2015

Routing and propagation in urban vehicular ad hoc networks

Abhinay Mukunthan
University of Wollongong, abhinay@uow.edu.au

ROUTING AND PROPAGATION IN URBAN VEHICULAR AD HOC NETWORKS

ABHINAY MUKUNTHAN

A dissertation submitted in partial fulfilment of
the requirements for the degree of

Doctor of Philosophy

University of Wollongong
School of Electrical, Computer and Telecommunications Engineering
Faculty of Informatics and Engineering

MARCH 2015

Supervisor:

Prof. Farzad Safaei

ICT Research Institute, University of Wollongong

Co-supervisors:

Assoc. Prof. Mehran Abolhasan

Dr. Daniel Franklin

University of Technology, Sydney

Faculty of Engineering and Information Technology

ACKNOWLEDGEMENTS

Firstly, I would like to acknowledge my supervisors, A. Prof Mehran Abolhasan, Dr. Daniel Franklin and Prof. Farzad Safaei, for without their support and encouragement, I would never have been able to complete this body of work. Special thanks to Daniel, for without his encouragement during my undergraduate years, I doubt I would have believed in myself enough to embark on a PhD at all! I would also like to thank Dr. Montserrat Ros for all her valuable input during the course of my thesis.

Thanks also go to Craig Cooper, Lidija Dordevski, Brett Hagelstein, Nidhal Odeh, Nalin Piyaratna and Vikram Sunkara for the support they offered me at various points of my PhD, specifically for providing an excellent sounding board for my rants and also helping me through the low points. Craig and Vikram were immensely helpful with various pieces of work, and this thesis would not be complete without the technical and mathematical help that both of them offered liberally. Nalin was also instrumental in encouraging me through ups and downs in my personal life, and I doubt I would have been able to get this far without his support.

I would like to thank my parents and grandparents for always supporting me on this endeavour, especially as it dragged on past the fifth year. They never lost their confidence in me and it was heartening to know that I always had their support.

I'm also ever so grateful to Chris and Kathy Cooper, Brett and Chris Hagelstein, Yuko Kaminishi, Richard and Emily Miller and Nalin Piyaratna for making sure that I always had a place to stay at during my many trips to Wollongong after

moving to Canberra. I doubt I would have been able to get as much work done if not for the knowledge that I had a place to stay whenever I needed to focus on my PhD uninterrupted. Special thanks to Chris and Kathy Cooper for always checking up on my work and making sure I was making progress and being such amazing hosts every time.

I would also like to thank Mr. Steve Tree from Rohde and Schwarz Australia for the loan of a spectrum analyser, which was critical in performing a range of experiments which form the basis of a chapter of this thesis.

Finally, I would like to thank Smart Services CRC for financing my PhD, and the heads of research, Julien Vayssière and Daniel Austin, who took an interest in every student's research and made the CRC an engaging place to be.

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) are distributed, infrastructure-independent wireless networks that are formed between both stationary and moving vehicles. Despite over a decade of research, there exist few practical deployments of large-scale vehicle to vehicle networks. The area of routing protocols is of particular interest since it forms the basis of inter-node communications in a distributed infrastructure-less network. Simulations are an important means of evaluating new VANET protocols and architectures since physical experiments require a large investment in resources and are beyond the reach of most researchers. Simulations rely on approximations of reality; the better the approximation, the more confidence may be placed in the results. The physical propagation model is one of the fundamental aspects of any network simulation since it defines how signals travel between nodes. A review of urban routing protocols revealed that minimal consideration has previously been given to the selection of a realistic propagation model when evaluating routing protocol performance. The use of simplistic propagation models was found to lead to a very inaccurate representation of performance, especially the relative performance difference between protocols. This thesis presents a series of experiments intended to characterise the propagation model in urban environments. The CORNER propagation model is studied and enhanced based on the data obtained from the experiments. A new greedy routing approach is then introduced, which uses knowledge of the signal propagation environment to make better routing decisions. Analytical results demonstrate the improved route-finding capabilities of the new approach over existing greedy and source-based routing protocols, with the new approach resulting in the selection of shorter paths in almost every situation. Simulation results establish

an improvement of between 90% and 300% over GPSR, a well known stateless greedy routing protocol. A semi-stateful fall-back mechanism is then proposed to augment the greedy routing approach when it fails and is found to result in a further 10-15% improvement in packet delivery ratios over the purely greedy approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF TERMS

