Previous Hop Routing: Exploiting opportunism in VANETs

by

AWOS KH. ALI

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

8th January 2018

Copyright 2018 AWOS KH. ALI

Abstract

Routing in highly dynamic wireless networks such as Vehicular Ad-hoc Networks (VANETs) is a challenging task due to frequent topology changes. Sustaining a transmission path between peers in such network environment is difficult. In this thesis, Previous Hop Routing *(PHR)* is poposed; an opportunistic forwarding protocol exploiting previous hop information and distance to destination to make the forwarding decision on a packet-by-packet basis. It is intended for use in highly dynamic network where the life time of a hop-by-hop path between source and destination nodes is short. Exploiting the broadcast nature of wireless communication avoids the need to copy packets, and enables redundant paths to be formed. To save network resources, especially under high network loads, *PHR* employs probabilistic forwarding. The forwarding probability is calculated based on the perceived network load as measured by the arrival rate at the network interface. We evaluate *PHR* in an urban VANET environment using *NS2* (for network traffic) and *SUMO* (for vehicular movement) simulators, with scenarios configured to reflect real-world conditions. The simulation scenarios are configured to use two velocity profiles i.e. Low and high velocity. The results show that the *PHR* networks able to achieve best performance as measured by Packet Delivery Ratio (PDR) and Drop Burst Length (DBL) compared to conventional routing protocols in high velocity scenarios.

Acknowledgements

This thesis represents a milestone in my academic life in more than four years of work at Loughborough University and specifically within the computer science department. Since my first day during Ph.D. on October 1st, 2013 I have felt at home at Loughborough. I have been given unique opportunities and taken advantage of them.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Iain W. Phillips, Director of Academic Staffing and Head of Computer Science department in Loughborough University, whose selfless time and care were sometimes all that kept me going. Iain has supported me not only by providing research guidance over the past four years, but also academically and emotionally through the rough road to finish this thesis. During the most difficult times when writing this thesis, he gave me the moral support and the freedom I needed to move on. Many thanks to him to make my Ph.D. experience productive and stimulating. The enthusiasm he has for his research was contagious and motivational for me, even during tough times. I am also thankful for the excellent example he has provided as a successful researcher.

I gratefully appreciate and acknowledge the funding and sponsorship of the Iraqi Ministry of Higher Education & Scientific Research (MOHESR) that made my Ph.D. work possible. With great appreciation I shall acknowledge all my colleagues and friends at Loughborough University, who constantly supported me throughout my Ph.D.

Last but not the least, I would like to thank my family for all their love and encouragement. For my parents Dr Khazal Ali & Salma who prayed for my success and raised me with a love of knowledge and supported me in all my pursuits. I am indebted to my brother Ahmed and my sisters Aseel & Areej who encouraged me and prayed for me during my the Ph.D. journey. And most of all for my loving, supportive, encouraging, and patient wife Israa whose faithful support during the all the stages of this Ph.D. is so appreciated.

I would like to thank my children, Jannat & Mohammed, who are the pride and joy of my life. I love you more than anything and I appreciate all your patience and support during daddy's Ph.D. studies. I dedicate this work to you all.

Thank you.
Awos Kh. Ali
8th January 2018

Contents

Abstract **2**

Acknowledgements **3**

1 Introduction **13**
1.1 Introduction . . . 13
1.2 Motivation . . . 14
1.3 Contributions . . . 16
1.4 Thesis Organisation . . . 18

2 VANET Applications **19**
2.1 Intelligent Transportation System (ITS) . . . 19
2.2 VANET Applications . . . 20
2.2.1 Potential applications and scenarios in the future for VANETs 21
2.2.2 Applications Classification . . . 21
2.3 VANET Challenges . . . 26
2.4 Summary . . . 28

3 VANET Routing **31**
3.1 Conventional Routing . . . 31
3.1.1 Ad hoc On Demand Distance Vector (AODV) . . . 32
3.1.2 Optimized Link State Routing protocol (OLSR) . . . 35
3.1.3 Greedy Perimeter Stateless Routing (GPSR) . . . 35
3.1.4 Dynamic Source Routing (DSR) . . . 37
3.2 Opportunistic Routing . . . 39
3.3 Opportunistic Routing Literature . . . 40
3.3.1 Taxonomy of Opportunistic Routing in Wireless Networks . 43
3.3.1.1 Opportunistic Routing for VANET based on Geographic position . . . 44
3.3.1.2 Link State Opportunistic Routing . . . 46
3.3.1.3 Probabilistic Opportunistic Routing . . . 48

3.3.1.4 Optimisation-Based Opportunistic Routing 49
3.3.1.5 Cross-Layer Opportunistic Routing 50
3.4 Opportunistic Routing versus Conventional Routing 52
3.5 Summary 53

4 Methodology **54**
4.1 Network Simulator NS2 54
4.2 Simulation of Urban MObility (SUMO) 55
4.3 Simulation Experiments 55
4.3.1 Performance Criteria 57
4.3.2 Simulation Configuration 57
4.3.2.1 The network traffic model 58
4.3.2.2 Mobility Model 58
4.3.2.3 802.11p standard 58
4.3.2.4 Network Traffic Model 59
4.3.2.5 Network topology mode 59
4.3.3 Propagation model 60
4.4 Summary 60

5 Evaluation Conventional Protocols **61**
5.1 Simulation Setup 61
5.1.1 Simulation description 63
5.2 Results and discussion 64
5.3 Summary 67

6 Previous Hop Routing PHR **74**
6.1 Motivation 74
6.2 PHR Design 75
6.2.1 Constraining Forwarding 76
6.2.2 Data and packet structure 76
6.2.3 Probabilistic forwarding 77
6.2.4 PHR Notation and algorithms 78
6.3 PHR Walkthrough 79
6.3.1 Simple scenario 79
6.3.2 Larger Scenarios 83
6.3.2.1 Grid Map Scenario 83
6.3.2.2 Dynamic Network Scenario 84
6.4 PHR versus other Opportunistic Routing protocols 88
6.5 Summary 88

7 PHR Implementation **90**
7.1 Sending Packet at Source Node . . . 90
7.2 Receiving Data Packet at Intermediate Node . . . 90
7.2.1 Unknown Forwarding . . . 91
7.2.2 Known Forwarding . . . 91
7.3 Receiving Packet at Destination Node . . . 91
7.4 Functions for Data Structure Management . . . 91
7.5 Analysis of PHR Trace Format . . . 92
7.5.1 Packet Drops Types . . . 93
7.5.2 Calculating PDR . . . 94
7.5.3 Calculating DBL . . . 94
7.5.4 Calculating C2C latency . . . 94
7.6 Implement different velocity scenarios . . . 95
7.7 Network Traffic Implementation . . . 95
7.8 Summary . . . 95

8 Evaluation and Discussion **96**
8.1 Simulation Experiments . . . 96
8.2 Probabilistic Forwarding Rate . . . 97
8.3 Results and Discussion . . . 97

9 Conclusions **110**
9.1 Summary . . . 110
9.2 Thesis Contributions . . . 111
9.3 PHR Evaluation . . . 113
9.4 Future Work . . . 113
9.4.1 More RSU involvement . . . 113
9.4.2 Various Network Traffic . . . 114
9.4.3 Additional Scenario . . . 114
9.4.4 Memory Usage . . . 114
9.4.5 Security Implications . . . 114
9.4.6 Jitter Delays . . . 115
9.4.7 Evaluate PHR against Opportunistic Protocols . . . 115

References **116**

A Integrate PHR source code into NS2 **124**

B Mobility Models Design **129**
B.1 London congestion zone and Leicester city centre . . . 129

B.2 Manhattan-style Model 130

C PHR Implementation Code **133**

C.1 PHR main actions code 133

C.2 Functions for Data Structure Management 133

C.3 Results calculation code 139

List of Figures

2.1 Geocast communication pattern[57]. 24
2.2 Beaconing communication pattern [57]. 24
2.3 Uni-cast communication pattern [57]. 24
2.4 Applications classification based on Network and Transport layers requirements. 30

3.1 Conventional routing classes 33
3.2 RREQ and RREP messages [1]. 34
3.3 RERR message. 34
3.4 MPRs Concept [27]. 36
3.5 Greedy forwarding example [33]. 36
3.6 Perimeter mode example [33]. 37
3.7 Route Discovery example: Node S is the sender, and node D is the destination. 38
3.8 Route Maintenance example: Node S is the sender, and node D is the destination. 39
3.9 Taxonomy of Opportunistic Routing in Wireless Networks 44

4.1 Basic architecture of NS2 [28]. 55
4.2 Graphical user interface of SUMO [6]. 56
4.3 Short and long DBL. 58

5.1 Manhattan mobility model map with RSUs in SUMO. 62
5.2 Part of the London congestion zone with RSUs in SUMO. 63
5.3 Part of the Leicester city centre with RSUs in SUMO. 64
5.4 Short Drop Burst of AODV, OLSR and GPSR with various number of connections in Man map (A zoomed portion). 65
5.5 Short Drop Burst of AODV, OLSR and GPSR with various number of connections in Czone map (A zoomed portion). 66
5.6 Short Drop Burst of AODV, OLSR and GPSR with various number of connections in Leicester city centre map (A zoomed portion). . . 67

5.7 Long Drop Burst of AODV, OLSR and GPSR with various number of connections in Man map (A zoomed portion). 69
5.8 Long Drop Burst of AODV, OLSR and GPSR with various number of connections in Czone map (A zoomed portion). 70
5.9 Long Drop Burst of AODV, OLSR and GPSR with various number of connections in Leicester city centre map (A zoomed portion). . . 71
5.10 Packet Delivery Ratio (PDR) of the selected protocols on the all maps. 71
5.11 CDF of delay/s for the selected protocols under various network loads in Man map scenario. 72
5.12 CDF of delay/s for the selected protocols under various network loads in Czone map scenario. 72
5.13 CDF of delay/s for the selected protocols under various network loads in Leicester city map scenario. 73

6.1 PHR packet structure 79
6.2 Scenario 1: D broadcast towards I 80
6.3 Scenario 2: I constrains flooding towards D 81
6.4 Scenario 1: S and D locations on the grid topology. 85
6.5 Scenario 1: S starts transmitting first packet towards D. 85
6.6 Scenario 1: D starts transmitting a packet towards S. 86
6.7 Scenario 2: starts transmitting first packet from S towards D. . . 87
6.8 Scenario 2: D sent to S using PHR forwarding strategy. 87

8.1 PDR against forwarding probability in case destination is known. . 98
8.2 PDR against forwarding probability in case destination is unknown. 99
8.3 Short Drop Burst of AODV, OLSR, GPSR and PHR with various number of connections (A zoomed portion). 101
8.4 Long Drop Burst of AODV, OLSR, GPSR and PHR with various number of connections (A zoomed portion). 102
8.5 CDF of delay for the PHR and the selected protocols under low and high network loads on Manhattan mobility model. 103
8.6 CDF of delay for the PHR and the selected protocols under low and high network loads on part of Leicester city centre map. 104
8.7 CDF of delay for the PHR and the selected protocols under low and high network loads on part of London congestion zone map. 105
8.8 Packet Delivery Ratio (PDR) of PHR and the selected protocols on the Man map with low and high mobility profiles. 107
8.9 Packet Delivery Ratio (PDR) of PHR and the selected protocols on the part of Czone map with low and high mobility profiles. 108

8.10 Packet Delivery Ratio (PDR) of PHR and the selected protocols on the part of Leicester city centre map with low and high mobility profiles. 109

List of Tables

2.1 VANET application characteristic [25] 22
2.2 Application classification based on network layer characteristics [31] [5] [25]. 23
2.3 Classifications of VANET applications based on Packet generation requirements [69] [31] [25]. 25

4.1 Some examples of VANET applications requirements[25]. [SC=Saftey Critical, CRS=Cooperative Road Safety, TM=Traffic Management, CM=Commercial, CO=Connection-oriented,CL=Connection-less,LW=Light-weight,HW=Heavy-weight(IP)], V2X=V2V or V2I 59

5.1 Drop ratio on both MAC and Network layers 68

7.1 Explanation of PHR trace file format. [RTR=routing layer, AGT=application layer, IFQ=Interface queue, MAC=mac layer CBR=constant bit rate pkt, s =SEND, r=RECEIVED, d=DROPPED, f=FORWARD] 93

Chapter 1

Introduction

1.1 Introduction

In the last few years, Vehicular Ad-hoc Networks (VANETs) have become an interesting research area because of the increasing demand of technology that makes the roads smarter. Vehicular Ad-hoc Networks (VANETs) represent a new class of Mobile ad hoc networks (MANETs) and one of the main topics in the communication system area, which grabs the attention of the research community. VANETs and MANETs share many of their characteristics, but there are key differences. Vehicles in the Intelligent Transportation System (ITS) act as routers, they send, receive and forward information on behalf of other cars to cut down the congestion on roads and improve safety. On the roads, vehicles communicating with each other and transmitting information such as direction and speed, and send alerts to each other, for avoiding an accident seems imminent. Some countries activated Vehicle-to-Vehicle (V2V) communication, such as the U.S. Department of Transportation. Vehicles need to be equipped with hardware to establish wireless connections with other vehicles or with Road Side Units (RSUs) which are fixed on the roadside and connected with backbone network to manage and provide internet connection to the network. Network connection in intelligent transportation system includes two types of configurations, which are Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I) and hybrid and over short distances [73].The characteristics of VANETs are unique compared with other mobile ad hoc networks. Network topology in VANETs suffer from frequent changes. Some of these changes are predictable as vehicles follow predefined paths (the road network) and velocity is limited by the local regulations and the capability of the vehicles themselves. A further challenge is the signal attenuation suffered by due to buildings and other obstacles. The density of vehicles in a an area can also vary greatly, e.g., consider an built up urban city centre compared with a countryside lane. As a

result, the communication links exist between vehicles is shortly lived and suffer from frequent route breakage, this could provide unreliable services for VANET applications. Accordingly, many challenges need to be addressed and resolved. One of the main challenges that should be considered in VANETs is the routing protocol. The research community proposed many techniques trying to resolve the routing issue and compute the best path in an attempt to deliver packets to their desired destination. These proposed techniques are varied in the process of finding the most appropriate path to a given destination. Some protocols use geographical location to direct the packets to their destination, while the others store all the pre-calculated routes to all the nodes in advance in several routing tables and keep these routing tables up to date. Other techniques, is to initiate a route discovery process when it needed to allow nodes to communicate with each other and maintains routes in use. Designing a routing protocol that provides stability and reliability in both VANET modes (V2V & V2I) is a key factor to deliver a decent service within a VANET environment, in other words, routing protocol should helps to deliver a high percentage of the sent messages to their destinations within a short latency. Typically MANET routing protocols are employed in VANETs to establish paths across the network from source to destination. The highly dynamic nature of the VANET often means that the paths are only stable for short periods of time, frequently breaking and causing the need for the routing protocol to intervene to find the new path. Zhang [79] states the maximum connection time of two vehicles and the speed gap 10 m/h is 107 s, if both vehicles are moving in the same direction. If they are moving in the opposite direction is maximum connection time is less than 10 s. However, in MANETs the node velocity is less than nodes in VANET, the connection could last for 480 s. Frequent route breakage is a key factor that should be considered in the designing of routing protocol that aim to satisfy VANET applications. Another key aspect needing to be taken into account during designing of the routing protocol to suit VANET environment is the density variation. Vehicle density could vary, depending on road conditions and time. Traffic congestion and accidents could occur on the roads, especially at peak times on weekdays. This could make the level of topology density varied. The vehicle density has an impact on the network connectivity of VANET, hence may influence on the performance of communication schemes dedicated for VANET.

1.2 Motivation

In the past, to avoid car accidents, drivers were using their hands, horn and observation of each others to handle and manage their behaviour. Increase the number

of cars on the roads, makes avoiding accidents increasingly difficult, especially in highly congested roads. The increased use of networked services within vehicles brings the need to use this technology on the roads. Vehicular Ad-hoc Networks (VANETs) employ 802.11 allowing vehicles to communicate. VANETs are different from other wireless networks by their own characteristics. The nodes in VANETs are limited to road layout while moving. This makes the predictability of the future position of a vehicle is easier and more accurate. This helps to identify which vehicle can afford more computing, communication and sensing capabilities as well as providing continuous transmission power. However, due to high mobility and rapid changes in the topology, the communication links exist between vehicles is shortly lived. In order to provide reliable services in VANETs, many challenges need to be addressed and solved. Stable and reliable routing in VANETs is one of the major issues. Deep research is needed in order to make VANETs more applicable. Vehicles have dynamic behaviour, high speed and mobility that make routing more challenging issue.

Routing history in VANETs starts with conventional mobile ad hoc network routing protocols such as AODV (Ad hoc on Demand Distance Vector Routing), Optimized Link State Routing protocol (OLSR) and Greedy Perimeter Stateless Routing (GPSR), they are considered efficient routing protocols for multi hop wireless ad hoc networks [27][37]. The differences in communication environments and mobility models between VANETs and MANETs, make current MANET routing protocols inefficient and unreliable to use them in VANETs. Inefficient means, does not fully satisfy VANETs application requirements. MANET routing protocols are either topology based or position aware. Topology based protocols are classified into proactive and reactive protocols. Proactive routing algorithms maintain routes by using tables. The weak point in proactive technique is routing information need to be exchanged frequently to keep routes to all nodes within the network are valid. Due to rapid changes in VANET network topology, the routes in the routing tables become invalid quickly. On the other hand, reactive routing algorithms are establishing a route only when it is needed. As the reactive approach is on-demand based routing, a route must be discovered before the first data packet is sent out towards the desired destination, consequently, increases the delivery time. Many studies have been conducted to assess network performance with MANETs routing protocols in a VANET environment, for example [3]. The results showed, neither of these two approaches are applicable in VANETs. The main issue with the reactive approach is that even there is an existing route to destination, that route may also be very short lived because of the mobile nature of VANETs. Another class of MANET routing approach is the position aware protocols. Position based routing, establish routes based on geographic position

of a destination, this could leads to frequent path breakage due to the mobility nature of VANETs.

One of the most basic requirements for traditional MANET routing protocols is that there must exist a fully connected path between the sender and the receiver for the duration of a communication session to make the communication possible. However, in case of high dynamic network topologies such as VANET, this is difficult to achieve.

A routing algorithm needed to sustain routes from source to destination despite the fact vehicles could travel under varying velocity and within a fading environment that buildings have impact on radio signals. This routing protocol should cope with different network loads and satisfying various types of VANET applications requirements. All these requirements and challenges should be considered during designing this routing protocol to suits various VANETs environment. Conventional MANETs routing protocols have satisfied applications? requirements to a certain extent. Due to rapid network topology changes, forwarding protocol needs to be resilient, ensuring high delivery rate.

1.3 Contributions

In this thesis, an investigation of routing protocols in VANET environments has been carried out. Number of conventional MANET routing protocols are tested in various VANET urban environments in different time of a day using *NS2* simulator (for network traffic) and Simulation of Urban MObility *SUMO* framework (for vehicles movement). The simulation scenarios are configured to reflect real-world urban environment conditions, such as the impact of building on the radio signal. Additionally, various level of network load is considered to emulate various density levels of a weekday.

The list of this thesis contributions are as follow:

- Drop-Burst Length (DBL): This is a novel measurement of the probability of drop a consecutive number of packets in each connection. It is the distribution of the lengths of the packet group drops. This provides with a richer indication as to the effects of performance the QoS of real-time traffic, which traditionally cope with single packet errors, but less so to burst drops. DBL is proposed to measures the packet drop impact on the network performance.

- Previous-Hop Routing (*PHR*): This is a probabilistic forwarding protocol using previous hop information to make the forwarding decision. *PHR* is exploiting the naive version of the epidemic protocol with control flooding

strategy towards the destination (strict pruning). The epidemic protocol works by transmitting each packet to every node in the topology. As a packet is passed from node to node, it is eventually delivered to the target node. One of the advantages of an epidemic protocol is that by trying every path towards the destination, it is guaranteed to try the best path. One of the disadvantages of an epidemic protocol is the extensive use of resources with every node needing to carry every packet and the associated transmission costs. *PHR* has the following features:

1. *PHR* provides a high level of robustness by selecting a limited number of forwarders those have valid topological information about the target node (nodes located in the path towards the destination).
2. With *PHR*, nodes transmit redundant copies of each message toward the destination node to ensure one of the copies will be delivered.
3. *PHR* selects the best forwarder instantly based on the distance from the destination.
4. *PHR* do not use any type on control message mechanism compared to the conventional routing protocols and some opportunistic protocols that need to update their routing tables or discover their neighbours node.
5. Due to the type on nodes' movements shape in VANET, the density of vehicles could make the roads congested to varying levels. The congested road may have an impact on the performance of the communication systems dedicated for VANET. This could lead to add an extra load to the network. *PHR* is adaptable to various network loads by exploiting probabilistic forwarding scheme.
6. *PHR* is adapted to the level of network topology changes and suits both VANET modes (V2V and V2I).

- Evaluating network performance with *PHR* against MANETs routing protocols: This thesis provides a simulation-based evaluation of *PHR* against number of selected conventional MANETs routing protocols. The simulation is designed and configured to reflect a real-world urban environment. The simulation is carried out using *NS2* (for network traffic) and *SUMO* (for vehicular movement) simulators. The network performance is evaluated using traditional metrics of packet delivery ratio (PDR) and latency (delay) alongside the proposed Drop-Burst Length (DBL).

The following papers are published based on this research.

- Awos Kh Ali, Iain Phillips, and Huanjia Yang. Evaluating VANET Routing in Urban Environments. 39th Int. Conf. Telecommun. Signal Process., pages 60-63, 2016.

- Awos Kh Ali, Iain Phillips, and Huanjia Yang. Drop-Burst Length Evaluation of Urban VANETs. Int. J. Adv. Telecommun. Electrotech. Signals Syst., 6(2):1-6, 2017.

1.4 Thesis Organisation

The rest of the thesis is organised as follows: Chapter 2 covers the description of Intelligent Transportation System (ITS) and VANET applications classifications and requirements. Routing literature, the three selected routing protocols (AODV, OLSR and GPSR) in addition to Dynamic Source Routing scheme and opportunistic routing are described in Chapter 3. Chapter 4 introduces the novel measurement of Drop-Burst Length(DBL) and the simulation model that used to evaluate the selected MANET routing protocols in VANET urban environment including. The simulation tools, network design(configurations) and VANET simulation scenario also described in this chapter. Chapter 5 provides an analysis of the simulation results of the selected routing protocols against the End-to-End Delay, Drop-Burst Length and Packet Delivery Ratio. Additionally, the key findings that we designed *PHR* protocol based on. Chapter 6 introduces the novel Previous-Hop Routing (*PHR*) protocol and its related key features. Chapter 7 presents the explanation of *PHR* implementation in the network simulator *NS2*. The Evaluation and the results discussion of the simulation scenarios with the *PHR* are outlined in chapter 8.

Finally, chapter 9 conclude the thesis, and the potential future research directions are suggested.

Chapter 2

VANET Applications

VANET, is a technology that uses moves vehicles as nodes in a network to create a mobile network. VANET turns every participating vehicle into a wireless router or node, allowing vehicles approximately 100 to 300 metres of each other to connect and, in turn, create a network with a wide range. As vehicles fall out of the signal range and drop out of the network, other vehicles can join in.

The ultimate goal of VANETs is to provide reliable network services to its applications. These applications could have highly diverse requirements, some of the applications could be delay tolerant while others demand $2-way$ communication. This chapter, discusses some of the literature related to Intelligent Transportation System (ITS), VANET applications and a brief description of VANET challenges,.

2.1 Intelligent Transportation System (ITS)

In the last decade, the increased use of wireless technology brings the ability to use this technology in vehicles on roads. VANET can support a range of applications including safety and non-safety applications such as enhancing navigation systems, vehicle safety, traffic management and finding the closest service centre [73]. Vehicles in the Intelligent Transportation System (ITS) act as routers; they send, receive and forward information between themselves to avoid congestion on roads and to provide safety. To achieve this, vehicles need to be equipped with hardware to establish wireless connections with other vehicles or with Road Side Units (RSUs) which are fixed on the road side and connected with backbone network to manage and provide Internet connection to the network. Network connection in intelligent transportation system includes two types of configurations, which are Vehicle- to-Vehicle (V2V) and Vehicle-to- Infrastructure (V2I) [73]. Public vehicles such as police cars could play a specific role and could be considered as a mobile infrastructure [32]. V2V communication mode, use multi-

hop to transfer information within the vicinity. The limitation of V2V connection mode, is a large amount of messages are generated and disseminated within the network and consequently, an increase in delivery time of these messages and a decrease in the delivery rate [73]. On the other hand, Vehicle-to-Infrastructure (V2I) connection mode is considers one hop broadcast. In V2I, access point sends information to vehicles within the vicinity, facilitate connections between nodes and provide high-bandwidth connectivity to vehicles, even if the network examines heavy traffic conditions [73].

2.2 VANET Applications

Various types of applications could be provided by VANETs. Some of these applications may perform successfully in urban areas, such as file sharing applications as they require high density topologies, while other applications may require a more open environment to avoid network congestion problem as they generate high network traffic rate. Moreover, applications could be categorised to many categories such as, road safety applications, traffic management applications and commercial applications. Some examples are presented as follows.

- Road Safety applications: the increasing number of cars on roads leads to an increasing number of accidents. Road safety is a priority in many countries. In order to reduce the number of accidents and provide safety on roads, several VANET applications have been developed to assist drivers in preventing collisions and be aware of road works. Another assistance could be provided by supporting drivers to avoid obstacles and notify weather information. Furthermore, Meraihi et al.[49] point out some services that could be provided by VANETs, including alert services, collaborative driving and parking management. In alert services, the driver is informed about the status of the road ahead and changes that may occur in its condition, for example, if he moves toward an accidents scene or traffic congestion. The alert services application may give the driver an alternative route to his destination, and it will send messages to all vehicles within the vicinity to notify drivers of the potential hazard. These messages could be transmitted by a vehicle that detects a traffic problem or an official vehicle. Another service is collaborative driving. This involves messages exchanging between vehicles in order to improve road safety, reducing the number of accidents and to realise the surrounding. In the parking management service, the system collects information about space availability and guides the car to find a free space [49]. Safety applications could be further categorised into

safety-critical and cooperative road safety.

- User Applications: Many comfort and entertainment services may be submitted to drivers or passengers by VANETs, for example, Internet access, messaging and network games [49].

- Car Torrent: the concept of car torrent application is to share data between vehicles (peer-to-peer). Vehicle can download parts of a file in a parallel way from different hosts and combine these parts together. These type of applications need high density topology, and high bandwidth [47].

2.2.1 Potential applications and scenarios in the future for VANETs

With the rapid developments in the network research area, especially in mobile networking technology, several services could be provided in the near future by VANETs to make roads safer. These potential services are as follows.

2.2.2 Applications Classification

In order to understand how to fulfil VANET application requirements, an investigation on application behaviour from different angles need to be carried out. This section provides deep analysis of VANET application requirements from different network stack layers' point of view and categorisation of these applications depending on their characteristic, as demonstrated in tables 2.1, 2.2 and 2.3. Table 2.1 illustrates VANET applications classification based on their general characteristics as follows.

- Connection characteristic: It represents, message exchanging nature.

- Connection range: It represents, the maximum wireless range between vehicles to exchange information successfully.

- Position: It represents, the applications need for position information.

- Messaging type: It represents, the trigger type of message creation.

- Priority: It represents, the application's priority.

Furthermore, these applications could be classified based on network layer perspective or on packet generation process.

Table 2.2 shows classification of VANET applications based on network layer characteristics. The difference between these applications is the requirement of

Table 2.1: VANET application characteristic [25]

Applications	Connection characteristic	Connection Range	Network Type	Position	Messaging Type	Network protocol	Priority	Serves type	No. of hop
malfunction in the car	Reliable	<100 m	V2V	Accurate positioning	Event-triggered,time limited broadcast	Non-IP	High	Safety	Multi
Car Status	Reliable	<300 m	V2I	Not required	Periodic broadcast	Non-IP or IP	Normal	Non-Safety	Single
Stolen car	Fast and reliable	<300 m	V2I	Accurate positioning	Event-triggered,time limited broadcast	Non-IP	High	Safety	Single
control house or office	Reliable	<1000 m	V2I	Not required	Event-triggered,time limited broadcast	IP	Low	Non-Safety	Single
Change in direction	Fast and reliable	<100 m	V2V	Accurate positioning	Event-triggered,time limited broadcast	Non-IP	High	Safety	Multi
Type of passengers	Reliable	<100 m	V2V	Accurate positioning	Event-triggered,time limited broadcast	Non-IP or IP	Normal	Safety	Multi
Entered school zone	Reliable	<300 m	I2V	Not required	Event-triggered,time limited broadcast	Non-IP or IP	Normal	Safety	Single
Driver health condition	Fast and reliable	< 300 m	V2V, V2I	Accurate positioning	Event-triggered, time limited broadcast	Non-IP	High	Safety	Multi
Weather condition	Reliable	<1000 m	V2V	Not required	Event-triggered,time limited broadcast	Non-IP or IP	Low	Safety	Multi
Emergency vehicle warning	Fast and reliable	< 100 m	V2V	Accurate positioning	Event-triggered,time limited broadcast	Non-IP	High	Safety	Multi
Emission warning	Reliable	<500 m	V2I	Not required	Periodic broadcast	Non-IP	Low	Non-Safety	Single
Auto parking	Fast and reliable	< 100 m	V2V	Accurate positioning	Event-triggered,time limited broadcast	Non-IP	High	Safety	Multi
Animals on road	Fast and reliable	<100 m	V2V	Accurate positioning	Event-triggered,time limited broadcast	Non-IP	High	Safety	Multi
Pre-defined trip	Reliable	<1000 m	I2V	Not required	Event-triggered,time limited broadcast	IP	Low	Non-Safety	Single

the packet generation process. Some applications need to use heavy weight packet format and others use light format.

Table 2.2: Application classification based on network layer characteristics [31] [5] [25].

Applications	No. of hops	protocol	Applcation Trigger	Communcati Pattern	Transport protocol	Packet Format
malfunction in the car	Single	Geocasat	Event-Driven	One-to-Zone	Connection-less	Light-weight
Car Status	Single	Uni-Cast	Periodic	One-to-One	Connection- Oriented	heavy-weight IP
Stolen car	Single	Uni-Cast	Event-Driven	One-to-One	Connection-less	Light-weight
control house or office	Single	Uni-Cast	User-Initiated	One-to-One	Connection- Oriented	heavy-weight IP
Change in direction	Multi	Geocast	Event-Driven	One-to- Zone	Connection-less	Light-weight
Type of passengers	Multi	Geocast	Periodic	One-to- Zone	Connection-less	Light-weight
Entered school zone	Single	Uni-Cast	Periodic	One-to- Zone	Connection-less	Light-weight
Driver health condition	Multi	Geocast	Event-Driven	One-to- Zone	Connection-less	Light-weight
Weather condition	Multi	Broadcast	Periodic	One-to-Many	Connection- Oriented	heavy-weight IP
Emergency vehicle warning	Multi	Broadcast	Periodic	One-to-Many	Connection-less	Light-weight
Emission warning	Single	Uni-Cast	Periodic	One-to-One	Connection- Oriented	heavy-weight IP
Auto parking	Multi	Geocast	Event-Driven	One-to- Zone	Connection-less	Light-weight
Animals on road	Multi	Geocast	Event-Driven	One-to- Zone	Connection-less	Light-weight
Pre-defined trip	Single	Uni-Cast	User-Initiated	One-to-One	Connection- Oriented	heavy-weight IP

For a better understanding of the network behaviour in the VANET environment, VANET applications have been classified into three distinct categories, safety applications group, traffic management group and commercial and entertainment applications group. In the safety applications group, the vehicle initiates broadcasting warning messages to all vehicles within a certain range when an event happened on the road, such as the detection of road hazards, driver health condition, vehicle collisions or a sudden break. If none of these events is happening, safety messages will not be exchanged. However, some safety applications have different characteristics, for example, cooperative collision warning and school zone notification. These applications rely on periodical broadcast messages in order to keep monitoring surrounding vehicles. On the other hand, traffic management and commercial applications use a triggered messages generation process initiated by the vehicle owner, rather than by a safety event or by the vehicle itself [5].

Moreover, every application in the VANET environment has its own needs and characteristics, for example, some applications use a one to many recipients pattern. These applications are more likely use broadcast strategy in the network layer, while the Unicast routing protocol is suitable for applications using a one to one communication pattern. Similarly, single hop packet dissemination (beaconing) is appropriate for limited region applications, when data packets are broadcast to all neighbours within the source range. In contrast, multi hop routing protocols are adequate for applications targeted to a medium or large region. Some VANET applications require high bandwidth and low latency which is difficult to achieve in high mobile network. In addition, the type of application determines the packet format. Usually, safety and traffic management applications use light-weight short

packet format in order to improve network performance and reduce latency, while commercial applications prefer to use heavy-weight IP packet format to suit existing Internet services. Most of the safety applications rely on information dissemination into a specific geographic region, this dissemination technique named Geo-cast or multi-hop Geo-cast communication strategy [32] [57], these applications use Geo-cast routing protocol because of the nature of safety applications and the way of warning other vehicles (One-to-Many) as shown in Fig 2.1 [57]. Whereas, cooperative collision warning and school zone applications use single hop broadcast scheme (Beaconing) to neighbours directly, Fig 2.2 illustrates beaconing communication pattern. Traffic management and commercial applications either use broadcast or Geo-cast to deliver messages in a specific region, for example, a traffic congestion notification, or use Unicast routing protocol to deliver data packets for a given destination, such as downloading a navigation map from other vehicles, as shown in Fig 2.3 [57].

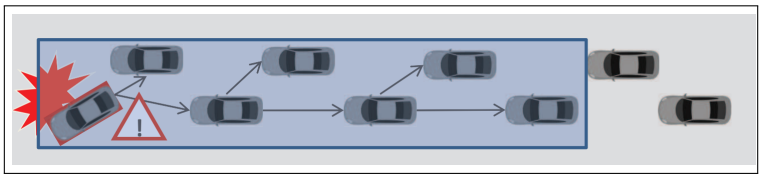

Figure 2.1: Geocast communication pattern[57].

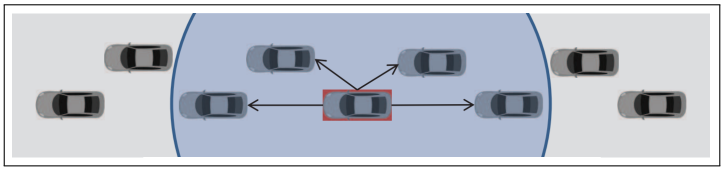

Figure 2.2: Beaconing communication pattern [57].

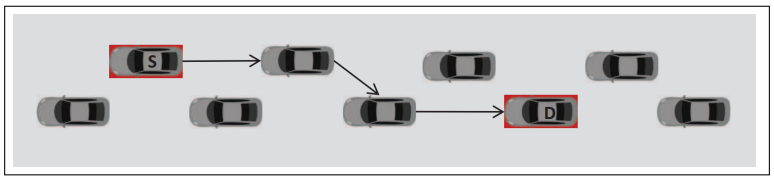

Figure 2.3: Uni-cast communication pattern [57].

