR-PDR as shown in Figure 4.9.

Simulation results also show that PDD in GPSR is high. Packets forwarded using GPSR encounter a high number of intermediate forwarders due to the perimeter recovery

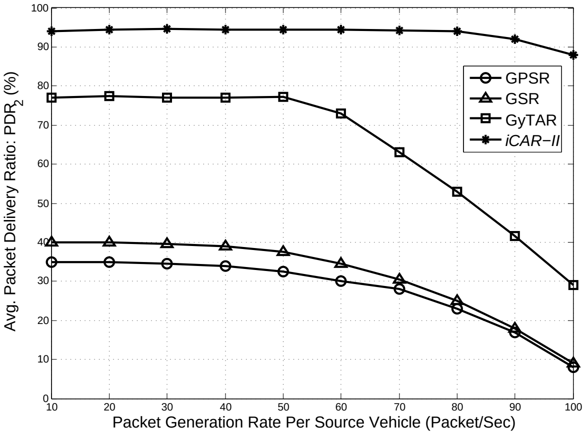

Figure 4.10: Average Packet Delivery Ratio ($PDR2$) using 4 RSUs and vehicle density of 70 vehicle/lane.klm

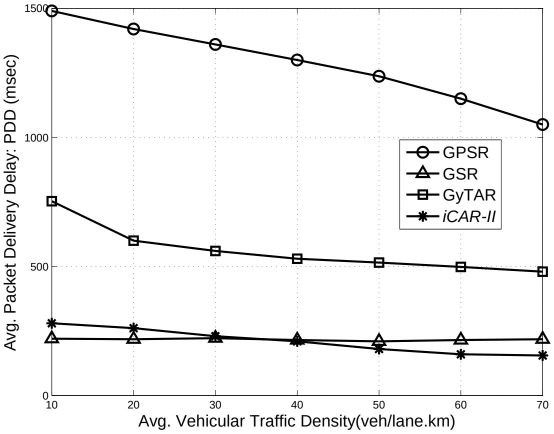

Figure 4.11: Average Packet Delivery Delay (PDD) using 4 RSUs and PGR = 10 packet/sec

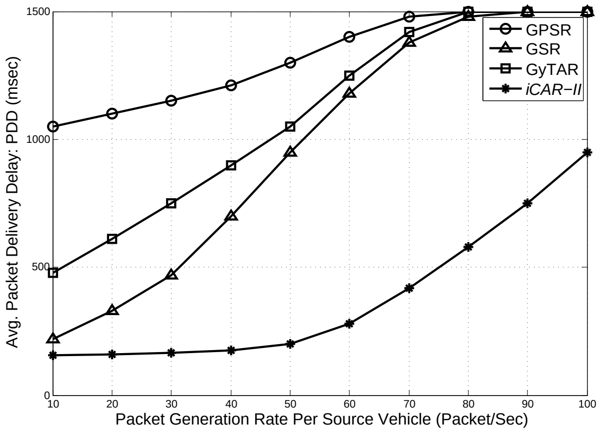

Figure 4.12: Average Packet Delivery Delay (PDD) using 4 RSUs and vehicle density of 70 vehicle/lane.klm

routing strategy and the experience of long routing paths. Moreover, increasing PGR leads to a significant PDD increase in GPSR, GSR and GyTAR, as shown in Figure 5.8. These protocols do not consider delivery delay in its routing, and when routes converge to a limited number of roads, data traffic congestion increases the delivery delay. It can be shown from Figure 4.10 and 5.8 that a considerable portion of data packets have been dropped due to reaching their expiry lifetime, which is set to be 1500 $msec$ in our study. As iCAR-II uses dynamic route selection considering the experienced delivery delay, it has a significantly reduced PDD.

Routing Overhead

We consider the additional routing control messages to measure and compare the introduced overhead by the different routing protocols. These control packets can be classified into three categories: 1) Beaconing messages; 2) LTE routing messages; and 3) Unicast routing control messages. Beacons are the main communication overhead introduced by any PBR protocol, as the broadcast beaconing messages use control channels periodically. However, beacons are required by safety applications, and as long as different protocols use the same inter-beacon interval/beaconing protocol, the effect of beacons on the networks is the same for the different protocols.

LTE communication overhead is an important evaluation metric, as accessing LTE channels cost more than VANETs DSRC channels. Figure 4.13 shows the simulation results of average LTE control messages used for each vehicular density scenario. In GSR and GyTAR, LTE routing messages are used to report entering new road segments (location updates) and to enquire about a destination location. iCAR-II has a slightly higher average of LTE control messages as it uses LTE channels to update road condition information. In GPSR, vehicles use LTE channels for location updates and location inquiry messages. The average LTE communication overhead, when GPSR is deployed, depends on the location update period, and it is always higher than the average overhead generated by the other protocols.This shows one of the advantages of intersection-based routing.

To better understanding the LTE communication overhead, Figure 4.14 shows the average number of LTE control messages sent by each vehicle per hour. It is shown that iCAR-II introduces higher LTE overhead specially in the sparse network cases. In fact, the increase of LTE overhead in iCAR-II as compared to the other protocols is still insignificant with respect to the increase in PDR shown in Figure 4.9. In the worst case, the difference is about 60 messages per vehicle per hour, which is a small cost to obtain connectivity information in a sparse network for data offloading or VANETs Internet access. With higher vehicle densities, roads become more connected with higher average RLL and less LCs updates accordingly.

The third type of communication overhead for routing control is the unicast packets sent

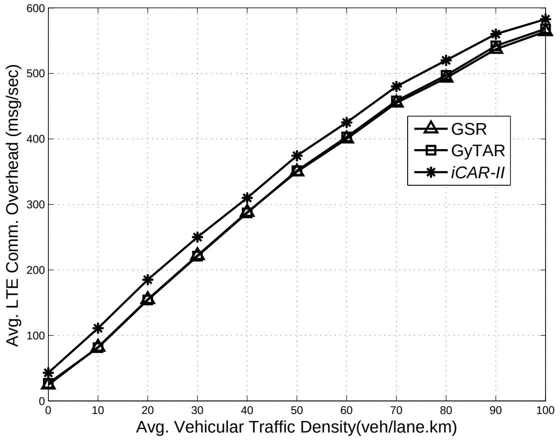

Figure 4.13: LTE Routing-Control Messages

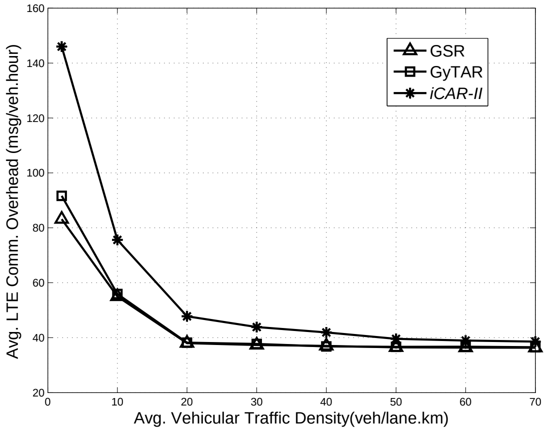

Figure 4.14: Average LTE Routing Messages per Vehicle per Hour

locally within roads to collect traffic information in GyTAR and to examine connectivity, collect links lifetime and calculate delivery delay in iCAR-II. Although iCAR-II introduces about double the number of these control packets compared to GyTAR, this overhead can be neglected as these packets are unicast, distributed, and in the worst case the average number of control packets does not exceed two packets per second for each road segment.

4.3 Summary

In this chapter, an efficient routing framework, iCAR-II, has been introduced to integrate VANET, cellular network, and location centers, in order to improve VANETs data routing and enable cellular network mobile data offloading. iCAR-II enables mobile users to proactively obtain VANET connectivity to the core network information. The availability of this information preserves VANET's bandwidth, in the cases of disconnectivity or data traffic congestions, and enables users to enjoy the low-cost VANET-based Internet access and mobile data offloading. iCAR-II utilizes the reliable communication channel of cellular network to construct a global real-time view of VANET's topology. It has been demonstrated that iCAR-II algorithms can provide real-time VANET information to mobile users, and an efficient and dynamic data routing, with a limited use of LTE messages per vehicle.

Chapter 5

D-CAR: Dynamic Connectivity-Aware Routing for Internet-based Services

---

Different from iCAR-II presented in the previous chapter, our goal in this chapter is to develop a routing protocol that proactively support vehicles with comprehensive connection information to RSUs without the assistance of the cellular network. The target connectivity information are: the existence of one or more possible paths to RSUs, the predicted connection residual time (CRT) for each route, and the expected delivery delay per alternative. To determine the existence of a connection to an RSU and its CRT, all links and different paths need to be considered. CRT depends on the possible paths which are comprised of links sequences. Thus, predicting link residual time (LRT) between each pair of vehicles is also needed.