General Terms	
CDF	Cumulative Distribution Function
GPS	Global Positioning System
IP	Internet Protocol
MAC	Medium Access Controller
MANET	Mobile Ad Hoc Network
OSI	Open-Systems Interconnect
PDF	Probability Distribution Function
TCP	Transport Control Protocol
Routing Related	
AODV	Ad Hoc On-Demand Vector Routing
A-STAR	Anchor-based Street and Traffic Aware Routing
BAHG	Back-bone Assisted Hop-Greedy routing
CAR	Connectivity Aware Routing
DSR	Dynamic Source Routes
GPCR	Greedy Perimeter Coordinator Routing
GPSR	Greedy Perimeter Stateless Routing
GRANT	Greedy Routing with Abstract Neighbour Table
GSR	Geographic Source Routes
PDR	Packet Delivery Ratio
RBVT	Road-Based routing using Vehicular Traffic
RERR	Route ERRor
RNG	Relative Neighbourhood Graph
RREP	Route REPly

RREQ	Route REQuest
SUMO	Simulation of Urban Mobility
Propagation Related	
FSPL	Free-Space Path-Loss
LOS	Line of Sight
NLOS	Non-Line of Sight
NLOS1	Non-Line of Sight path around 1 corner
NLOS2	Non-Line of Sight path around 2 corners
PS	Probabilistic Shadowing
TRGR	Two-Ray Ground Reflection Model
Simulation Related	
CBR	Constant Bit Rate (Network Traffic)
NS-2	Network Simulator (version 2)
PDR	Packet Delivery Ratios
RWP	Random WayPoint
STR	Static Transmission Range
SUMO	Simulation of Urban MObility

Chapter 1

INTRODUCTION

As the sophistication and intelligence of vehicles and road infrastructure increases, vehicular ad hoc networks (VANETs) have emerged as an important area of research, in particular due to their potential for contributing to improved road safety and transport efficiency. Ad hoc networks are distributed infrastructure-independent interconnected collections of nodes, characterised by their ability to automatically self-organise an arbitrary spatial distribution of nodes, despite variable node density and potentially transient node disconnection or failure. When some or all nodes can potentially move, an ad hoc network is termed a *mobile* ad hoc network (MANET); the vehicular ad hoc network (VANET) is a variety of MANET in which some nodes are attached to road vehicles, and are therefore subject to the same mobility constraints as road traffic. The structure of road networks and the organised movement of the vehicular nodes results in a high degree of predictability and determinism for vehicular traffic, which can be exploited by communications protocols for improved performance.

1.1 Applications of VANETs

VANETs are proposed to serve a wide range of applications, each with its own specific set of requirements. One of the key applications is road safety, which involves the dissemination of information across the network via neighbouring vehicles. This information is usually distributed as small packets and has stringent requirements for delay and data integrity. The other classes of applications, such as convenience, comfort and entertainment applications may also use multi-hop routing in the network, as long as they do not interfere with the aforementioned

requirements.

Comfort and entertainment applications may easily be provided over the existing infrastructure such as cellular networks. However, there are already a large number of mobile phones and computers which rely on such networks for access to the Internet. The addition of vehicular devices places an even greater strain on these networks. Ad Hoc networks are ideal in such conditions since they provide an easy means of disseminating information without imposing an additional burden on the cellular infrastructure. Furthermore, active research in clustering and cooperative caching may even lead to a means of decreasing the load on cellular networks by sharing commonly sought out information between a local group of peers once the information exists in the local neighbourhood.

VANETs are therefore an important area of study going into the future, especially as the number of mobile devices increases beyond the capacity of existing networks.

1.2 Evaluation of VANETs

Experimental evaluation of protocols and applications for VANETs is challenging due to the need for a high critical mass of vehicular nodes, limiting the feasibility of large-scale study of real-world performance. Therefore, realistic simulations are critical for development, testing and validation of new ideas in the field. Simulations are built on a series of assumptions that are designed to *approximate* reality; in the case of VANETs, the key considerations for accurate simulations are:

1. The models and parameters used for traffic networks and vehicular behaviour (for example, road maps (real or synthesised), road rules (and the degree of compliance), traffic density);