It is observed that, all safety and traffic management applications, follow connection-less protocol in the transport stack such as Wireless Access in Vehicular Environments (WAVE) Short Message Protocol (WSMP) or User Datagram Protocol (UDP), while commercial applications often use connection oriented mode such as Transmission Control Protocol (TCP). Some safety applications use the traditional kind of beaconing message, SAE J2735 [58] describe 50 types of applications messages. One of these important messages is Basic Safety Message

(BSM) which represents the heart beat message, it announces vehicle information periodically to its neighbours. BSM contains of two parts of data element. The first part contains the main data elements, including vehicle position, speed, heading, acceleration, steering wheel angle, and vehicle size, which are transmitted about 10 times per second. The second part of BSM contains a variable set of data elements, which are selected based on an event triggers, such as Antilock Brake System (ABS) status, vehicle type, weather information, lights status and GPS status. The second part of BSM data elements are added to first part and broadcast as part of the BSM message, but the second part is transmitted less frequently. BSM is used by V2V applications because it delivers data packets in low latency, these applications might be safety applications or non-safety applications [25].

On the other hand, VANET applications could be classified based on the packet generation process as shown in table 2.3. These applications are classified depending on the packet requirements, as illustrated in the Vehicle Safety Communications Project Task 3 Final Report by the U.S. department of transportation [25].

Table 2.3: Classifications of VANET applications based on Packet generation requirements [69] [31] [25].

Applications	Packet/sec	Packet size(byte)	Packet transmition duration	Range of communication(m)	Allowable latency
malfunction in the car	1	Up to 1500	Limited duration	400	5 sec
Car Status	1	Up to 1500	Periodic	400	5 sec
Stolen car	1	100-200	Limited duration	250	1 sec
control house or office	1	Up to 1500	Limited duration	400	N/A
Change in direction	10	100-200	Limited duration	300	100 ms
Type of passengers	1	500	Periodic	100	0.5 sec
Entered school zone	1	100-200	Periodic	200	1 sec
Driver health condition	6 - 8	28	Limited duration	1000	5 sec
Weather condition	2	Up to 1500	Periodic	400	0.5 sec
Emergency vehicle warning	1	100-200	Periodic	1000	1 sec
Emission warning	1	Up to 1500	Periodic	400	N/A
Auto parking	10	100 -200	Limited duration	150	100 ms
Animals on road	1	100-200	Limited duration	200	1 sec
Pre-defined trip	1	Up to 1500	Limited duration	250	N/A

Each application has its own packet process requirements, for example, create data packets, send them and receive packets. Table 2.3 categorise VANET applications depend on these requirements.

- Packet/sec: It describes packets sending rate.
- Packet size: It describes the size of each packet.

- Packet transsmition duration: It describes the time of packet transmission process.
- Range of communication: It describes the required communication distance between the sender and receiver to support a specific application.
- Allowable latency: the maximum allowable duration of time between sending information and receiving it.

The packet format is determined by the type of VANET application. Regularly, safety and traffic management applications use small packet payload in order to improve network efficiency and to reduce the number of dropped packets [25]. Conversely, commercial applications prefer to use traditional big size IP packet format to fulfil the requirements of existing Internet services. For the sake of providing more in-depth analysis of VANET applications requirements, the need to classify these applications, based on network layer and transport layer requirements, should take place to provide reliable services to the application layer. Fig. 2.4, illustrates the required performance level of the network and transport layers. In the active road safety class applications, the application layer should be provided with maximum packet delivery rate, short drop burst and low latency. The requirements of the cooperative traffic efficiency class (Traffic management) are more tolerant than safety applications in terms of latency. The transport layer is therefore connection-less type such as User Datagram Protocol (UDP) format. Other applications could require high transport layer performance, however, more tolerant in network layer performance because the transport layer uses a retransmission mechanism in case of packet not received such as TCP. These kinds of applications are not time-sensitive applications, such as instant messaging and media downloads.

This thesis is targeting safety applications class in term of requirements and testing the network performance that carrying safety messages.

2.3 VANET Challenges

Since VANET is a subset of MANET with different characteristic and mobility pattern, researchers have faced many challenges in designing convenient routing protocol, that have the ability to handle these characteristics. Some challenges are described briefly as follow:

- Data Dissemination: Due to the high mobility pattern in VANET topology, it could be impossible to sustain unicast/geocast connection. Data exchange strategies need to be adaptive to this frequent changes. Broadcast seems to

be the most effective solution for this problem [16]. However, the simple broadcasting model generates a huge number of messages, and these messages are delivered to nodes within the network at a regular interval which lead to heavy traffic in the network and rise in delivery time [73]. Nevertheless, a wide range of safety and traffic management applications require geocast transmission mode rather than broadcast.

- Routing protocols: VANET routing protocols have been investigated and studied widely in the last few years. Most of MANETs' routing protocols are tested and evaluated in order to use them in VANET environment. Nodes in VANET have different characteristics compare to nodes in MANETs. In VANETs, nodes have good capability and power consumption is not a big problem since the network's hardware is fitted in a vehicle. Furthermore, nodes in VANET move in a predictive way and follow roads. Some road infrastructures such as traffic lights and speed limit signs, could have effects on vehicles' behaviour. As a result, this change in nodes' speed could affect on network topology, which makes designing an efficient routing protocol a challenging task [16]. Moreover, a wide range of VANET applications need to be supported to perform better under different circumstances. To satisfy applications' requirements, an appropriate routing protocol need to be designed that has the ability to support these applications.

- Low Bandwidth: In VANETs, low bandwidth is considered as a serious problem due to some dependencies of safety and non-safety applications on fast reactions from roadside infrastructure or other vehicles, especially in urgent cases such as accidents and traffic jams. During peak hours in a day if a large number of vehicles in a small area (high vicinity) need to exchange information at the same time this could result a network congestion, for this reason, routing protocol should be designed to take all these conditions into consideration [61].

- High Mobility Pattern: Since nodes in VANET are vehicles, they move in a very fast manner which is why designing an effective routing protocol to handle such types of mobility patterns is a challenging task. In addition, a mobility model could be varied based on the environment, such as on a highway and within a city or town. In the highway environment, the speed of the nodes (vehicles) might be very high compare to the nodes' speed inside a city or a town [32].

- Different Communication Environment: In alignment with the mobility model, VANET has two communication models. The first model is in a highway

road, which is considered a very simple model and does not have serious barriers to block communication signal. However it could examine a link failure between nodes or between nodes and access points due to high velocity. The second model is when nodes are within city conditions. The roads in the city could be separated by buildings, trees and other obstacles. Therefore to share information between nodes, a routing protocol needs to use an indirect link [32].

Routing is considered as a main key to deliver data packets successfully and support applications especially with such a high applications requirements and frequent topology changing rate. Therefore, a new routing technique required to handle all these challenges. Moreover, to design such a routing protocol, we need a metric that measures the impact of the network performance and the high mobility rate on VANET applications.

The following sections describe routing in VANET.

2.4 Summary

This chapter has identified VANET applications' requirements from different perspectives based on their characteristics. These applications are categorised into three main classes, i.e., safety applications class, traffic management applications class and entertainment applications class. VANET application classes, require different needs to perform efficiently in an environment that suffers frequent topology changes. Overall, VANET applications are classified based on their general characteristics such as, data exchanging nature, connection range, location, messaging trigger type and priority. For better understanding of the applications requirements and how the network should behave to satisfy these requirements, VANET applications have been investigated according to network layer characteristics. For example, some VANET applications require that information be transmitted to a certain location, the network layer protocol need to use the Geocast technique to deliver this information to the targeted area, while another application may need to send a data stream to a specific group, therefore, it should employ the multicast techniques. Furthermore, VANET applications have been further investigated to identify the type of the generated network traffic, particularly, the type of packet in terms of size, the transmission interval and transmission duration. Finally, more in-depth analysis of VANET application requirements from both network and transport layers has been provided. The investigation of VANET applications requirements has shown that these applications have different functionalities, characteristics and requirements. These different factors have a crucial

impact on network performance. A scalable and reliable routing engine is required to maintain and give a better performance, despite the variation in the network environment and applications requirements. The upcoming chapter discusses the routing proposal for Mobile Ad-hoc Network (MANET) and the related subclasses in more detail.

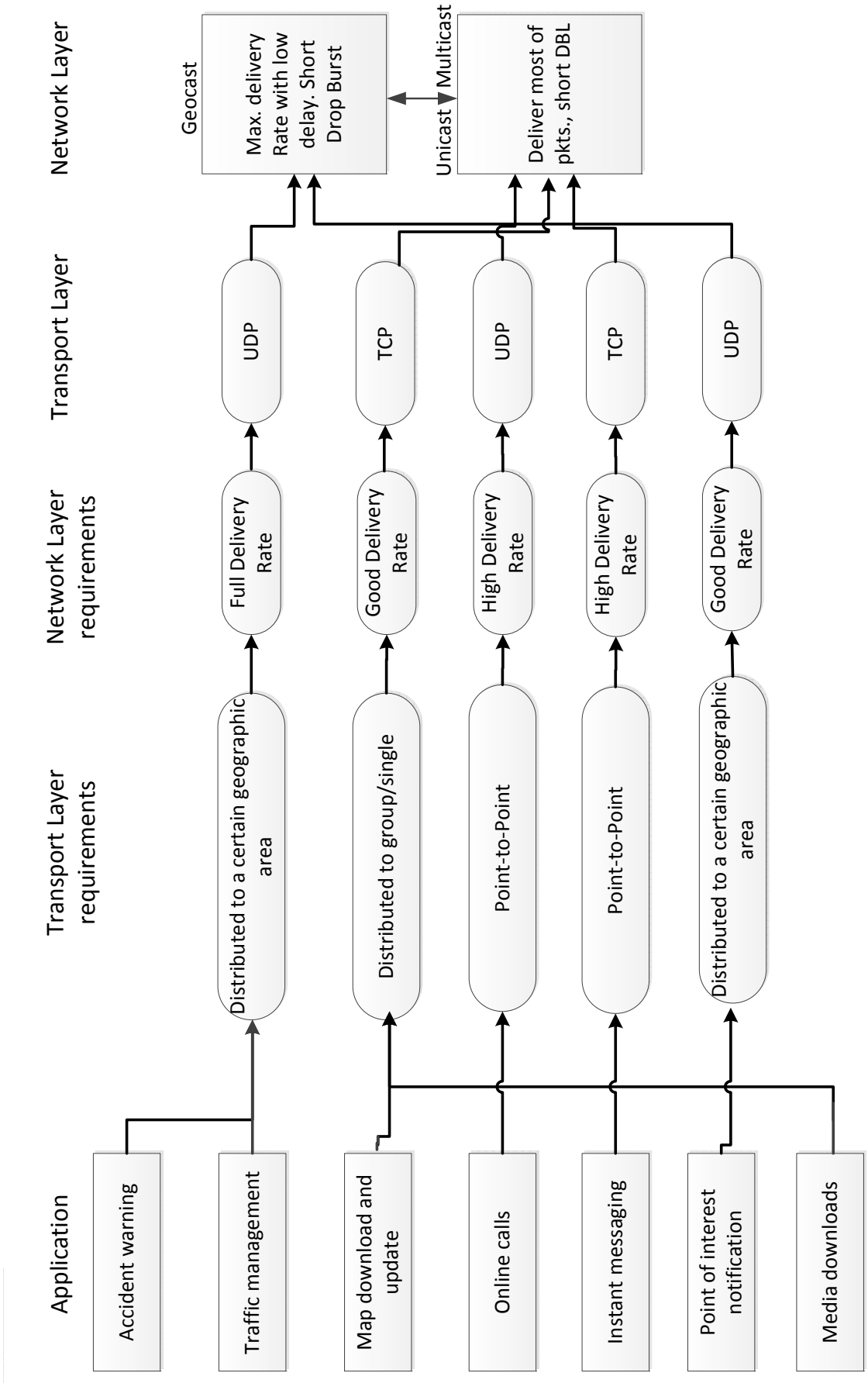

Figure 2.4: Applications classification based on Network and Transport layers requirements.

Chapter 3

VANET Routing

Vehicular Ad-hoc Networks (VANETs) are an emerging class of Mobile Ad-hoc Network (MANETs) where nodes include both moving vehicles and fixed infrastructure. VANETs aim to make transportation systems more intelligent by sharing information to improve safety and comfort. Efficient and adaptive routing protocols are essential for achieving reliable and scalable network performance. However, routing in VANET is challenging due to the frequent high-speed movement of vehicles, which results in frequent network topology changes. This chapter provides a survey of the routing proposals with emphasis on opportunistic routing strategies. Finally, a brief comparison between traditional routing and opportunistic routing protocols.

3.1 Conventional Routing

Routing in VANET is a vital factor to support the requirements of different applications types in different environments. There are a few variations between MANET and VANET. The main difference between them is the mobility pattern. Nodes in VANET move along certain type of mobility pattern, such as, follow predefined route (streets), follow speed limit signs, traffic lights and traffic conditions. All these factors have an impact on topology shape and network performance. The same cannot be said about nodes in MANET, due to the different mobility behaviour and limited computation resources, they move randomly or as groups. However, nodes in VANET and MANET have mutual features, they manage information by themselves without a centralised authority, and they are self-organised. They act as servers and/or clients to exchange information like peers [43].

Since VANET is a special type of MANET, conventional MANET protocols could be applied in VANET. These conventional routing proposals could be cat-

egorised into two main classes; topology based which could be further classified as reactive (on-demand) and proactive (table-driven), and position based protocols. Routing protocols that fall under topology based use link information in the network to deliver packets to their destinations. Whilst position based protocols utilise geographical position information to choose the shortest path to a destination [18].

In reactive routing protocols, routes are established when needed to allow nodes to communicate with each other and nodes maintain routes in use. Consequently, it reduces the amount of network overhead caused by broadcasting routing information. Reactive routing protocols start a route request process in order to find an active route to a given destination by flooding query messages into network [36]. In case of an application decids to send data packets, no packet will be sent until a route is formed. Whereas, proactive technique determines routes to all nodes in the network in advance by storing these routes in one or several routing tables, hence, routes to all nodes always available whenever they needed. Nodes in a topology based update their routing tables periodically in order to discover all routes by exchanging routing messages. As a result, the route update process causes large network overhead [35].

On the other hand, position based routing protocols utilise geographical information for each node in topology to make all routing decisions, thus, each node needs to announce its position, to do that, each node periodically broadcasts small packets called beacons contain geographical information of the node; this could compromise the privacy. Increased node velocity leads to inaccurate position information and high topology changing rate could cause route disconnect. Furthermore, position based protocols work well in dense networks, however, it fails in sparse network due to some regions without nodes (voids)[54].

Ad hoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) are an example of reactive routing, while Optimized Link State Routing protocol (OLSR) represents a proactive category, Greedy Perimeter Stateless Routing (GPSR) stand for geographic aware routing protocols. Fig 3.1 illustrates the classes of the conventional routing protocols. The following sections describe the routing mechanisms for these four routing protocols.

3.1.1 Ad hoc On Demand Distance Vector (AODV)

Ad hoc On Demand Distance Vector (AODV) is one of the most popular routing algorithms in mobile ad-hoc networks (MANTEs) area. It is an on-demand routing protocol which means that it establishes routes and maintains them to deliver a

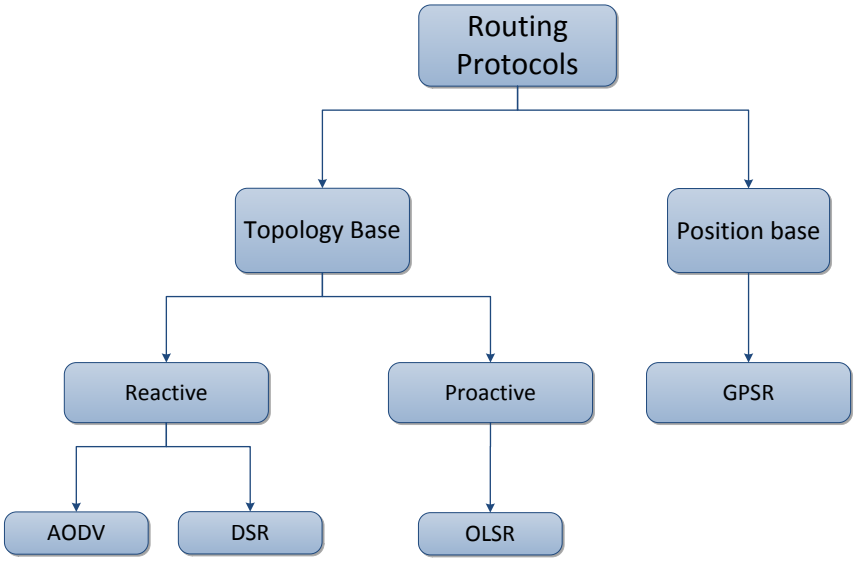

Figure 3.1: Conventional routing classes

packet stream if a source node needs to. Each node maintains its routing tables those contain information about neighbours nodes by employing HELLO messages [52] [53]. When a source node wants to communicate with another node within the network, it starts a route discovery process, by broadcasting a Route Request message (RREQ) to all its neighbours. RREQ message contains the following fields (source address; source sequence no.; broadcast id; destination address; destination sequence no.; hop count). To avoid route loop problems, whenever a source node issues a RREQ, the broadcast-id is incremented, due to nodes could receive a number of copies of the same RREQ packets from several neighbours. When an intermediate node receives a RREQ, the broadcast-id and source address will be checked. If the RREQ received before, it will drop and do not rebroadcast it.

Moreover, intermediate nodes should use an active route entry in their routing tables rather than an expired one. To achieve that, RREQ messages are checked by comparing the sequence number of a given destination in their routing tables with the destination sequence number in the RREQ. If the sequence number in the RREQ is greater than in the routing tables, the intermediate nodes must not use their route information as a reply to the RREQ. Instead, the intermediate node rebroadcasts the RREQ to its neighbours. The Route Reply RREP message must unicast to a neighbour who receives the RREQ from, only if the intermediate node has a route with a sequence number that is greater than or equal to that contained in the RREQ message this means the intermediate node has a fresh route to destination. A RREP packet contains the following fields: (source address;

destination address; destination sequence no.; hop count; life time of route) [52] [53].

If the intermediate node receives multiple RREP it updates its routing information and forwards RREP packet toward a destination if the RREP has a greater sequence number than the previous RREP message, or if it has the same sequence number with the smallest hop count.

Furthermore, due to the mobility nature of nodes in MANET and VANET environment, the probability of link breakage is increasing. In AODV when a node detects a link failure, it propagates a Route Error (RERR) message to source node and its neighbours to inform them about unreachable destination. Neighbours update their routing tables and delete the route entry. Source node re-initiates a route discovery process after receiving the RERR message [52] [53]. Fig 3.2 and Fig 3.3 illustrate route discovery process with three types of messages (RREQ, RREP and RERR). Marina and Das proposed On-demand Multipath Distance Vector Routing in Ad Hoc Networks (AMODV) [48], which is a multi-path extensions to AODV and has similar routing mechanism.

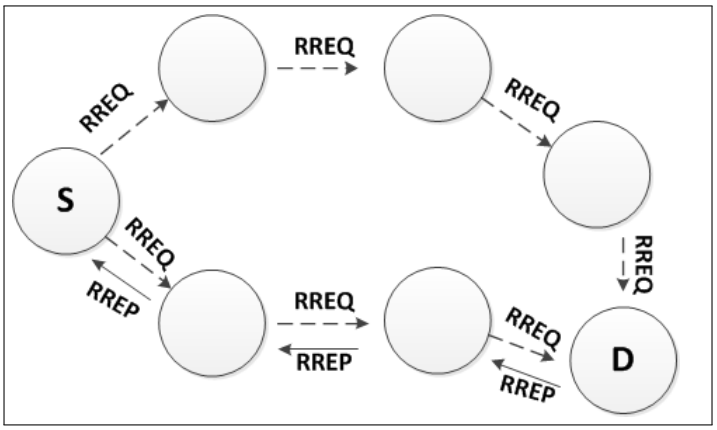

Figure 3.2: RREQ and RREP messages [1].

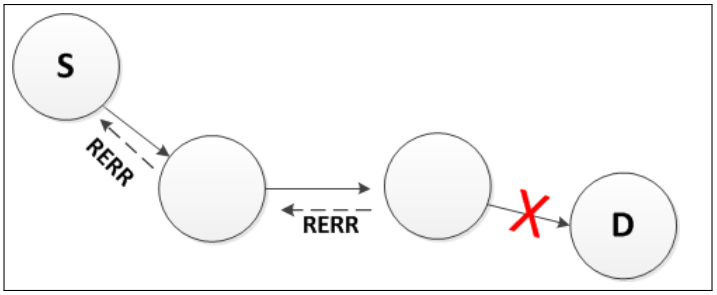

Figure 3.3: RERR message.

3.1.2 Optimized Link State Routing protocol (OLSR)

Optimized Link State Routing Protocol (OLSR) is a table driven routing protocol, i.e. exchange routing information with other nodes in topology continuously to keep the routing table up to date. OLSR is designed based on link state algorithms [29]. It provides immediate availability of route whenever needed because of its proactive nature. Due to mobility behaviour in MANET and VANET, network topology experiences rapid changes. These changes in network topology cause a flood of topological information to all nodes in the network. In order to reduce the amount of network overhead caused by route update messages, OLSR uses Multipoint Relays (MPR) strategy [27]. The concept of MPR is to make every node within a network select a set of one hop neighbour nodes to retransmit its topology update messages to them. This set of chosen nodes called Multipoint Relays (MPRs) of that node [29].

Additionally, neighbours nodes those are not in the MPR set of a given node can receive and process broadcasted messages but cannot retransmit them. In order to select MPRs, every node broadcast a list of its one hop neighbours periodically using 'Hello' message; from the list of nodes in the 'Hello' messages, the MPRs selector are determined, the list of neighbours, cover nodes with a distance two hop neighbours [17]. Beside 'Hello' message, OLSR keeps maintaining routes to all destinations in the network by using Topology Control message (TC). TC messages are broadcasted periodically to a whole network by MPRs nodes, unlike 'Hello' messages, which are broadcasted to MPRs nodes only, and they are one hop away. [27]. TC message contains a list of MPRs nodes, and a sequence number is used to avoid the loop problem because of infinite retransmission of the message. Although TC message does not contain a list of all neighbours, only MPRs nodes, but this information is sufficient to build the topology of network that provides the shortest path to destination [29]. Fig 3.4 illustrates MPRs concept [34].

3.1.3 Greedy Perimeter Stateless Routing (GPSR)

The Greedy Perimeter Stateless Routing (GPSR) is a well known position-based routing protocol which can be used in vehicular ad hoc networks. GPSR utilise vehicles positions to make packet forwarding decisions. It employs greedy and perimeter forwarding models. The greedy model makes forwarding decision using information about a router's immediate neighbours in the network topology. In other words, a forwarding node can choose locally the best next hop. In particular,

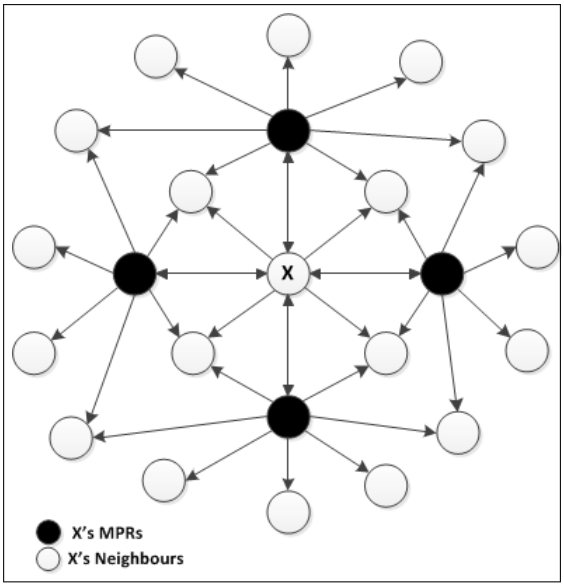

Figure 3.4: MPRs Concept [27].

a node chooses the next hop which is the geographically closest neighbour to the packet destination. An example of greedy model in GPSR shown in Fig 3.5. Here emptyX forwards the packets to Y because the distance between Y and D is less than that between D and any of X's other neighbours. This forwarding mechanism keeps repeats until packet reach D. All nodes in the topology maintain their neighbours table (which is stores the addresses and locations of their single-hop radio neighbours) by broadcasting beacons periodically. In the case of not receiving a beacon from a neighbour for longer time out of interval, a GPSR node assumes that the neighbour gone out of range or has failed [33].

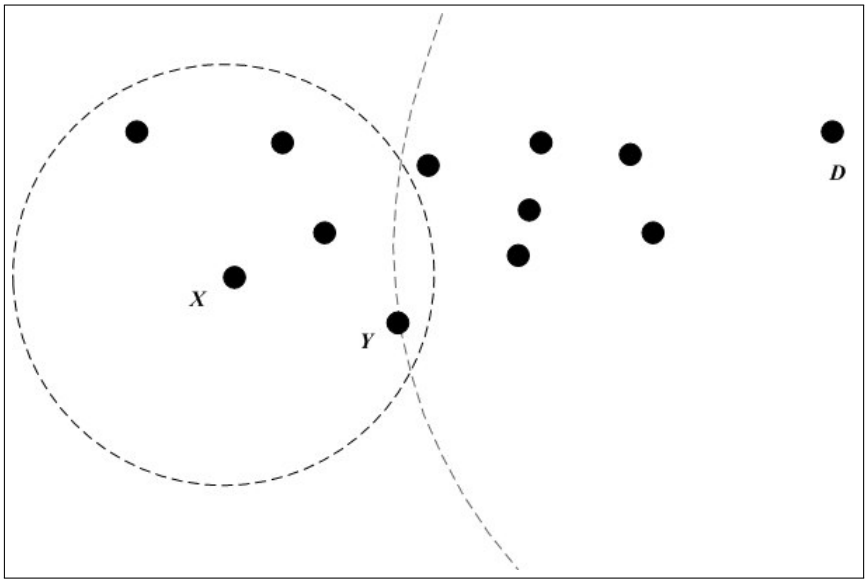

Figure 3.5: Greedy forwarding example [33].

In case of receiving a greedy-mode packet for forwarding, a node searches its

neighbour table for the geographically closest neighbour to the packet destination. If no neighbour is found, packet marked into perimeter mode. This perimeter mode packet is forwarded using simple planar graph traversal, when each node receiving a packet marked as in perimeter mode uses the right-hand rule to forward it to nodes, which are located counterclockwise to the line joining forwarding node and the destination. Fig 3.6 shows the perimeter forwarding example, where D is the destination; X is the node where the packet enters perimeter mode; forwarding hops are solid arrows; the dashed line is the failed greedy route. Each node checks the present distance to the destination, If the current distance is less, packet is routed through greedy forwarding repeatedly from that point onwards, otherwise, keep a packet on perimeter mode[33].

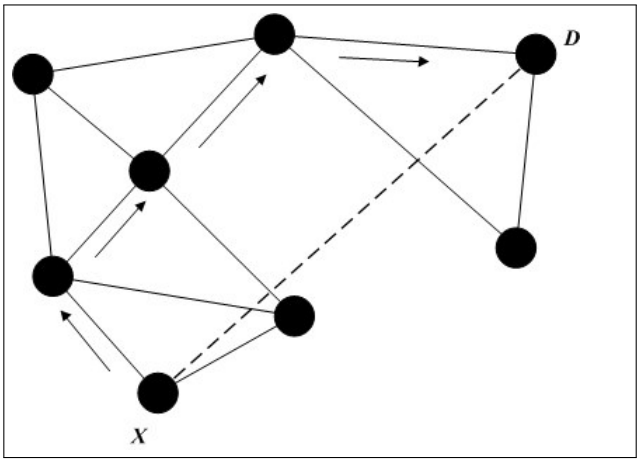

Figure 3.6: Perimeter mode example [33].

3.1.4 Dynamic Source Routing (DSR)

The Dynamic Source Routing protocol (DSR) is proposed by Johnson et al. [30]. DSR is an on-demand routing protocol designed for MANETs. DSR composed of two main phases of route discovery and route maintains, which routes to be computed when necessary then maintain them using the source route technique. In the source routing, the sender node determines the complete sequence of hops, which the packet has to traverse. The routing discovery phase similar to that in AODV [53]. The sender flood the Route Request message (RREQ) over the network until the RREQ reach the destination node. Fig 3.7 shows an example of Route Discovery, in which node S is attempting to discover A route to node D by transmitting a Route Request message as broadcast packet, which, received by all the nodes within the wireless range of S. Each node RREQ message contains the sender ID, the destination ID and a unique ID determined by the sender to avoid forwarding loop problems. The route request message also contains a

list of each intermediate node through which particular copy of the RREQ has been forwarded. This route list initialised to an empty list by the sender of the Route Discovery. When the RREQ is received by an intermediate node that not received this RREQ before, this node appends its own address to the route list and propagates the RREQ message by broadcast it. This process continues until the RREQ message arrive the destination node or an intermediate node has a valid route to the destination.

The destination node reply back to the sender with Route Replay message (RREP) which contains the accumulated route record that the destination fetch it from the RREQ message. The sender upon receiving the RREP message, stores the route to the destination in its route cache for subsequent routing. In the Route maintenance phase, each node transmitting the message is responsible for confirming that the packet has been received by the next hop; the packet is retransmitted up to a maximum number of attempts until this confirmation of receipt is received, otherwise, the transmitter node returns a Route Error message (RERR) to the sender of the packet, identifying the link is no longer available (broken). The sender node removes the broken route form its routing cache and look up in its routing cache for an alternative route to the desired destination, if there, then send message using the new route immediately, otherwise, the sender performs a new Route discovery. For example, in the scenario illustrated in Fiq 3.8, node S is sent a packet to node D using a source route through intermediate nodes A,B and C. In case of number of retransmission is exceeded and no confirmation is received by B, means node B unable to deliver the packet to the next hop C, stating that the current link between S and D is broken. Node B returns a Route Error message (RERR) to the original sender of the packet, node S. The sender (S) removes this broken route from its cache, If node S has in its cache another route to D (form additional Route Replays), it can sends the packet using the new route. Otherwise, it perform a new Route Discovery for this destination.

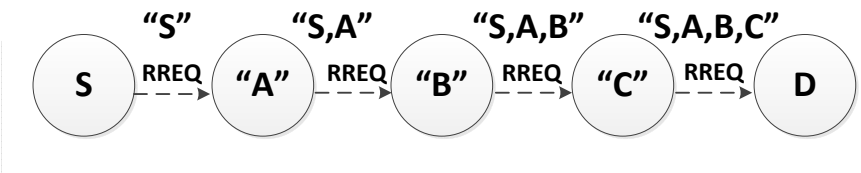

Figure 3.7: Route Discovery example: Node S is the sender, and node D is the destination.

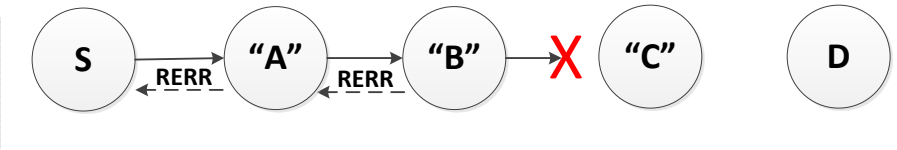

Figure 3.8: Route Maintenance example: Node S is the sender, and node D is the destination.

3.2 Opportunistic Routing

Over the last few years many studies have been carried out in the data routing field to improve network performance in wireless networks. Conventional routing protocols for wireless networks perform path discovery process and select best path before the actual transmission starts. This strategy applies the main principles inherited from routing in wired networks. As a matter of fact, conventional routing protocols exploit different strategies to be adaptive in a highly dynamic environment such as using the RERR message in DSR and AODV in case of a route breakage occurs. However, these strategies are not good enough to cope with highly dynamic topologies such as VANETs. Consequently, these routing protocols use excessive MAC layer retransmissions trying to build alternative routes, waste of network resources, and could lead to network collapse.

In addition, most of conventional routing protocols see the wireless medium as a an obstruction that is difficult to deal with due to its temporal variation. In other words, conventional routing protocols are unable to update the link costs at the right time that the wireless link variations occur. They then force to use inaccurate or outdated costs, which limits the network performance [15].

The shared wireless medium should be considerd as an opportunity rather than a limitation. This developed the concept of opportunistic routing. The key idea behind opportunistic routing is to overcome the weakness of unreliable wireless transmission by using the advantage of the broadcast nature of the wireless medium.

Instead of pre-selecting a specified relay node at each forwarding process, opportunistic routing broadcasts a data packet so that it is overheard by multiple neighbours which later decide to forward or not based on the protocol mechanism.

As been shown in previous chapters, conventional MANET routing protocols could satisfy VANET applications requirements to a certain extent. This brings the need for new techniques to route data packets to their destinations in a VANET environment. It seems the best way to achieve high throughout in a rapid network topology is to use flooding techniques. This is done by making all the nodes receives the same packets including the destination. However, flooding data packets

all over the network causes a network congestion problem and consumes network resources. In order to overcome these shortcomings, in this thesis, an investigating of the Opportunistic Routing (OR) techniques is conducted. Flooding could be considered as a naive version of opportunistic routing.

Opportunistic Routing (OR) aims to increase the level of efficiency and reliability by reducing the number of forwarding for each packet. Forwarding can be defined as a process of retransmitting a packet when an intermediate node receives it. In OR there is no predefine route to forward a packet to a next hop, it is rather that each node acts individually and takes decision based on current circumstances and available information. In most OR algorithms, each node seeks for better qualified next hop to forward the data packet by setting up a Candidate Set (CS). CS contains a list of selected nodes those who qualified to be the best next hop forwarders based on a specific metric. If none is available (CS is empty) it will wait until the right opportunity to forward the packets. In the following section some of the related literature is described briefly.

The concept of opportunistic routing was firstly proposed in 2005 in Opportunistic Multi-Hop Routing for Wireless Networks (ExOR) [9], the aim of this routing protocol is to improve conventional routing performance by exploiting broadcast nature of the wireless medium. The main functionalities of opportunistic routing are implemented in this scheme. More about opportunistic routing schemes in the following section.

3.3 Opportunistic Routing Literature

As mentioned earlier, employing opportunistic routing enables builds routes on fly via selective opportunistic forwarders. Opportunistic routing broadcasts data packets to a set of candidate nodes rather a single pre-selected forwarder. Candidate nodes that received the data packet execute a selective algorithm to select best nodes among other receivers according to the value of a specific metric to forward the packet. These steps are iterated until the packet is received by the destination. The main functions in opportunistic routing are [15]:

- Candidate set selection.
- Broadcast data packets to the candidate nodes.
- Best forwarder selection.
- Data packet forwarding

In addition to these four functions, other issues need to be taken into consideration in the design of opportunistic routing protocol such as candidate set prioritisation. More about opportunistic main functions are as follow.

A. Candidate Set Selection

Each node uses opportunistic routing, broadcasts a packet to multiple nodes neighbours simultaneously, and so if the transmission to one of the next hops fails, an alternative neighbour that successfully receives the packet can forward it on. This set of multiple next hops is defined as a Candidate Set and denote it as CS throughout this thesis. When the source node broadcasts a packet to the CS, many candidates might receive the same packet. In order to avoid forwarding duplication, one neighbour of these candidates is selected to do the forwarding. To do that, each node in the CS is assigned a priority value that is calculated according to a predefined metric. if the candidate with highest priority value receives the packet, it will forward the packet to its destination. Otherwise, the candidate with the second highest priority will forward the packet, and so on. The rest on the neighbours in CS will discard the packet. The CS selection has four main operations as follow.