In this chapter, we propose D-CAR, an efficient routing protocol that is able to deliver instant deterministic connectivity information to connected vehicles. By obtaining key connectivity parameters, such as minimum connection duration and average delivery delay, in-vehicle Internet-based applications can decide whether to start a low-cost Internet access via VANETs, reschedule the transmission until a better connectivity condition becomes available, or use an alternative network. This will save network's bandwidth, increase packet delivery ratio, and allow vehicles to dynamically select routes with reduced delivery delay, which certainty improve the overall VANETs performance. D-CAR uses a microscopic mobility prediction model run by individual vehicles to predict LRTs with neighbouring vehicles. A distributed beaconing-based algorithm is run across neighbouring vehicles to extract CRTs to one or more RSUs. Probe control packets (CPs) frequently traverse each road segment to examine its delivery delay.

This chapter is organized as following: D-CAR protocol description is presented in Section 5.1. The protocol is described by its three main frameworks: a) A framework for Link Lifetime Prediction, b) A framework for Network Connectivity Prediction, and c) Data Packets Routing. Section 5.2 presents the evaluation of D-CAR components and the network performance followed by a chapter summary in Section 5.3.

5.1 D-CAR: Protocol Description

D-CAR utilizes beaconing messages to extract and disseminate mobility and routing information. Using neighbouring vehicles' mobility vector information, vehicles predict and update LRT for each neighbour in their routing tables. Starting from vehicles passing by RSUs, CRTs to RSUs are calculated and included in beacons. Every vehicle checks the received CRTs in beacons, and the associated LRTs for links to beacons sources, in order to update CRTs in its own beacons. While CRTs are disseminated and updated along road segments, paths are recorded by the means of intersections IDs. Moreover, average packet delivery delay per road segment is calculated in a distributed and probabilistic manner. Available delivery delay information for each road helps vehicles in finding and updating the expected delivery delay for each routing path. When a road segment is connected, by the means of intermediate vehicles, to an RSU or more, path(s) information is carried in beacons. The information of each path includes: 1) the path by the means of consecutive intersection IDs, 2) CRT associated with this path, 3) the experienced delivery delay via this path, and 4) the timestamps of this information.

5.1.1 Framework for Link Lifetime Prediction

Assuming two mobile vehicles are within the transmission range of each other, their mobility will eventually increase the distance between them until it exceeds the transmission range distance causing communication link breakage. It is required to predict the time left for these two vehicles before the communication becomes no longer possible. Giving mobility information at time t_0, we want to predict the remaining time, LRT, before the first possible disconnection occurs due to vehicles mobility. In D-CAR, we do not consider vehicles to be aware of driving routes or traffic lights; however, they are aware of their own and their neighbouring vehicles' locations, speed, driving directions, and turning signals' status, by sharing periodic one-hop beacons.

Within an urban road segment, vehicles follow different driving patterns according to drivers behaviour, driving routes, traffic density, distance to intersection, traffic light status, and other factors; which make the microscopic mobility prediction a challenging task. To mitigate the effects of these factors, we propose grouping neighbouring vehicles in a vehicle's routing table according to: 1) the location and the road segment they belong to, 2) driving direction, and 3) turning signal status. A subgroup called *leading vehicles*, LV, has a special importance in influencing a vehicle's mobility pattern as it includes the set of neighbouring vehicles within the same road segment, having the same signalling status and driving direction, and are in front of the said vehicle.

Unlike highway scenarios, vehicle's speed and acceleration change frequently within short time windows in the city scenarios. Beacons report instant driving conditions which might not be suitable for applying directly to mobility prediction models. The two averaged speed parameters introduced in Chapter 4, ES_m and S_{max} are used in the proposed LRT prediction model. ES_m is the predicted average speed of a vehicles v_m during the prediction time window W. ES_m is considered to be a more stable speed during the target time-window and more suitable to use as compared to the reported speed S_m in v_m's beacon. In D-CAR, we propose using neural networks based model to find ES_m. Pre-defined neural networks parameters are generated for on-line calculation of ES_m with respect to the following inputs: the vehicle's speed S_m, the distance $d_{m,i}$ to the next intersection I_i, the number and the average speed of the leading vehicles, and the number of lanes in the road segment. ES_m is reported in the vehicle's beacons along with S_m.

In some mobility scenarios where vehicles belong to different road segments or have an active turning signal, S_{max} is used instead of ES_m to represent cases that have a vehicle changing its speed from 0 m/s to the maximum speed within the prediction window. S_{max} is a constant design parameter that represents the averaged maximum speed which considers mobility with a maximum acceleration from a stationary condition to the maximum speed in the city roads and maintaining it for a total time of W seconds. R and $\hat{R}$ are also used in D-CAR to represent the effective transmission ranges for the line-of-sight and non-line-of-sight predicted mobility scenarios, respectively.

Short-term Speed Prediction Module

Using local information at v_m's routing table, it is required to predict the average speed of v_m, ES_m, for the next W seconds, where W is the prediction time frame. One fast and easy method to find ES_m is by using neural networks as an efficient data driven approach that relates observed traffic conditions with past traffic data. Neural network (NN) approaches show the capability to map non-linear input and output patterns which make them suitable in solving the complicated non-linear traffic related prediction problems [57, 58, 59]. In our NNs model, three networks have been designed based on the turning signal status. We train the networks to make connections, or weights, between the different factors, or inputs, that we consider. For each network, the following five inputs are considered:

- Vehicle's speed S_m: The current speed of a vehicle is an important factor in mobility prediction, especially in low-traffic scenarios, as it reflects the current driving condition and the driving attitude of the driver;
- Number of leading vehicles: It has been shown by the different traffic model based prediction approaches [23] that leading vehicles have a direct impact on the mobility of the subject vehicle. We limit the concept of leading vehicles in our model to the subset of vehicles that share the same turning signal for more accurate prediction;
- Average speed of the leading vehicles: High variation between S_m and the average speed of leading vehicles can be related to the driver's attitude or a change in the traffic light status, especially when it is considered with the traffic density and the distance to the next intersection;
- The distance to the next intersection $d_{m,i}$: As vehicles mobility patterns can noticeably change around intersections, considering $d_{m,i}$ can give a better prediction

accuracy. Also, because the set of leading vehicles LV considers only vehicles within the transmission range R, $d_{m,i}$ can be related to the traffic density when $d_{m,i} < R$;

- Number of lanes in the road segment: The number of lanes is related to the leading vehicles density.

For neural network learning process, and for constructing weight and bias vectors, MATLAB Neural Networks tool is used together with processed vehicular trace data files. Three NNs are used for the three possible turning signal values: idle, right-turn, and left-tun turning signal indicator statuses. For each NN, a two-layer feed-forward supervised data fitting network is used with five inputs, twenty sigmoid hidden neurons, and a single output, the predicted ES_m. Trace data files are generated using the microscopic vehicle traffic simulator VISSIM [26], and NN input/output patterns are extracted from these files to feed the NN. The NN is trained with Levenberg-Marquardt backpropagation algorithm. The generated parameters, weights and bias vectors, are deployed for real-time prediction of ES_m. More details about the generation of the NNs parameters are presented in Section 5.2.1.

Minimum Link Lifetime Prediction Model

The aim of this model is to predict the time, $t_{m,n}$, of the first possible link break between two communicating vehicles, v_m and v_n. By deploying static map information and mobility vectors information from vehicles routing tables, the model defines a practical prediction approach to be used in VANETs environment. First, the potential driving speeds and directions are extracted as a set of different possible mobility scenarios. Then, the earliest link breakage time is calculated accordingly. In the following, we find the general $LRT_{m,n}(\kappa)$ formula for a mobility scenario κ between v_m and v_n.

Let the Cartesian coordinates of v_m and v_n, at the prediction instant t_0, be $Loc_{m,0}(x_{m,0}, y_{m,0})$ and $Loc_{n,0}(x_{n,0}, y_{n,0})$ respectively. Without loss of generality, assume that v_m and v_n are moving with constant velocity vectors $u_m = (u_{m,x}, u_{m,y})$ and $u_n = (u_{n,x}, u_{n,y})$, respectively, for the following W seconds. With the help of the Pythagorean theorem, the distance d_0 between v_m and v_n at t_0 can be found as follows:

$$d_0 = \sqrt{\Delta x_0^2 + \Delta y_0^2} \tag{5.1}$$

where $\Delta x_0 = x_{m,0} - x_{n,0}$ and $\Delta y_0 = y_{m,0} - y_{n,0}$. It follows that the predicted distance $d_{\Delta t}$ between v_m and v_n after $\Delta t \leq W$ seconds is given by:

$$d_{\Delta t} = \sqrt{(\Delta x_0 + \Delta u_x \Delta t)^2 + (\Delta y_0 + \Delta u_y \Delta t)^2} \tag{5.2}$$

where $\Delta u_x = u_{m,x} - u_{n,x}$ and $\Delta u_y = u_{m,y} - u_{n,y}$. Assuming that the link between v_m and v_n is always functioning while $d_{\Delta t} \leq R$ (or $d_{\Delta t} \leq \hat{R}$ in the non-line-of-sight potential mobility scenarios), $LRT_{m,n}$ is the value of Δt that makes $d_{\Delta t}$ equal to the effective transmission range. By setting $d_{\Delta t} = R$ in Equation 5.2, and solving for Δt, the theoretical link residual time duration for a mobility scenario κ is given by:

$$LRT_{th(m,n)}(\kappa) = \frac{-(\Delta x_0 \Delta u_x + \Delta y_0 \Delta u_y) \pm \sqrt{R^2(\Delta u_x^2 + \Delta u_y^2) - (\Delta x_0 \Delta u_y - \Delta y_0 \Delta u_x)^2}}{\Delta u_x^2 + \Delta u_y^2} \tag{5.3}$$

and the predicted link residual time for that scenario is given by:

$$LRT_{m,n}(\kappa) = min\{LRT_{th(m,n)}, W\} \tag{5.4}$$

According to v_m and v_n locations, mobility directions and signals status, there are different possible potential mobility scenarios to consider. In the following, we present the rules for scenario-generating, which aim to identify the worst possible mobility scenario for the communication link:

1. For a vehicle v_m that has an idle turning signal status, a scenario with a speed of ES_m and the original reported location and driving direction is considered.