2. The models and parameters adopted for signal propagation (for example, urban, suburban or rural); and

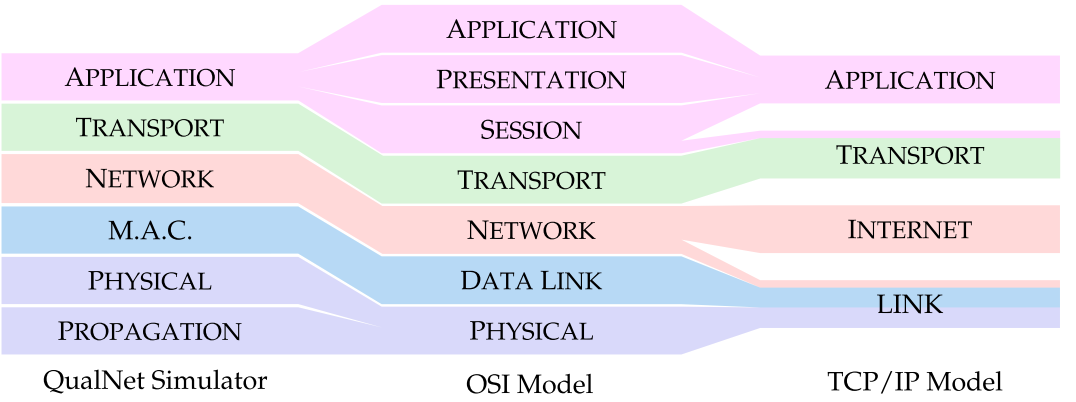


Figure 1.1: Networking Models

3. The simulation framework used to model the various communication protocols (typically split into physical, MAC, network, transport and application layers).

Most wireless network simulation platforms focus on the latter two elements, without explicit support for the first. However, provided that a network simulator can implement a generic node mobility function, then a specialised vehicular behavioural simulator can be used to control node mobility for the network simulator, providing a complete simulation framework which can be used for the evaluation of VANET protocols and applications.

A specific characteristic of network *simulators* as opposed to physical *implementations* of wireless networks is the need to accurately model the signal propagation environment. Network simulators normally implement this as a distinct layer beneath the physical layer of the traditional OSI network stack, which defines the way in which bits are represented as an analogue waveform. Figure 1.1 shows the typical structure of a network simulator (QualNet in this case [1]) alongside the OSI network model and the simplified TCP/IP protocol stack [2, 3].

A wide variety of channel models have been used in wireless network simulators, ranging from simple free-space and two-ray models to the sophisticated Okumura-Hata empirical models [4]. However, to date, few channel models have focused

on vehicular propagation, particularly in built-up urban environments. More recently, the CORNER propagation model has been proposed [5], which offers a number of advantages over classical approaches for modelling vehicular channels. Specifically, the CORNER propagation model is a situation-aware model. Situation-aware models provide an estimate of path-loss between a transmitter and receiver, taking the specific profile of the terrain between the two nodes into account. This is in contrast to empirical models like the Okumura-Hata models which provide a *median* estimate. Situation aware models are preferred when designing higher-layer protocols since they allow protocol designers to study the impact of specific terrain on path-loss.

While the CORNER propagation model shows promise in modelling an urban environment, it requires independent validation. The original implementation of CORNER was validated by the authors at 2.4 GHz using MAC layer metrics, specifically packet delivery ratios between a source and destination node travelling around a city block. This is an indirect validation of CORNER since the model only predicts the signal-strength between vehicles. MAC-layer metrics are prone to interference from other devices operating on a similar MAC layer, especially given the large number of wireless devices that operate at the tested frequency. Additionally, CORNER's classification system, which classifies paths between two nodes as being in line-of-sight (LOS) or non-line-of-sight (NLOS) ignores any paths that are spaced more than two intersections apart. This rather arbitrary decision is not explained in the original paper and therefore warrants further investigation.

Moving up from the propagation model, the physical and MAC layers of vehicular networks have now been standardised as IEEE 802.11p, which was formally published in July 2010 [6]. 802.11p operates at the 5.9 GHz frequency of the spectrum and the physical and MAC layers are designed for the highly mobile and dynamic nature of VANETs. It should be noted that this thesis does not make use of these standards since they were unavailable at the time of com-

mencement. All the studies undertaken in this thesis are instead performed on the IEEE 802.11b PHY and MAC layers [7] which operate at 2.4 GHz. This standard was chosen due to its wide availability and compatibility with multiple programmable embedded platforms.

Other than the frequency, the physical layer of 802.11p mainly differs from 802.11b in the use of 10 MHz channels instead of 22 MHz. This is chosen specifically to reduce the impact of inter-symbol interference from the numerous reflectors that are present in urban environments. In view of the applications mentioned earlier in this section, the MAC layer of 802.11p is designed to carry different priorities of traffic by implementing multiple weighted queues [8]. Furthermore, the MAC layer of 802.11p also features slightly modified inter-frame and back-off timings optimised for the vehicular environment.