I. Candidate Discovery (Neighbour Discovery): To discover the neighbour nodes, periodic or non-periodic broadcast could be used. The selection on a neighbour is done according on the radio link quality which is highly unstable and variable. At the beginning, all the discovered neighbours are included in the CS, then the potential forwarders are sorted according to predefined metric. This CS could be further refined by removing the candidates those may degrade the performance.

II. Candidate Prioritisation: After the CS is formed, priority values are assigned to the candidate nodes according to a specific metric. Thus, the selection of a metric has a large impact on the opportunistic routing performance. The metrics that are widely used in the literature are, the geographical distance [71], the hop count [51], the expected transmission time (ETT) [13], the expected transmission count (ETX) [8], and the coding gain [70]. Some metrics have been used in conventional routing protocols as well, such as ETT and EXT. The metric selection depends on the application requirements that the opportunistic routing is aimed to support.

III. Candidate Set Optimisation: The number of neighbours nodes in the CS provides higher flexibility. According to [15], the increase size of CS increases the number of candidates that are unable to hear each other,

consequently, duplicate transmission could occur. Zeng et al [75], [76] claim that it is often better to limit the number of candidate nodes to avoid duplication and minimise overhead Therefore it is better to remove some nodes from CS, especially in large network topologies, this process known as candidate filtering. The most popular candidate filtering approach removes the candidates that are worse than the sender according to a specific metric. However, this simple approach cannot guarantee optimal performances. Therefore, other considerations such as connectivity, duplicate likeliness and node contribution have been considered from the research community.

B. Best forwarder selection and forwarder announcement

The selection of the best forwarder among other candidates should be done to maximise the network performance in terms of a selected metric such as reliability, throughput, latency and energy saving. The best forwarder could be selected in a deterministic or probabilistic fashion [72] [67]. The most common deterministic selection approach is the priority-based selection. Another technique could be found in [67] when each node in a pre-build forwarding list must serve as a forwarder. On the other hand, probabilistic forwarder selection performs random candidate selection process. Each candidate calculates its own probability of forwarding after it receives the packet. Despite the fact this approach is simple, it could cause a significant increase of duplicate transmission if the CS size is big and the probability is not properly selected.

Forwarder Announcement (The Coordination Method) The aim of coordination method is to select the best forwarder and also helps the other candidate nodes to decide which one to forward and which to discard the packet. The best coordination method choose the best forwarder who considers the smallest cost in term of latency, overhead and packet duplication.

C. Data packet forwarding

After the candidate node has been selected, this candidate of the CS performs the packet forwarding process. As a result, every packet follows one path only to reach its destination. This path is determined on the fly based on which nodes receive the packet successfully at each hop. Most opportunistic routing protocols use acknowledgements on the successful packet reception. Broadcast packets in opportunistic routing do not implement link layer acknowledgments. Therefore a couple of acknowledgment mechanisms have been used. Bissau et al [8] and Rozner et al [55] exploit end-to-end

acknowledgement generated by the final destination, while Bhorkar et al [7] and Lee et al [40] considered hop-by-hop acknowledgment generated by the forwarder. The hop-by-hop technique is designed to reduce the latency but cause more overhead. The end-to-end however reduces packet overhead but could lead to higher latency since the forwarding process depends on the acknowledgment generated by the final destination.

3.3.1 Taxonomy of Opportunistic Routing in Wireless Networks

The first opportunistic routing scheme was introduced in 2005, namely Extremely Opportunistic Routing (ExOR) [9]. The aim of ExOR is to improve the performance of conventional routing protocols by exploiting the broadcast nature of the link layer. ExOR selects the next hop opportunistically in packet-by-packet basis. Prior to that, most conventional routing scheme proposed for Mobile Adhoc Networks such as AODV [52], OLSR [29] and GPSR [33] select a shortest path between source and destination pairs and then forward each packet through a sequence of a predefined intermediate hops. The current literature on opportunistic routing uses different techniques to solve conventional routing problems from a different perspective. Most of these proposed schemes are sub-categories of five major classes [15]:

- Geographic Opportunistic routing: Proposals that are based on nodes' location are listed under this category.
- Link state aware: This category covers approaches that aim to improve network performance, reliability and throughput by considering the link quality and bandwidth in opportunistic routing design.
- Probabilistic Opportunistic routing: Protocols under this category trying to tackle the problems of frequently changing network topology by using link availability and quality prediction.
- Optimisation-based Opportunistic routing: This category covers the protocols that use optimisation tools from convex programming, game theory and machine learning theory to shape the problem of opportunistic routing and optimise some of its design components like candidate relays selection and prioritisation.
- Cross layer Opportunistic routing: In this class protocols designed to exploit the information exchanged between different stack layers (the Network

layer, the Physical (PHY) layer and/or the MAC layer) to get more accurate routing metrics measurements and scheduling-aware routing decisions.

Fig 3.9 shows the taxonomy of OR protocols. In the following sections, we review some of exiting literature based on the opportunistic routing taxonomy.

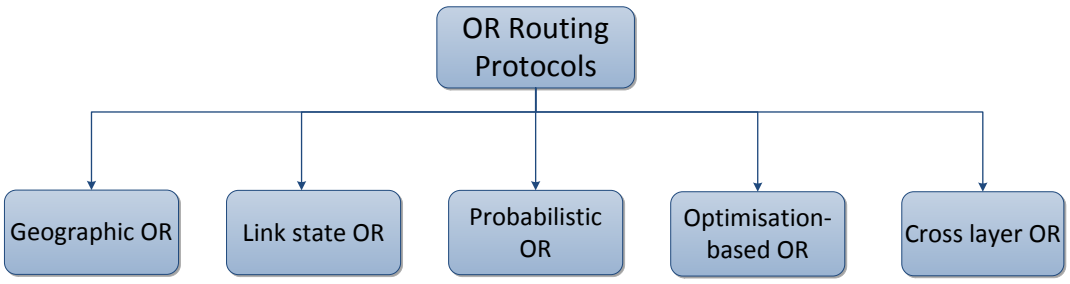

Figure 3.9: Taxonomy of Opportunistic Routing in Wireless Networks

3.3.1.1 Opportunistic Routing for VANET based on Geographic position

Geographic opportunistic routing provides an alternative solution for the lack of infrastructure in Mobile ad-hoc networks. In this section we review opportunistic routing proposals that build their forwarding decision based on location information. Fussler et al. [20] proposed Contention-Based Forwarding (CBF), is an opportunistic geographic routing protocol. In CBF the closest node to the destination among all the neighbours nodes is chosen to forward the data packet. The forwarder selection process is performed opportunistically through contention between other packet receivers. The source node first broadcasts Request-To-Forward (RTF) message to all its neighbours and wait, the neighbours compete with each other to reply Clear-To-Forward (CTF) message. The CTF packet transmission scheduled using a distance-based timer, so the closest neighbour to destination replies first. Once the neighbour replies with CTF packet to the source, the source forwards the data packet to the selected forwarder. This forwarding process continues until the packet reach its destination. Obviously, the CBF protocol does not need any topology in formation exchange compared with the conventional routing protocols. However, it causes an additional delay due to the sending of the RTF and waiting for the CTF to be replied.

Geographic Opportunistic Routing (GOR) [74] is another geographic opportunistic protocol. The authors claimed that assigning high priority to nodes close to the destination does not guarantee the best performance. Therefore, they proposed a local metric named Expected One-hop Throughput (EOT) to balance the

trade-off between the benefit (i.e. packet advancement and transmission reliability) and the cost (i.e. medium time delay). They employ candidate selection algorithm based on EOT. The algorithm includes a new node in the CS without changing the priorities among the already selected candidates; thus, the EOT of the new CS increases. In terms of coordination, GOR uses the ACK-based method.

Authors in [41] introduced GeoDTN+Nav which is a hybrid geographic routing technique to protect routing from frequent disconnection by using the vehicular mobility and on-board vehicle navigation systems to deliver packets in more reliable pattern, even in a partitioned network. The authors introduced virtual navigation interface (VNI) to find out which vehicle is able to forward packets in disconnected environments. The routing works based on the GPSR routing algorithm.

To secure location information during communication,a set of geographic opportunistic protocols hase been proposed. For example, Yuan et al. [72] design Resilient Opportunistic Mesh Routing (ROMER) protocol. ROMER try to maximise the resilience of the network by redundant packet transmission and randomise the selection of forwarder each time. The concept is to assign forwarding probability to each candidate. If a packet received by candidate is located on the shortest path, the candidate forwards the packet with the probability. Otherwise with the candidate not on the shortest path, the forwarding decision will be made based on the throughput of the nodes downstream and the desire level of packet redundancy. Each packet carries a credit and allows the maximum distance to deviate from the shortest distance toward the destination.

Nassr et al. design Directed Transmission Routing Protocol (DTRP) [51] for Wireless Sensor Networks. The forwarding mechanism in DTRP is similar to ROMER, however, calculation of the forwarding probability of candidate nodes is different. Sensor nodes located on the shortest path between the source and the sink perform forwarding with high probability. However, other nodes forward the received packet with a probability of the extra cost of an extra number of hops to the sink. The furthest forwarder from the shortest path, the lowest probability of forwarding. Another aspect should be mentioned, DTRP uses beacons exchange to get the hop count between sensor nodes and the sink.

Leontiadis and Mascolo present GeOpps protocol [44]. It is a geographical delay tolerant routing algorithm that exploits information from the vehicless navigation system to route messages to a specific location. In order to forward a packet to its destination, neighbour vehicles that follow suggested routes from the navigation system to their drivers's destination calculate the closest point to the packet destination. Next, nodes use the nearest point and their map in a utility function that expresses the minimum estimated time that this packet would

need in order to reach its destination. Finally, the node that is nearest to the destination and can deliver the packet become the next packet forwarder.

Another type of geographic opportunistic routing is the cross layer opportunistic geographic routing, which considers physical propagation dynamics during the forwarding process. For example, Wang et al. introduce Cooperative Opportunistic Routing protocol (CORMAN) [66]. The distance between nodes is calculated via exploiting the Received Signal Strength Indicator (RSSI) measure of the received packets. While Context Aware Opportunistic Routing (COR) [80], makes routing decisions based on the node's position, its mobility information (direction and speed) and link quality measured using the RSSI indicator. The radio transmission range in this protocol is represented by irregular shapes that reflects the signal quality variation.

Xuelian et al. [12] proposed Link State aware Geographic Opportunistic routing protocol (LSGO) which exploits a combination of geographic location and the link state information as the routing metric. In LSGO protocol, the forwarder is selected based on a vehicle's location and link's quality toward the destination.

Kevin et al. [42] proposed Topology-assisted Geo-Opportunistic routing (TO-GO) for VANET. In this approach, to deliver packets from source to a given destination, the sender determines the best target forwarder using 2-hops beaconing. The best target is defined as the farthest away node or a node located at a crossroad. If the best target is on a cross road, the node can forward packets in any direction. Then, the sender selects a set of candidates that can hear the target node and other candidate transmissions. Finally, after transmitting the packet, each candidate uses timer-based coordination to decide to forward the packet or not.

Wang et al. proposed opportunity routing protocol for data forwarding based on vehicle mobility association (OVMA) [65]. in OVMA, packets can be forwarded without passing through many extra intermediate nodes by not forwarding them to a certain range. The forwarding decision is adaptable to the vehicle densities by allowing to each vehicle carries the only replica information to record its associated vehicle information.

3.3.1.2 Link State Opportunistic Routing

As mentioned earlier, the main aim of the Opportunistic routing is to increase the reliability in packet delivery process via benefiting from the broadcast nature of the wireless medium by choosing next hop opportunistically. Opportunistic routing reduces the number of transmission failure and guarantees packet progress toward the destination. Many researchers exploit this reliability feature by using Packet

Delivery Ratio (PDR) based metric to show the efficiency of their opportunistic routing proposals. A review of some proposals that consider wireless link's delivery probability will be discussed below.

Biswas et al. propose the first link state opportunistic routing scheme, ExOR [9]. The beginning of opportunistic routing principles were shaped in ExOR. It combines link and routing layers functionalities. Essentially, the source node broadcasts a batch of 10 to 100 packets which contains a list of potential forwarders. Each node in the forwarding list receives this batch and waits its turn to perform the forwarding. ExOR uses TDMA-like MAC scheduling approach to assign the forwarding priority in the forwarding list, so, the candidate with higher priority performs the forwarding. Other candidates do the forwarding only if all candidates with higher priority failed to do so. The priority is calculated using the expected number of transmission required to deliver a packet to its destination (EXT). The EXT is calculated based on the loss-rate between the nodes.

Hus et al. propose a duplicate free opportunistic protocol (Economy) [26]. Economy eliminate duplicate transmission that might occur due to forwarders which can not hear each other. This is done by removing the those forwarders that cause the duplication and establishing a fully connected path by passing a token along the path for scheduling. The forwarders that hold the token allowed to perform the forwarding. Economy causes a considerable amount of overhead due to token exchanging mechanism.

In order to mitigate the overhead problem in the opportunistic routing proposals, the research community turned to and adopted network coding to solve this issue. For instance, Chachulski et al. [14] design a MAC-Independent Opportunistic Routing and Encoding Protocol (MORE), which uses intra-flow network coding and opportunistic routing to increase the network throughput. MORE exploiting network coding to select the best candidate to perform packet forwarding. MORE separates data packets into batches, each batch contains K packets. The source node continuously broadcasts a random linear combination of the K original data packets in the same batch. The receiving candidate that is closest to the destination in the sense of ETX is selected as the next forwarder. This forwarder only keeps innovative packets (i.e. packets that are linearly independent of the previously received packets in the same batch) to avoid packet duplication in forwarding process toward the destination. Following, it creates a random linear combination of this innovative packet and previously- received coded packets from the same batch, and broadcasts it. Based on the receipt of K linearly independent packets, the destination node restores the original data packets and sends an acknowledgment (ACK) back to the source, this allows it to move on to the next batch.

3.3.1.3 Probabilistic Opportunistic Routing

Probabilistic opportunistic routing protocols suits applications that are characterised by frequent nodes mobility like Delay Tolerant Networks (DTNs). Opportunistic routing address the mobile applications' challenges related to reducing the communication delay and ensuring reliable traffic delivery. Below some examples of probabilistic opportunistic routing protocols.

The main and important function in opportunistic routing is the selection of the best forwarder among other candidate nodes to forward the packet on. The frequent changes in high dynamic environments such as MANET and VANET makes it difficult to decide if the selection of a best forwarder at the selective time is accurate. Thus, the early versions of opportunistic routing exploit blind opportunistic forwarding such as Epidemic Routing [64]. In Epidemic Routing, each forwarder transmits the packet to every encountered node that does not receives a copy of it before. Each forwarder performs this process until its copy of packet times out. This scheme achieves high throughput, however, it increases the overhead cost that cannot be accepted under high network load conditions. On the other hand, Spyropoulos et al. propose an enhancement scheme of Epidemic Routing, named it Spray and Wait [60] via controlling the number of copies of a packet in the network. This is achieved by assigning a number of logical ticket to each packet. Hence, the number of each packet copies will not exceed the assigned number. Whilst the Spray and wait technique reduces the routing overhead caused by packet redundancy, it achieves poor network performance in term of the packet delivery ratio (PDR). To cope with these challenges, recent opportunistic routing proposals use statistics and learning tools to make an online interface of candidate relays availability and quality in MANET [15].

Lindgren et al. proposed Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [45]. PRoPHET forwarding is made depending on the delivery predictability value that calculated based on encounter history between nodes. Delivery predictability implies the probability of future contacts between nodes. The forwarding performed only if the delivery predictability to a destination node of the next hop node is larger than that of the transmitting node. Each node updates its delivery predictability values by increasing the probability for the nodes that have been met. Delivery predictability value determines which node is more likely to deliver the message.

Burgess et al. introduce the first probabilistic approach in [11] for DTN networks. The authors propose MaxProp protocol. This protocol schedules the stored packets of each node based on its assigned cost. The cost calculation is based on

an estimated value of packet delivery likelihood. This value is adjusted by each node at each new encounter with another node. In fact, each node maintains a list of estimations of the probability of meeting other nodes in the network. Guo et al. propose opportunistic flooding routing protocols in [22] for Wireless Sensors Networks. It is a cross-layer opportunistic routing tailored for low-duty-cycle networks with unreliable wireless links and predetermined working schedules. The concept is, the sender performs the forwarding process probabilistically. The probability is estimated based on the delay distributions of the next hop nodes. Firstly, the nodes are shaped as an energy-optimal forwarding tree. Secondly, using the MAC layer, a statistical delay characterisation is made upon the tree structure. This protocol can only be applied in duty-cycled stationary wireless networks.

The research in opportunistic routing field reaches new levels after the early mentioned opportunistic routing protocols which range from geographic to link state aware and probabilistic opportunistic schemes. Recently, the research focuses on applications requirements such as reliability, latency, mobility and energy rather than opportunistic routing features via the optimisation of its building blocks and the integration of cross-layer interactions as shown in the upcoming sections respectively.

3.3.1.4 Optimisation-Based Opportunistic Routing

In recent years, there has been an increasing amount of literature on Optimisation-Based routing to enhance opportunistic routing performance by keeping routing practical and simple. The proposals in this class trying to smooth the building blocks process (The selection process of candidate forwarders) using optimisation programming [78], game theory [77], machine learning [7] and some approaches inspired from graph theory [38]. It would be considered an example from each approach proposals. Zhang and Li propose Optimised Multi-path Network Coding protocols (OMNC) [78]. OMNC exploits a rate control protocol to improve the lossy wireless networks throughput. OMNC employs multiple paths to push coded packets to the destination and uses the broadcast MAC to deliver packets between neighbouring nodes. The coding and broadcast rate is allocated to forwarders by a distributed optimisation algorithm that maximises the advantage of network coding while avoiding congestion. Using game-theory approach, Zhang et al. propose DICE [77] to optimise opportunistic routing decisions and rate allocation to multiple flows using negotiation or competition. In DICE, the authors use two different game modes in which the source node is modelled to be a cooperative player and selfish player respectively. A player's rule is to assign the encoding and broadcast rates to the source and intermediate forwarders, in

order to increase its own payoff or the social payoff. Laufer et al. propose Shortest Multi-rate Any-path First (SMAF) [38]. SMAF have the same time complexity as traditional short-path algorithms such as Dijkstra's algorithm and Bellman-Ford's algorithm. SMAF is a generalisation of the Dijkstra and Bellman-Ford single path routing algorithms to any-path routing. To make this generalisation, the authors exploit hyper-graph to model the network topology. This hyper-graph consist of directed hyper-edges that links a given node to a set of other nodes. Each link has a weight which represents the delivery ratio of that hyper-link. As a result SMAF is capable of choosing the bit rates and optimal forwarders set that minimise the cost of Expected Anypath Transmission Time (EATT) toward the destination. Bhorkar et al. propose an adaptive opportunistic routing approach AdaptOR [7]. AdaptOR protocol reduces the mean of the per-packet routing cost in the absence of links' quality and network topology information. AdaptOR uses a reinforcement learning framework to dynamically learn the probabilistic models of the network connections. AdaptOR has four main steps:

- The sender broadcasts the data packet to its CS.
- Nodes in the CS acknowledge the receipt of the data packet. This acknowledgement contains the value of the estimated best score (EBS).
- The sender decides the routing action based on a randomised rule. This action will be either to select a forwarder to do the forwarding or the termination of packet transmission.
- The sender updates its score list as well as its own EBS using the EBS values received from its neighbours.

The updated values of EBS are used to make future opportunistic routing decisions. As a result, the consistency of these learning-based opportunistic routing decisions depends on the successful delivery of the control packets that carry the EBS-related values. Consequently, AdaptOR is exposed to some performance degradation problems that may be caused by the control packets loss.

3.3.1.5 Cross-Layer Opportunistic Routing

The proposed opportunistic routing protocols in this category take the interaction between the network layer and underlying layers (Link and Physical layers) into consideration in designing the opportunistic protocol. Below are sections describe few examples of cross-layer opportunistic routing.

A. Physical-Aware Opportunistic Routing

This sub-class discuss the proposals that aim to improve the network throughput by considering physical interface or/and Channel State Information(CSI).

Lee et al. propose an iterative forwarding packets mechanism. If a recipient receives a packet, it acknowledge the sender at each hop. This protocol named Simple and Practical Opportunistic Routing protocol [40]for multi-hop wireless networks. Once the packet reaches its destination, the latter broadcasts an acknowledgement to stop any retransmission of this packet. This protocol causes considerable amount of latency and network overhead due to acknowledgement basis mechanism.

Zhao et al. propose a Topology and Link quality-aware Geographical opportunistic routing protocol (TLG) [81]. TLG uses multiple network metrics, including network topology, link quality, and geographic location to coordinate the opportunistic routing mechanism. At the beginning, the candidate sets created based on the nodes' location. The forwarders are selected according to the metric consist of the link quality (Received Signal Strength Indicator-RSSI), remaining energy and weighted sum of the progress. The main propriety in TLG is the ability of selecting the forwarder with higher priority according to this metric in short time compared with the other protocols.

B. Mac-Aware Opportunistic routing

Bruno et al. propose MaxOPP [10], an opportunist routing protocol for Wireless Mesh Networks (WMN). MaxOPP combines opportunistic forwarding with packet scheduling to support multiple simultaneous flows. It selects the forwarding node at runtime and per-packet basis by employing a localised routing decision process to opportunistically leverage any transmission opportunity generated by the short-term channel dynamics. In other words, MaxOPP does not precompute any candidate set list, and it does not force selected forwarders to transmit during preassigned time windows. On the contrary, wireless diversity generates multiple receivers for each packet transmission, and any of those receivers may be used as an alternative forwarder to deliver the packet to its destination.

C. Physical and Mac Aware Opportunistic Routing The proposals in this subclass benefit from the information that provided from link and physical layers added to the network layer features to deliver the packets in more efficient way to their destinations.

Lu et al. propose Protocol for Retransmitting Opportunistically (PRO) [46]. PRO is an opportunistic-forwarder based Mac-layer retransmission protocol

that has the capability of overhearing nodes to perform as forwarders that retransmit packets on behalf of the source node, after they learn about a failed transmission. In order to achieve that, firstly the PRO estimate the link quality towards the destination using calibration process. Secondly, they filter out all the poor forwarders using a local qualification process. Finally they, choose the best list of eligible forwarders among all qualified forwarders and prioritises them using a distributed forwarders selection algorithm. To insure high priority forwarders transmit with high probability, 802.11e Enhanced Distributed Channel Access (EDCA) is leveraged.

Zuo et al. propose Cross-Layer Aided Energy-Efficient Opportunistic Routing protocol [82]. This protocol exploits the benefits of cross-layer information exchange, including the knowledge of the Frame Error Rate (FER) in the physical layer, the maximum number of retransmissions in the Medium Access Control (MAC) layer and the number of forwarders in the network layer to perform opportunistic forwarding based on energy-efficient. The building of a candidate set and perform routing based on the energy consumption metric.

3.4 Opportunistic Routing versus Conventional Routing

One of the main disadvantages in conventional routing solutions is the need to maintain network topology information regularly. This information is collected in a periodic fashion. Therefore, whenever a change happens in the network topology, it will trigger a flood of messages exchange throughout the whole network. This massive routing update exchange leads to significant degrades in the network resources. In mobile networks where the topology is rapidly changing, it is impossible to establish any hierarchy or logical structure for routing. MANET reactive routing protocols, like AODV and DSR, need to cache all routes to destinations, these routes are discovered on demand each time a packet needs to be routed to a particular destination. Moreover, these routes would likely become outdated quickly, even sometimes before starting the forwarding, due to the frequent mobility of nodes. Conventional routing protocols suffer from high maintenance cost, reliance on static routes for the proactive protocols and high risk of using outdated information for the reactive protocols.

On the contrary, opportunistic routing ends the need of topological information exchange process on the large scale fashion to find out the best route toward a par-

ticular destination. In fact, nodes do not have to establish routes and worry about maintaining them since only the nodes whom successfully received the packet participate in the forwarding process. Furthermore, packets transmitted from same source to same destination could follow different paths based on opportunistic reception of these packets, this property gives more flexibility and freedom of adaption in highly dynamic network topologies.

3.5 Summary

This chapter has provided the current literature on routing protocols in mobile ad-hoc networks in general. Some of well known existing protocols have been explained in both conventional and opportunistic routing. Conventional protocols are categorised into two main classes of position-based and topology-based protocols. A separate classification is into reactive (on-demand) and proactive (table-driven). Topology-based protocols use link state information in the network to deliver packets to their destinations, While position based protocols utilise geographical position of the intermediate nodes. Four examples of the conventional protocols have been described (AODV, OLSR, GPSR and DSR) as representatives of conventional routing classes. Opportunistic routing proposals have been categorise into five main sub-classes according to the literature, including, geographic, link state aware, probabilistic, optimisation-based and cross-layer opportunistic routing. This chapter has also provided the key differences between conventional routing and opportunistic routing protocols.

Chapter 4 presents the proposed simulation model that are used to evaluate the performance of some MANET routing protocols(AODV, OLSR and GPSR) in urban environments. In addition, the simulation tools and the configurations are also listed.

Chapter 4

Methodology

This chapter introduces a new measurement criteria (Drop-Burst Length) as well as discusses the proposed simulation model to evaluate the network performance with the selected routing protocols in VANET urban environment. The employed tools (NS2 and SUMO) and the parameters that used to configure the simulation model are described also.

4.1 Network Simulator NS2

Network simulator version 2 (*NS2*) is a simulation tool used to simulate all types of networks. It has been developed at UC Berkely, it is a discrete event simulator written in C++ and Object-oriented Tool Command Language OTcl [23]. *NS2* is an open-source simulation tool that can be run under different platforms. C++ is used to define the internal mechanism of the simulator. OTcl is used to set-up and configure the simulation objects as well as scheduling the events. It is widely used in networking research as the capability of providing varying types of routing protocols, IP protocols, TCP and UDP, for example and multicast protocols over wired and wireless networks. One of the advantages of *NS2* is to support many protocols and has the ablility to visualise network topology. Moreover, *NS2* provides multiple algorithms in queuing such as Drop Tail, RED and CBQ. After each simulation, it dumps two types of files, a text-based file called trace file and an animation-based file called NAM file. The Network AniMator *NAM* tool is used to visualise the network topology [28]. Fig. 4.1 shows the basic architecture of *NS2*.

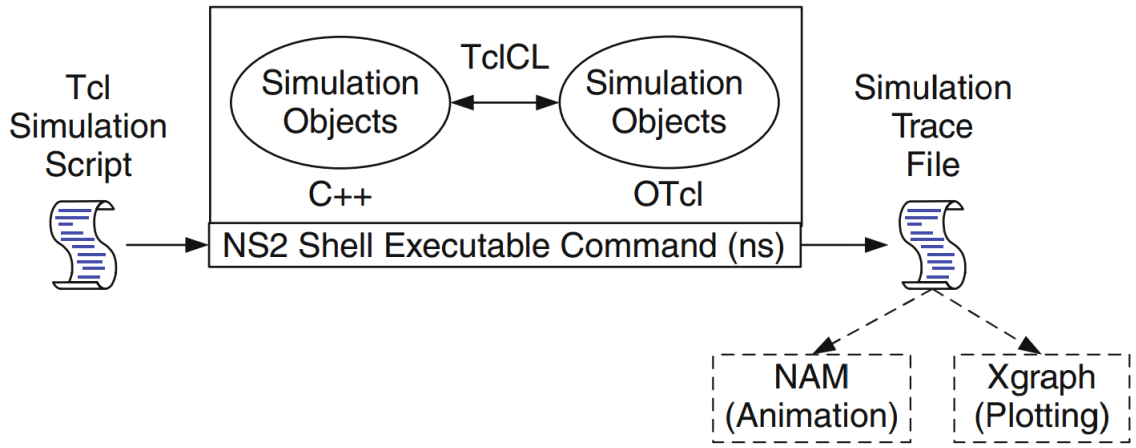

Figure 4.1: Basic architecture of NS2 [28].

4.2 Simulation of Urban MObility (SUMO)

Simulation of urban mobility (SUMO) is open source traffic simulation package developed by the German Aerospace Centre (DLR) in 2001. It is supported road traffic simulation to allow the researchers implement their own algorithms. In order to simulate Vehicular ad hoc networks (VANET), it is observed that SUMO considers as a most popular road traffic application among other mobility generator applications due to it is generated realistic mobility models for VANET simulations. Usually, it generates mobility file that uses later with other communication simulators such as *NS2*, *NS3* and *OMNET++* to simulate VANETs. In addition, SUMO has a graphical user interface and allows setting up a particular road parameter such as speed limits signs, connections across intersections, and traffic lights [6]. Figure 3.2 illustrates the graphical user interface of SUMO.

4.3 Simulation Experiments

For simulation to be effective to evaluate the performance of a network it must be configured to be representative of reality. Factors that increase simulation realism in the case of VANETs are the application network traffic model, the mobility model (vehicle traffic model), the medium access (MAC) protocols and the model of the impact of an urban area obstacles on radio signals together with fading of the radio channel. One or more of these is often neglected as in [68][4], consequently, results are less likely to be truly representative. Moreover, the majority of the evaluation studies used traditional metrics to measure network

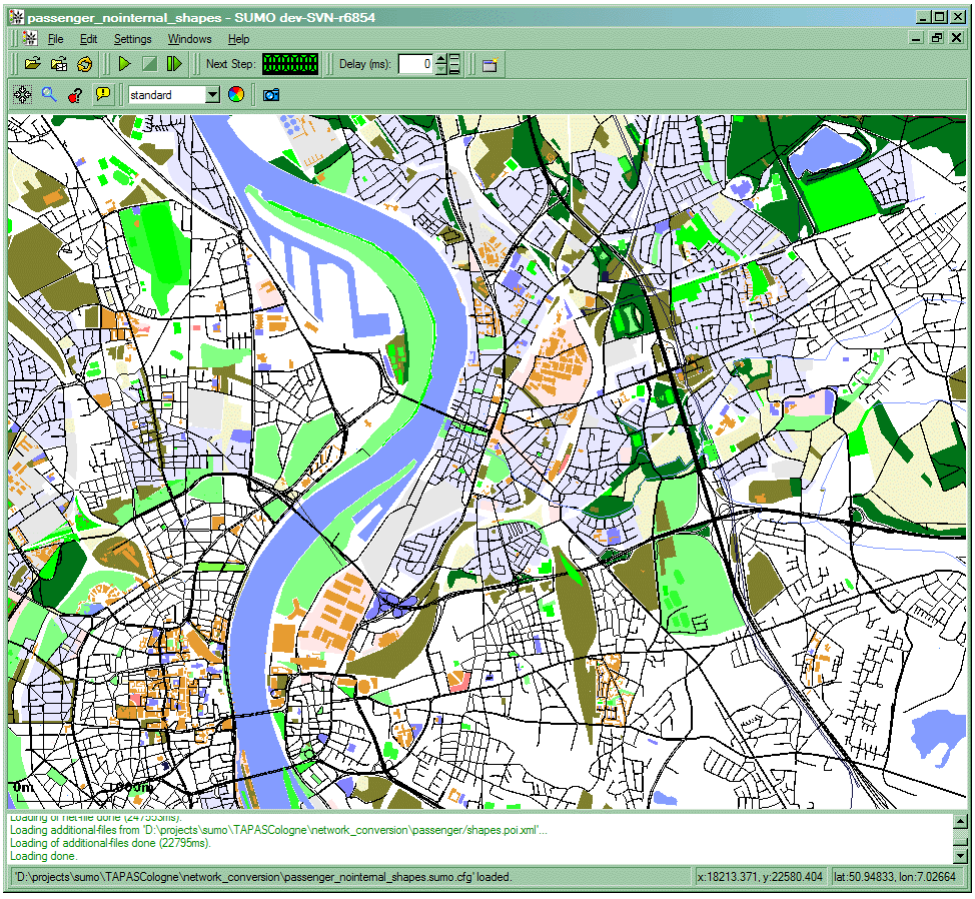

Figure 4.2: Graphical user interface of SUMO [6].

performance with different routing protocols such as average end-to-end delay and average packet loss. While there are instances where these have some value, for example, a safety critical message must be delivered within a short delay, these metrics do not fully reflect actual network performance as perceived by the application and user; they measure averages sometimes losing vital information in the calculation. To overcome these issues, we have introduced Drop Burst Length (DBL). This measures the probability of drop a consecutive number of packets in each connection. Real time traffic is more susceptible to burst drops so this metric provides a better indication of performance. Fig. 4.3 illustrates short and long DBL and how it has an impact on application performance. DBL is calculated as a probability distribution of each packet drop burst.

4.3.1 Performance Criteria

Three performance criteria were used to evaluate the performance of AODV, OLSR and GPSR routing protocols in the experiments are as follows.

- Drop Burst Length (DBL): It is a probability distribution of drop a particular number of consecutive data packets during the whole simulation time. This metric has crucial impact on applications' efficiency, especially real time applications. Some applications have the capability to cope with small drop burst length (DBL) such as one or two consecutive packets, however, when DBL is increased this could has a bad influence on application performance. Fig 4.3 shows the concept of DBL.

- Packet Delivery Ratio (PDR): The ratio of the data packets that successfully delivered to the application layer in destinations compared to the data packets that have been sent by a source [24].

- Car-to-Car (C2C) delay: it is the time taken for a packet to be transmitted across a network from the application layer in a car to another including infrastructure acting as forwarders nodes.

The following subsections describe the network parameters (standards and configurations) that used in the simulation experiments.

4.3.2 Simulation Configuration

To ensure some realism in the simulation, the following factors are considered:

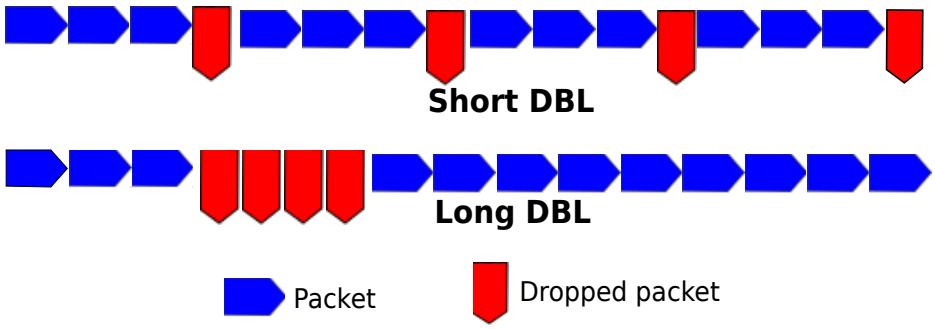

Figure 4.3: Short and long DBL.

4.3.2.1 The network traffic model

According to the U.S. Department of Transportation report [25], the shape of the network traffic depends on an application requirements, different VANET applications create different network traffic. VANET applications can be categorised into three major classes (safety applications, traffic management applications and commercial applications). Table 4.1 presents typical application requirements. Each category has a set of requirements that should be provided to the applications, safety-critical class for example, it required a minimum 10 messages to be sent every second with small packet size using connection-less transport protocol, and to be delivered within 100ms.