2. For a communicating vehicle v_m that has an active turning signal, three scenarios are considered: a) proceeding with the same driving direction and the speed of ES_m, b) stopping at the reported location (i.e., waiting to make a turn), and c) changing the driving direction, according to the turning signal, and proceeding with the maximum average speed S_{max} immediately.

3. For v_m and v_n that are not within the same road segment, and v_m is moving toward the common intersection, two scenarios are added: a) v_m stops immediately (i.e., for a red traffic light) and b) v_m proceeds with the speed of S_{max}.

After generating a set $\mathbb{K}$ of N possible scenarios for a certain case, Equation 5.3 and Equation 5.4 are applied for each scenario $\kappa \in \mathbb{K}$, with respect to substituting the velocity information according to the aforementioned rules. Notice that R is also substituted by $\hat{R}$ in scenarios that consider at least one vehicle's turning or perpendicular mobility directions for communicating vehicles. Applying $LRT_{m,n}$ equation to the different scenarios results in a set of possible correspondent LRTs. The minimum predicted link lifetime between v_m and v_n is the minimum predicted $LRT_{m,n}$ in the set. $t_{m,n}$, the actual predicted time for the earliest link break is given by:

$$t_{m,n} = t_0 + LRT_{m,n} \tag{5.5}$$

The predicted link break times for the different neighbouring vehicles are maintained in the vehicle's routing table. This information is updated frequently to preserve accurate LRTs.

5.1.2 Framework for Network Connectivity Prediction

The framework of network connectivity prediction aims to define light-weight distributed approaches to support individual vehicles with three key routing metrics: 1) a communication path, P, to an RSU, if there is one, 2) the minimum predicted connectivity duration of P, CRT, and 3) the expected delivery delay when packets are forwarded via P. We model the road map as a graph $G(V, E)$ of vertices set V, representing road intersections, and edges set E, representing road segments. Assuming RSUs to be located at road intersections, every routing path $P_z(i)$ consists of a set of consecutive vertices bounded by a vertex that has an RSU, where $P_z(i)$ is a possible path, with an identifier z, from the intersection I_i to the infrastructure.

In order to eliminate additional routing control packets and reduce communication overhead, $P_z(i)$ connectivity information is shared via beacons locally at I_i and its adjacent road segments, i.e., roads that intersect at I_i. Considering RSUs as stationary VANET-s nodes, with LRTs values to their neighbouring nodes, a heuristic approach is used to initiate path information from RSUs to every connected road segment, and calculate associated CRTs accordingly. Packets originators and forwarders inset timestamps of packet originating and the time that packets enters a new road segment, to enable a distributed on-the-fly updating of the average of experienced delivery delay, per road segment, based on two probabilistic methods. Connectivity information is distributed over road segments via vehicles' beacons providing proactive dynamic route alternatives for vehicles on the connected roads.

Dynamic Connectivity-Awareness Model

D-CAR enables vehicles to receive instant connectivity information when one or more RSUs are reachable via multi-hop routing. In this section, we are presenting different procedures that allow D-CAR to find, distribute, and update the paths to RSUs, the minimum connectivity duration of each path, as well as the expected data packet delivery delay per path. These procedures run distributively and simultaneously. In the following, we describe them consecutively.

a) Constructing Routing Paths

Each vehicle v_m attaches, in its periodic beaconing messages, a set $\mathbb{P}$ of available routes to the core network and the associated connectivity information to each path. Paths in D-CAR are anchor-based routes by the means of intersections. Each path $P_z(i) \in \mathbb{P}$ consists of an ordered sequence of intersections, or junctions, starting from the closest one, I_i, and identified by a locally-unique randomly generated identifier z. Paths are initiated and updated by vehicles at intersections, and carried to the other connected intersections via vehicles beacons. Consider a path $P_z(i) = \{(i,i)\}, I_i \in V$, that has been initiated at intersection I_i by a vehicle that is connected to an RSU at I_i. This path information is piggybacked and distributed via vehicles beacons until it reaches adjacent connected intersections. A vehicle that receives $P_z(i)$ and identifies itself to be located at an adjacent intersection I_j, updates the path to include the new intersection, i.e., the updated path becomes $P_z(j) = \{(j,i)\}$. Similarly, a vehicle at a next intersection I_k that receives $P_z(j)$ updates it to be $P_z(k) = \{(k,j),(j,i)\}$, and so on. While updating a path P_z by a vehicle v_m, the following rules should be observed:

- If v belongs to $e_{i,j} \in E$, v extracts $\mathbb{P}$ only from neighbouring vehicles belonging to $e_{i,j}$, I_i, and I_j

- When v belongs to $e_{i,j}$, and it has received a beacon from a vehicle in I_i, a received path $P_z(i)$ is excluded if there is a path $\hat{P}_w(j)$ at v's routing table such that $P_z(i) = \{\hat{P}_w(j) \cup (i,j)\}$
- When v belongs to I_i, if the intersection I_i is found in an inner junction in a received path $P_z(j)$, $P_z(j)$ is excluded as it would have a redundant sub-route from I_i
- When v belongs to I_i and it has received a beacon that includes a path $P_z(j)$, v changes the path's id, z, while updating the path information if there is another path $\hat{P}_z(i)$ in its routing table and $\hat{P}_z(i) \neq \{P_z(j) \cup (i,j)\}$

b) Minimum Connection Duration

In D-CAR, every connected vehicle v_m, that is connected to the core network, maintains in its routing table the connectivity information that includes the set $\mathbb{P}$ of the different paths and the associated expected time of expiry, $T_z(i)$ of each $P_z(i) \in \mathbb{P}$, as well as the expected time $t_{m,n}$ for the first link break between v_m and every neighbouring vehicle v_n. When initiating paths information by vehicles that are directly connected to RSUs, CRTs of the generated paths are the LRTs between these vehicles and RSUs. The expiry time of the path $P_z(i)$ in this case is given by:

$$T_z(i) = t + CRT_z \tag{5.6}$$

where t is the current system time and CRT_z is the connectivity residual time of the path at $T_z(i)$ generating instant. Connectivity information, or paths information, is distributed and updated among vehicles via their beaconing messages. Each vehicle v_m updates $T_z(i)$ value of each path $P_z(i)$ received from a neighbouring vehicle v_n according to the previous $\hat{T}_z(i)$ value in its routing table, the received $T_z(i)$ from v_n, and $t_{m,n}$ as follows:

$$T^*_z(i) = max\{\hat{T}_z(i), min\{T_z(i), t_{m,n}\}\} \tag{5.7}$$

where $T_z^*(i)$ is the updated $T_z(i)$ value at v_m which will be stored in its routing table and included in its upcoming beacons. Equation 5.7 enables D-CAR to check all the possible connections between vehicles and maintain only the information about the expected long-life connections.

c) Expected Delivery Delay

Delivery delay in D-CAR is calculated using distributed probabilistic methods. Junction-to-junction delivery delay, $D_{i,j}$ is considered, and expected path delivery delay $D_z(i)$ is updated at I_i according to the most recent $D_{i,j}$ available at the intersection. D-CAR deploys two methods in order to find the experienced delay $D_{i,j}$ per road segment $e_{i,j}$: average delivery delay for a set of data packet samples forwarded via $e_{i,j}$, and delivery delay of a *probe* message traversing $e_{i,j}$. First, D-CAR requires data packets forwarders at intersections to attach and update timestamps at packets' headers for the time that packets pass by the most recent routing junction. A vehicle v_m at intersection I_j that is forwarding packets from $e_{i,j}$ towards the next routing junction, utilizes, with a probability of p_{SD}, these timestamps to find the average delivery delay $D_{i,j}$ for M data packets forwarded from I_i to I_j. v_m includes $D_{i,j}$ in its beacons together with the time of its originating. Vehicles within I_j and $e_{i,j}$ re-broadcast the most recent $D_{i,j}$ information in their beacons. p_{SD} is related to the originating time of the last known $D_{i,j}$ in v_m's routing table, t_D. We design p_{SD} to be calculated as follows:

$$p_{SD} = \frac{1}{1 + e^{(t_D + \frac{\tau_D}{2} - t)}} \tag{5.8}$$

where t is the system actual time and τ_D is a control constant representing the time window for a major change in data traffic or vehicular traffic volume at a road segment in the city environment, which controls the frequency of $D_{i,j}$ updates.