While the PHY and MAC layers of the VANET stack have been standardised, substantial opportunities remain for improvements in performance at the network layer. Inter-vehicle routing is a particularly important element of VANETs since it provides a means to rapidly disseminate critical information across a large, distributed, ad hoc network of vehicles. Routing protocols which make routing decisions on the basis of source, relay and destination position are of particular interest due to the near-ubiquitous availability of positioning information through the use of GPS/AGPS and other positioning schemes.

Due to the use of simplistic propagation models in simulating urban vehicular networks, routing protocols have thus far employed design decisions that cater to these models. Existing routing protocols have largely assumed either a very small radio-range or assumed that propagation is restricted merely to line-of-sight paths. Both of these approaches are designed to model buildings as completely radio-opaque objects, obscuring signals completely. A significant amount of research [9, 10, 11, 12] has shown this to be a false assumption. This assumption has led to protocols that greedily forward packets to intersections and ignore any other available paths, leading to inefficiencies in the path-finding process. A

protocol designed to account for a realistic propagation environment is therefore a necessary next step in the evolution of urban VANET routing protocols.

1.3 Research Problem

Routing and Propagation are inextricably linked to each other. The selected propagation environment has a significant impact on routing performance, since many of the fundamental aspects of commonly used routing strategies are based on assumptions about the propagation environment. The research problem is therefore to study the urban propagation environment in significant detail, perform experiments to validate the recently proposed CORNER model and then use these results to design a better urban propagation model and routing protocol.

1.4 Aims of the Thesis

Despite propagation modelling being a critical area of VANETs, it remains a poorly understood one. While the CORNER propagation model shows significant promise, it's testing methodology, relying on MAC-layer observations, raises questions about its efficacy. A key aim of this thesis is to therefore perform a thorough, signal-strength based validation of CORNER to determine if the model is an accurate reflection of reality. Similarly, an experimental study on the effects of fading in urban environments is also performed to better characterise the phenomenon. Once propagation is better understood, the characteristics of the urban propagation environment are used to design a routing algorithm catered to such environments. A study of different urban routing approaches revealed that present protocols are often built with the assumption that signals cannot permeate buildings. Additionally, a significant majority of protocols still characterise propagation using a static transmission range, which fails to represent the varying urban terrain present in most cities. This leads to a significant number of viable forwarding options being ignored.

A greedy algorithm which uses the propagation model proposed from the experiments is therefore proposed and tested. Since greedy approaches often exhibit a failure case due to the presence of local maxima, which are areas where the greedy approach fails to find a path to the destination despite a valid path existing, a fall-back mechanism that leverages the unique characteristics of the urban environment is also proposed and tested.

1.5 Thesis Overview

This section outlines the chapters of this thesis and their contents.

Chapter 2 presents a detailed overview of routing protocols for Vehicular Ad Hoc Networks. The focus is mainly on position-based VANET routing protocols, which leverage real-time positioning information from an on-board satellite positioning system to aid with routing decisions. A wide array of protocols are surveyed and their methods and parameters are studied. The simulation environments under which these protocols are surveyed are also studied with an aim to identify deficiencies in the testing methodology, specifically in the area of signal propagation.

Chapter 3 opens with a brief overview of propagation models used in the literature, with a specific focus on the CORNER propagation model. The CORNER model is independently validated with a series of signal-strength measurements carried out in three different sites in Wollongong and Sydney, Australia. Deficiencies in the model are noted and improved respectively. This material was published in [12] and also used as the basis for [13] and [14]. The impact of the propagation model on routing performance is then studied, identifying a need to consider the propagation environment when designing routing protocols. This material was published in [15].

Chapter 4 discusses the consideration of the propagation model in the development of a routing protocol. A new greedy routing approach is proposed, which

leverages propagation estimates instead of pure Cartesian distance to make routing decisions. An analytical model is proposed, which measures routing performance of the new greedy algorithm against both a Cartesian distance based greedy approach and a source routing approach. A simulation based study of the new routing algorithm is also carried out, comparing it to the well established GPSR protocol. The content of this chapter has been submitted to IEEE GlobeCom 2015 for review.