4.3.2.2 Mobility Model

To simulate VANET environment in a realistic way, a suitable mobility model should be applied. Several mobility models are proposed to simulate VANET urban environment. Most evaluation studies employ even an artificial mobility model or digital maps of an urban area. In order to obtain accurate results from the simulation, the simulations should considers the both models (artificial and real map).

4.3.2.3 802.11p standard

The Institute of Electrical and Electronics Engineers (IEEE) proposed a new modification of the IEEE 802.11 standard in order to add wireless access in vehicular environments (WAVE) to solve problems such as reflections and high speed of the vehicles. In order to operate with extremely high quality of service and support the nature of the automotive applications (reliable broadcast), IEEE define enhancements to 802.11 which is the basis of products marketed as Wi-Fi required to support Intelligent Transportation Systems (ITS) applications specificities, this includes data exchange between high-speed cars and between cars and RSUs. The

Table 4.1: Some examples of VANET applications requirements[25]. [SC=Saftey Critical, CRS=Cooperative Road Safety, TM=Traffic Management, CM=Commercial, CO=Connection-oriented,CL=Connection-less,LW=Light-weight,HW=Heavy-weight(IP)], V2X=V2V or V2I

Application	Category	Conn. mode	Allowable latency (ms)	Minimum message freq. (Hz)	Transport protocol	Packet Format
Braking Warning	SC	V2X	100	10	CL	LW
Emergency vehicle warning	SC	V2X	100	10	CL	LW
Roadwork warning	CRS	I2V	100	2	CL	LW
Weather condition	CRS	V2V	500	2	CO	HW
Intersection management	TM	I2V	500	2	CL	LW
Time to traffic light change	TM	I2V	100	1–10	CL	LW
Electronic commerce	CM	I2V	500	1	CO	HW
Media downloading	CM	I2V	500	1	CO	HW

new modification is called 802.11p, it support delay critical and high priority VANET applications [63][21][39].

4.3.2.4 Network Traffic Model

In VANETs, network traffic models are varied. The shape of the network traffic depends on an application requirements, different VANET applications required different network traffic, it depends on which class that an application belongs to. VANET applications could be categorised into three major classes (Safety applications, traffic management applications and commercial applications). Table 4.1 represents the network traffic requirements for some VANET applications.

4.3.2.5 Network topology mode

In order to add more realism to the to the VANET simulation based research, another factor should be taken into account which is the network topology mode. This factor has an effect on the network performance. In this thesis, a hybrid architecture mode is considered to emulate a real world network topology. Hybrid architecture combines both network topology modes, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). In V2V, a vehicle can communicate with another vehicle either on single hop or multi-hop mode, depending on the distance between them or routing protocol architecture, the same can be said about V2I but the connection will be between vehicle and road side unit [18].

4.3.3 Propagation model

In an urban environment, radio frequency (RF) suffers from severe fading due to the presence of buildings or other obstacles, these act as barriers for radio signals. Consequently, it is unlikely that line of sight between transmitter and receiver exists. Therefor, the characteristic of an urban environment and its impact on the radio signal should be taken into account of the simulation studies.

4.4 Summary

In this chapter, the proposed metric of the drop-burst length (DBL) and the potential impact on the network performance is provided. The methodology of evaluation network performance with the selecting routing protocols is proposed. It covers the shape of the simulation model, the simulation tools, the network traffic pattern (apply VANET safety application traffic) and mobility pattern. The simulation experiments consider the urban area as a simulation environment. Two urban mobility models are involved to simulate urban area, i.e., artificial map and real map. A real urban obstacle representative fading model is integrated to reflect the impact of buildings and obstacles that result in the scattering, reflection and diffraction of the radio signal.

In the following chapter, the extracted results from the simulation experiments will be discussed and analysed based on several different criteria to evaluate the performance of the network with AODV, OLSR and GPSR as routing protocols.

Chapter 5

Evaluating conventional routing in VANET urban environment

This chapter presents the simulation setup to evaluate the performance of the selected routing protocols alongside the discussion and the analysis of the simulation results. This chapter illustrates the simulation test-bed that compare different routing protocols in a VANET networks.

5.1 Simulation Setup

We employ between 5 s to 20 s flows of 10 packets per second. Each simulation is for a random length of time with the total number of flows varying from 200 (low) to 1000 (high).

For simplicity, Geocast and multicast traffic model are not considered. A unicast network traffic with safety critical application requirements has been used as it is suits most VANET application requirements. This type of network traffic is exploited to assess the network performance on a larger scale scenarios.

The simulations involved two mobility models (artificial and real map) as follows.

- Real street map: we used part the London congestion zone and Leicester city centre maps as a representative of real road network maps, we generate random vehicle trips over those maps. The OpenStreetMap website enables to capture a real world map in different format, it is used to capture part of the London congestion zone map and Leicester city centre. In order to generate random trips on the captured map, Simulation of Urban MObility (SUMO) framework [6] is involved. It is an open source traffic simulation package developed by the German Aerospace Centre (DLR) in 2001.

- Manhattan mobility model is considered one of the most popular mobility models that represents an urban environments because it contains a grid of streets that organized vertically and horizontally. In the Manhattan model, nodes follow a probabilistic approach in the selection of its direction, since at each intersection a vehicle chooses to keep moving in the same direction or change it. The probability of going straight is 0.5 and taking a left or right is 0.25 each. It can be noted that this model is not suitable for highway systems.

Fig. 5.1 , 5.2 and 5.3 illustrate simulation scenarios in a configured Manhattan map, portion of the London congestion zone and portion of the Leicester city centre respectively using SUMO. 100 vehicles move at speeds up to $20\,\mathrm{m\,s^{-1}}$, with 13 fixed roadside units.

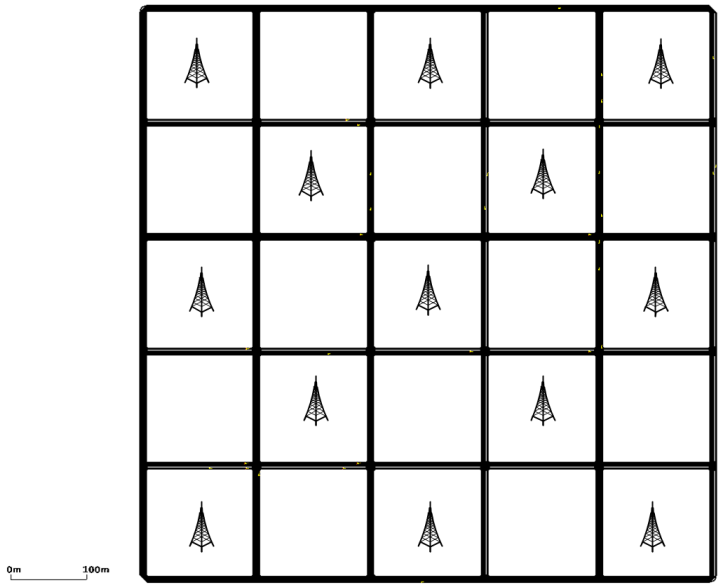

Figure 5.1: Manhattan mobility model map with RSUs in SUMO.

In order to reflect the characteristic of an urban environment, we employ the Nakagami fading model. This propagation model is a mathematical modelling of a radio channel with fading. It represents the impact of building on the wireless signal, it has more configurable parameters compared with other propagation models such as two-ray ground and shadowing. The Nakagami propagation model has the ability of simulate various levels of fading on wireless channel, from a free space channel to severe attenuation channel in urban environments by changing shaping factor values[50] [56]. to configure the Nakagami propagation model in urban area, we use the following parameters ($m_0, m_1, m_2 = 1.0$, $use_nakagami_dist_ = false$, $\gamma_0, \gamma_1, \gamma_2 = 2.0$ and $d_{0\gamma}, d_{1\gamma} = 200, 500$ respectively) [50].

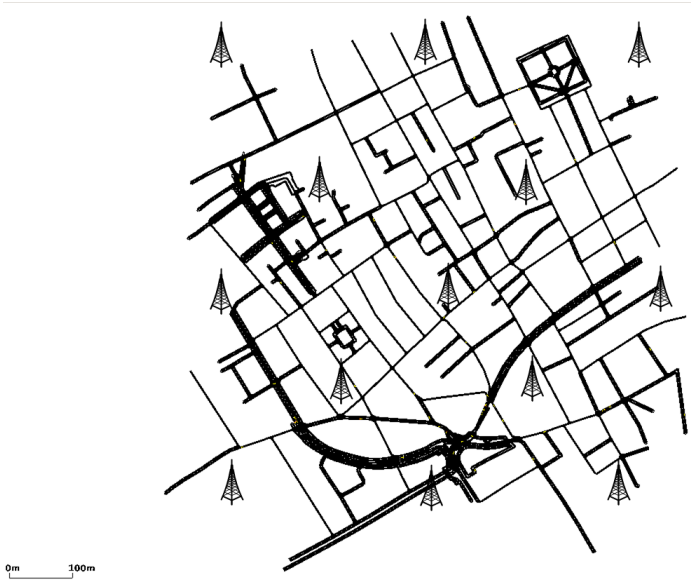

Figure 5.2: Part of the London congestion zone with RSUs in SUMO.

5.1.1 Simulation description

The simulations are configured using NS-2.35 platform [62][28]. The mobility traces models are generated using SUMO road traffic simulator [6] for the both scenarios (Artificial map scenario and Real World map scenario). 802.11p standard [21][39] has been integrated in all nodes within the simulation including infrastructure units to enable them to add Wireless Access in the Vehicular Environment (WAVE). Moreover, squared urban areas of 850x850m are considered for the all maps, 100 vehicles are deployed randomly and move freely on the roads with three lanes respecting traffic lights regulations on each intersection for 200 seconds (simulation time), 13 infrastructure are deployed to represent the hybrid network environment(V2V and V2I). In order to simulate the impact of buildings and other obstacles on wireless signal in the urban areas, Nakagami propagation model has been parametrised. Regarding network traffic model, constant bit rate (CBR) traffic sources and 100 byte UDP data packet payload is used with 10Hz packets per second as a message frequency to simulate a critical-safety application in VANET, each connection last for random period of time between (5, 20) seconds. To evaluate AODV, OLSR and GPSR in various network conditions that represents different time of a day (peak time and off-peak time), different network loads have been tested (low, medium and high), it has been assumed that low network load represents by 200 and 400 pair of connections, while 600 connections stand for medium network load, for high network load 800 and 1000 pairs of connections are considered. The reason behind choosing these number of connections as a representative of low, medium and high network load is we did simulate a different number of connections which is not included in this thesis, and we draw

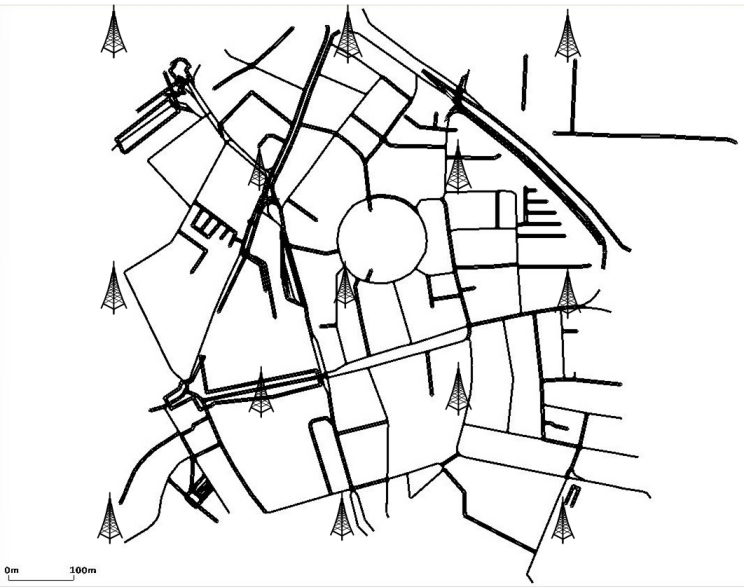

Figure 5.3: Part of the Leicester city centre with RSUs in SUMO.

a conclusion which is these number of connections are more representative to the real world scenario.

5.2 Results and discussion

The analysis of the network performance using DBL, PDR, C2C delay as described in earlier. Simulations were undertaken with increasing load, i.e. numbers of traffic flows (connections). Each flow has random duration (5 s to 20 s) at 10pps, on each map. Each run was performed five times with the same random source and destination selections for each flow on each run. Fig. 5.4, 5.5, 5.6 5.7, 5.8 & 5.9 show the DBL for three loads. We observe the performance of the selected routing protocols (AODV, OLSR and GPSR) is similar on the all maps.

Each protocol shows different performance:

- GPSR achieves the shortest C2C delay because it considers the closest neighbour that has a route to destination. Fig. 5.11, 5.12 & 5.13 illustrate this delay in low, medium and high loads.

- Fig. 5.11, 5.12 & 5.13 also show that with AODV packets take longer to be delivered under different network load on all maps. These longer delays are due to its route initialisation mechanism, it takes time to set-up a route to destination (sending a RREQ and waiting for a RREP). This leads to packets being queued and dropped before transmission and the probability of dropping consecutive packets with AODV increases along the simulation.

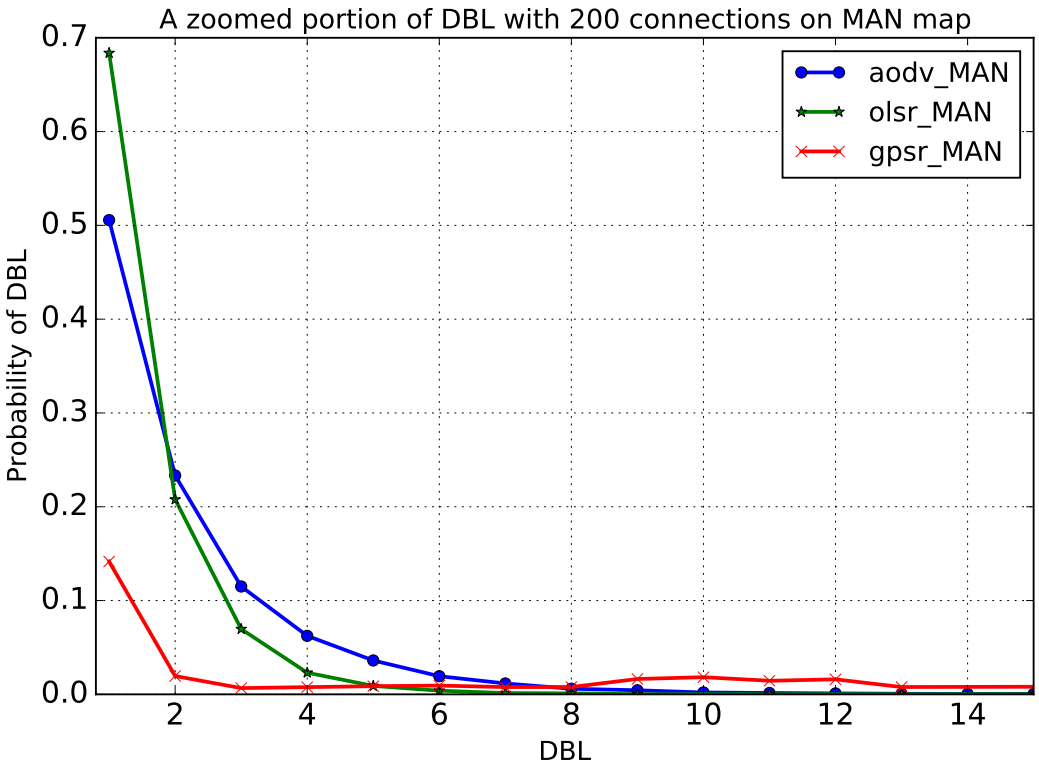

Figure 5.4: Short Drop Burst of AODV, OLSR and GPSR with various number of connections in Man map (A zoomed portion).

- OLSR provides a route to a destination immediately, and source node with GPSR already has the closest neighbour that has a route to destination, this can give an advantage for those protocols over AODV in terms of delay and DBL (Fig. 5.4, 5.5, 5.6, 5.7, 5.8 & 5.9), especially at the start of the connection.

- Using DBL we observe that long packet burst drops are avoided (Fig. 5.7, 5.8 & 5.9). OLSR and AODV recover a broken route quickly when a failure is detected, despite the fact that they have higher probability of one packet DBL under low network among other protocols see Fig. 5.4, 5.5 & 5.6. In other words, with OLSR, the network provided better QoS performance to the safety applications than other protocols.

- GPSR shows a worse performance in term of DBL. The probability of dropping the entire flow is much higher compared with AODV and OLSR, see Fig. 5.7, 5.8 & 5.9, although it performs much better under low network.

- OLSR outperforms AODV and GPSR in terms of DBL and PDR under low, medium and high network load. However, as load increases, the performance reduces as the drop ratio on MAC layer increases

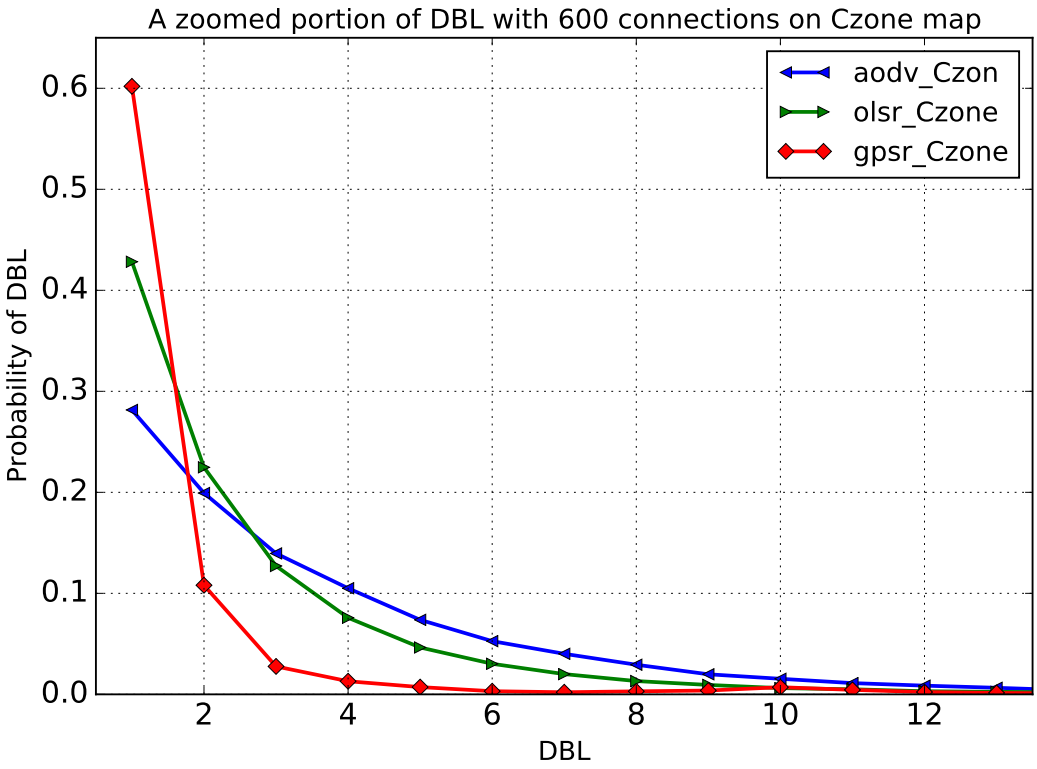

Figure 5.5: Short Drop Burst of AODV, OLSR and GPSR with various number of connections in Czone map (A zoomed portion).

- With AODV, the poor performance of the network is due to unavailability of routes to the next hop (NR), so the drop ratio increases at the network (routing) layer as shown in the Table 5.1. AODV failed to calculate paths from source to destination under high network load as a consequence of incapability of handling the growth in routes demanding.

- The reason behind of the most dropped packets with OLSR is MAC getting busy due to the frequent updates of OLSR routing tables. As network load increases OLSR failed to provide paths towards destinations.

- Despite the weakness with GPSR performance in terms of PDR under low network load, it shows a better performance under medium and high load (Fig. 5.10).

- Each protocol failed to provide high delivery rate due to a reason. Theses reasons are varied based on route building mechanism. Table 5.1 provides drop information for each simulated protocol on the selected maps. AODV failed to handle the increase demand on routes in the network, therefore, it is obvious that most of the dropped packets is due no route available to the desire destination (NR). While, the drop with OLSR is occurring

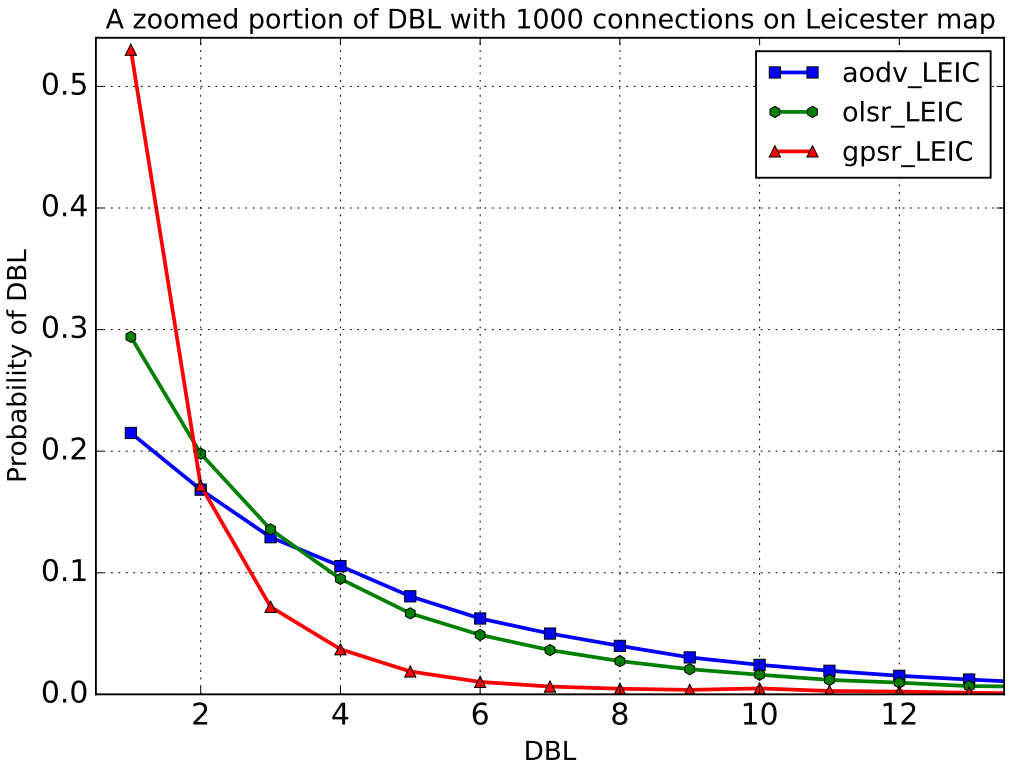

Figure 5.6: Short Drop Burst of AODV, OLSR and GPSR with various number of connections in Leicester city centre map (A zoomed portion).

in the MAC layer as the medium getting busy with increase load on the network alongside the routing tables update scheme. As the network load is increasing, the routing table could be outdated as routing messages not delivered and being dropped due to the busyness of the medium.

5.3 Summary

The results indicate that the variation of the selected urban maps configured Manhattan map, the London congestion zone and Leicester city centre maps have little influence the performance network traffic for these simulations.

Using our performance metric (DBL) we find OLSR outperforms AODV and GPSR. With OLSR packet drops more commonly due to a busy MAC layer with AODV the failure to establish a path to the destination. With GPSR the network experiences a stable performance and the delay is the shortest among other protocols.

While no protocols provide all the requirements of a safety critical system, this lead to address key required to design a new routing algorithm that has the capability to cope with VANET characteristics. These key findings are as follows:

Table 5.1: Drop ratio on both MAC and Network layers

AODV	Manhattan		Czone		Leicester	
	Dropped		Dropped		Dropped	
Load	NR	MAC	NR	MAC	NR	MAC
200	0.67	0.33	0.68	0.31	0.68	0.32
400	0.77	0.22	0.78	0.21	0.78	0.21
600	0.81	0.18	0.82	0.17	0.82	0.17
800	0.84	0.15	0.84	0.15	0.84	0.15
1000	0.85	0.13	0.86	0.13	0.86	0.13
OLSR	Manhattan		Czone		Leicester	
	Dropped		Dropped		Dropped	
Load	NR	MAC	NR	MAC	NR	MAC
200	0.13	0.86	0.15	0.85	0.15	0.85
400	0.25	0.75	0.20	0.80	0.17	82
600	0.39	0.61	0.34	0.66	0.32	0.67
800	0.49	0.51	0.44	0.55	0.44	0.56
1000	0.57	0.43	0.53	0.47	0.52	0.48
GPSR	Manhattan		Czone		Leicester	
	Dropped		Dropped		Dropped	
Load	NR	MAC	NR	MAC	NR	MAC
200	0.62	0.37	0.45	0.55	0.49	0.51
400	0.44	0.56	0.32	0.68	0.38	0.62
600	0.34	0.66	0.27	0.73	0.31	0.69
800	0.32	0.68	0.22	0.77	0.25	0.75
1000	0.28	0.71	0.20	0.80	0.22	0.78

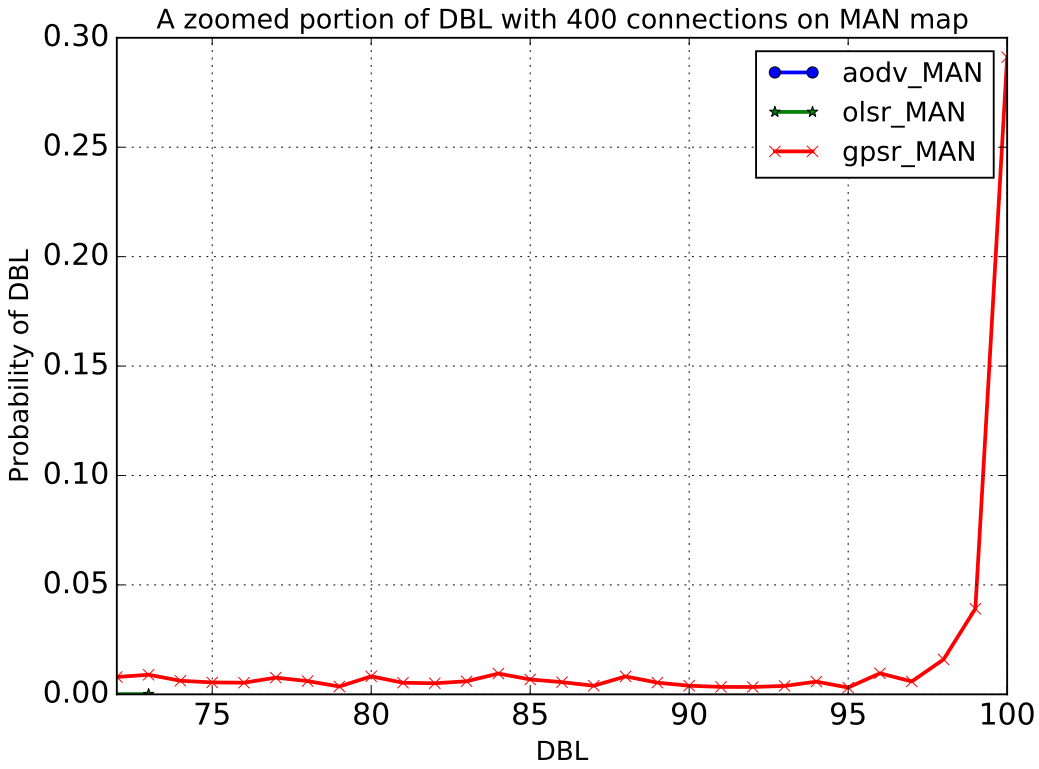

Figure 5.7: Long Drop Burst of AODV, OLSR and GPSR with various number of connections in Man map (A zoomed portion).

- Route set-up time has a crucial influence on network performance especially when the connection time is short.
- Geographic location information could be utilised to reduce packets delivery time, nevertheless, this could lead to frequent route disconnection due to a rapid topology change.
- Unicast routing fulfils some VANET applications requirements, however, it is not sufficient to satisfy all the applications.
- The choice of routing protocol has an effect on DBL, this could have an impact on applications performance, especially real-time applications.

To conclude, the selected routing protocols have fulfilled VANET safety applications' requirements as shown in chapter 2 to a certain extent. A new routing and forwarding technique needed to provide robustness and increase network throughput.

This chapter outlined the network performance with the three selected routing protocols against the proposed DBL metric and traditional metrics (Delay and

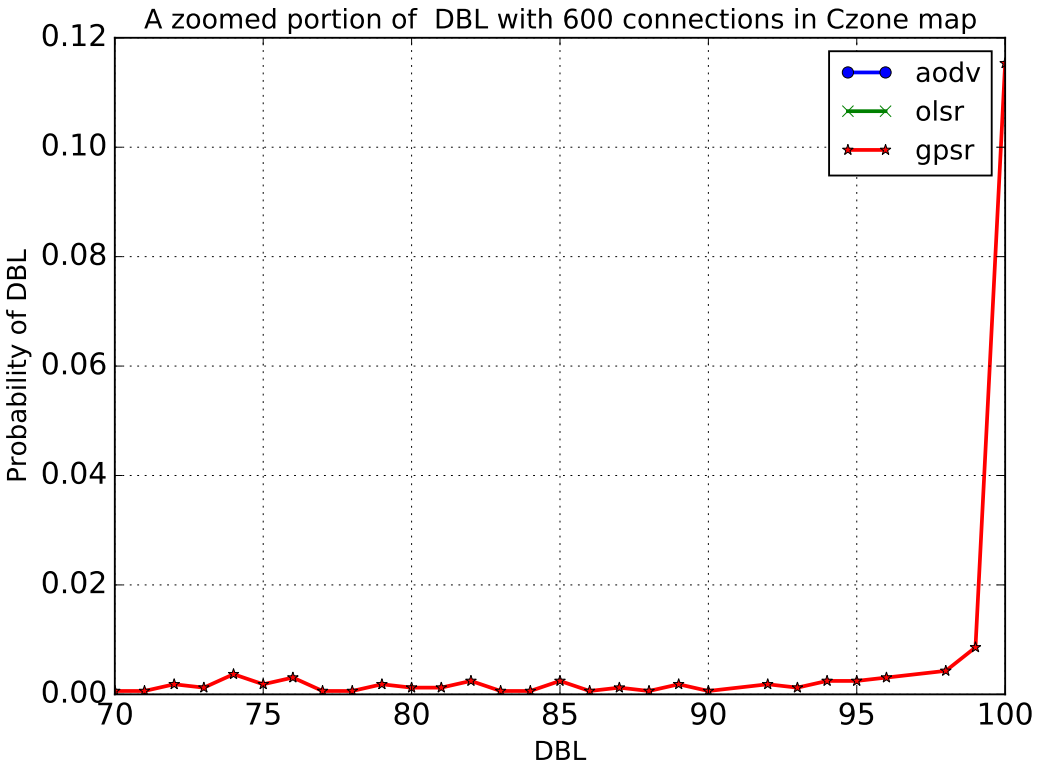

Figure 5.8: Long Drop Burst of AODV, OLSR and GPSR with various number of connections in Czone map (A zoomed portion).

PDR). The selected protocols are being investigated and tested in the VANETs environment. The relative performance and behaviour of each protocol in the scenarios are explained and highlighted.

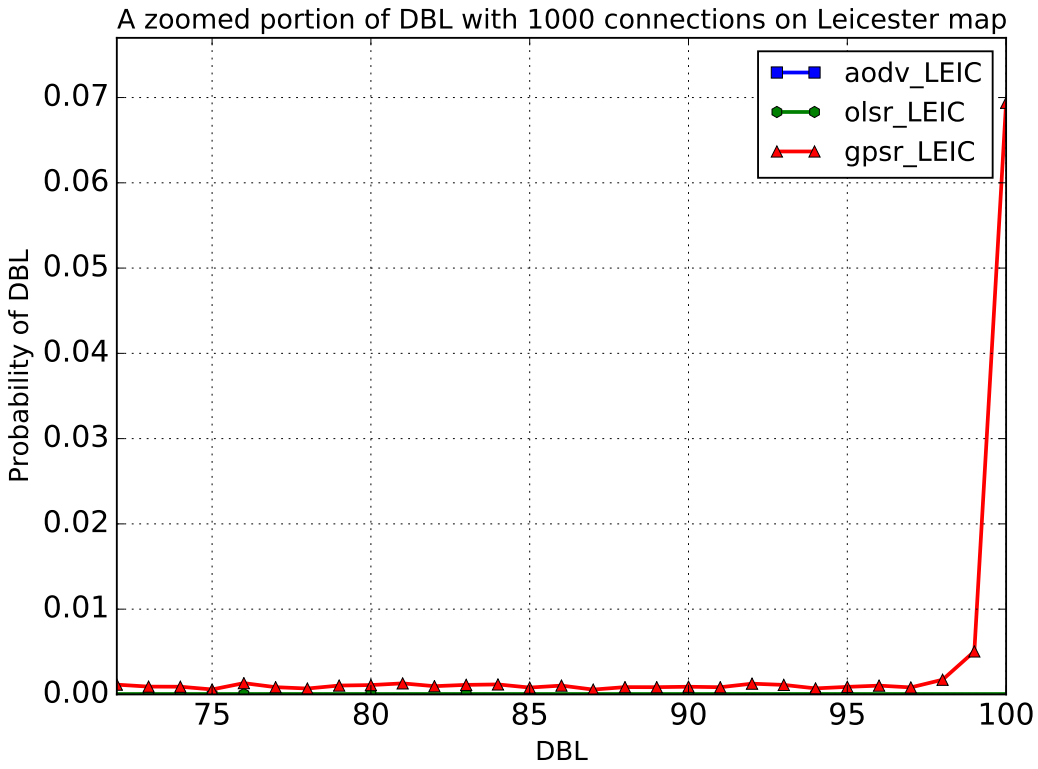

Figure 5.9: Long Drop Burst of AODV, OLSR and GPSR with various number of connections in Leicester city centre map (A zoomed portion).

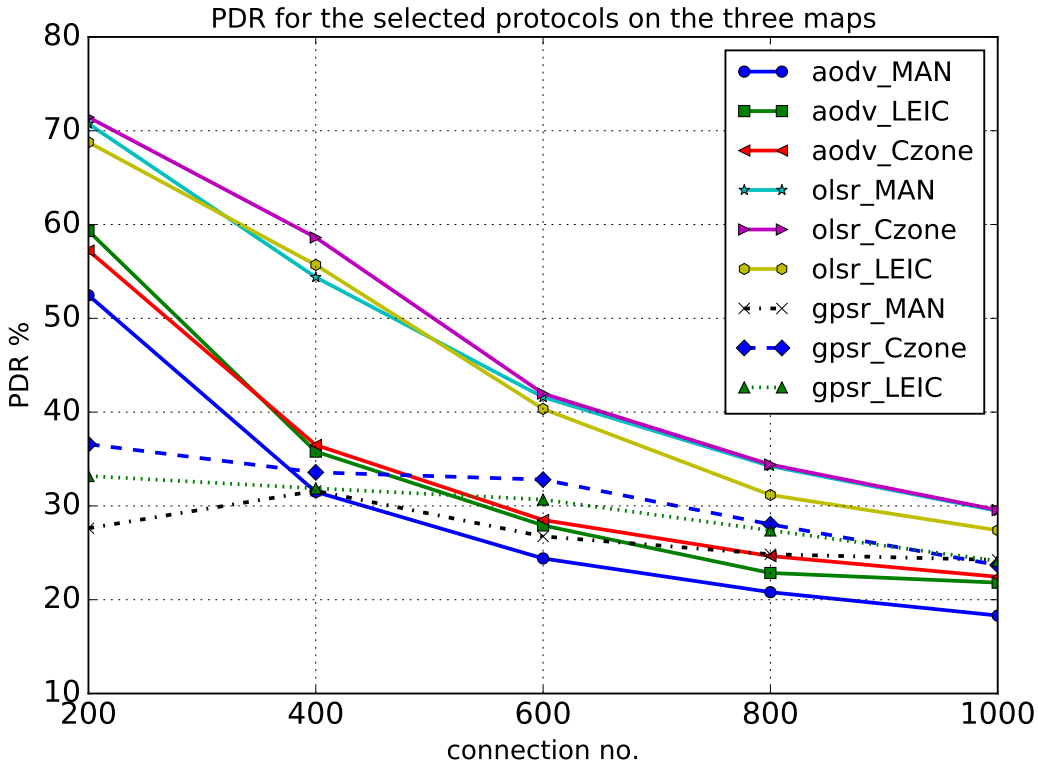

Figure 5.10: Packet Delivery Ratio (PDR) of the selected protocols on the all maps.