Second, a vehicle v_m that enters $e_{i,j}$ send a *probe* unicast message to I_j location with a probability $p_{prob} = p_{SD}/|LV|$ where $|LV|$ is the number of v_m's leading vehicles. The probe message consists of a number of packets with the same priority of data packets that are used to experience the delivery delay locally at road segments. The closest vehicle to the location of the target junction receives the probe packets and calculates $D_{i,j}$. Probe message is used to support D-CAR with $D_{i,j}$ when there is no data traffic at a certain road segment and reflect the other communication attributes that affect the delivery delay such as road length and number of intermediate forwarders.

Finally, when a vehicle v_m at intersection I_j updates $\mathbb{P}$ information, for a $\mathbb{P}$ forwarded from a vehicle at $e_{i,j}$, v_m utilizes the available $D_{i,j}$ to assign $D_z(j)$ for every received $P_z(i) \in \mathbb{P}$. Given $D_z(i)$ that has been carried in beacons together with $P_z(i)$ from I_i, $D_z(j)$ is calculated as follows:

$$D_z(j) = D_z(i) + D_{i,j} \tag{5.9}$$

in other words, the expected delivery delay of a path $P_z(i)$ is the sum of the experienced packets delivery delay via the individual road segments consisting $P_z(i)$. Vehicles do not distribute connectivity information for paths that have CRT_zs below a certain threshold ε_{CRT}, or when $D_z(i)$ exceeds a certain threshold ε_D. Path parameters are always updated with the most recent information based on attached timestamps.

5.1.3 Data Packets Routing

D-CAR is a layer 3 protocol that is responsible to efficiently route data packets among mobile OBUs and stationary RSUs. D-CAR can cooperate with layer 2 and upper layers protocols for reliable Internet packets delivery. The details of core network architecture, mobility and handover management, and inter-domain cooperation are out of the scope of our work. However, we will briefly discuss the compatibility of D-CAR with existing mobile Internet protocols such as Proxy Mobile IPv6 (PMIPv6) [60]. Then, we will describe the different operations of D-CAR for data packet routing in VANETs.

PMIPv6 is a mobility management protocol standardized by IETF to allow mobile nodes (MNs) to change their points of attachment to the Internet without changing their IP addresses. In PMIPv6, MNs are associated with mobile access gateways (MAGs) which are connected to a local mobility anchor (LMA). MAGs detect connection and perform the required signalling with LMA which manages all traffic from and to MNs, maintains the routes to MNs, and manages MNs prefixes in each administrative domain. In predictive and reactive fast handover for MIPv6 (PFMIPv6), MAGs buffer data packets, forward them to MNs after new connections are established, and initiate tunnels between previous and next MAGs to route packets in the core network during the handover process. Other adaptations to PMIPv6 dedicated to vehicular environment has also been proposed as in [61].

D-CAR is compatible with the architecture of PMIPv6 as MAGs can be implemented at RSUs, and LMAs can be either added to the VANETs structure or be implemented at VANETs location servers. However, in the simple VANETs drive-thru Internet model, where a vehicle has Internet access when passing by an RSU, PMIPv6 will encounter a large number of handover calls due to the high speed of vehicle's mobility and the short communication range of OBUs and RSUs. However, as D-CAR provides connection to RSUs information and enables multi-hop routing, vehicles can be associated to same RSUs for longer periods which significantly reduces handover between RSUs. Moreover, supporting vehicles with path alternatives allows vehicles to predict handover, prior to path changing, which facilitates PFMIPv6.

In PBR, data packets are routed using its final destination's location and identifier. Location Service is one of the vital components of any PBR protocol to obtain the location of packets destination. In D-CAR, location server is required, e.g., LMA, and vehicles periodically report their locations to the server while they are connected to the core network. Source routing is used in both uplink and downlink routing where source node attaches the selected path, by the means of intersections IDs, to packets' headers. In uplink routing, packets are forwarded from junction to junction until reaching the designated RSU. Similarly, in downlink routing, the associated RSU uses the same path, in addition to the road segment ID of the destination vehicle. When packets reach the destination road segment, each forwarder checks the destination vehicle's ID in its routing table while packets traverse the road segment.

When a packet cannot be delivered via its original path in uplink routing, the forwarder encapsulates the packet and uses another available path if the road segment is connected to the core network; otherwise, the packet is dropped. Among the available paths, a routing path is selected by a source vehicle with respect to the required time to transmit the message, the CRT of each path, and the expected delivery delay. This information helps source applications to check the connection quality before transmission, which supports the variations of traffic types and QoS requirements in multi-hop VANETs. Within a road segment, packets are forwarded towards the next intersection with respect to the channel quality between the forwarder and each neighbouring vehicle, LRT, and the progression to the next intersection each potential next-forwarder can make. The next-forwarder is the neighbouring vehicle that has acceptable channel conditions, sufficient LRT, and makes maximum progression.

When a vehicle is receiving a downlink packets stream and enters a new road segment, it appoints a representative to forward the incoming packets to it in the new road segment while updating its location at the location server. The representative vehicle is a neighbouring vehicle that is entering the previous road segment of the appointing vehicle. It sets a timer for being a representative, announces its representation in its beacons, receives the appointing vehicle's packets and uses source routing to forward the packets to the new destination of the appointing vehicle.

5.2 Performance Evaluation

In this section, we will evaluate the performance of the individual components of D-CAR, as well as the overall network performance with the protocol deployed, via intensive simulation scenarios and analysis. First we will examine the capability of the designed NN to estimate a vehicle's average speed during a certain time window. Then, we will evaluate the ability of the proposed framework in Section 5.1.1 to predict a minimum LRT between communicating vehicles. In addition, the overall D-CAR performance will be evaluated by means of end-to-end packet delivery ratio (PDR) and packet delivery delay (PDD).

5.2.1 Simulation Setup

A city scenario simulation platform has been developed using the microscopic vehicular traffic generator VISSIM [26] and MATLAB. In this setup, and for more realistic routing scenarios, the area around University of Waterloo has been simulated with 18 road segments and 6 controlled intersections as shown in Figure 5.1. Vehicular traffic generation follows the setup in [62]. Turning signals are set to be activated 30 m before making turns. Simulation scenarios with different vehicular traffic densities, data traffic densities, and RSUs number have been designed to evaluate the proposed protocol. Packets sources, in these scenarios, are always 10% of the vehicles, and data traffic densities are controlled by the packet generating rate (PGR). Simulation scenarios consider the deployment of one RSU at the southern university entrance, 2 RSUs at the southern and the northern

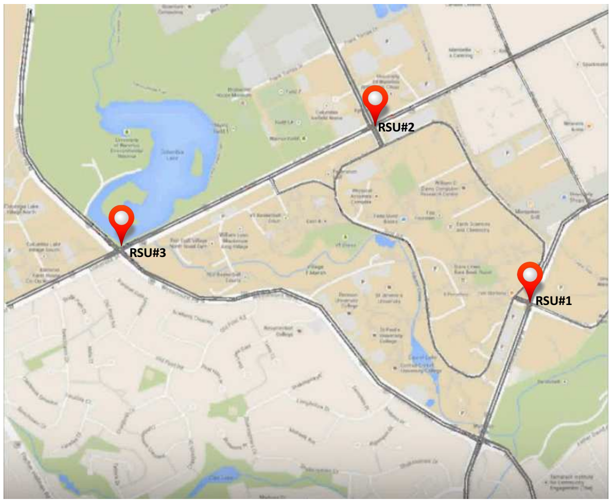

Figure 5.1: The Simulated Area around University of Waterloo

entrances, and 3 RSUs at the two campus entrances in addition to the main intersection at the west of the road map as shown on Figure 5.1.

D-CAR is implemented to route data packet streams from source vehicles to connected RSUs. Other PBR protocols, GPSR [18], GSR [20] and GyTAR [21], are also deployed for comparison purposes. GPSR, GSR and GyTAR are PBR protocols that have been widely applied as benchmarks for new protocols evaluation in the VANETs context. GPSR uses distributed position-based greedy routing, GSR uses map-based source routing while GyTAR utilizes map information and real-time traffic information for intersection-based distributed routing. Different scenarios have been designed with different vehicular and data traffic densities and prediction periods.

With the focus on the routing evaluation purposes of this simulation, ideal physical channel is considered, a simple DCF MAC is applied, and a FIFO data packets queue is deployed at each vehicle. For NN design, VISSIM trace files are used to train, validate, and test the NN in order to obtain the NN's *weight* and *bias* values. These values are

Table 5.1: Simulation Parameters for D-CAR Evaluation

Parameter	Value	Parameter	Value
Scenario Duration	40 s	W	10, 15, 20, 25 sec
Scenario Repetition	5-12 times	S_{max}	14 m/sec
R	250 m	τ_D	3 sec
$\hat{R}$	150 m	$\epsilon_D, \epsilon_{CRT}$	1 sec
Packet Lifetime	1000 ms	Mobility speed	0-16 m/sec

Table 5.2: Correlation and RMS between the NNs Model Outputs and Target ES

W (Seconds)	5	10	15	20	25
Correlation	0.97	0.96	0.93	0.87	0.80
RMS	0.02	0.04	0.08	0.11	0.18

used in the evaluation of the LRT model as well as the overall D-CAR performance. Other parameters that have been used in this evaluation are listed in Table 5.1.

5.2.2 Simulation Results

Link Lifetime Prediction

Results of the designed NNs model demonstrate the ability of the model to relate the different known inputs with the average speed of a vehicle in the time period ahead. Table 5.2 presents the correlation and RMS between the actual average speed of the test sample vehicles and the corresponding NNs outputs, with different prediction periods. Figure 5.2 also shows the correlation between the normalized estimated average speed and the target values for a certain prediction time period. The results show that the accuracy of the model is dependent on the prediction time window W. However, even with a larger time

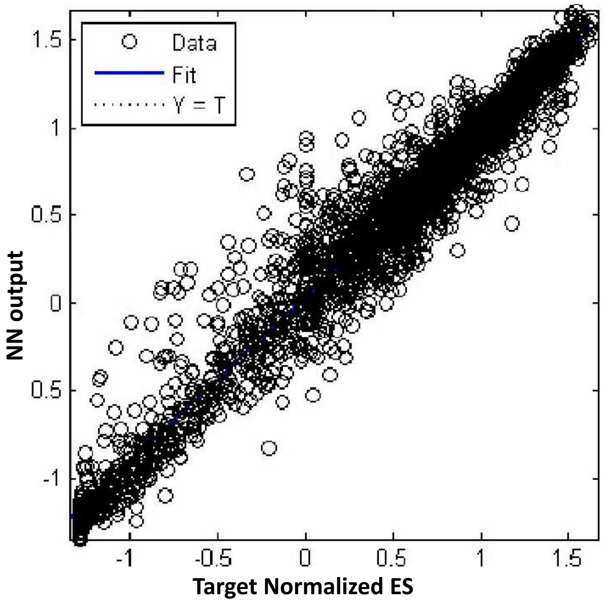

Figure 5.2: Correlation between the NNs outputs and the Target ES with $W = 10$ *sec*

window, e.g., $W = 20$ *sec*, the performance of the model is acceptable and the estimated speed, ES_v, correlates with the actual average speed of a vehicle much better than the reported speed S_v.

Moreover, the results show that the minimum link lifetime prediction model successfully provides prior validity link information for more than 97% of the cases for the different values of W for up to 20 seconds. In other words, the neighbouring communicating vehicles does not experience link breakage during the predicted link lifetime between them. However, although the LRT model successfully predicts links failure within W, simulation shows that the actual link lifetime between vehicles can be much longer than the predicted values. This is because the LRT model selects the worst anticipated mobility scenario, with respect to the communication link, which does not always occur. The results indicate the validity of using estimated average speed for a vehicle in the city scenario, instead of relying on the actual reported speed and acceleration, to predict mobility, as well as the ability of the proposed rules in Section 5.1.1 to predict the worst mobility scenario for a certain communication link. Only in few cases ($< 3\%$), the link has been broken just before the end of the predicted LRT due to inaccuracy of the estimated average speed. The frequent calls of LRT prediction process can maintain accurate and updated link information at vehicles routing tables.

Network Connectivity

We examine the ability of D-CAR to supply vehicles with connectivity information to the core network. Routes information are checked at vehicles routing tables at random instances, with different traffic densities and RSUs placement simulation scenarios. Figure 5.3 shows the percentage of connected road segments as reported by D-CAR. The number of connected road segments to RSUs increases with the increase of vehicles density and the number of RSUs, as the network becomes more connected. With the deployment of only one RSU at the southern part of the simulation area, and with lighter vehicular traffic, limited number of adjacent road segments to the RSU is reported to be connected to the core network. However, once the vehicular traffic density becomes sufficient to connect the northern road segments to the southern parts, a sharp increase occurs in the number of connected road segments as shown in Figure 5.3.

Figure 5.4 shows the average CRTs, found by D-CAR, at the moment of updating routes information, with respect to different traffic densities and prediction window values. The results show that D-CAR is able to predict route lifetimes that are sufficient for multi-hop data delivery and mobile data packets offloading. Higher vehicular traffic and increased prediction periods increase the average predicted CRTs, however, high W values affect the network performance as will be shown in the next section.

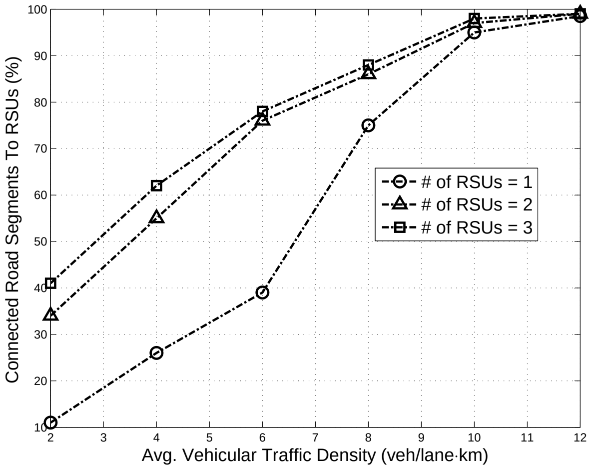

Figure 5.3: Percentage of Connected Road Segments to RSUs with $W = 15$

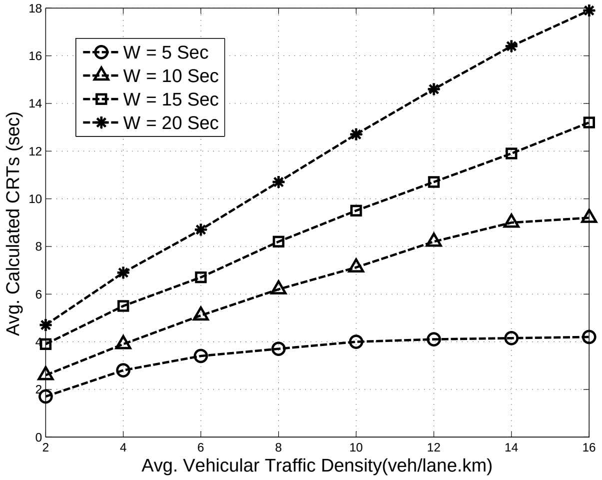

Figure 5.4: Average CRT in the Network with 2 RSUs

Packet Delivery Ratio

By defining packet delivery ratio (PDR) to be the ratio of data packets that are successfully delivered to RSUs, to the total data packets sent by OBUs, D-CAR is expected to have a high PDR as compared to other PBR protocols, as vehicles buffer their data packets if the connection to RSUs is not confirmed. Buffering data packets significantly impact the overall VANETs performance as it conserves the network bandwidth and reduces the queuing delay. The simulation results confirmed the expected performance of D-CAR and its advantage over the other protocols under consideration.

Figure 5.5 shows the average percentage of buffered, sent, and received data packets, with respect to different vehicular traffic densities. With light traffic densities, where the network is intermittent and the relaying resources are limited, more packets are buffered and PDR is noticeably high (PDR$>$ 0.85). With more vehicular traffic, the network becomes more connected and more packets are sent. It is shown that with higher vehicular density PDR is further improved as the routes become stable and less sensitive to individual vehicles' mobility.

When compared to other PBR protocols, D-CAR has a much higher PDR as it is a connection-aware routing protocol. In GPSR, GSR, and GyTAR, packets are always sent whether the vehicle is connected to an RSU or not. Figure 5.6 presents the simulation results of PDR for the different routing protocols with respect to vehicular density, as well as the PDR of D-CAR with different W values. The results validate the advantage of connectivity awareness over the traditional PBR performance.