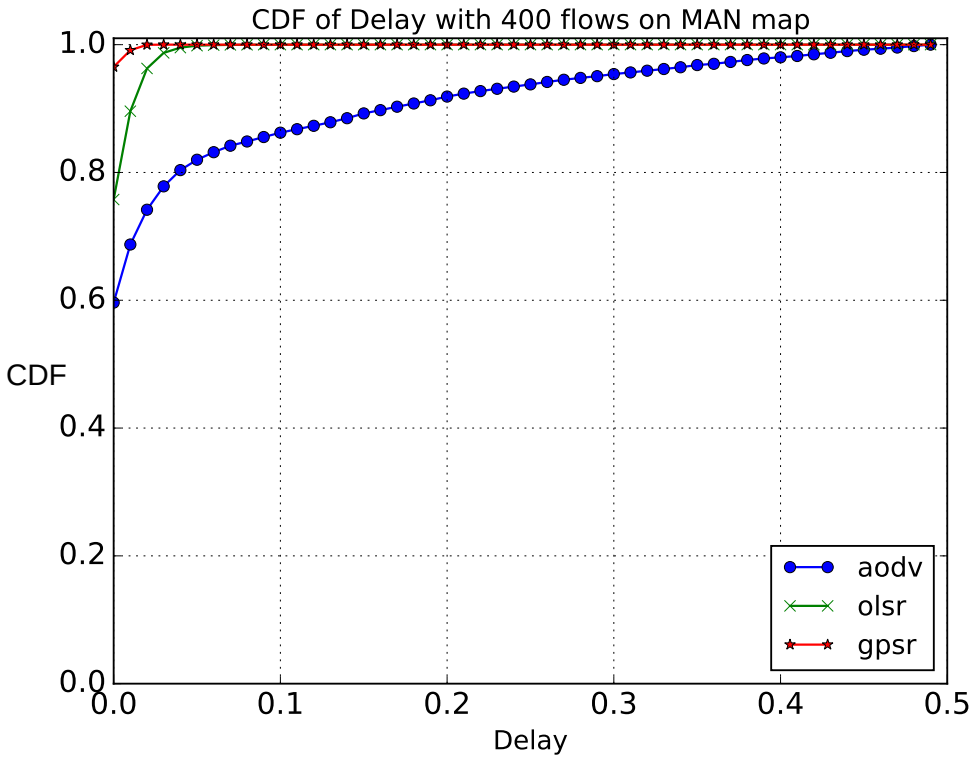

Figure 5.11: CDF of delay/s for the selected protocols under various network loads in Man map scenario.

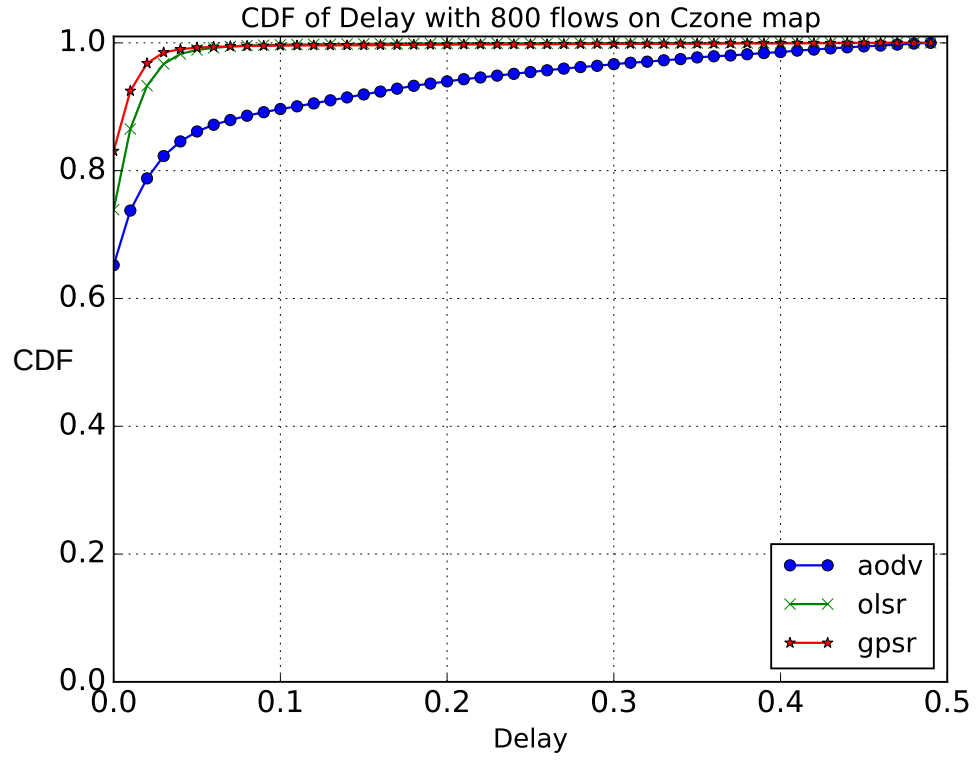

Figure 5.12: CDF of delay/s for the selected protocols under various network loads in Czone map scenario.

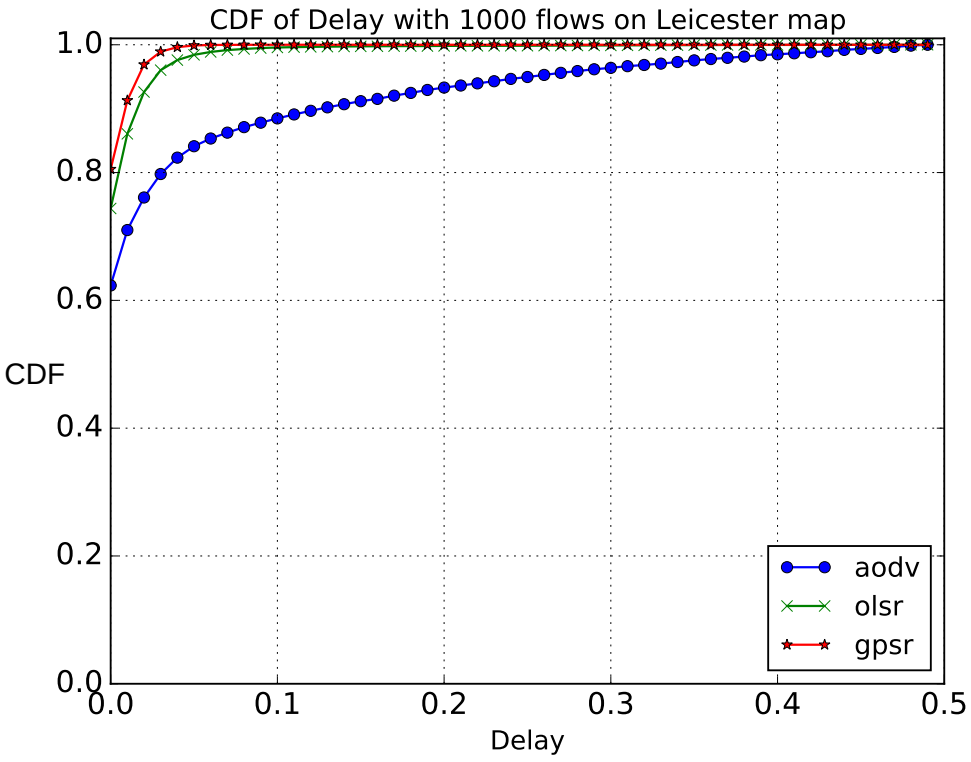

Figure 5.13: CDF of delay/s for the selected protocols under various network loads in Leicester city map scenario.

Chapter 6

Previous Hop Routing PHR

6.1 Motivation

As previously stated, routing in VANETs environments is very challenging issue. Each routing algorithm needs to sustain routes from source to destination even though nodes travel under varying velocity and with a fading environment that buildings have impact on radio signals. Routing protocol should cope with different network loads and satisfying various types of application requirements. Network loads could be varied in VANETs environments depends on the time of the day, the network could be dense at peak-time and sparse in off-peak time. All these conditions should be considered during designing a routing protocol to suits various VANETs environment. Conventional MANETs routing protocols have satisfied applications' requirements to a certain extent. Due to rapid network topology changes, forwarding protocol needs to be resilient, ensuring high delivery rate. One of the proposed techniques that might suits frequent network topology changes is the opportunistic forwarding scheme (OR). The Majority of proposed OR protocols employ candidate selection algorithm to specify which node should perform the forwarding. Many researchers proposed candidate selection algorithms based on link state, geographic location of nodes, Expected Number of Transmissions (ExNT) or using history of encounters and transitivity and the like as been described previously in chapter 3 section 3.2.

After identifying VANET applications requirements and challenges, in this thesis, a new concept of OR is designed by exploiting broadcast characteristic of the wireless medium. Each node after successful receiving a packet, decide whether going to forward this packet or not based on a probabilistic value and some network topological information. The probabilistic values is estimated based on the network busyness. This new technique named Previous-Hop Routing (*PHR*). The aim of (PHR) is to forward the packets towards the desired destination in

constraining flooding pattern by making the intermediate nodes who successfully receive a packet decide whether to forward it or not. This forwarding decision is taken based on valid topological information.

6.2 PHR Design

PHR exploits the broadcast nature of radio communication whereby several receivers, i.e., those in range, receive a message from a transmitter. This is similar to opportunistic routing (OR) as found in delay tolerant networking (DTN) [19][59]. However, there are some key differences. Firstly we are aiming to work in real-time so as to be able to support time dependent network traffic, such as video or safety-related accident information. Secondly OR protocols pay particular attention to the selection of the next hop node from a candidate set, using their limited knowledge of the topology towards the destination. In PHR each node makes a forwarding decision itself, based on information available. This information includes the destination, source and previous hop of the current message.

The simplest way to send a message to a particular destination is to flood. Each node rebroadcasts every packet as it is received, unless it has received the message before. The disadvantage of such an approach is the load placed on the network by unnecessary transmissions. *PHR* aims to constrain the rebroadcast (forwarding) decision to ensure lower load on the networking, yet still delivering the packets.

PHR protocol utilises the benefit of flooding in opportunistic scheme on a packet-by-packet basis to spread packet stream towards destination to ensure robustness and high throughput, at the same time *PHR* reduces the number of forwarding significantly to avoid network congestion problem. The actual forwarding nodes are also transmitting redundant data copies probabilistically in a controlled manner to ensure resiliency against lossy links and to deal with high frequent mobility changes. The existing body of research on opportunistic routing suggests that to select one forwarder to perform the packet forwarding process among other candidates those were selected before. Moreover, much of the current literature on opportunistic routing pays particular attention to the mechanism of the selection process of the forwarder node from a candidate set rather than the shape of the route toward the destination. However, with *PHR*, limited the number of nodes decide to forward the received packets. The forwarders have valid topological information about the destination (forwarders those close to the destination). Thus, with *PHR*, network provides a high level of robustness at much lower cost (retransmission cost) compared with traditional routing and other opportunistic routing techniques. Also, with *PHR*, node exploits multi directional forwarding,

by transmitting to multiple receivers (rebroadcast), at least one of which is more likely will receive the transmitted packet. Leveraging opportunistically forwarding also enables nodes to decide if it is going to forward or not at the moment. In contrast, conventional protocols cannot adapt at the millisecond time scale since its routing metrics are updated every 3s-10s.

6.2.1 Constraining Forwarding

There are two main mechanisms to constrain the forwarding decision. Firstly, to only forward messages if it is known that the message is closer to the destination than the previous hop. Secondly, given it have some redundancy afforded by the broadcast nature of the communication, forward only a random fraction of the messages at a particular node. The principle of the latter decision is as there are several receivers it can assume that not every one needs to forward.

6.2.2 Data and packet structure

To achieve constrained forwarding each node maintains the following information:

- *Message id list.* This is used in conjunction with the source node id to prevent forwarding of messages previously sent. Entries expire after a short period of time (here we fixed the timing to 5 seconds).
- *Known list.* This is a list of nodes that the node in question has received packets from. It is filled with source node ids and previous hop node ids from received packets. Again entries expire after a suitable interval (being fixed to 2 seconds). Within the known list we store the number of hops to that node. This information is used to determine whether a node is closer to the destination than the packet's previous node.
- *Packet Arrival Rate.* Each node monitors the number of packets arriving at its network interface and stores it. This is used to measure the busyness of the network and forms part of the probabilistic forwarding decision. Both the current real-time rate and the maximum rate observed are recorded.

As the protocol runs this information is built up in each node and as more is store, decisions can become more appropriate. To build these tables each packet needs to contain the following information:

- The node ids of the *source*, *destination* and *previous hop* of the message.
- The number of hops the message has travelled so far (*hops*), this is used as the best guess of the number of hops to the source node and stored in the known list.

- The number of hops to the destination at the previous node (*dest distance*).
- A flag (*known*) which indicates that a packet has passed through a node that knows about the destination node. The details are explained later.

In conventional routing protocols (such as AODV and DSR), packets are relayed from hop to another along a chain leading from the source to the packet's destination based on predetermined route by the source node. However, in opportunistic networks such as Delay Tolerant Network (DTN), the forwarding mechanism is different. Every node selects the best next forwarder among other nodes those in a predefined candidate set. Therefore, there is no predefined route from the source towards the destinations. In *PHR*, the forwarding decision is taken on a per-node, per-packet basis. No long term determination of paths is made, so the protocol is robust to the situations that would previously have caused path breakage.

6.2.3 Probabilistic forwarding

In trying to reduce the number of transmitted data packets, to avoid severe wireless channel collisions that drain the network resources and affect performance, each forwarder determine firstly considers forwarding the packet is appropriate and if so makes a probabilistic forwarding decision.

To determine if forwarding is appropriate we follow the following mechanism:

1. If the sender has no information about the destination, the first packet is transmitted with the known flag set to false. This is referred to as *unknown forwarding*. Packets are flooded through the network and the destination node is reached. Every intermediate node receives a copy of the packet, looking up the destination in the known list. If there, then subsequent rebroadcasts of the packet have the known flag set to true; this is referred to as a *known forwarding*. If not there, then the packet is forwarded with a false known flag. If the packet destination is in the sender's list, then the packet is sent with a known flag set to true.

2. If a packet is received with a true known flag and the destination is not in the known list, then the packet is not forwarded. The packet has moved from the set of node that know about the destination to the set that do not.

3. If a packet is received with a true known flag and the destination is in the known list, then the node compares the local distance with the previous hop distance (from the packet header) and performs a known forwarding if the

distance is shorter, otherwise the packet is dropped. The aim is only to forward packets that are getting closer to the destination.

4. If a packet is received by an intermediate node an a false known flag then it is checked against the known list as in 1 above.

Once a packet has been identified as suitable for known or unknown forwarding we now make a probabilistic decision as to whether to forward the packet. the packet will be forwarded with added jitter delay to avoid packet collisions at the MAC layer. Clearly if the network is highly loaded we don't wish to add significantly more load. Here we use the packet arrival rate (P) and forward with a probability of:

$$1 - \frac{P}{2P_{max}} \tag{6.1}$$

6.2.4 PHR Notation and algorithms

PHR employs a constrained flooding strategy using the available information in each node about the destination. Each node in the network maintains a list named (the *known list*) during operation. For each packet that arrives at a node, the packet's source node and previous hop node are added to the *known list*. Until a node receiving a packet has some knowledge of the closeness of the destination node the packet will be fully flooded over the network.

Using the following notation for packet $[i, S, D, H, k, p_h, p_d]$ and other data:

- Nodes: Source S, Destination D, Previous hop H.
- Packet identification i, unique to each packet.
- Known Flag k: means that the packet has traveled through a node that has the packet's destination in its known-list.
- Hops so far p_h: hops traveled since the packet left its source, used to update the sources entry in the known-list.
- Previous distance to destination p_d: The number of hops between the packet's previous hop node and the destination.
- $N.kl$ is the known list for node N, a map indexed by destination node.
- $N.rl$ is the received list for node N, a list of packet ids.
- $N.kl[D].d$ is the distance to destination for node D at N.

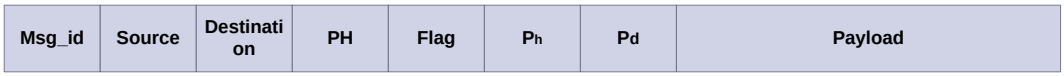

Figure 6.1: PHR packet structure

Fig 6.1 illustrates the *PHR* packet structure.

The protocol floods packet across the network until they reach nodes are close to the destination, then the full flood stops and packets forwarding focuses on enabling the packets to reach the destination node itself. The hops-so-far and previous-distance information is used to throttle the forwarding.

At the beginning of transmission, if the destination not in the source's *Known-list*, the flag k is set to False (meaning the packet has not been through a node that *knows* about the destination). Algorithms 1 and 2 shows initial node and forwarding decisions.

In algorithm 2 there are three reasons for discarding packets:

1. This is a previously received packet.
2. The node knows about this packet's destination, but the previous node was closer.
3. The known flag is set, but the node doesn't know about the destination. This means that the previous node knows more than the current node, so the packets destiny should be managed there.

Algorithm 1: Source node algorithm. The *pbroadcast* operation has two parameters, the first is the packet header and the second is True for a probabilistc broadcast of False for always broadcast.

input : new packet ready to send
if $D \in S.kl$ **then**
  $pbroadcast([i, S, D, null, \text{True}, 0, S.kl[D].p_d], \text{True})$
else
  $pbroadcast([i, S, D, S, \text{False}, 0, \infty], \text{True})$

6.3 PHR Walkthrough

6.3.1 Simple scenario

Here there is no previously known information in each of nodes, i.e., the *known-list* is empty. A packet is sent from sender S to destination D. Fig 6.2 shows the network topology.

Algorithm 2: Forwarding decision.

input: Receive packet: $[i, S, D, H, k, p_h, p_d]$

$me.kl[S] = (p_h, now)$
$me.kl[H] = (1, now)$
if $i \in N.rl$ **then**
  $discard^1$
else
  if D *in* $N.kl$ **then**
    if $N.k[D].dist \leq p.p_d)$ **then**
      $pbroadcast([i, S, D, N, \text{True}, p_h + 1, N.k[D], dist], \text{False})$
    else
      $discard^2$
  else
    if k **then**
      $discard^3$
    else
      $pbroadcast([i, S, D, N, \text{False}, p_h + 1, \infty], \text{False})$

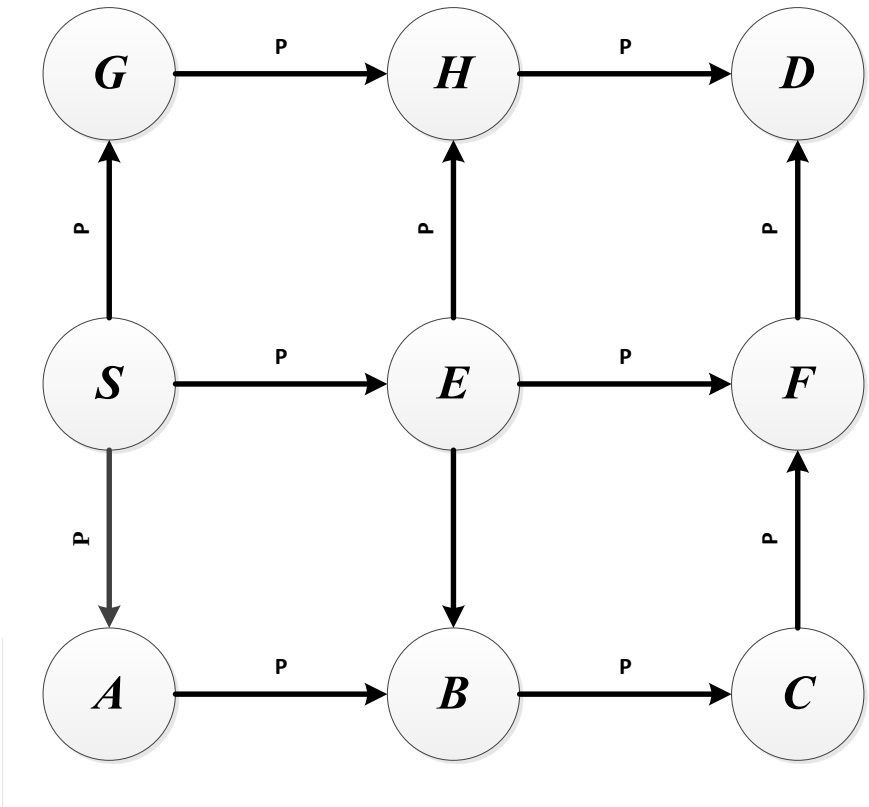

Figure 6.2: Scenario 1: D broadcast towards I

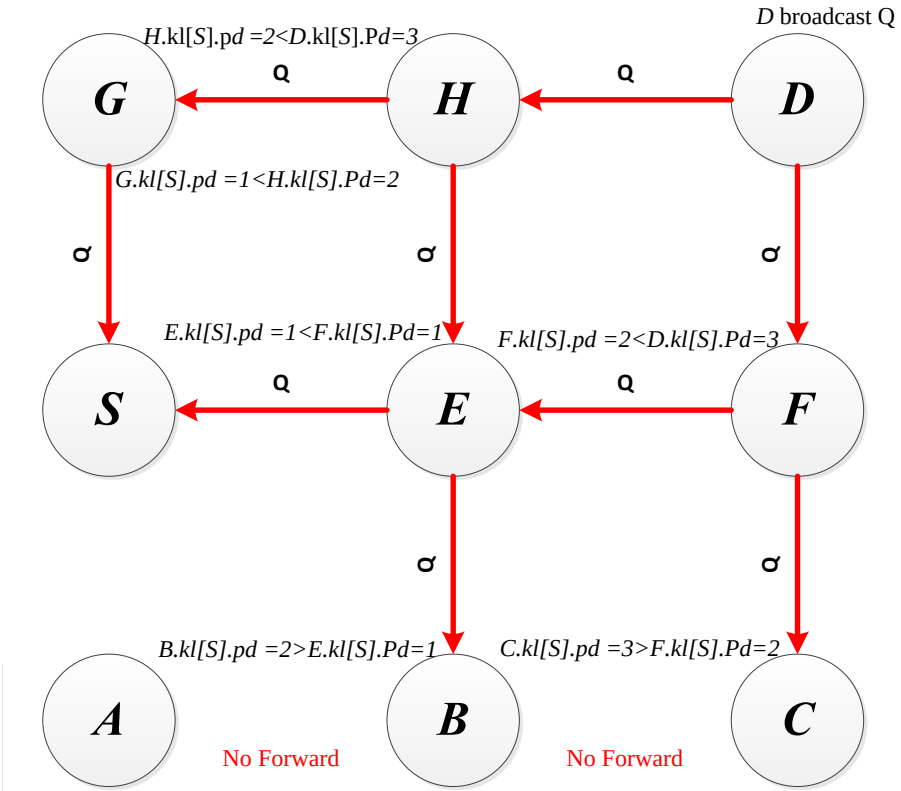

Figure 6.3: Scenario 2: I constrains flooding towards D

Algorithm 3: Probabilistic broadcast.

```
1 Function pbroadcast(P, a)
2     if a ∨ (P / 2P_max < uniform.rand(0, 1)) then
3         broadcast(P)
```

1. S sends a packet (P). D not in S's *Known-list*.
2. P is broadcast with header $[i_P, S, D, S, \text{False}, 0, \infty]$.
3. P is received by E, G and A. Each run the forwarding algorithm and as they do not know about D, they must broadcast as well.
4. This process continues on all the nodes in the topology with P being received by all nodes, one of which will be D.
5. At each node the local *known-list* is updated with information about S and the previous hop, using information from the packet header.

Fig 6.2 illustrates the first scenario and how S broadcasts P towards D.

There is then another packet Q from D to node S.

1. Since D received a packet from S before, it knows where S is.
2. Q's packet header is $[i_Q, D, S, D, \text{True}, 0, D.kl[S].p_d]$, True), here D.kl[S].p_d = 3 as its that number of hops to S.
3. All of D's neighbours F and H receive Q. Both add D to their *known-list*.
4. Both F and H compares the distance they know to S with the previous hop distance in Q's packet header. If less than or equal to they rebroadcast, which in this scenario they both do.
5. Q is then received by G, E, and C and also back at D. D simply discards the packet as it is a receive of a previous packet.
6. Nodes G and E are closer to S than the packet has been so far, so they forward. However, node C is not, so it does not forward.
7. Q progresses this way. It will reach nodes B and S. B will not forward. Q will never reach A.

Some points to note:

- The protocol provides multiple paths across the network. This provides redundancy in the case of packet loss. It also allows for flexibility if the nodes are mobile.
- The repeated packet transmissions are delayed by a small random amount to avoid broadcast clashes.

- To save further on the radio medium usage, a probabilistic approach is taken on the forwarding decision. If the load on the network is high (based on observed packet arrival rate), then the probability of transmission is reduced.
- The *known-list* entries expire after are certain period of time, as it is assumed that node mobility will mean that information is out-of-date.
- With a rapidly changing topology the *known-list* information may become out-of-date. Topology-change speed can be measured as the rate of new nodes entered into the *known-list*. By fixing the length of this list the age of the oldest entries will reduce as the rate of topology change increases.

6.3.2 Larger Scenarios

As illustrating above about how *PHR* performers in a simple scenario, it need to be testedted in a larger scale network topology. It has been designed and implemented a test case to assess *PHR*. The simulation is implemented using Python. This test case has two different urban maps includes (Artificial grid map and part of Leicester city centre map). The description of the *PHR* test phase is in the following sub-sections.

6.3.2.1 Grid Map Scenario

In this test case a grid map is considered. Each node in this grid topology is fixed and linked to its neighbours, having four neighbours utmost based on its position in the topology. In order to see the packet flow and validate the network with *PHR*, two nodes S and D have been selected to be as a sender and a receiver. S and D are far apart from each other, each node is placed on the opposite edge of the grid map. Fig. 6.4 shows nodes' locations on the grid scenario. The nodes with *PHR* apply same previously mentioned algorithms as follows.

- S sends packet and it is fully broadcasts in the network to reach it destination to D.
- The receiving nodes have changed their colour to be darker indicating they have received the first packet.
- Each node in the topology does not know where the D is (did not hear from D before) including S.
- Each node saves information about S and previous hop into its *Known-list* and rebroadcasts the packet again.

- The packet is progresses until reaches D. D has a red colour telling that it receives the first packet coming from S.

Fig. 6.5 shows the grid map scenario and how S broadcasts packet to received by all nodes, one of which will be D. There is then D sends a new packet to S.

- D knows where the S is and how far it is (D received a packet from S before).
- D broadcast a packet back to be received by its neighbours. This packet contains the distance towards S.
- All the D's neighbours add D to their local *known-list*.
- The neighbours compares the distance to S in their *known-list* with the previous hop distance in the packet's header. If it is less than or equal to they forward, which in this scenario, only the yellow nodes do.
- In case of the local distance to S is bigger than the previous hop distance, the packet will not be forwarded as have been performed by the black nodes in Fig. 6.6.
- The packet is progresses this way until reach S.

Fig. 6.6 illustrates the directed flooding from D to S based on *PHR* mechanism.

6.3.2.2 Dynamic Network Scenario

Two factors could cause network dynamics. First, the wireless propagation environment variation caused by channel fading. This factor has an impact on the link quality and reliability. Second, topology changes caused by the node's mobility. The latter affects the nodes' availability and consequently the route construction process or the forwarder selection process in case of opportunistic routing. *PHR* tackles these two issues by employing controlled broadcast.

In order to test and validate *PHR* in a more realistic and dynamic environment, the same scenario that used in the grid topology example has been used. The sender and the receiver nodes (S & D) are moving in the opposite dierctions. In the same way as the fixed nodes scenario, it has been used the same network model by choosing two nodes (sender and receiver) as follows.

- S broadcasts packet and the destination is the node D. D not in S's *known-list*.

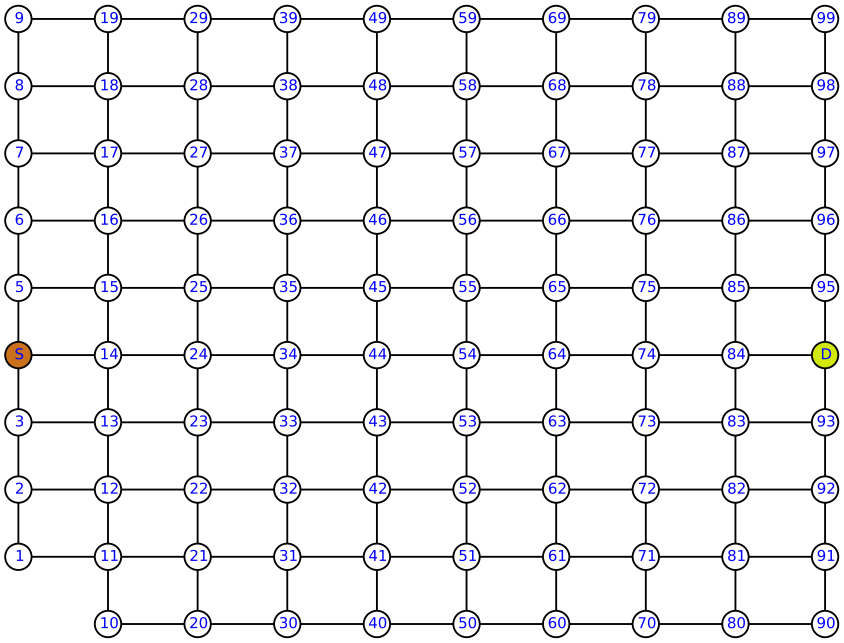

Figure 6.4: Scenario 1: S and D locations on the grid topology.

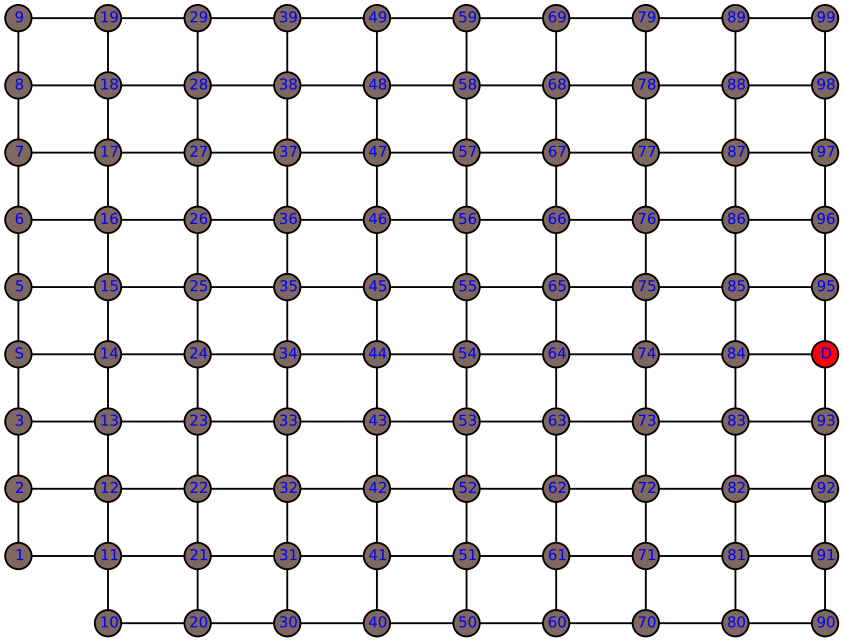

Figure 6.5: Scenario 1: S starts transmitting first packet towards D.

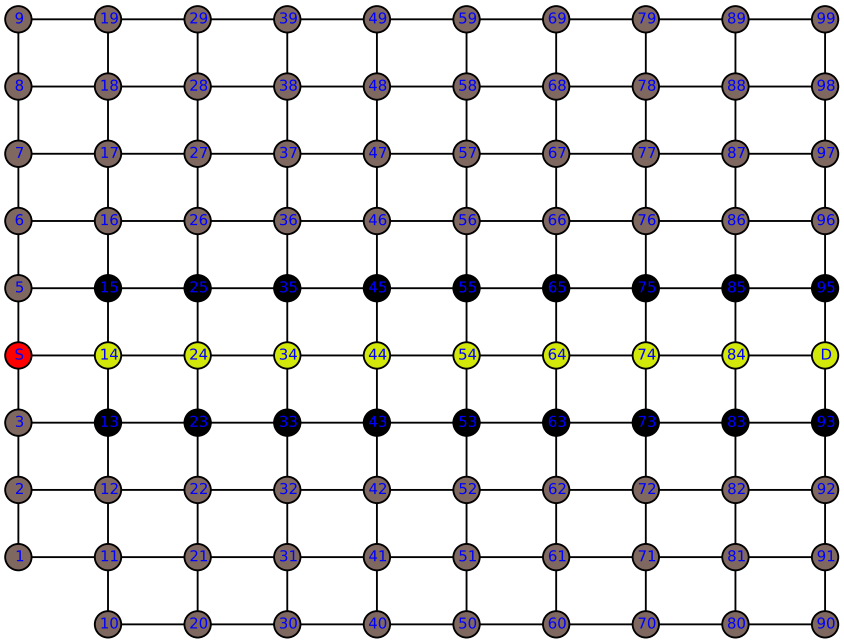

Figure 6.6: Scenario 1: D starts transmitting a packet towards S.

- Since none of the node in the topology have receives a message from D before, they must broadcast as well.
- Each node receives the message successfully, it update its local *known-list* with information about S using the message header.
- This process continue until the message received by all the nodes including D.

Fig 6.7 illustrates the dynamic network topology scenario and how S broadcasts packet to received by all nodes, one of which will be D. There is then D send another message to S.

- Now D knows where S is. The message contains the distance to S.
- Each node receives the message compares it is local distance to S with the previous hop. If it is less than or equal to, they rebroadcast the message, which is in this scenario the yellow node do. Fig 6.8 shows the send a new message journey from D to S
- Otherwise, the message will be discarded, which is the black nodes do.

All the previously described scenarios considers the network condition is load free. Therefore, the probability of forwarding is 1.0 (Always forward).