In low traffic densities, the PDRs using GPSR, GSR, and GyTAR are homogeneous, and most of the transmitted packets are dropped due to the lack of a connection to the core network. With high traffic, GPSR exhibits the worst performance as it does not

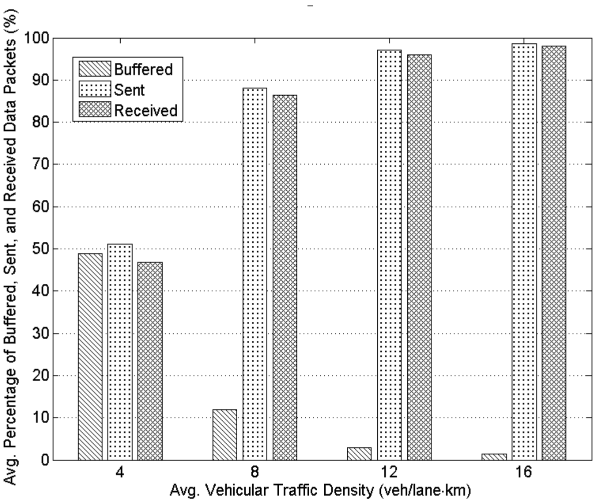

Figure 5.5: Percentage of Buffered, Sent, and Received Data packets with 2 RSUs and W= 15 Seconds

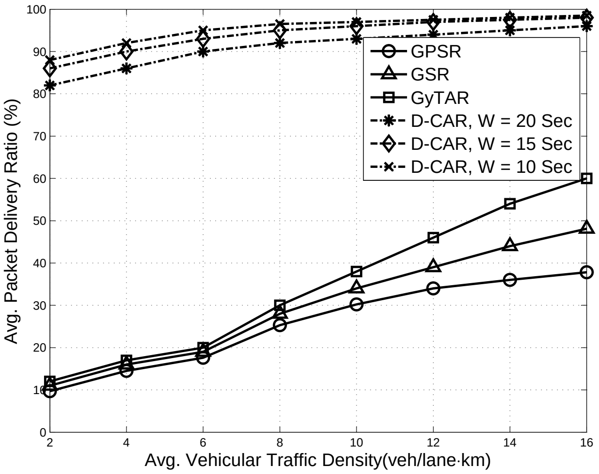

Figure 5.6: Average Packet Delivery Ratio with 2 RSUs

consider road map or overlay routing, and packets encounter a high number of intermediate forwarders. Packets are dropped when the routing protocol fails to find a next forwarder according to the forwarding strategy, or when the packets reach their expiry time. GyTAR shows an improved PBR performance as it selects routes according to the real-time traffic information, yet, its connectivity awareness is limited to one junction only and does not extend to the core network.

As packets in D-CAR are only transmitted if a connection to the core network is reported, the number of packets sent by D-CAR is low, especially in low traffic density scenarios, and the PDR is higher, as compared to using the other PBR protocols. The results also show that higher value of W can degrade PDR in D-CAR as the reported connection duration becomes less accurate. Prediction windows shorter than 15 *secs* show comparable packet delivery performance.

Packet Delivery Delay

Offering delivery delay information is a major feature in D-CAR which enables source vehicles to select connected paths with reduced expected delivery delay. This feature supports applications that have known QoS delay constraints. Simulation results confirmed that D-CAR is able to maintain a significantly reduced average packet delivery delay (PDD) with respect to both vehicular traffic density and data traffic density.

PDD is defined to be the time consumed from the transmission of the data packet by its originator until it is received by an RSU. Average PDD is the average packet delivery delay for the successfully delivered packets to the core network. With low data traffic and light deployment of RSUs, average PDD is affected mainly by the number of intermediate forwarders. It is shown in Figure 5.7 that in the sparse network scenarios, the different

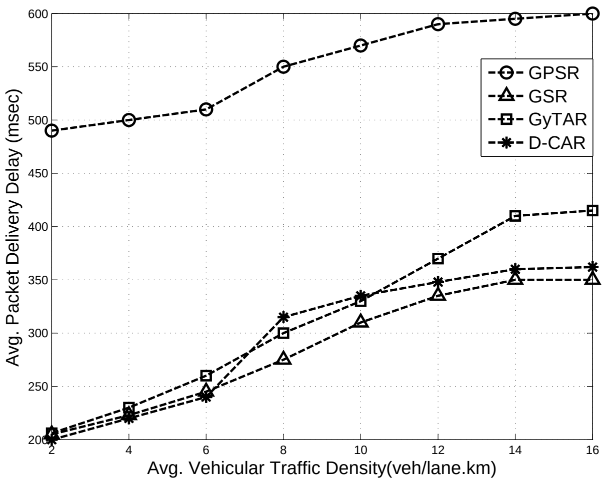

Figure 5.7: Average Packet Delivery Delay with 1 RSU and low data traffic

protocols have analogous average PDD, except GPSR which suffers from the frequent calls of perimeter routing in the sparse VANETs which causes higher number of intermediate hops. With a moderate traffic density and slightly connected roads, D-CAR shows a reasonably higher average PDD compared to GyTAR and GSR. Packets in these scenarios travel through longer paths, yet, connected. As a result, a noticeable increase in the sent packets and PDR is observed in these scenarios as shown in Figure 5.5. In high vehicular densities, D-CAR maintains a high PDR with a reduced PDD. GSR shows a slightly less average PDD compared to D-CAR as it considers a shortest path with a small number of intermediate forwarders. However, this routing strategy is inefficient and shows a low PDR as shortest paths are not always connected.

Figure 5.8 shows the effect of increasing data traffic on the average PDD in a connected network. These scenarios examine the capability of a routing protocol to cope with data congested routes. D-CAR is designed to dynamically adjust routing paths based on the experienced delivery delay per road segment. Thus, the results show a significant reduction

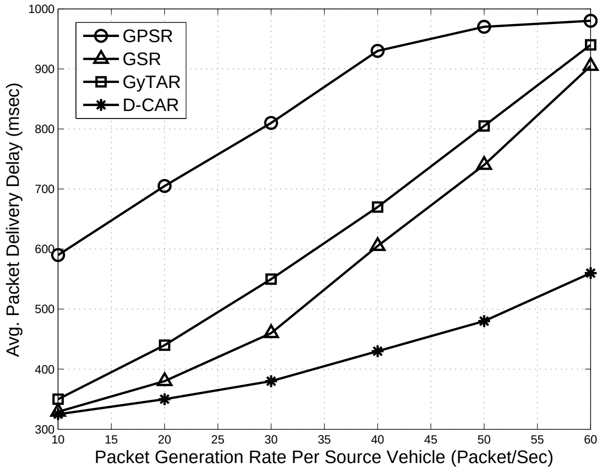

Figure 5.8: Average Packet Delivery Delay with 2 RSUs and moderate vehicular traffic

in average PDD for D-CAR compared to other protocols, especially in high PGR scenarios. RSUs here are the bottlenecks of the network which determine the increase in the average PDD in high PGR scenarios. GSR has a sharper increase in PDD with respect to the increase of PGR as it uses static routes causing data congested roads and long queuing delay.

Routing Overhead

Routing protocols commonly introduce communication and computation overhead to function. Communication overhead is defined to be the extra bytes transmitted for the purpose of enabling the data packet routing process. In D-CAR, it is required to share mobility, paths, and delay information among vehicles. In order to minimize the communication overhead, D-CAR utilizes beaconing messages to distribute this information. As mobility information is required to be included in beacons by the safety applications, only paths, delay, and associated timestamps information is added to the beacon's payload.

Probe messages are another type of routing control messages introduced by D-CAR to collect delay information in certain situations, however, probe packets are unicast, sent locally within road segments, and sent only with low probability when there are no enough data packets traversing a road segment as shown in Section 5.1.2. Thus, its effect is very limited and can be negligible.

In order to enable a wide network connectivity view to individual vehicles, D-CAR involves many simple and distributed calculation operations to be performed by vehicles. Each vehicle has to apply some computations on their one-hop neighbouring vehicles information in order to obtain supporting routing parameters. These parameters include the link residual time with each neighbour, whether the neighbouring vehicles is a leading vehicle or not, and the estimated speed of the vehicle itself. These operations are done with much less frequency than the beaconing rate. For simplicity we have used a random interval to update routing table's entries for each neighbour within a time frame of 3 seconds. More advanced information update techniques can be used considering the change in relative speed, moving distance and/or local vehicular traffic density. OBUs are considered to have sufficient computation power to perform such operations. Moreover, NN training, for different roads and intersections types, is an off-line operation and only the prediction stage is required by a vehicle's OBU for its ES prediction.

5.3 Summary

In this chapter, we have introduced a routing protocol that proactively supports vehicles with connectivity information to nearby RSUs in order to enable multi-hop Internet access and mobile data offloading. The proposed protocol, D-CAR, is capable of utilizing the one-hop mobility awareness and the predictable vehicles movement for better estimation of link residual time between vehicles, different routes to RSUs, the minimum connection lifetime for each route, as well as its expected delivery delay. Proactive link lifetime information and connectivity awareness help VANETs users to take better transmission decisions and preserve the network bandwidth. D-CAR dynamically suggests routing the packets via one or more paths, or postponing the transmission, based on the real-time changes on VANETs topology.

Providing alternative routes with associated expected delivery delay maximizes the utilization of the network resources by selecting connected paths, including longer paths, with reduced delivery delay. We have shown that D-CAR successfully supports vehicles with connectivity and routes information sufficient for efficient multi-hop data delivery and packets offloading within the city environment. It is found that with the prediction window of $W = 15$ *sec*, D-CAR can maintain an average speed prediction accuracy of 93%, valid LRT for more than 98% of the links, up to 13.6 *sec* average connection lifetime, and a high average PDR. Compared to other PBR protocols, D-CAR shows significant improvement in VANETs performance associated with the advantages of connectivity awareness and dynamic routing.