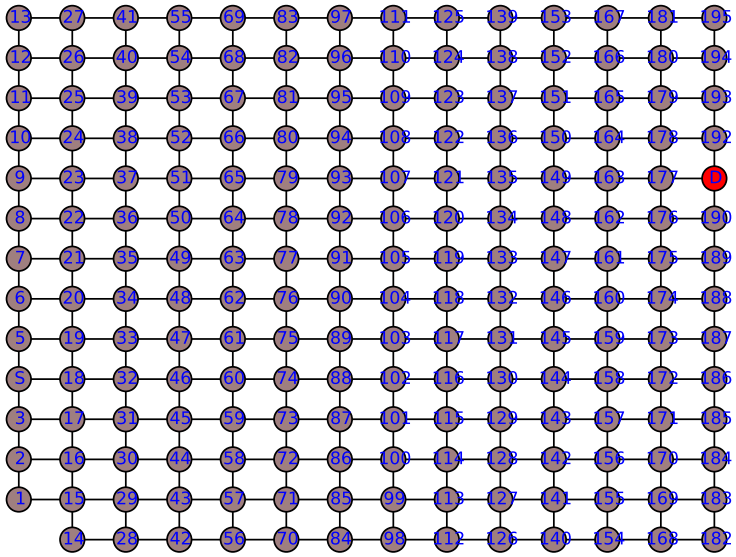

Figure 6.7: Scenario 2: starts transmitting first packet from S towards D.

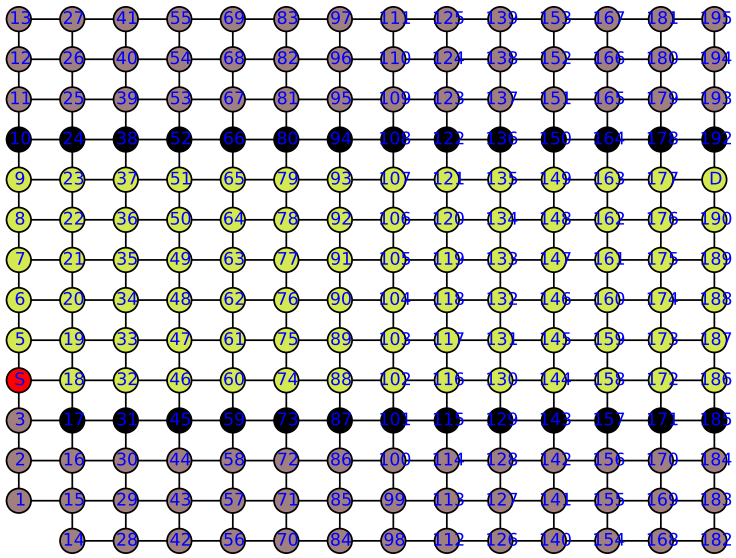

Figure 6.8: Scenario 2: D sent to S using PHR forwarding strategy.

6.4 PHR versus other Opportunistic Routing protocols

Most opportunistic routing (OR) protocols are primarily designed to route and forward traffic in DTNs. Nodes running OR protocols select the most appropriate next hop from amongst a set of other candidate nodes. This set is formed based on topological information exchanged between neighbour nodes using special types of messages (beaconing or hello messages), or depending on some specific information such as link quality or a history of encounters. Applications of DTNs do not expect real time performance, while most VANET applications are time-sensitive and require packets to be delivered within a specific timing, otherwise, the delivered information will be no longer valid.

The coordination overhead among the candidate nodes could lead to place more load on the network and drain the network resources. In highly dynamic environments such as VANETs, pre-selecting the next appropriate hop could not remain accurate enough as the next hop could move away from the transmitter or the destination.

In OR the majority of the forwarding protocols forward an individual packet down a single path from source to destination. In PHR, employing broadcast MAC transmission allows for a single packet to be replicated down multiple paths. The benefit of this is to ensure high delivery rate in highly dynamic environment.

However, the disadvantage is the large amount of message transmissions may lead to extensive the network resources and congest the network with unnecessary retransmission.

In order to reduce the number of unnecessary retransmission, *PHR* uses probabilistic forwarding to control the forwarding rate and minimise the overhead and reduces the number of retransmissions.

Nodes running *PHR* make their own forwarding decision based on the available topological information and the forwarding probability. However, most of the OR protocols, the next forwarder is determined by the current hop.

To cope with the highly dynamic environment, *PHR* is designed that when a packet reaches a node with knowledge of the destination, it is no longer forwarded by nodes that do not have this knowledge.

6.5 Summary

This chapter describes the proposed *PHR* forwarding strategy for high dynamic networks in details. *PHR* is a variant of the epidemic routing protocol for wire-

less networks that operates by throttle back the full flood to minimise resource usage while still attempting to achieve the best case forwarding of epidemic routing. *PHR* uses opportunistic forwarding technique to ensures robustness against unstable path between nodes in high dynamic environment, and exploits path diversity toward mobile destination to maximise the end-to-end PDR. *PHR* has exploit two techniques in its design. It explores opportunistic forwarding to transmit redundant packet copies along multi-path toward the destination. It is also form the path on the fly as the packet moves toward the destination. To control the overhead, especially in high network loads situations, *PHR* also uses probabilistic forwarding at each intermediate hop. In order to show how *PHR* works, a simplified scenario of six fixed nodes network topology with one sender and one receiver is considered. Then moved to more advance scenarios by conducting the grid topology model with non-mobile nodes, this topology is implemented in Python. In addition, it has been used a simple network traffic model, the sender sends one packet towards the destination, then the destination sends a new message back to the sender. The second scenario is the part of Leicester city centre map scenario, were the nodes are mobile, it has been employed the same network traffic model that is used in the first scenario to illustrates how *PHR* works in mobile scenario.

In the next chapter, *PHR* is implemented and integrated to the Network Simulator *NS2* to be tested in more realistic network environment.

Chapter 7

PHR Implementation

The proposed Previous-Hop Routing scheme was described in details in chapter 6. This chapter provides the *PHR* source code. *PHR* is implemented in Network Simulator *NS2* environment that is previously described in chapter 4. Many reasons after choosing *NS2*, the capability of providing all the network interfaces in each node, well known and trusted network simulator in the research community and the ability of emulating the impact of obstacles on wireless signal. The following sections describe the source code of *PHR* forwarding strategy.

7.1 Sending Packet at Source Node

When a packet arrives at the network layer and the source ID in the packet header matches the current node ID, it means the current nodes is the source node. Prior the packet transmission, the source node checks if it know where the destination is, to include the distance to the destination in the packet header and set the known flag. The packet is prepared to be transmitted by setting all the packet header fields in both cases (*known and unknown forwarding*).

7.2 Receiving Data Packet at Intermediate Node

This part of the code handles the receiving packet event at intermediate nodes. When a packet is successfully received on the network interface, the *PAR* counter is increased. Every second the accumulative number of receiving packet stores in a list to be used later on in the forwarding probability calculation as Packet Arrival Rate (*PAR*). Then the counter is reset to re-calculate *PAR* to adjust the forwarding probability value according to *PAR*. The calculation of *PAR* is done in recv_data(Packet * p) function.

After handling *PAR* value, the packet should be checked if it received before to be discarded as this could cause a loop problem. Prior discarding the same receiving packet, it compares if the packet coming from a new neighbour, if so, the previous hop (the new neighbour) is added in the *known-list*. Otherwise, it resets the expire timer of the previous hop in the *known-list*, then discard the packet. This technique helps nodes discover their surroundings. The packet dropping process is implemented in recv_data(Packet * p) function.

The calculation of forwarding probability based on *PAR* is implemented as shown line 2 in algorithm 3 in chapter 6.

7.2.1 Unknown Forwarding

Subsequent receiving packet that not received before and calculation of the forwarding probability, the packet is ready for further processing. The packet's destination will be looked up in the local *known-list*. If it is not there and the known flag is not set, the packet will broadcast probabilistically with a false known flag. Otherwise, the packet will be discarded.

7.2.2 Known Forwarding

In case of the destination in the local *known-list* and the distance to the destination is less than or equal to the distance from previous hop to the desired destination, the packet flag is set to be true and based on the previously calculated probability, the forwarding decision will be taken.

7.3 Receiving Packet at Destination Node

Upon arriving a packet to its desired destination node, the packet header is decapsulated and the information about the sender and previous hop will be updated in the destination's *known-list* and then pass the packet to the upper layers. The packet handling process at its destination is done in recv_data(Packet * p) function.

7.4 Functions for Data Structure Management

As mentioned earlier in chapter 6, *PHR* need to maintains a number of data structure lists, i.e. *Message id list*, *known-list* and *PAR list*. Each list requires

three functions, to insert into, read from and to expire entries. The management function for each data structure as follows:-

- *Message id list*:
  1. Insert into *Message id list* in app_pkt_insert(nsaddr_t id, u_int32_t bid) function.
  2. Read from *Message id list* in app_pkt_lookup(nsaddr_t id, u_int32_t bid) function.
  3. Expire *Message id list* entries in app_pkt_purge() function.
- *PAR list*:
  1. Insert into *PAR list* in sum_insert(double PAR) function.
  2. Read from *PAR list* in sum() function.
  3. Expire *PAR list* entries in sum_purge() function.
- *known-list*:
  1. Insert into *known-list* in known_insert (nsaddr_t src, u_int8_t hopcount) function.
  2. Read from *known-list* in known_lookup(nsaddr_t dst) function.
  3. Expire *known-list* entries in known_purge() function.

7.5 Analysis of PHR Trace Format

At the end of each simulation, *NS2* dumps a trace file contains all the events that occurred during the simulation. in order to extracts the results and analyse them, it is essential to understands the format of *PHR* trace file.

Each field in the trace line describes a propriety or event in the packet journey from its source until reach the destination or getting dropped. The following examples of a trace file illustrate three main events, i.e. sends, forwards and receives

- Send example at application layer in a source node:-

```
1 s -t 9.892139000 -Hs 35 -Hd -2 -Ni 35 -Nx 311.21 -Ny 76.58 -Nz
    0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is
    35.1 -Id 52.1 -It cbr -Il 100 -If 0 -Ii 1 -Iv 32 -Ph 0 -Phsf 0
    -Pn cbr -Pi 0 -Pf 0 -Po 0
```

- Forward example at routing layer:-

Table 7.1: Explanation of PHR trace file format. [RTR=routing layer, AGT=application layer, IFQ=Interface queue, MAC=mac layer CBR=constant bit rate pkt, s =SEND, r=RECEIVED, d=DROPPED, f=FORWARD]

Field No.	Action	Value
1	The current event	s, r, d, f
2	Time at which event occurred	-t
4	Node at which event occurred	-Hs
6	Next hop address	-Hd (-1 for broadcast)
18	Layer at which the event occur	AGT , RTR, IFQ, MAC
20	Drop reason(if drop performed)	LOOP, CLOSER, DKNOW
34	Packet Type	CBR ,TCP, RTS, ARP
44	Previous hop id	[-Ph 103, 72 in the examples]
46	Number of hops so far	-Phsf
50	Sequence No.	- Pi
52	No. of forwarding times	-Pf

```
1 f -t 9.895572109 -Hs 103 -Hd -1 -Ni 103 -Nx 375.00 -Ny 75.00 -Nz
    0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 23
    -Mt 0 -Is 35.1 -Id -1.1 -It cbr -Il 120 -If 0 -Ii 1 -Iv 31 -Ph
    103 -Phsf 2 -Pn cbr -Pi 0 -Pf 1 -Po 0
```

- Receive example at application layer in a destination node:-

```
1 r -t 8.903753480 -Hs 35 -Hd -1 -Ni 35 -Nx 311.21 -Ny 76.58 -Nz
    0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md ffffffff -Ms 5b
    -Mt 0 -Is 52.0 -Id 35.0 -It cbr -Il 120 -If 0 -Ii 0 -Iv 29 -Ph
    35 -Phsf 3 -Pn cbr -Pi 0 -Pf 3 -Po 0
```

- Drop example at routing layer:-

```
1 d -t 10.908405779 -Hs 75 -Hd -1 -Ni 75 -Nx 144.62 -Ny 334.38 -Nz
    0.00 -Ne -1.000000 -Nl RTR -Nw DKNOW -Ma 0 -Md ffffffff -Ms 3d
    -Mt 0 -Is 52.2 -Id -1.2 -It cbr -Il 120 -If 0 -Ii 2 -Iv 30 -Ph
    75 -Phsf 2 -Pn cbr -Pi 0 -Pf 2 -Po 0
```

Table 7.1 explains the main trace format fields.

7.5.1 Packet Drops Types

As mentioned in chapter 6 section 6.2.4, *PHR* packet has three reasons to be dropped by the receiving node as follows:-

- Previously received packet: The function that handle this event is in Listing C.3 line 16.

Listing 7.1: The three drop reasons definition in *cmu-trace.h*

```
1 #define DROP_RTR_ROUTE_LOOP  "LOOP"    //Loop.
2 #define DROP_PHR_PH_CLOSER   "CLOSER" //PH is closer to D.
3 #define DROP_PHR_DKNOW       "DKNOW"  //Know flag set and don't know
    about D.
```

- Previous hop is closer to destination: this event is handles in Listing C.6 line 36.
- The flag is set and the destination not in the node's *known-list*: Drop event function is in Listing C.5 line 40.

These three drop events are traced in using use *cmu-trace* objects, which is the output of compiling *cmu-trace.h* and *cmu-trace.cc*. Listing 7.1 shows the definition of these three drops reasons.

7.5.2 Calculating PDR

Packet Delivery Ratio (PDR) is the ratio of the successfully received packets at the application layer to the number of packets that have been sent at the application layer as well. To calculate the PDR of the simulation experiments, a Python code is used. This code reads a trace file that is dumped by *NS2* simulation and calculate the number of sent packets at the application layer to those who received at the same layer.

7.5.3 Calculating DBL

Drop-Burst Length (DBL) measures the probability of drop a consecutive number of packets in each flow. Two Python code is used to calculate the DBL of each *NS2* trace file. First file reads the trace file and list out the packet status in a consecutive order, whether it is delivered or dropped. This list of packets' destiny is fed into the second code, which is calculating the accumulative of dropped packets (drop length), then finds the probability of each drop length.

7.5.4 Calculating C2C latency

The definition of packet delay (latency) is the time that is taken by a packet to travel across a network from the application layer at the source node until being delivered at the same layer at the destination. Delay for each packet can be calculated as shows in equation 7.1, where St is a packet send time and Rt is

denote to a packet received time:

$$p_delay = Dt - St \tag{7.1}$$

7.6 Implement different velocity scenarios

Two mobility scenarios are implemented and used on the selected urban maps to evaluate *PHR* i.e. high and low dynamic scenarios. Both scenarios have the identical mobility model, the only key difference between them is the velocity rate. In this thesis, a Python-based script is written to change the velocity of all the nodes according to the desired model.

7.7 Network Traffic Implementation

In order to evaluate *PHR* in an urban environment, a network traffic that represents a VANET applications should be formed. In chapter 2 VANET applications are classified based on the network traffic requirements. The selected conventional routing protocols are evaluated using safety-critical application traffic as has been described In chapter 4. To generate an *NS2* network traffic file that can be used in the simulation, a Python-based script is implemented to generate random peers connections.

7.8 Summary

This chapter presents the implementation of *PHR* in the Network Simulator (*NS2*). The code is described briefly at each step of the packet forwarding process and how node acts as being in different roles (source, destination or forwarder). Additionally, the source code of the functions that manage (insert into, read from and expire entries) *PHR* data structure is listed. To give a better understanding how the results are extracted from simulation trace file, a quick trace format illustration is carried out, showing the main features and how the drop events are implemented. Finally, the source code of the employed metrics calculation and the pre-simulation files (Velocity changing rate and Network traffic generator scripts) are presented in brief way.

The evaluation of *PHR* performance against the selected protocols is presented in the following chapter.

Chapter 8

Evaluation and Discussion

The proposed Previous-Hop Routing scheme was described in details in chapter 6. This chapter illustrates the simulation experiments that carried out to asses the network performance with the proposed *PHR* in highly dynamic topology. In order to make the network environment more challenging, we have slightly changed the simulation configurations that have been used in chapter 4. The network performance is analysed and evaluated with *PHR* against the selected MANET routing protocols using the previously mentioned performance metrics in chapter 5. This chapter also, give a better understanding of how the *PHR* adapts under various network loads and tunes the forwarding probability value to suit the network load. It is depicting the impact of forwarding probability for known and unknown destinations on the network performance.

8.1 Simulation Experiments

In order to evaluate and assess *PHR*, we employ same simulation configuration from previous experiments in chapter 4[3]. We employ a safety critical network traffic model, which required a minimum 10 messages to be sent every second with small packet size using connection-less transport protocol according to U.S Department of Transportation report [25]. Each flow has random duration (5 s to 20 s) at 10 pkt/s, on each map with the total number of calls varying from 200 (low) to 1000 (high). For a communication model, we employ 801.11p as MAC layer with RTS/CTS (for point-to-point communication). The network performance is evaluated by employing two velocity profiles i.e. high velocity and low velocity. Nodes' velocity in the highly dynamic scenario could reach up to 30 m/s, while in lower dynamic scenario, the velocity could be up to 10 m/s. Both models are simulated on the previously configured maps (London congestion zone, Leicester city centre and an artificial Manhattan-style mobility model). Additionally, the net-

work traffic pattern consists of bi-directional calls, to better represent real network application traffic. Moreover, the proposed Drop-Burst Length (DBL)[2] along side traditional assessment metrics (Packet Delivery Ratio and Delay) are employed to evaluate network performance with *PHR* against conventional MANET routing protocols (AODV, OLSR and GPSR).

8.2 Probabilistic Forwarding Rate

PHR employs a probalistic forwarding mechanism based on the apparent network load ($\frac{P}{P_{max}}$). This means that under high load the probability of a message being forwarded will reduce. Fig 8.1 and 8.2 show some results from varying the rate of this reduction. It has been observed that the best PDR performance is achieved when this rate is set to $0.5 - 0.6$, which is used (see eqn 6.1 in chapter 6).

8.3 Results and Discussion

The analysis of the network performance is carried out using DBL, PDR, C2C latency. Simulations were undertaken with increasing load, i.e. number of traffic flows (calls). Simulation runs were performed five times with the same random source and destination selections for each flow on each run, but different seeds for the propagation model.

- Fig. 8.5, 8.6 and 8.7 show that OLSR achieves the shortest C2C latency because it pre-computes all the routes to the all nodes in the topology.
- Fig. 8.5, 8.6 and 8.7 also shows that PHR achieves the second best C2C latency because PHR does not need spend time to determine the path to the the destination as with AODV and OLSR. There is longer delay in PHR is due to the added jitter delay on each packet forwarding process to avoid packet collisions.
- With AODV, packets take longer to be delivered under different network load on all maps. These longer delays are due to its route initialisation mechanism, it takes time to set-up a route to destination (sending a RREQ and waiting for a RREP). This leads to packets being queued and dropped before transmission and the probability of dropping consecutive packets with AODV increases along the simulation.
- OLSR provide routes immediately to destinations, and the nodes with GPSR already forward to the closest neighbour, this can give an advantage for those

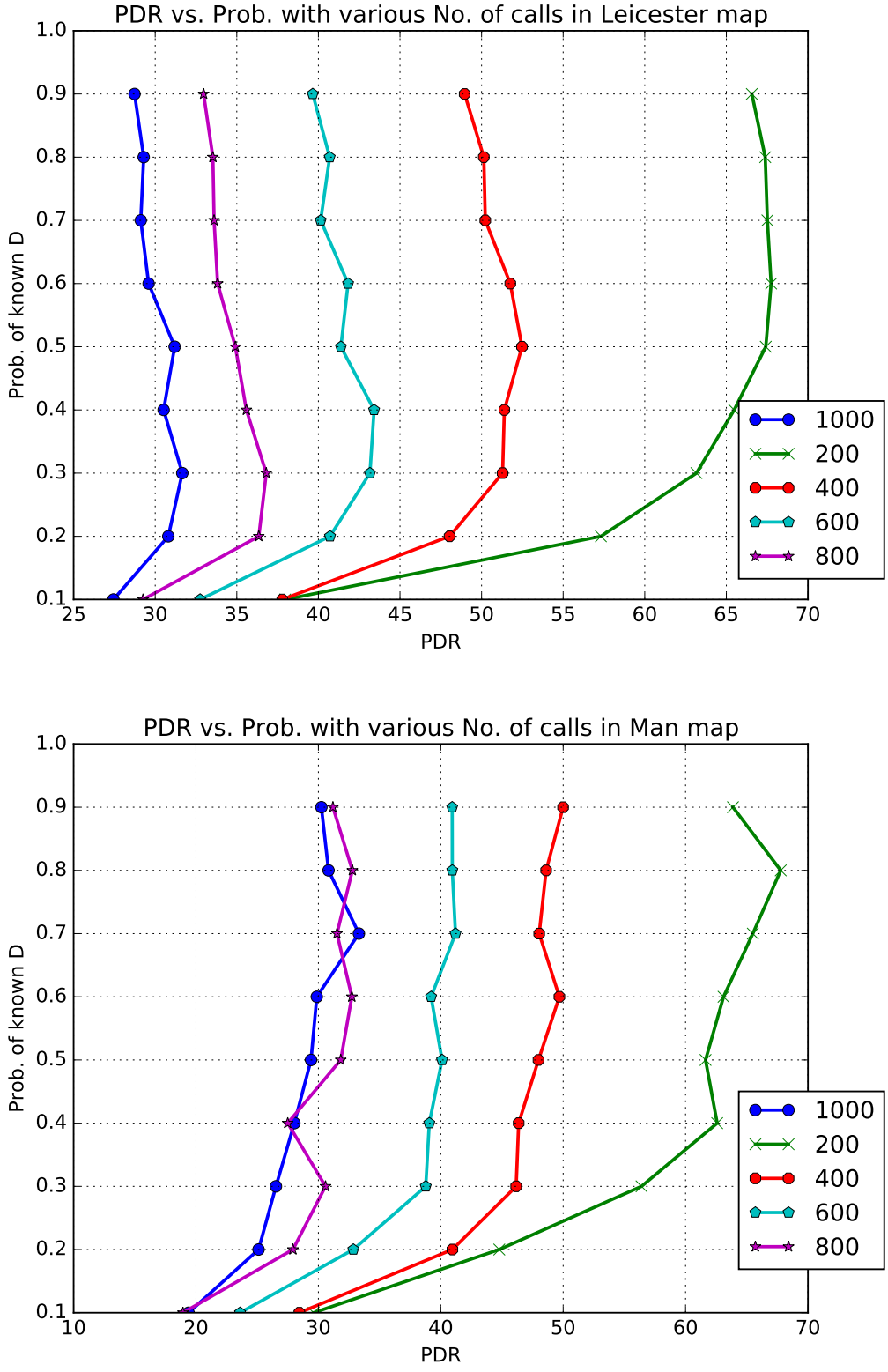

Figure 8.1: PDR against forwarding probability in case destination is known.

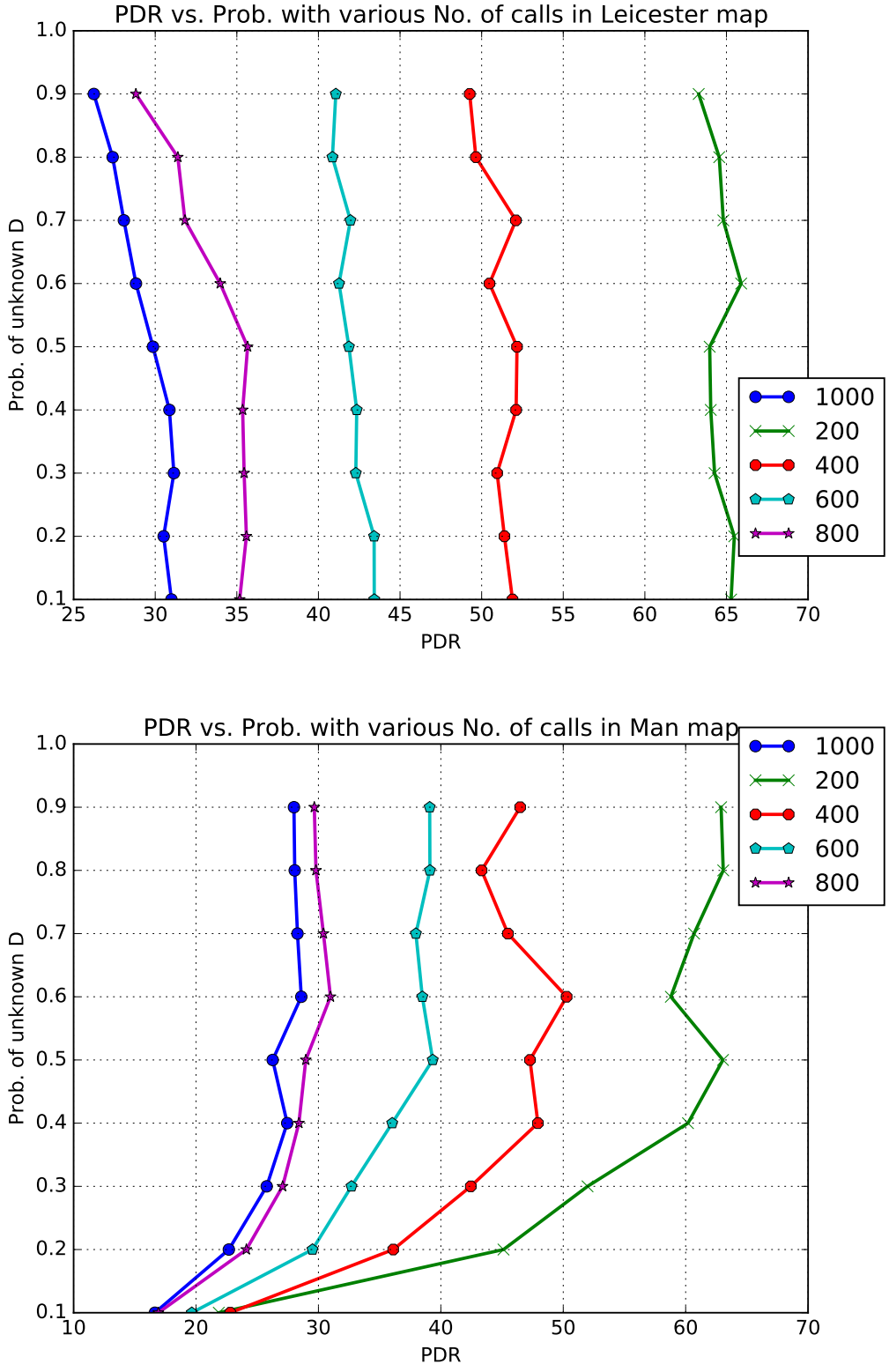

Figure 8.2: PDR against forwarding probability in case destination is unknown.

protocols over AODV in terms of delay and DBL. However, they still employ a unicast link between source and destination, this leads to increase the DBL along the simulation in contrary to PHR (see Fig. 8.3 & 8.4).

- With PHR, long DBL is less likely to occur comparing with the other protocols (Fig. 8.4) because nodes transmitting redundant copies of each packet towards the destination, despite the fact that PHR has high probability of short DBL under low and high network loads (lower than OLSR) compare to AODV and GPSR see Fig. 8.3. This satisfies VANET applications needs mentioned in chapter 2.

- With GPSR, the network shows a worse performance in term of DBL. The probability of dropping the entire flow is much higher compared with the other protocols due to the rapid position changing, although it performs much better in term of latency. This leads routes to getting out dated quickly (see Fig. 8.4).

- PHR outperforms OLSR, AODV and GPSR in terms of DBL and PDR under low, medium and high network load (more support for VANET application performance). However, as load increases, the performance reduces as the drop ratio increases due to the network getting busy as a consequences of transmitting large number of packets' copies.

- Using PDR, we observe that *PHR* out performs the other protocols under various network loads in high dynamic model as illustrated in Fig. 8.8,8.9 and 8.10 although, the performance drops as the network load increases. PDR performance is affected by the load that placed on the network and topology dynamic rate. However, this impact varies depending on the routing strategy employed.

- In low mobility scenarios, *PHR* is outperformed by OLSR on Leicester and London congestion zone maps (see 8.9 & 8.10). As the topology changing rate decreases the selected protocols performing more efficient (delivers a high number of packets to the destination) in terms of PDR as exploit the MAC retransmission technique to delivers packets to the next hop in contrary with *PHR* that employ the MAC broadcasts technique. However, on Manhattan-style model *PHR* shows a better performance compared with OLSR, AODV and GPSR under low network load (see 8.8).

- With PHR, the impact of placing more loads on the network is low compared with the other protocols, especially on the MAN map (see Fig. 8.8). This

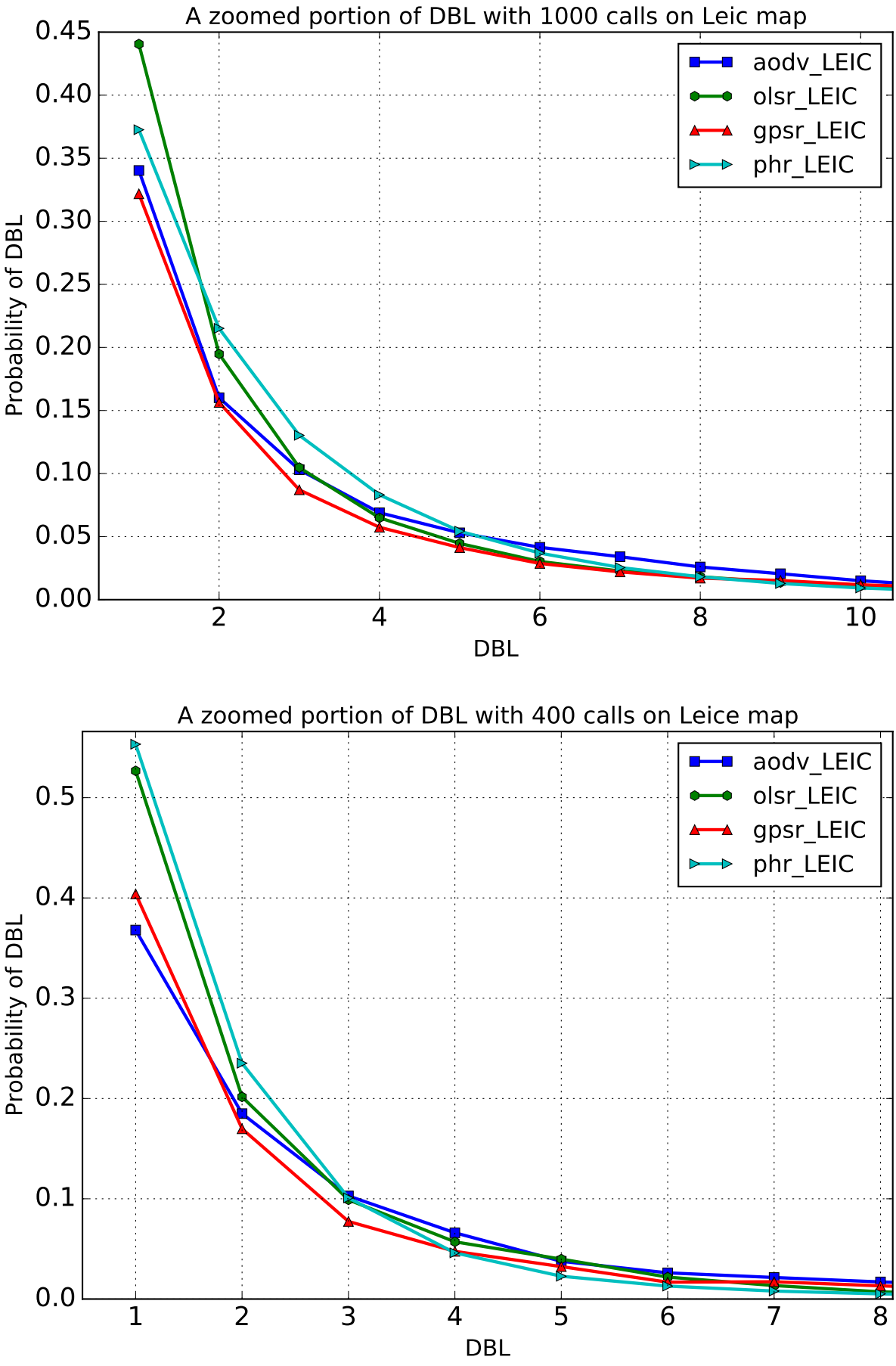

Figure 8.3: Short Drop Burst of AODV, OLSR, GPSR and PHR with various number of connections (A zoomed portion).

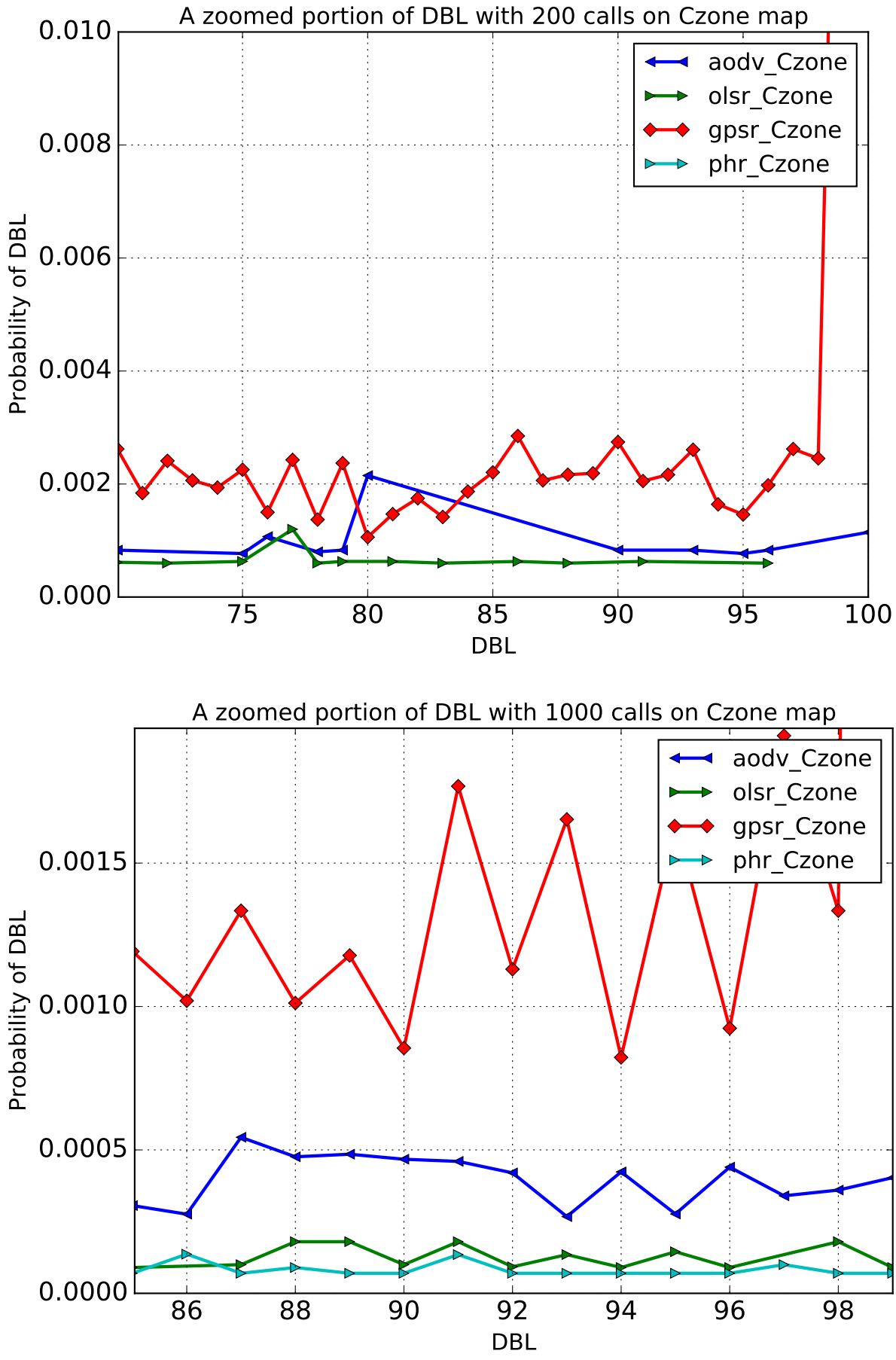

Figure 8.4: Long Drop Burst of AODV, OLSR, GPSR and PHR with various number of connections (A zoomed portion).