Chapter 6

Conclusion and Future Work

6.1 Summary and Conclusion

Motivated by the promising applications of multihop VANETs, and the increasing demand of mobile data, this thesis set out to address the existing challenges in VANETs routing in order to design a routing protocol that has the ability to efficiently support VANETs users with proactive routing information. In order to resolve routing problems related to the high dynamics of VANETs topology in the traditional topology-based routing, and the limited topology awareness in the position-based routing, we have proposed three connectivity-aware routing protocols: iCAR, iCAR-II, and D-CAR. Different algorithms and models have been introduced to predict communicating vehicle's mobility in order to estimate links residual time between each pair of vehicles, as well as connectivity residual time from a certain location to one gateway RSU or more.

First, we have considered broad connectivity, or road-level connectivity, awareness in iCAR. iCAR is an intersection-based connectivity-aware routing protocol that uses distributed routing to forward data packets from junction-to-junction based on the available connectivity and score information of the adjacent junctions, as well as location information of the adjacent junctions and packets' destination. Broad connectivity information, including the estimated minimum road-level connectivity duration, is proactively calculated by iCAR and becomes available at each intersection. Intersections are ranked by iCAR according to some dynamically updated parameters such as the experienced packet delivery delay and vehicular traffic density per road segment. Although iCAR has investigated some key routing parameters in urban VANETs, and shown better performance than its PBR counterparts, it does not provide comprehensive connectivity information to VANETs users and, accordingly, can not be efficiently deployed for Internet access or mobile data offloading.

iCAR-II, on the other hand, have considered comprehensive connectivity to the infrastructure in order to enable global network topology awareness. With the assistance of the reliable cellular network channels, vehicles at intersections dynamically reports broad connectivity and packet delivery delay information to LCs. LCs update the network topology and send customized NSI packets to representative vehicles at intersections. NSI is distributed locally, attached to vehicles beacons, for proactive connectivity awareness. In iCAR-II, an improved deterministic mobility prediction model is introduced that takes into consideration the different possible mobility scenarios that each pair of neighbouring vehicles can follow, according to their current mobility case. Due to comprehensive connectivity awareness, iCAR has shown a significant improvement in the overall VANETs performance on the cost of using cellular network channels partially for routing purposes.

Different than iCAR, we have proposed D-CAR to support VANETs users with comprehensive connectivity information, yet, without the assistance of cellular networks. In iCAR-II and D-CAR, VANETs applications utilize connectivity information to take transmission decisions. With a prior knowledge of a confirmed CRT and experienced delivery delay, VANETs applications can initiate low-cost communication sessions using VANETs, use alternative networks, or reschedule their transmission. D-CAR improves the accuracy of LRT estimation by predicting vehicles speed, during a certain prediction time-window, with the help of current traffic conditions and NNs. The analysis and simulation-based performance evaluation demonstrated that the proposed algorithms and protocols can provide real-time connectivity information to VANETs users, and efficient and dynamic data routing. The proposed protocols have shown significant improvement in PDR and PDD, as well as network's bandwidth saving which improves the overall VANETs performance.

The work in this thesis verifies the validity of using VANETs for multi-hop Internet access and mobile data offloading, even with a light deployment of RSUs. For example, it is found that with the prediction window of $W = 15$ *sec* and 2 RSUs deployed in the vicinity of University of Waterloo, D-CAR can maintain an average speed prediction accuracy of 93%, valid LRT estimation for more that 98% of the links, up to 13.6 *sec* of reported average connection lifetime, and a high average PDR. Although the focus of this work was only on the routing problem and performance, this work gives a base to resolve other challenges, such as secure routing and IP sessions management, before multi-hop VANETs is fully enabled for Internet-base services and data offloading. We highlight some of these applications in the following section.

6.2 Future Work

The proposed protocols and algorithms can be utilized to extend the research work in several directions, such as:

Cross-layer Link Residual Time Prediction

In our current work, we have considered information extracted from network layer and upper layers, such as mobility information, to predict the availability of radio resources for communication during a prediction time window. Channel status information has not considered in the LRT estimation process. As vehicles, with variable sizes, and vehicular traffic densities, can degrade the link quality between neighbouring vehicles, considering lower layers information can improve the LRT estimation. Combining the proposed prediction model with a physical-layer prediction method, such as the work in [45], to include the estimated channel condition per each link, is an interesting research direction that enhances the accuracy of the prediction method. Predicting the status of the channel between two vehicles will not only help in better LRT and CRT estimation, but also can be combined with the greedy routing for an optimized next-hop selection strategy.

IP-Session Management in Multihop VANETs

Our focus in this thesis was on confirming the availability of valid network layer paths to route packets from source vehicles to the infrastructure, in order to support Internet-based services and data offloading applications. With confirmed connection resources and known $CRTs$, IP-session management becomes a key research subject to enable Internet access in multi-hop VANETs. In Section 5.1.3 we have introduced the compatibility of our work with IPv6 structure, and the promising enhancements that could be achieved by combining our work with PMIPv6 [60] in session handover from an RSU to another, and the overall session management.

Location Server Management with known Link and Connectivity Information

Throughout our work, LCs have been treated as a one logical unit. As VANET is a large scale network, maintaining reachable and up-to-date location information for VANETs users requires the design of an efficient LCs management scheme. Adding the tasks for topology-awareness and routes providing to LCs complicates the design of this scheme. In Section 1.5, we have highlighted one way to design such scheme, taking into consideration the geographically distributed nature of VANET. In this scheme, the city-road map can be divided into a number of vicinities and each server is responsible for one or more vicinities. Adjacent vicinities can exchange connectivity information, delivered by our protocols, for better reach to destinations as well as a wider real-time network topology awareness.

Fast Data and User Authentication in Secure Multi-hop Routing

For secure routing, we have previously proposed a cryptographic scheme for data integrity and user authentication which is based on digital signature and hash-chain functions [63]. As cryptographic operation introduces significant delay, multihop forwarders authentication can degrade the network performance. The proposed LRT model in this work can be used to accelerate the encryption-based user and data authentication phases in secure routing. As the communication link is guaranteed to be valid for a certain period of time, the number of signature verifications per beacon or message can be reduced, and hash chain values can be disclosed per packet transmission for authentication during LRT. This will reduce the verification delay from milliseconds scale to microseconds, which significantly supports the multihop delivery of Internet packets.

References

[1] "Car2Car Communication Consortium." [Online]. Available: http://www.car-to-car.org

[2] IEEE Working Group, "Standard specification for telecommunications and information exchange between roadside and vehicle systems-5 GHz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications," *ASTM DSR STD E2313-02*, 2002.

[3] "National Highway Traffic Safety Administration (NHTSA)." [Online]. Available: www.nhtsa.gov/

[4] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency Band," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, Oct 2007.

[5] F. Rebecchi, M. D. de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 580–603, Secondquarter 2015.

[6] X. Wang, C. Wang, J. Zhang, M. Zhou, and C. Jiang, "Improved rule installation for real-time query service in software-defined Internet of vehicles," *IEEE Trans. on Intell. Trans. Sys.*, vol. 18, no. 2, pp. 225–235, Feb. 2017.

[7] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE WMCSA*, 1999, pp. 90–100.

[8] D. B. Johnson, D. A. Maltz, and J. Broch, "Ad hoc networking." Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, ch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pp. 139–172. [Online]. Available: http://dl.acm.org/citation.cfm?id=374547.374552

[9] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, Jun. 2007.

[10] C. A. T. H. Tee and A. C. R. Lee, "Survey of position based routing for Inter Vehicle Communication system," in *Proc. IEEE DFmA*, 2008, pp. 174–182.

[11] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and road safety service support in vehicular networking: From a communication perspective," *Mech. Syst. Signal Process., Special Issue Integr. Veh. Dyn.*, vol. 25, no. 6, pp. 2020–2038, Aug. 2011.

[12] J. Yan, M. Zhou, and Z. Ding, "Recent advances in energy-efficient routing protocols for wireless sensor networks: A review," *IEEE Access*, vol. 4, pp. 5673–5686, 2016.

[13] C. Suthaputchakun and Z. Sun, "Routing protocol in intervehicle communication systems: a survey," *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 150–156, Dec. 2011.

[14] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *IEEE ICC 2000*, vol. 1. IEEE, 2000, pp. 70–74.

[15] Z. J. Haas, “A new routing protocol for the reconfigurable wireless networks,” in *IEEE 6th ICUPC*, vol. 2, Oct 1997, pp. 562–566 vol.2.

[16] J. Zhao and G. Cao, “VADD: Vehicle-assisted data delivery in vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1910–1922, 2008.