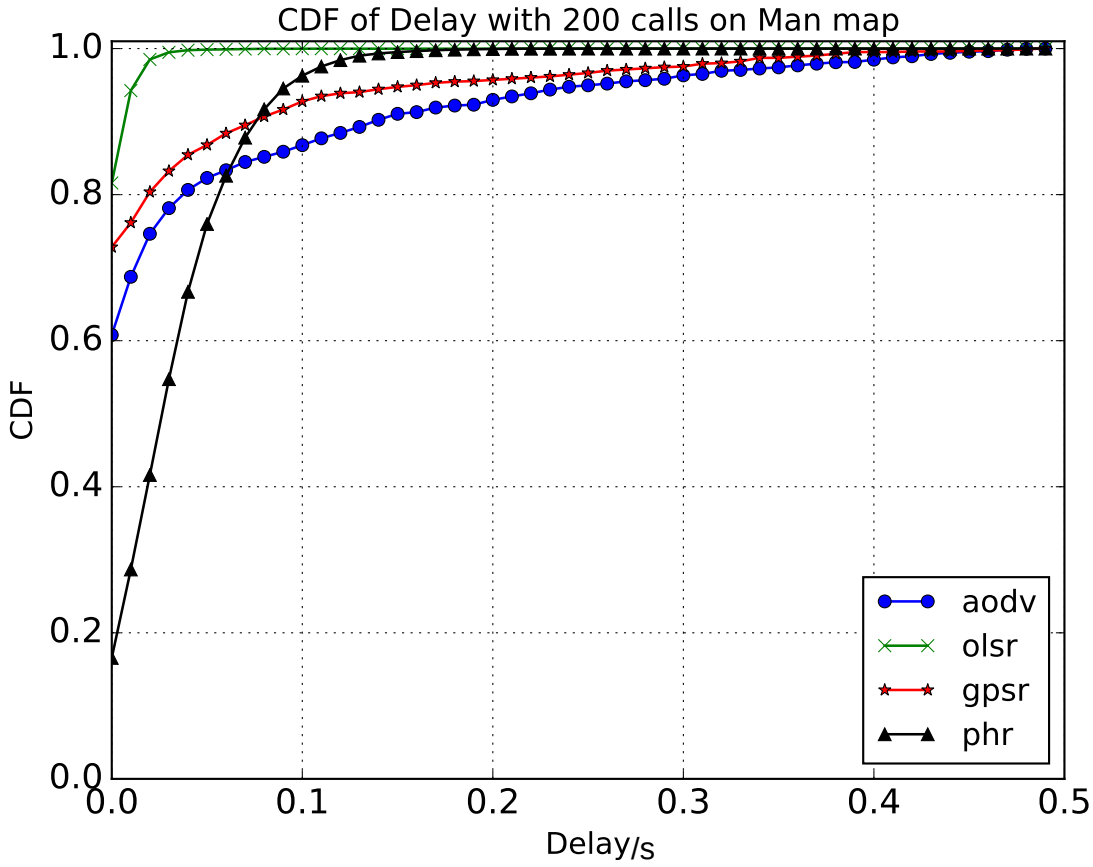

Figure 8.5: CDF of delay for the PHR and the selected protocols under low and high network loads on Manhattan mobility model.

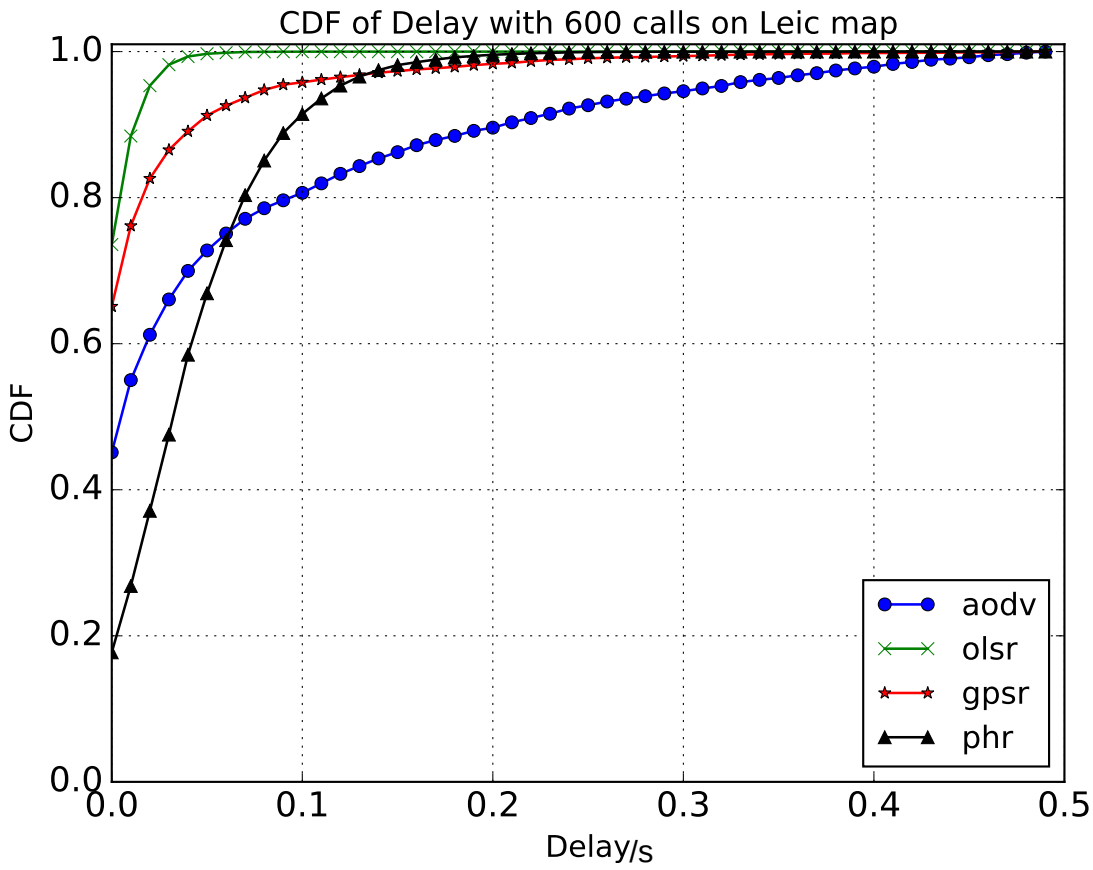

Figure 8.6: CDF of delay for the PHR and the selected protocols under low and high network loads on part of Leicester city centre map.

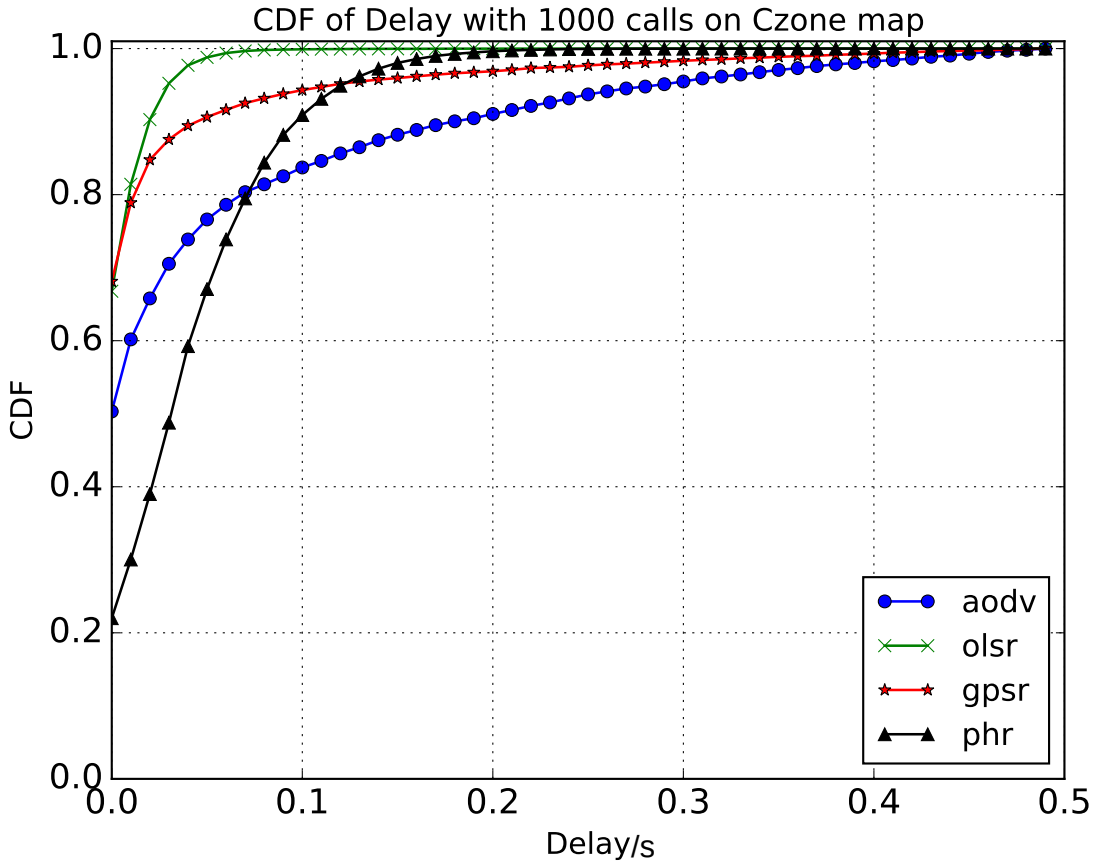

Figure 8.7: CDF of delay for the PHR and the selected protocols under low and high network loads on part of London congestion zone map.

is because, the probabilistic forwarding decision with *PHR* is taken depending on the load adaptive technique by measuring perceived *Packet Arrival Rate*(P).

- We observe that the network performance (PDR) varies on different maps, although the simulation scenarios are identical. The variation is due to the properties of the road networks under study, with sometimes roads, and thus vehicles, being closer to each other in places.

- Despite this *PHR* shows a more stable performance on the all selected maps, as *PHR* employs probabilistic forwarding to provide a better congestion control under load (see Fig. 8.8,8.9 and 8.10).

- AODV outperforms OLSR with low network load (200 calls) as shown in Fig. 8.9 and 8.10 in high dynamic scenario. However, the performance is drops as the load grows. This drop is due to the inability to cope with the increased number of path building messages.

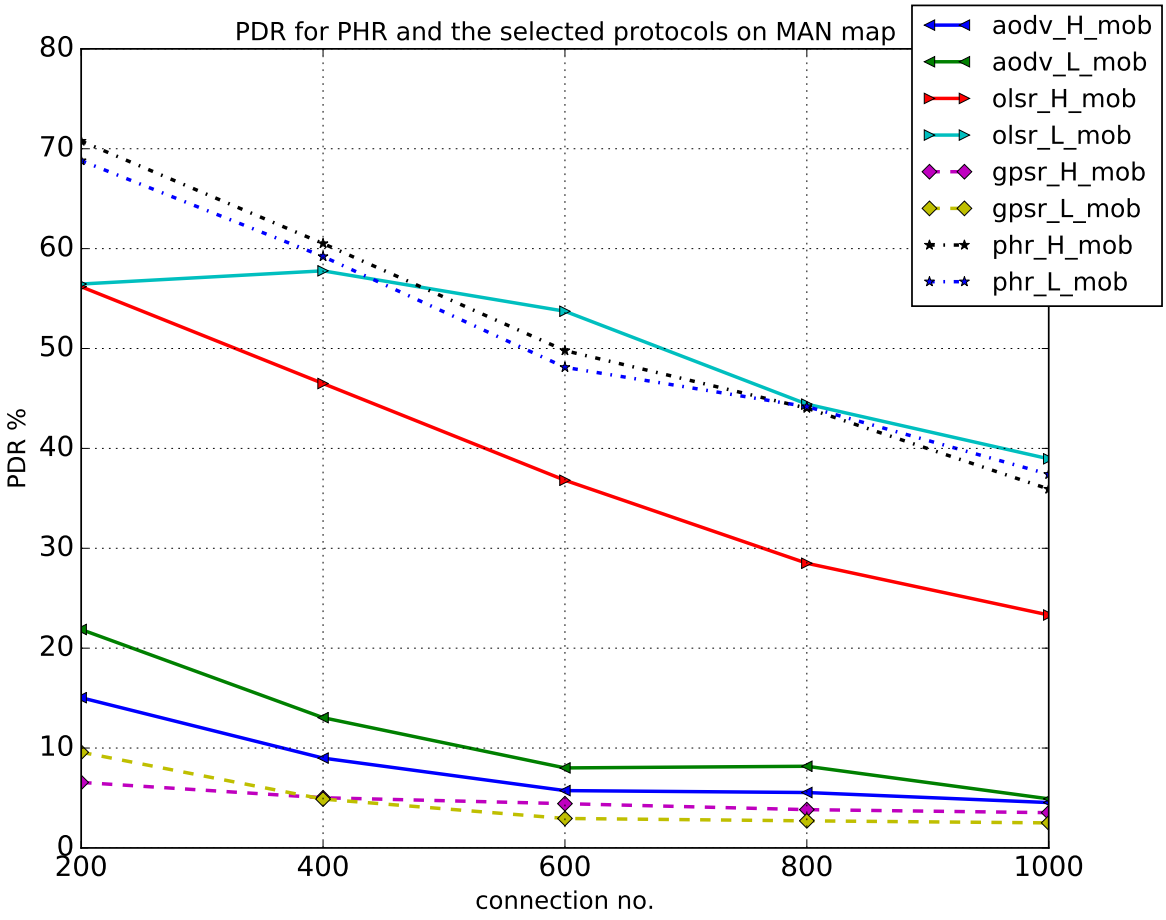

Figure 8.8: Packet Delivery Ratio (PDR) of PHR and the selected protocols on the Man map with low and high mobility profiles.

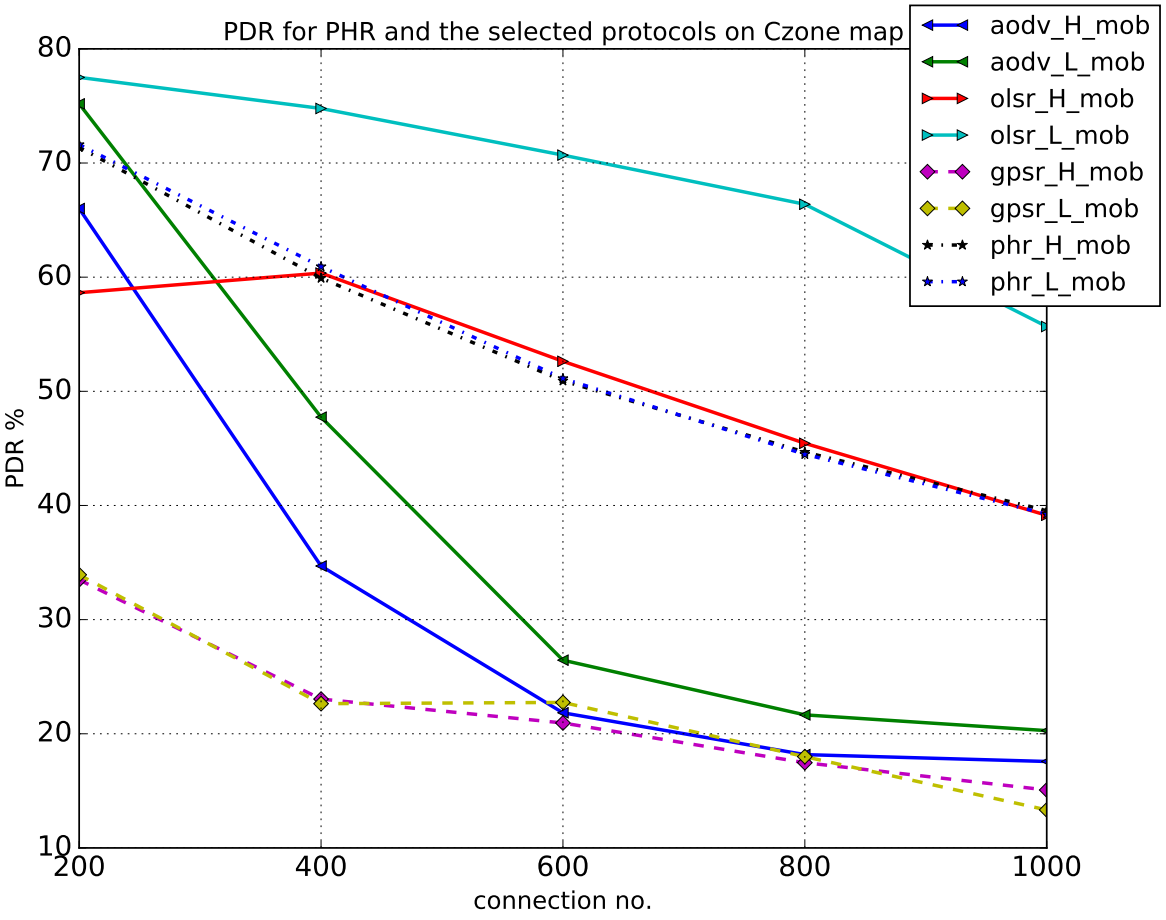

Figure 8.9: Packet Delivery Ratio (PDR) of PHR and the selected protocols on the part of Czone map with low and high mobility profiles.

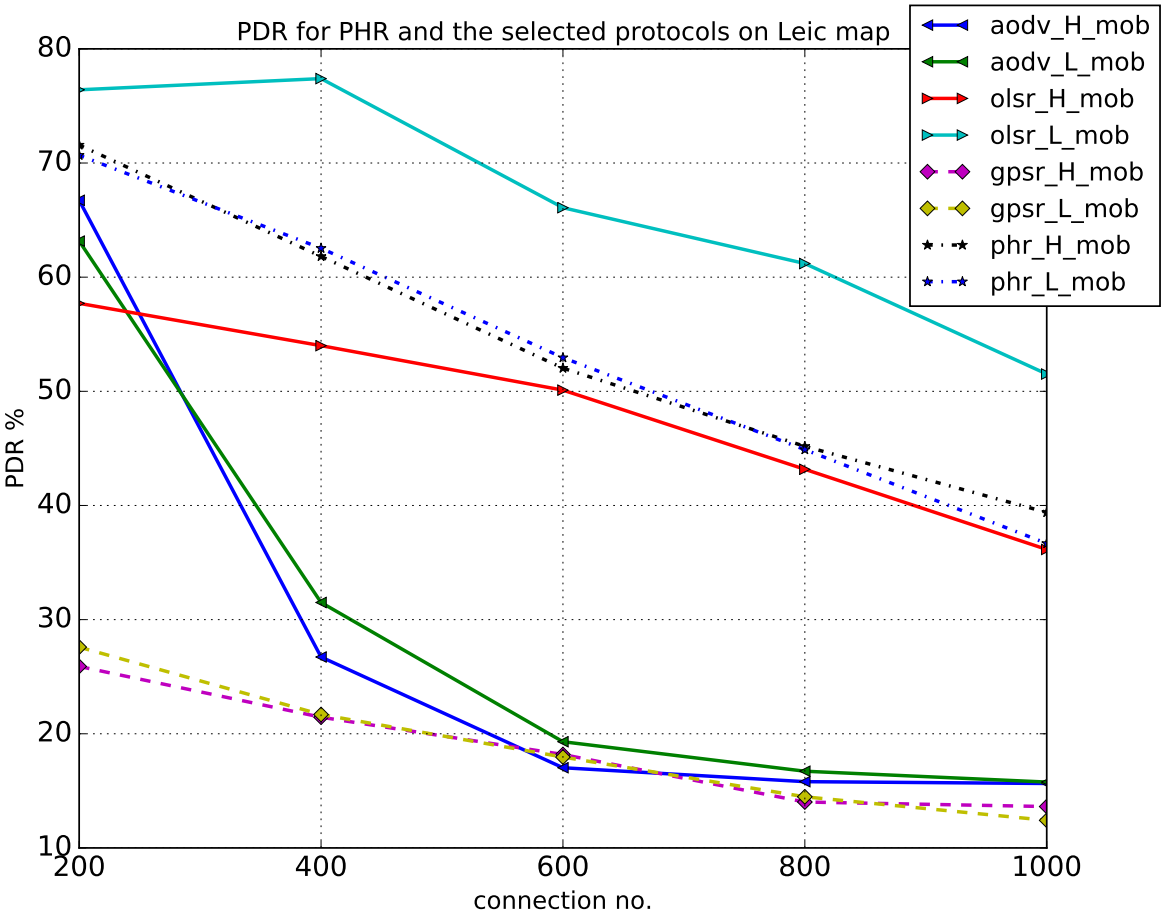

Figure 8.10: Packet Delivery Ratio (PDR) of PHR and the selected protocols on the part of Leicester city centre map with low and high mobility profiles.

Chapter 9

Conclusions

This chapter summarises the work that has been done so far. Furthermore, the direction of the further research is presented.

9.1 Summary

This thesis includes 9 chapters as follows.

- Chapter 1 outlined the research area that interests this research (Routing in highly dynamic network environment such as VANETs).
- Chapter 2 identified VANET applications classification and their related requirements based on different perspectives (packet generation process and network interface).
- The literature on Mobile Ad-hoc Networks (MANETs) routing protocols (conventional protocols), opportunistic routing protocols, and the key differences between these two routing strategies are provided in chapter 3.
- Chapter 4 provides the description of the proposed simulation model to assess the conventional routing protocols in VANET urban environments. Additionally, Drop-Burst Length (DBL) measurement is introduced.
- Chapter 5 provides the analysis and the evaluation of the network performance with the selected conventional routing protocols (AODV, OLSR and GPSR) in the configured VANET urban environment.
- Previous Hop Routing *(PHR)* is introduced in chapter 6, which is a new routing and forwarding protocol for highly dynamic networks such as VANETs.
- The implementation of *PHR* in the network simulator *NS2* is illustrated and explained in chapter 7.

- The evaluation of the network performance with *PHR* against the selected conventional protocols is done in chapter 8. The evaluation is achieved by employing the proposed metric (DBL) along side traditional metrics (Packet Delivery Ratio and Delay).

- This chapter draws the conclusions of this research.

9.2 Thesis Contributions

An up to date source-destination path may be shortly lived in highly dynamic wireless networks such as VANET. This frequently network topology changes makes the routing task challenging. Conventional routing protocols (MANET routing protocols) are widely used in VANETs. They satisfy VANET applications requirement to a certain extent. The potential solution for such high mobile network is to rebroadcast (flood) each message all over the network. The main disadvantage of the flooding approach is the extensive use of the network resources by placing load on the network by unnecessary retransmission. In order to solve the routing problem in highly dynamic topology, This thesis brings an alternative routing/forwarding paradigm to ad-hoc networking, namely one that employs the value of broadcast networks and inspired by work in opportunistic networking. The protocol is tested in a number of scenarios against existing routing protocols. It has been observed that *PHR* shows some increased performance under high loads and high mobility.

For more in depth details, this research proposes the following.

Drop-Burst Length (DBL): a novel metric to measure the probability of drop a consecutive number of packets in each connection. DBL gives a better indication as to the effects of performance the QoS of real-time traffic.

Previous-Hop Routing(PHR): a new opportunistic based routing and forwarding protocol. *PHR* exploits the broadcast nature of radio communication whereby several receivers, are in range to receive a message from a transmitter. *PHR* has a key difference compared to the existing opportunistic proposals in Delay Tolerant Networks (DTN). *PHR* aim to perform in real-time network traffic such as safety-related VANET information. In addition, OR protocols exploit the limited knowledge of the network topology in the selection of the next hop from a candidate set. However, with *PHR*, each node makes the forwarding decision itself, based on the available information (source node, destination node, previous hop) of the current message. *PHR* constrains the flooding towards the destination to gain the

advantage of flooding without paying the cost that could cause a network congestion problem. There are two main mechanisms to constrain the forwarding decision as follows:

1. Forward the message, only if it is closer to its destination than the previous hop.
2. Forward only a random fraction of the message at a particular node. In other words, not all the receivers need to forward.

In order to constrain the forwarding towards the destination, each node in the topology maintains the following information.

- *Message ID list*: use to prevent re-forwarding the same received message again. Entries expire after a short period of time.
- *Known list*: Contains list of source node IDs and previous node ID's from received messages. Each entry in this list expires after a suitable interval.
- *Packet Arrival Rate*: contains the number of the arrival message at the network layer to be used in making part of probabilistic forwarding decision.

The forwarding decision is made based on the following mechanisms.

- *unknown forwarding*: this is referring to as the sender has no information about the destination. The first packet transmitted with known flag set to be false. The packet is flooded in the network and the destination is reached. Each intermediate node receives a copy of the packet, looks up the packet's destination in its *known-list*, if there, the packet known flag is set to be true, this known as *known forwarding*. If the packet's destination is not there, the the packet is forwarded with a false known flag.
- If a packet received with a true known flag and the packet's destination not in the *known-list*, then the packet is discarded. This is mean, the packet is away from its destination.
- *known forwarding*: the aim is only to forward the packets that are getting close to their destinations by comparing the local distance with the previous hop distance to the destination. If the local distance is shorter, the node performs a known forwarding, otherwise, the packet is discarded.

- To avoid placing more load on the network due to transmitting redundant copy of each packet, here *Packet Arrival Rate* is employed to be used in probabilistic forwarding, once the packet is ready to be forwarded.

Performance evaluation in VANET urban: This thesis provides a comprehensive performance evaluation with *PHR* against MANET conventional routing in low and high dynamic VANET urban environment. The simulation scenarios are parametrised to represent real-world conditions. The performance evaluation is conducted by employing traditional performance metrics i.e. Packet Delivery Ratio (PDR), Delay and the proposed metric of the probability of Packet Drop-burst Length (DBL).

9.3 PHR Evaluation

In order to validate the proposed protocol in VANET environment, *PHR* has been implemented using Network Simulator (*NS2*) as part of this research. The reason of choosing *NS2* is that the capability of providing all the network interfaces in each node and simulate network environment obstacles, and hence, provides more realistic results. The results indicate that *PHR* outperforms OLSR, AODV and GPSR using the proposed DBL performance metric and the traditional PDR in high dynamic environment. In terms of delay, *PHR* is outperformed by OLSR due to the added delay to each packet to avoid packet collision problems by transmitting redundant copies of each packet.

9.4 Future Work

Further research could be conducted to monitor how the network performs with *PHR* as follows.

9.4.1 More RSU involvement

Despite the fact that the conducted simulation scenarios are hybrid mode (contain V2V and V2I), however, the role of the RSUs has been just forwarding network traffic in inter-vehicles manner rather than provide services like Internet access, information services and information on weather or traffic. One interesting scenario could be, is to connect the vehicles with a cloud infrastructure or with the Internet throughout the RSU.

9.4.2 Various Network Traffic

The shape of the network traffic has a significant influence on the network performance, especially in dynamic network topology. Application level performance depend on the network traffic class. In this thesis, the evaluation of the network performance is conducted by exploiting safety-related VANET application traffic. It would be interesting to exploit a mixture of various VANET application traffic into one scenario and monitor the network behaviour with *PHR* with such a heterogeneous network traffic. Furthermore, it would be valuable if the *PHR* is evaluated with Geocast and multicast network traffic.

9.4.3 Additional Scenario

Although the same general network principles need to be applied for both urban and motorway topologies. The type of topology has a big influence on the network performance. In urban area, high density in both of number of vehicles and of the number of network flows could cause a network collapse. However, the road pattern in rural areas could lead to sparse network topology problem. In this thesis, the urban environment model is selected to evaluate the network performance with *PHR*. It is worth observing *PHR* performance in a larger scale scenario and different mobility model such as motorway scenario.

9.4.4 Memory Usage

Memory usage has a linear relationship with the number of communications in PHR. As previously mentioned, nodes maintain different types of information, this requires a memory to store this information to be involve later in the forwarding decision making. As the number of connections increase, so does the memory consumption. With *PHR*, information table entries (Known-list, Message id list and Packet Arrival Rate) are expired based on a specific timing. It could be interesting to expire the information table entries based on the available memory alongside the specific duration, to reduce the memory consumption and study the impact of the available memory on the performance.

9.4.5 Security Implications

During the designing of *PHR*, we have not consider the security implications of the protocol, including the effects that malicious nodes could have on the performance.

9.4.6 Jitter Delays

As *PHR* forwarding strategy depending on the broadcast nature of the wireless medium. This could leads to collision problem if two or more nodes perform forwarding at the same time. To avoid the collisions, *PHR* added a random jitter to each forwarding process. It is worth investigating the jitter delays and the impact on the performance and delay.

9.4.7 Evaluate PHR against Opportunistic Protocols

Opportunistic protocols are designed based on Delay Tolerant Networks (DTN). The aim of DTNs is to support the disruption of connectivity and/or long delivery delays. DTN uses the store and forward strategy. Nodes carry messages and store them in their memory while moving and forwarding these messages when they find an opportunity or encounter with other nodes. Despite the fact that the *PHR* is designed to route and forward real-time traffic, it worth comparing PHR against well known opportunistic protocols such as PRoPHET [45] and MaxProp [11]. Additionally, Evaluate *PHR* against established OR protocols designed for VANET.

References

[1] Asar Ali and Z Akbar. *Evaluation of AODV and DSR routing protocols of wireless sensor networks for monitoring applications*. PhD thesis, 2009.

[2] Awos Kh Ali, Iain Phillips, and Huanjia Yang. Evaluating VANET Routing in Urban Environments. *39th Int. Conf. Telecommun. Signal Process.*, pages 60–63, 2016.

[3] Awos Kh Ali, Iain Phillips, and Huanjia Yang. Drop-Burst Length Evaluation of Urban VANETs. *Int. J. Adv. Telecommun. Electrotech. Signals Syst.*, 6(2):1–6, 2017.

[4] Shuja Ansari, Tuleen Boutaleb, Sinan Sinanovic, Carlos Gamio, and Ioannis Krikidis. Vehicular Multitier Gateway Selection Algorithm for Heterogeneous VANET Architectures. pages 180–185, 2017.

[5] Fan Bai, Hariharan Krishnan, and Varsha Sadekar. Towards Characterizing and Classifying Communication-based Automotive Applications from a Wireless Networking Perspective. *Proc. IEEE Work. Automot. Netw. Appl.*, pages 1–25, 2006.

[6] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. SUMO - Simulation of Urban MObility - an Overview. *Proc. 3rd Int. Conf. Adv. Syst. Simul.*, (c):63–68, 2011.

[7] Abhijeet A Bhorkar, Mohammad Naghshvar, Student Member, Tara Javidi, and Bhaskar D Rao. Adaptive opportunistic routing for wireless ad hoc networks. *IEEE\slash ACM Trans. Netw.*, 20(1):243–256, 2012.

[8] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Comput. Commun. ...*, 34(1):69, 2004.

[9] Sanjit Biswas and Robert Morris. ExOR: Opportunistic Multi-Hop Routing for Wireless Networks. *ACM SIGCOMM Comput. Commun. Rev.*, 35(4):133, 2005.

[10] Raffaele Bruno, Marco Conti, and Maddalena Nurchis. MaxOPP: A novel Opportunistic Routing for wireless mesh networks. *IEEE Symp. Comput. Commun.*, pages 255–260, 2010.

[11] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. *Proc. - IEEE INFOCOM*, 00(c), 2006.

[12] Xuelian Cai, Ying He, Chunchun Zhao, Lina Zhu, and Changle Li. LSGO : Link State aware Geographic Opportunistic routing protocol for VANETs. *EURASIP J. Wirel. Commun. Netw.*, pages 1–10, 2014.

[13] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. *ACM SIGCOMM Comput. Commun. Rev.*, 37(4):169, 2007.

[14] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. *ACM SIGCOMM Comput. Commun. Rev.*, 37(4):169, 2007.

[15] Nessrine Chakchouk. A Survey on Opportunistic Routing in Wireless Communication Networks. *IEEE Commun. Surv. Tutorials*, 17(4):2214–2241, 2015.

[16] Gayathri Chandrasekaran. VANETs: The Networking Platform for Future Vechicular Applications. *Csrutgersedu*, 35(1):43–65, 2008.

[17] T Clausen and P Jacquet. Optimized link state routing protocol (OLSR). *Rfc 3626*, pages 1–75, 2003.

[18] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel A F Mini, and Antonio A F Loureiro. Data Communication in VANETs: Survey, Applications and Challenges. Technical report, 2014.

[19] Kevin Fall. A delay-tolerant network architecture for challenged internets. *Proc. 2003 Conf. Appl. Technol. Archit. Protoc. Comput. Commun. - SIGCOMM '03*, page 27, 2003.

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

[21] IEEE 1609 Working Group. IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. *IEEE Std 1609.0-2013*, (IEEE P1609.0/D7.0):1–78, 2014.

[22] Shuo Guo, Yu Gu, Bo Jiang, and Tian He. Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links. 63(11):133–144, 2009.

[23] Ibrahim F. Haddad and David Gordon. Network Simulator 2: a Simulation Tool for Linux, 2002.

[24] S Hamma, E Cizeron, H Issaka, and J.-P. Guedon. Performance evaluation of reactive and proactive routing protocol in IEEE 802.11 ad hoc network. *Next-Generation Commun. Sens. Networks 2006*, 6387:38709, 2006.

[25] D O T Hs. Vehicle Safety Communications Project Task 3 Final Report. Technical Report March, 2005.

[26] C.-J. Hsu, H.-I. Liu, and W. Seah. Economy: A duplicate free opportunistic routing. *Proc. 6th Int. Conf. Mob. Technol. Appl. Syst. Mobil. '09*, pages 0–5, 2009.

[27] Aleksandr Huhtonen. Comparing AODV and OLSR Routing Protocols 2 Ad hoc On Demand Distance Vector. *Telecommun. Softw. Multimed.*, (HUT T-110.551):1–9, 2004.

[28] Teerawat Issariyakul and Ekram Hossain. *Introduction to network simulator NS2*, volume 9781461414. Springer US, Boston, MA, 2012.

[29] Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Top. Conf. 2001. IEEE INMIC 2001. Technol. 21st Century. Proceedings. IEEE Int.*, pages 62–68. Ieee, 2001.

[30] David B Johnson, David A Maltz, and Josh Broch. DSR : The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Ad Hoc Netw.*, pages 139–172, 2001.

[31] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Commun. Surv. Tutorials*, 13(4):584–616, 2011.