[17] J. He, L. Cai, P. Cheng, and J. Pan, “Delay Minimization for Data Dissemination in Large-Scale VANETs with Buses and Taxis,” *IEEE Trans. on Mobile Comput.*, vol. 15, no. 8, pp. 1939–1950, Aug 2016.

[18] B. Karp and H.-T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in *Proc. 6th ACM MobiCom*, 2000, pp. 243–254.

[19] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, “A-star: A mobile ad hoc routing strategy for metropolis vehicular communications,” in *Proc. NETWORKING*, 2004, pp. 989–999.

[20] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, “A routing strategy for vehicular ad hoc networks in city environments,” in *Proc. IEEE IVS*, 2003, pp. 156–161.

[21] M. Jerbi, S. Senouci, T. Rasheed, and Y. Ghamri-Doudane, “Towards Efficient Geographic Routing in Urban Vehicular Networks,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5048–5059, Nov. 2009.

[22] K. Abboud and W. Zhuang, *Mobility Modeling for Vehicular Communication Networks.* Springer, 2015.

[23] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Commun. surveys & Tutorials*, vol. 11, no. 4, pp. 19–41, Dec. 2009.

[24] R. R. Roy, *Handbook of mobile ad hoc networks for mobility models.* Springer Science & Business Media, 2010.

[25] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo–simulation of urban mobility," *Int. J. On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.

[26] "VISSIM transport and traffic simulation," http://vision-traffic.ptvgroup.com/en-uk/home/, accessed: 2016-11-16.

[27] S. Olariu and M. C. Weigle, *Vehicular Networks: From Theory to Practice*, 1st ed. Chapman & Hall/CRC, 2009.

[28] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for automobile users," in *IEEE INFOCOM 2004*, vol. 1, Mar. 2004, p. 373.

[29] W. L. Tan, W. C. Lau, O. Yue, and T. H. Hui, "Analytical Models and Performance Evaluation of Drive-thru Internet Systems," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 207–222, Jan. 2011.

[30] T. Luan, X. Ling, and X. Shen, "MAC in Motion: Impact of Mobility on the MAC of Drive-Thru Internet," *IEEE Trans. on Mobile Comput.*, vol. 11, no. 2, pp. 305–319, Feb. 2012.

[31] M. Wang, Q. Shen, R. Zhang, H. Liang, and X. Shen, "Vehicle-Density-Based Adaptive MAC for High Throughput in Drive-Thru Networks," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 533–543, Dec. 2014.

[32] H. Zhou, B. Liu, F. Hou, T. Luan, N. Zhang, L. Gui, Q. Yu, and X. Shen, "Spatial Coordinated Medium Sharing: Optimal Access Control Management in Drive-Thru Internet," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2673–2686, Oct. 2015.

[33] K. Ota, M. Dong, S. Chang, and H. Zhu, "MMCD: Cooperative Downloading for Highway VANETs," *IEEE Trans. on Emerg. Topics Comput.*, vol. 3, no. 1, pp. 34–43, Mar. 2015.

[34] B. Zhang, X. Jia, K. Yang, and R. Xie, "Design of Analytical Model and Algorithm for Optimal Roadside AP Placement in VANETs," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, 2015.

[35] H. Omar, W. Zhuang, and L. Li, "Gateway placement and packet routing for multihop in-vehicle Internet access," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 3, pp. 335–351, Sept. 2015.

[36] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in Internet of vehicles: A review," *IEEE Trans. on Intell. Trans.n Sys.*, vol. 16, no. 5, pp. 2339–2352, 2015.

[37] N. Cheng, N. Lu, N. Zhang, X. Shen, and J. W. Mark, "Vehicular wifi offloading: Challenges and solutions," *Veh. Commun. (Elsevier)*, vol. 1, no. 1, pp. 13–21, 2014.

[38] G. El Mouna Zhioua, H. Labiod, N. Tabbane, and S. Tabbane, "VANET Inherent Capacity for Offloading Wireless Cellular Infrastructure: An Analytical Study," in *Proc. IEEE 6th NTMS*, 2014, pp. 1–5.

[39] G. el Mouna Zhioua, J. Zhang, H. Labiod, N. Tabbane, and S. Tabbane, “VOPP: A VANET offloading potential prediction model,” in *IEEE WCNC’14*, Apr. 2014, pp. 2408–2413.

[40] W. Su, S.-J. Lee, and M. Gerla, “Mobility prediction and routing in ad hoc wireless networks,” *Int. J. of Netw. Manag.*, vol. 11, no. 1, pp. 3–30, 2001.

[41] J. Tang, G. Xue, and W. Zhang, “Reliable ad hoc routing based on mobility prediction,” *J. of Combl Optim.*, vol. 11, no. 1, pp. 71–85, 2006.

[42] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, “A Stable Routing Protocol to Support ITS Services in VANET Networks,” *IEEE Trans. on Veh. Technol.*, vol. 56, no. 6, pp. 3337–3347, Nov. 2007.

[43] H. Menouar, M. Lenardi, and F. Filali, “Improving Proactive Routing in VANETs with the MOPR Movement Prediction Framework,” in *IEEE ITSC ’07*, Jun. 2007, pp. 1–6.

[44] N. Sofra and K. K. Leung, “Estimation of link quality and residual time in vehicular ad hoc networks,” in *IEEE WCNC’08*, Mar. 2008, pp. 2444–2449.

[45] N. Sofra, A. Gkelias, and K. Leung, “Route Construction for Long Lifetime in VANET-s,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3450–3461, Sept. 2011.

[46] M. Ayaida, L. Afilal, H. Fouchal, and H. E. Mehraz, “Improving the link lifetime in VANETs,” in *IEEE 36th LCN Conference*, Oct. 2011, pp. 905–912.

[47] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, “MURU: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks,” in *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services*, Jul. 2006, pp. 1–8.

[48] W. Viriyasitavat, F. Bai, and O. Tonguz, “Dynamics of network connectivity in urban vehicular networks,” *IEEE J. on Select. Areas in Commun.*, vol. 29, no. 3, pp. 515–533, Mar. 2011.

[49] X. Wang, C. Wang, G. Cui, and Q. Yang, “Practical link duration prediction model in vehicular ad hoc networks,” *Int. J. of Dist. Sensor Networks*, vol. 2015, 2015.

[50] H. Menouar, M. Lenardi, and F. Filali, “Movement Prediction-Based Routing (MOPR) Concept for Position-Based Routing in Vehicular Networks,” in *IEEE 66th VTC*, Sept. 2007, pp. 2101–2105.

[51] R. Tavakoli and M. Nabi, “TIGeR: A Traffic-Aware Intersection-Based Geographical Routing Protocol for Urban VANETs,” in *Proc. IEEE 77th VTC*, 2013, pp. 1–5.

[52] M. Boban, T. T. V. Vinhoza, M. Ferreira, J. Barros, and O. K. Tonguz, “Impact of Vehicles as Obstacles in Vehicular Ad Hoc Networks,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 15–28, Jan. 2011.

[53] H. Xia, “A simplified analytical model for predicting path loss in urban and suburban environments,” *IEEE Trans. Veh. Technol.*, vol. 46, no. 4, pp. 1040–1046, 1997.

[54] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, “LTE for Vehicular Networking : A Survey,” *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May 2013.

[55] N. Alsharif and X. Shen, “*i*CARII: Intersection-based connectivity aware routing in vehicular networks,” in *Proc. IEEE ICC*, 2014, pp. 2731–2735.

[56] F. K. Karnadi, Z. H. Mo, and K.-c. Lan, “Rapid generation of realistic mobility models for VANET,” in *Proc. IEEE WCNC*, 2007, pp. 2506–2511.

[57] H. Dia, “An object-oriented neural network approach to short-term traffic forecasting,” *European Journal of Operational Research*, vol. 131, no. 2, pp. 253–261, 2001.

[58] E.-M. Lee, J.-H. Kim, and W.-S. Yoon, “Traffic speed prediction under weekday, time, and neighboring links speed: back propagation neural network approach,” in *Advanced Intelligent Computing Theories and Applications with Aspects of Theoretical and Methodological Issues.* Springer, 2007, pp. 626–635.

[59] J. Park, D. Li, Y. L. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips, “Real time vehicle speed prediction using a neural network traffic model,” in *IEEE IJCNN’11.* IEEE, 2011, pp. 2991–2996.

[60] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy mobile IPv6,” 2008.

[61] S. Cspedes and X. Shen, “On Achieving Seamless IP Communications in Heterogeneous Vehicular Networks,” *IEEE Trans. on Intell. Trans. Sys.*, vol. 16, no. 6, pp. 3223–3237, Dec. 2015.

[62] H. Omar, W. Zhuang, A. Abdrabou, and L. Li, “Performance evaluation of VeMAC supporting safety applications in vehicular networks,” *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 69–83, 2013.

[63] N. Alsharif, A. Wasef, and X. Shen, “ESPR: Efficient Security Scheme for Position-Based Routing in Vehicular Ad Hoc Networks,” in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–5.