[32] MND Karande and MKK Kulkarni. Efficient routing protocols for vehicular adhoc network. *Int. J. Eng. ...*, 2(1):1–8, 2013.

[33] Brad Karp and H. T. Kung. Gpsr. *Proc. 6th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '00*, pages 243–254, 2000.

[34] Imran Khan and Amir Qayyum. Performance evaluation of AODV and OLSR in highly fading Vehicular Ad hoc Network environments. In *INMIC 2009 - 2009 IEEE 13th Int. Multitopic Conf.*, pages 1–5, 2009.

[35] K. U R Khan, Rafi U. Zaman, and A. Venugopal Reddy. Performance comparison of on-demand and table driven ad hoc routing protocols using NCTUns. *Proc. - UKSim 10th Int. Conf. Comput. Model. Simulation, EUROSIM/UKSim2008*, pages 336–341, 2008.

[36] Rakesh Kumar and Mayank Dave. A Comparative Study of Various Routing Protocols in VANET. *J. Comput. Sci.*, 8(4):6, 2011.

[37] Sunil Kumar and Jyotsna Sengupta. AODV and OLSR Routing Protocols forWireless Ad-hoc and Mesh Networks. *2010 Int. Conf. Comput. Commun. Technol.*, pages 402–407, 2010.

[38] Rafael Laufer, Henri Dubois-Ferriere, and Leonard Kleinrock. Polynomial-time algorithms for multirate anypath routing in wireless multihop networks. *IEEE/ACM Trans. Netw.*, 20(3):742–755, 2012.

[39] Marie-Ange Lèbre, Frédéric Le Mouël, Eric Ménard, Julien Dillschneider, and Richard Denis. VANET Applications: Hot Use Cases. 2014.

[40] Goo Yeon Lee and Zygmunt J. Haas. Simple, practical, and effective opportunistic routing for short-haul multi-hop wireless networks. *IEEE Trans. Wirel. Commun.*, 10(11):3583–3588, 2011.

[41] Kevin C. Lee and Mario Gerla. Opportunistic vehicular routing. *2010 Eur. Wirel. Conf. EW 2010*, pages 873–880, 2010.

[42] Kevin C Lee, Uichin Lee, and Mario Gerla. Routing in Urban Vehicular Grids. 2009.

[43] Kevin C. Lee, Uichin Lee, and Mario Gerla. Advances in Vehicular Ad-Hoc Networks. pages 149–151. 2010.

[44] Ilias Leontiadis and Cecilia Mascolo. GeOpps: Geographical opportunistic routing for vehicular networks. *2007 IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WOWMOM*, 2007.

[45] A Lindgren, A Doria, E Davies, and S Grasic. Probabilistic Routing Protocol for Intermittently Connected Networks. Technical report, RFC Editor, 2012.

[46] Mei-Hsuan Lu, Peter Steenkiste, and Tsuhan Chen. Design, implementation and evaluation of an efficient opportunistic retransmission protocol. *Proc. 15th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '09*, page 73, 2009.

[47] Gustavo Marfia, Giovanni Pau, and Marco Roccetti. On developing smart applications for VANETs: Where are we now? Some insights on technical issues and open problems. *2009 Int. Conf. Ultra Mod. Telecommun. Work.*, 2009.

[48] Mahesh K. Marina and Samir R. Das. On-demand Multipath Distance Vector Routing in Ad hoc Networks. *Proc. 9th IEEE Int. Conf. Netw. Protoc.*, pages 14–23, 2001.

[49] Rabah Meraihi, Sidi Mohammed Senouci, Djamal Eddine Meddour, and Moez Jerbi. Vehicle-to-Vehicle Communications: Applications and Perspectives. In *Wirel. Ad Hoc Sens. Networks*, pages 285–308. 2010.

[50] Minoru Nakagami. The m-distribution - A General Formula of Intensity Distribution of Rapid Fading. *Stat. Methods Radio Wave Propag.*, pages 3–36, 1960.

[51] Matthew S. Nassr, Jangeun Jun, Stephan J. Eidenbenz, Anders A. Hansson, and Angela M. Mielke. Scalable and reliable sensor network routing: Performance study from field deployment. *Proc. - IEEE INFOCOM*, pages 670–678, 2007.

[52] CE Perkins, EM Belding-Royer, and SR Das. Ad hoc on-demand distance vector (AODV) routing. *Req. Comments 3561*, pages 1–37, 2003.

[53] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. WMCSA'99. Second IEEE Work. Mob. Comput. Syst. Appl.*, number 3, pages 90–100, 2009.

[54] Kiah Mlm Qabajeh, Liana Khamis. A Qualitative Comparison of Position-Based Routing Protocols for Ad-Hoc Networks. *Int. J. Comput. Sci. Netw.*, 9(2):131–140, 2009.

[55] Eric Rozner, Jayesh Seshadri, Yogita Ashok Mehta Mehta, and Lili Qiu. {SOAR}: Simple Opportunistic Adaptive Routing Protocol for Wireless Mesh Networks. *IEEE Trans. Mob. Comput.*, 8(12):1622–1635, 2009.

[56] Lorenzo Rubio, Juan Reig, Vicent M. Rodrigo-Pearrocha, and Narcis Cardona. A semi-deterministic propagation model for predicting short-term fad-

ing statistics in urban environments based on the Nakagami-m distribution. *AEU - Int. J. Electron. Commun.*, 61(9):595–604, 2007.

[57] Elmer Schoch, Frank Kargl, Michael Weber, and Tim Leinmüller. Communication patterns in VANETs. *IEEE Commun. Mag.*, 46(11):119–125, 2008.

[58] Society of Automotive Engineers. Dedicated Short Range Communications (DSRC) Message Set Dictionary (SAE J2735), 2009.

[59] Libo Song and David F Kotz. Evaluating Opportunistic Routing Protocols with Large Realistic Contact Traces. *Work. Challenged Networks*, pages 35–42, 2007.

[60] T Spyropoulos, K Psounis, and C S Raghavendra. Spray and Wait : An Efficient Routing Scheme for. *Direct*, pages 252–259, 2005.

[61] Sandeep Tayal and Malay Ranjan Triphathi. VANET-challenges in selection of vehicular mobility model. *Proc. - 2012 2nd Int. Conf. Adv. Comput. Commun. Technol. ACCT 2012*, pages 231–235, jan 2011.

[62] The Network Simulator (ns-2). http://www.isi.edu/nsnam/ns/.

[63] Intelligent Transportation, Systems Committee, Ieee Vehicular, and Technology Society. *IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture IEEE Vehicular Technology Society*, volume 2010. 2013.

[64] Amin Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. *Tech. Rep. number CS-200006, Duke Univ.*, (CS-200006):1–14, 2000.

[65] Leilei Wang, Zhigang Chen, and Jia Wu. An opportunistic routing for data forwarding based on vehicle mobility association in vehicular ad hoc networks. *Inf.*, 8(4), 2017.

[66] Zehua Wang, Yuanzhu Chen, and Cheng Li. CORMAN: A novel cooperative opportunistic routing scheme in mobile ad hoc networks. *IEEE J. Sel. Areas Commun.*, 30(2):289–296, 2012.

[67] Cédric Westphal. Opportunistic routing in dynamic ad hoc networks: The OPRAH protocol. *2006 IEEE Int. Conf. Mob. Ad Hoc Sens. Syst. MASS*, pages 570–573, 2007.

[68] Previous Work and I N Vanet. Network Performance in IEEE 802.11 and IEEE 802.11p Cluster Based on VANET. pages 495–499, 2017.

[69] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle safety messaging in DSRC. *Proc. first ACM Work. Veh. ad hoc networks - VANET '04*, pages 19–28, 2004.

[70] Y. Yan, B. Zhang, H. T. Mouftah, and J. Ma. Practical Coding-Aware Mechanism for Opportunistic Routing in Wireless Mesh Networks. *IEEE Int. Conf. Commun.*, pages 2871–2876, 2008.

[71] Shengbo Yang, Feng Zhong, Chai Kiat Yeo, Bu Sung Lee, and Jeff Boleng. Position Based Opportunistic Routing for Robust Data Delivery in MANETs. *GLOBECOM 2009 - 2009 IEEE Glob. Telecommun. Conf.*, pages 1–6, 2009.

[72] Yuan Yuan, Hao Yang, Starsky H Y Wong, Songwu Lu, and William Arbaugh. ROMER : Resilient Opportunistic Mesh Routing for Wireless Mesh Networks. *IEEE 1st Work. Wirel. mesh networks*, 2005.

[73] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular ad hoc networks (VANETS): status, results, andchallenges. *Telecommun. Syst.*, 50(4):217–241, dec 2010.

[74] Kai Zeng, Wenjing Lou, Jie Yang, and Donald R. Brown. On throughput efficiency of geographic opportunistic routing in multihop wireless networks. *Mob. Networks Appl.*, 12(5-6):347–357, 2007.

[75] Kai Zeng, Wenjing Lou, and Hongqiang Zhai. Capacity of Opportunistic Routing in Multi-Rate and Multi-Hop Wireless Networks. 7(12):5118–5128, 2008.

[76] Kai Zeng, Zhenyu Yang, and Wenjing Lou. Opportunistic routing in multihop wireless networks. *IEEE Trans. Wirel. Commun.*, 9(11):3512, 2010.

[77] Xinyu Zhang and Baochun Li. Dice: a Game Theoretic Framework for Wireless Multipath Network Coding. *Proc. ACM Int. Symp. MobiHoc*, pages 293–302, 2008.

[78] Xinyu Zhang and Baochun Li. Optimized Multipath Network Coding in LossyWireless Networks. *IEEE J. Sel. AREAS Commun.*, 27(5):622–634, 2009.

[79] Zengzhe Zhang. Fast propagation of messages in VANETs and the impact of vehicles as obstacles on signal propagation. 2015.

[80] Zhongliang Zhao, Denis Rosário, Torsten Braun, and Eduardo Cerqueira. Context-aware opportunistic routing in mobile ad-hoc networks incorporating

node mobility. *IEEE Wirel. Commun. Netw. Conf. WCNC*, 3:2138–2143, 2014.

[81] Zhongliang Zhao, Denis Ros??rio, Torsten Braun, Eduardo Cerqueira, Hongli Xu, and Liusheng Huang. Topology and Link quality-aware Geographical opportunistic routing in wireless ad-hoc networks. *2013 9th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2013*, pages 1522–1527, 2013.

[82] Jing Zuo, Chen Dong, Hung Viet Nguyen, Soon Xin Ng, Lie Liang Yang, and Lajos Hanzo. Cross-layer aided energy-efficient opportunistic routing in ad hoc networks. *IEEE Trans. Commun.*, 62(2):522–535, 2014.

Appendix A

Integrate PHR source code into NS2

PHR directory contains three files, i.e. *phr.cc*, *phr.h* and *phr_packet.h*. In order to integrate and compile *PHR* protocol in *NS2*, the following steps should fulfilled.

- It should install a fresh copy of *NS2.35*.
- Download and copy *PHR* directory into */ns-allinone-2.35/ns-2.35/*.
- Add *case PT_PHR* to */ns-allinone-2.35/ns-2.35/queue/priqueue.cc* from line 94.
- *PHR* packet header need to be defined, */ns-allinone-2.35/ns-2.35/common/packet.h* file should modified accordingly by adding *#define HDR_PHR(p) (hdr_phr::access(p))* after line 62.
- Modifying same file, PT_NTYPE should change to 74, and for *PHR* protocol PT_PHR = 73. If you have already installed another routing protocol. Just make sure PT_NTYPE is last, and protocol number is ordered sequentially. Code in A.2 shows the changes to *packet.h*.
- Add the *type == PT_PHR* as shown in A.2 at line 280 of the same *packet.h*. Then add *name_[PT_PHR]="PHR"* in line 420.
- In order to provide a trace functionality into the simulation, it should enable *NS2* to trace all the events in the simulation, to do that, */ns-allinone-2.35/ns-2.35/trace/cmu-trace.h & cmu-trace.cc* files need to be modified.

Listing A.1: Packet header file changes

```
1 static const packet_t PT_PHR = 73;
2 static packet_t PT_NTYPE = 74; // This MUST be the LAST one
```

Listing A.2: Make packet has high priority

```
type == PT_PHR ||
type == PT_MDART)
```

Listing A.3: Define drop reasons

```
#define DROP_PHR_PH_CLOSER          "CLOSER"//PH is closer to D
#define DROP_PHR_DKNOW              "DKNOW"// know flag set and don't
    know about D.
```

- First, define drop reasons by adding lines in A.3 into *cmu-trace.h* at line85.
- Define trace function in *cmu-trace.h* at line 165 as shown in A.4.

Listing A.4: Define trace function

```
void    format_phr(Packet *p, int offset);
```

- The implementation of the trace function should be added in *cmu-trace.cc* at line 1182 as shown in A.5.

Listing A.5: Main body of PHR trace function.

```
#include <phr/phr_packet.h> //PHR protocol
// main body of the trace function.\
void
CMUTrace::format_phr(Packet * p, int offset)
   {struct hdr_phr *phr = HDR_PHR(p);
   struct hdr_phr_bc *bc = HDR_PHR_BC(p);
   switch (phr->pkt_type) {
   case PHR_BC:
      if (pt_->tagged())
         {sprintf(pt_->buffer() + offset,
            "-PHR:t %x -PHR:h %d -PHR:b %d -PHR:s %d "
            "-PHR:ts %f "
            "-PHR:c PHR ",
            bc->bc_type,
            bc->bc_hop_count,
            bc->bc_bcast_id,
            bc->bc_src,
            bc->bc_timestamp);
      } else if (newtrace_)
         {sprintf(pt_->buffer() + offset,
```

```
21              "-P phr -Pt 0x%x -Ph %d -Pb %d -Ps %d -Pts %f -Pc PHR ",
22              bc->bc_type,
23              bc->bc_hop_count,
24              bc->bc_bcast_id,
25              bc->bc_src,
26              bc->bc_timestamp);
27        } else {sprintf(pt_->buffer() + offset,
28              "[0x%x %d %d [%d] [%f]] (PHR)",
29              bc->bc_type,
30              bc->bc_hop_count,
31              bc->bc_bcast_id,
32              bc->bc_src,
33              bc->bc_timestamp);
34        }
35        break;
36     default:
37 #ifdef WIN32
38        fprintf(stderr,
39           "CMUTrace::format_phr: invalid PHR packet typen");
40 #else
41        fprintf(stderr,
42           "%s: invalid PHR packet typen", __FUNCTION__);
43 #endif
44        abort();
45     }
46 }
```

- After changing C++ files, TCL files also need to be changed to create *PHR* routing agent to be used in TCL file. This is done by modifying */ns-allinone-2.35/ns-2.35/tcl/lib/ns-packet.tcl*.
- Add *PHR* at line 172
- Set routing agent by modifying */ns-allinone-2.35/ns-2.35/tcl/lib/ns-lib.tcl* at line 639 as shown in A.6.
- Set port numbers of *PHR* agent (sport is the source port, dport is destination port) by adding code in A.7 to */ns-allinone-2.35/ns-2.35/tcl/lib/ns-agent.tcl* at line 201.
- Modify */ns-allinone-2.35/ns-2.35/tcl/lib/ns-mobilenode.tcl* by adding code in A.8 at line 204.

Listing A.6: Set PHR agent

```
PHR {
  set ragent [$self create-phr-agent $node]
}
\item At line $870$ code in \~\ref{tcl} should be added.
\begin{lstlisting}[caption= Create PHR agent \label{tcl}, float]
Simulator instproc create-phr-agent {node}
    {set ragent [new Agent/PHR [$node node-addr]]
   $self at 0.0 "$ragent start"
   $node set ragent_ $ragent
   return $ragent
}
```

Listing A.7: Set ports of PHR agent

```
Agent/PHR instproc init args
    {$self next $args
}
Agent/PHR set sport_ 0
Agent/PHR set dport_ 0
```

Listing A.8: Set ports of PHR agent

```
# Special processing for PHR
set phronly [string first "PHR" [$agent info class]]
if {$phronly!= -1}
   {$agent if-queue [$self set ifq_(0)] ;# ifq between LL and MAC
}
```

- Modify */ns-allinone-2.35/ns-2.35/Makefile* by adding *phrphr.o* after *pumapuma.o* line to the list of object files for *NS2*.

Now, *NS2* should be ready to be recompiled. To do so, run make command in */ns-allinone-2.35/ns-2.35/*. When the compilation is done, *NS2* ready to be tested with *PHR*.

Appendix B

Mobility Models Design

The urban mobility models that used in this thesis is implemented using Simulation of Urban Mobility(SUMO). SUMO enables to generate random vehicle trips on the selected maps. As mentioned in the early chapters, three maps are implemented to represent the urban environment i.e. part of London congestion zone, part of Leicester city centre and Manhattan-style map. The following subsections show the selected maps implementation method.

B.1 London congestion zone and Leicester city centre

The real urban map is captured using Open Street Map page that provides a geographic data such as street maps. After selecting a rectangular area in the web browser, the selected map will be exported into OSM (OpenStreetMap file) format. The following steps show the method to generate a mobility model of an urban map for *NS2*

- In order to make the map file format usable by SUMO, it must convert it into a SUMO network file. Typically you do this with NETCONVERT that is provided by SUMO. NETCONVERT extracts the simulation-related information from the OpenStreetMap file and puts it out in the SUMO network file. The following command converts the OSM map file into a SUMO network file:- *netconvert –osm-files map.osm -o map.net.xml*

- Now the map is ready to add vehicles moving around in a random way fashion. SUMO provides a Python-based tool to add random trips on the map, this tool is *randomTrips.py*. The following command added a number of vehicles moving around on the map:-

python /home/ /sumo-dir/tools/trip/randomTrips.py -n map.net.xml -r map.rou.xml(routes file) -e 100(End time)

- Before export the mobility file in TCL format that is can *NS2* reads, it need to save the complete network states into file that contains all the simulation events as follows:-

  sumo -n map.net.xml (network file) -r map.ru.xml (routes file) –fcd-output netstate.xml (states file)

- The states file now is ready to be exported as a *NS2* configuration files. SUMO provides a Python-based tool *traceExporter.py* to produce these configuration files as shown in the following command line:-

  python /home/sumo-dir/tools/bin/traceExporter.py –fcd-input netstate.xml (states file) -p 1 (cars export rate) -b 0 (begin) -e 100 (End) –ns2activity-output a.tcl (NS2 activity) –ns2config-output config.tcl (configuration) –ns2mobility-output mobility.tcl (mobility)

The produced configuration files are now ready to be used in the *NS2* simulation.

B.2 Manhattan-style Model

In real-city map generation, it has been used Open Street Map web page. However, to build an artificial map and configure it to match the selected real-city map's dimensions, A Python-based script is written as illustrated in B.1.

Listing B.1: Manhattan-style map script

```
#This script generate a Manhattan-style map.\
import xml.etree.cElementTree as ET
nodes=ET.Element("nodes")
scale_x=850
scale_y=850
average_dist = 150
block_num=scale_x / average_dist +1
id_num=0
layout=[]
for i in range(0,block_num):
    for j in range(0,block_num):
        x_cord=average_dist*j
        y_cord=average_dist*i
        if x_cord==0:
            x_cord=1
```

```
16          if y_cord==0:
17              y_cord=1
18          node=ET.SubElement(nodes, "node")
19          node.set("x", str(x_cord))
20          node.set("y", str(y_cord))
21          if x_cord==1 or y_cord==1 or x_cord==scale_x or y_cord==scale_y:
22              myType = "priority"
23          else:
24              myType = "traffic_light"
25          node.set("type", myType)
26          node.set("id", "node"+str(id_num))
27
28          layout.append({"id":id_num,"x":x_cord,"y":y_cord})
29          id_num = id_num + 1
30  tree = ET.ElementTree(nodes)
31  tree.write("ex1.nod.xml")
32
33  def findID(x,y):
34      for node in layout:
35          if node["x"]==x and node["y"]==y:
36              return node["id"]
37  print layout
38  edge_id=0
39  edges=ET.Element("edges")
40  for node in layout:
41      if node["x"]==1:
42          x_r_cord=average_dist
43      else:
44          x_r_cord=node["x"]+average_dist
45      y_r_cord=node["y"]
46      if x_r_cord<=scale_x:
47          dest_id=findID(x_r_cord,y_r_cord)
48          edge_right=ET.SubElement(edges, "edge")
49          edge_right.set("id", "edge"+str(edge_id))
50          edge_right.set("from", "node"+str(node["id"]))
51          edge_right.set("to", "node"+str(dest_id))
52          edge_right.set("numLanes", "3")
53          edge_right.set("priority", "75")
54          edge_right.set("speed", "20")
55          edge_id=edge_id + 1
56
```

```
    x_t_cord=node["x"]
    if node["y"]==1:
        y_t_cord=average_dist
    else:
        y_t_cord=node["y"]+average_dist
    if y_t_cord<=scale_y:
        dest_id=findID(x_t_cord,y_t_cord)
        edge_top=ET.SubElement(edges, "edge")
        edge_top.set("id", "edge"+str(edge_id))
        edge_top.set("from", "node"+str(node["id"]))
        edge_top.set("to", "node"+str(dest_id))
        edge_top.set("numLanes", "3")
        edge_top.set("priority", "75")
        edge_top.set("speed", "20")
        edge_id=edge_id + 1
tree=ET.ElementTree(edges)
tree.write("ex1.edg.xml")
```

To generate vehicles movements and export it as *NS2* mobility files, the same previously mentioned steps should be followed.

Appendix C

PHR Implementation Code

This appendix provides the *PHR* implementation code alongside the scripts that used to perform the simulation experiments and extract the results.

C.1 PHR main actions code

The main functions that PHR uses to perform the opportunistic forwarding is listed in this section.

- Sending packet at source node as shown in C.1:
- Receiving data packet at intermediate node as illustrates in C.2.
- Drop previously received packets (see C.3).
- Calculating forwarding probability as shown in C.4.
- Unknown Forwarding as illustrates in C.5.
- Known forwarding. see C.6.
- Receiving packet at destination node as shown in C.7.

C.2 Functions for Data Structure Management

1. *Message id list*:
    - Insert into essage id list. see C.8.
    - Read from *Message id list* see C.9.
    - Expire *Message id list* entries as shown in C.10.
2. *PAR list*:

Listing C.1: Sending Packet at Source Node

```
// If I am the source node
if (ih->saddr() == index)
{//fill phr pkt header at source node
ch->prev_hop_ = index;
ch->addr_type() = NS_AF_NONE;
ch->direction() = hdr_cmn: :DOWN;
ch->next_hop_ = MAC_BROADCAST;
ph->dest_ = ih->daddr();
ih->daddr() = MAC_BROADCAST;
ph->src = ih->saddr();
ph->pkt_id = ch->uid();
ph->hops_so_far = 1;
//Look up for theDest.
KnownList * kn = known_lookup(ph->dest_);
if (kn!= NULL)
   {//set the packet for known forwarding
   ph->dist_to_dest = kn->hops_to_dest;
   ph->known_flag = true;
} else {//set the packet for unknown forwarding
   ph->dist_to_dest = NETWORK_DIAMETER;
   ph->known_flag = false;
}
forward(p, MAC_BROADCAST, 0.0);
}
```

Listing C.2: Receiving Data Packet at Intermediate Node

```
void
PHR::recv_data(Packet * p)
   {struct hdr_ip *ih = HDR_IP(p);
   struct hdr_cmn *ch = HDR_CMN(p);
   struct hdr_phr *ph = HDR_PHR(p);
   //increase no.of received pkts.\
   rece_pkts += 1;
   //Save PAR value every 1 sec.\
   time_slot=CURRENT_TIME - time_gab;
   if (time_slot > (1.0))
      {time_gab = CURRENT_TIME;
      sum_insert(rece_pkts);
      //PAR
      rece_pkts=0;
      //reset the counter
   }
```

Listing C.3: Discard previously received packet

```
41 // Check if the pkt received before.\
42 if (app_pkt_lookup(ih->saddr(), ch->uid()))
43    {//if packet come from different neigh.\, save the neigh.\ and
          discard.\
44     KnownList * kn1=known_lookup(ph->ph);
45     if (kn1==NULL)
46     {known_insert(ph->ph, 1);
47     } else {//just update the neighbour info.\
48    kn1->hops_to_dest = 1;
49    kn1->phr_expire=CURRENT_TIME + DEFAULT_ENTRY_EXPIRE;
50 }
51 drop(p, DROP_RTR_ROUTE_LOOP);
52 return;
```

Listing C.4: Calculation of Forwarding Probability

```
53 // call PAR from the stored list to calc.\ forwarding probability.\
54 PAR = sum();
55 if (PAR > 0)
56     {if (max_par < PAR)
57      {max_par=PAR * 2;
58    }
59    //Calculate the forwarding probability.\
60    prob_prob=1.0 - (PAR / max_par);
61 } else {prob_prob=1.0;
62 }
```

Listing C.5: Unknown Forwarding

```
63 // Generate random float value(0, 1]
64 r=((float)rand()) / (float)(RAND_MAX);
65 prob_forwarding= true;
66 nsaddr_t neighbour=ph->ph;
67 KnownList *kn=known_lookup(ph->dest_);
68 if (kn==NULL) {//add to Known list and update phr pkt fields.
69 KnownList *nor = known_lookup(neighbour);
70 if (nbr==NULL)
71    {known_insert(neighbour, 1);
72 } else {nbr->phr_expire=CURRENT_TIME+DEFAULT_ENTRY_EXPIRE;
73 }
74 KnownList *knsrc=known_lookup(ph->src);
75 if (knsrc==NULL)
76    {known_insert(ph->src, ph->hops_so_far);
77 } else {knsrc->hops_to_dest=ph->hops_so_far;
78    knsrc->phr_expire=CURRENT_TIME+DEFAULT_ENTRY_EXPIRE;
79 }
80 broadcast=prob_forwarding;
81 broadcast=broadcast && (r <= prob_prob);
82 broadcast=broadcast && (ph->known_flag == false);
83 if (broadcast)
84     {ph->hops_so_far +=1;
85    ph->dist_to_dest -=1;
86    forward(p, MAC_BROADCAST, DELAY);
87 } else {//Drop if the flag is set and prob value is low.\
88    drop(p, DROP_PHR_DKNOW);
89    return;
90 }
91 }
```

Listing C.6: Known Forwarding

```
92 // if the distance to the D is <= the distance from previous hop to D
93 if (kn->hops_to_dest <=ph->dist_to_dest)
94     {//prepare the packet for forwarding
95    ph->hops_so_far +=1;
96    ph->known_flag=true;
97    ph->dist_to_dest=kn->hops_to_dest;
98    //Check if I received from this src before.
99    KnownList * kn2=known_lookup(ph->src);
100   if (kn2==NULL)
101       {known_insert(neighbour, 1);
102      known_insert(ph->src, ph->hops_so_far);
103   } else {//update the entry
104      kn2->phr_expire=CURRENT_TIME+DEFAULT_ENTRY_EXPIRE;
105   }
106   //send the pkt.probabilistically.\
107      if (prob_forwarding && r < prob_prob)
108       {forward(p, MAC_BROADCAST, DELAY);
109   } else {//Discard the packet Packet: :
110      free(p);
111      return;
112   }
113 } else {//Drop if previoushop is closer to D
114   drop(p, DROP_PHR_PH_CLOSER);
115   }
```

Listing C.7: Receiving Packet at Destination Node

```
116 if (index==ph->dest_)
117   {//look up for the src.
118   KnownList * kn=known_lookup(ph->src);
119   if (kn==NULL)
120       {known_insert(neighbour, 1);
121      //Adding neighbour to the known - list
122      known_insert(ph->src, ph->hops_so_far);
123   } else {//update the Known list
124      kn->hops_to_dest=ph->hops_so_far;
125      kn->phr_expire=CURRENT_TIME+DEFAULT_ENTRY_EXPIRE;
126   }
127   //Send the packet to the upper layers.\
128   dmux_->recv(p, 0);
129 }
```

Listing C.8: Insert into *Message id list*

```
130 void
131 PHR::app_pkt_insert(nsaddr_t id, u_int32_t bid)
132    {Broadcastcbr  *b=new Broadcastcbr(id, bid);
133    double      now=CURRENT_TIME;
134    assert(b);
135    b->expire=now+BCAST_ID_SAVE;
136    b->src=id;
137    b->id=bid;
138    LIST_INSERT_HEAD(&cbrhead, b, link);
139 }
```

Listing C.9: Read from *Message id list*

```
140 bool
141 PHR::app_pkt_lookup(nsaddr_t id, u_int32_t bid)
142    {Broadcastcbr  *b=cbrhead.lh_first;
143    for (; b; b=b->link.le_next)
144        {if ((b->src==id) && (b->id==bid))
145          return true;
146    }
147    return false;
148 }
```

Listing C.10: Expire *Message id list* entries

```
149 void
150 PHR::app_pkt_purge()
151    {Broadcastcbr  *b=cbrhead.lh_first;
152    Broadcastcbr  *bn;
153    double      now=CURRENT_TIME;
154    //expire entries every interval
155    for (; b; b=bn)
156        {bn=b->link.le_next;
157       if (b->expire <=now)
158           {LIST_REMOVE(b, link);
159          delete  b;
160       }
161    }
162 }
```

Listing C.11: Insert into *PAR list*

```
163 void
164 PHR::sum_insert(double PAR)
165    {SumList *sm=new SumList(PAR);
166    sm->sum_expire=CURRENT_TIME+ENTRY_EXPIRED;
167    LIST_INSERT_HEAD(&sumhead, sm, sum_link);
168 }
```

Listing C.12: Read from *PAR list*

```
169 float
170 PHR::sum()
171    {float  S=0.0;
172    SumList  *s=sumhead.lh_first;
173    for (; s!=NULL; s=s->sum_link.le_next)
174        {S +=s->summ;
175    }
176    return S;
177 }
```

- Insert into *PAR list* in C.11.
- Read from *PAR list* in C.12.
- Expire *PAR list* entries in C.13.

3. *known-list*:

- Insert into *known-list* in C.14.
- Read from *known-list* in C.15.
- Expire *known-list* entries in C.16.

C.3 Results calculation code

- Calculating PDR as shown in C.17
- Calculating DBL. Listing C.18 shows the first phase of the DBL calculation while C.19 shows the second phase.
- Calculating C2C latency. see C.20.

Listing C.13: Expire *PAR list* entries

```
178 void
179 PHR::sum_purge()
180    {SumList *b=sumhead.lh_first;
181    SumList *bn;
182    double    now=CURRENT_TIME;
183    for (; b; b=bn)
184        {bn=b->sum_link.le_next;
185       if (b->sum_expire <= now)
186           {LIST_REMOVE(b, sum_link);
187          delete  b;
188       }
189    }
190 }
```

Listing C.14: Insert into *known-list*

```
191 void
192 PHR::known_insert (nsaddr_t src, u_int8_t hopcount)
193 {KnownList *kn=new KnownList (src, hopcount);
194   kn->hops_to_dest=hopcount;
195   kn->phr_expire=CURRENT_TIME + DEFAULT_ENTRY_EXPIRE;
196   LIST_INSERT_HEAD (&knhead, kn, kn_link);
197  }
```

Listing C.15: Read from *known-list*

```
198 KnownList*
199 PHR::known_lookup(nsaddr_t dst)
200    {KnownList *r=knhead.lh_first;
201    for (; r; r=r->kn_link.le_next)
202        {if (r->phr_dst==dst)
203          return r;
204    }
205    return NULL;
206 }
```

Listing C.16: Expire *known-list* entries

```
207 void
208 PHR::known_purge()
209    {KnownList     *b=knhead.lh_first;
210    KnownList     *bn;
211    double      now=CURRENT_TIME;
212    for (; b; b=bn)
213        {bn=b->kn_link.le_next;
214       if (b->phr_expire <= now)
215           {LIST_REMOVE(b, kn_link);
216          delete  b;
217       }
218    }
219 }
```

Listing C.17: PDR calculation

```
1  with gzip.open(trace_file, 'rb') as f:
2      for readlines in f:
3     split1=readlines.split()
4           #Count no.\ of sent packets.\
5          if split1[0]=='s' and split1[18]=='AGT'and split1[34]=='cbr':
6             sent[split1[40]]='s'
7          #Count no.\ of received packets.\
8          if split1[0]=='r' and split1[18]=='AGT'and split1[34]=='cbr':
9             receive[split1[40]]='r'
10 #calculate PDR
11 PDR = (len(receive)/len(sent))*100
```

Listing C.18: Produce packets' status list for DBL calculation

```
with gzip.open (trace_file, 'r') as f:
    for line in f:
       try:
        split=line.split()
        pkt_id=int(split[40])
        layer=split[18]
        fid=int(split[38])
        pkt_typ=split[34]
        time=float(split[2])
        event=split[0]
        if event=='s' and layer=='AGT' and pkt_typ=='cbr':
            if fid not in sent:
                sent[fid]={}
                sent[fid][pkt_id]=time
            else:
                sent[fid][pkt_id]=time
        elif event=='r' and layer=='AGT' and pkt_typ=='cbr':
                if fid not in rece:
                    rece[fid]={}
                    rece[fid][pkt_id]=time
                else:
                    rece[fid][pkt_id]=time
       except: IndexError,
for flow in sent:
    if flow in rece:
        for pkt_id in sent[flow]:
            if pkt_id not in rece[flow]:
                print (pkt_id, [flow, 'Dropped', sent[flow][pkt_id]])
            else:
                print (pkt_id, [flow, 'Received', rece[flow][pkt_id]])
```

Listing C.19: Calculate the probability of DBL

```
f=open(packet_list, 'r')
lines=f.readlines()
for line in lines:
    split1=line.split()
    fid=int(re.search("\d+",split1[1]).group())
    status=str(re.search("\w+",split1[2]).group())
    t=(fid, status)
    pairs.append(t)
output={} # (f,t) -> [dbl, dbl,] (f:from t: to)
last={}   # (f,t) -> [dropped/received]
for (fid, s) in pairs:
    if fid in output:  # if a pair exists in the output dict
        if s=='Dropped': # D->D or R->D
            output[fid][-1] +=1
        elif last[fid]=='Dropped': # D->R
            output[fid].append(0)
    else:
        output[fid]=[0]
        if s=='Dropped':
            output[fid]=[1]
    last[fid]=s
alldbls=[]
for x in output:
   if output[x][-1]==0:
        del output[x][-1]
   alldbls=alldbls+output[x]
a=Counter(alldbls)
DD=[]
DD=sorted(a.items())
A=0
L1=0
for L in DD:
   L1=L[1]+L1
for k in DD: #calculate the probability
    K=k[1]/L1
     A=A+K
```

Listing C.20: Calculate Delay

```
#The following code calculates delay for one protocol only.
#Same code is used for the other protocols.\
with gzip.open(sys.argv[1], 'r') as f:
    for line in f:
        split1=line.split()
        event=split1[0]
        layer=split1[18]
        pkt_typ=split1[34]
        if event=='s' and layer=='AGT'and pkt_typ=='cbr':
           sendid[int(split1[40])]= float(split1[2])
        if event=='r' and layer=='AGT'and pkt_typ=='cbr':
           rece[int(split1[40])]= float(split1[2])
pdf_delay_protocol=[]
for i in sorted(sendid.keys()):
   if j in sorted(rece.keys()):
      delay=((float(rece[j]) - float(sendid[i])))
      pdf_delay_protocol.append(float(delay))
      delay=0
```