Chapter 5 studies the failure cases of the greedy routing approach proposed in the previous chapter with a view to establishing a fall-back strategy. Existing fall-back strategies are analysed and a new fall-back strategy is proposed which addresses the deficiencies of existing approaches. The complete model is also extensively tested in simulations against both AODV, a stateful routing protocol and GPSR, a completely stateless routing protocol. The content of this chapter, is currently being prepared for submission, along with the content of Chapter 4 to the IEEE Transactions on Intelligent Transportation Systems for review. It is expected that this paper will be submitted in April 2015.

Chapter 6 summarises the important results from the thesis and concludes the thesis, along with a summary of future avenues for this line of research.

Chapter 2

A REVIEW OF URBAN VANET ROUTING PROTOCOLS

Vehicular Ad Hoc Networks (VANETs) have been an active area of research for well over a decade. VANETs are a specialised variety of the more general Mobile Ad Hoc Network (MANET), in which some or all nodes are able to move around their environment, and routing decisions are fully decentralised. While MANETs are most commonly characterised by unconstrained but relatively slow movement, VANETs tend to exhibit much faster node velocities, albeit in much more constrained and predictable ways.

Figure 2.1 illustrates a very simple set of scenarios which are common to VANETs. The simplest, an 'open field', is the scenario most commonly associated with MANETs. All nodes have a significant degree of freedom of movement and there are almost no obstructions present which may significantly affect the transmission range of individual nodes. This scenario is most applicable to vehicles such as aircraft or tanks which exhibit mobility that is not constrained by any particular terrain or topology. The 'highway' scenario limits the mobility of nodes to a single degree of freedom. Each highway only permits movement in a single direction, with a parallel highway often permitting movement in the reverse direction. An example of this type of terrain is a dual-carriageway. Nodes within a highway also tend to form into distinct clusters, where intra-cluster movement is relatively slow compared to inter-cluster movement. Finally, the urban and rural scenarios are differentiated by the density of traffic and the presence of obstructions in the form of buildings. However, nodes in both scenarios move in similar, predictable ways, bound by street topology and traffic rules.

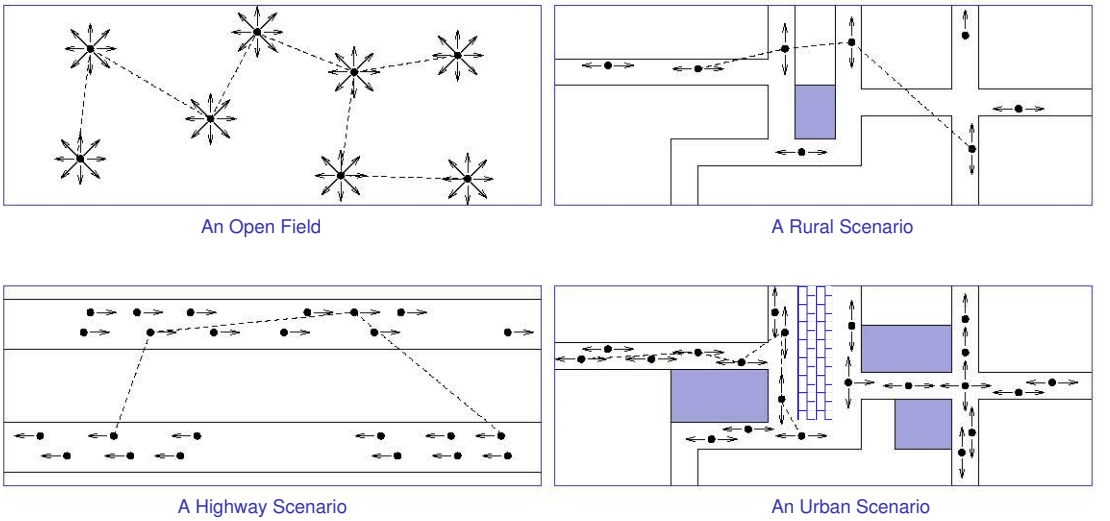


Figure 2.1: VANET Routing Scenarios

In 2010, the IEEE developed a set of vehicular enhancements to both the physical and MAC layers of the 802.11 wireless network standard, resulting in the creation of the 802.11p standard for VANETs [6]. Despite these developments, routing remains an open problem. A multitude of routing paradigms have been proposed for VANETs, most attempting to specifically address one of the scenarios described in Figure 2.1.

VANET routing protocols almost universally assume the presence of a location determination system such as a satellite navigation system, which allows nodes to accurately estimate their own location and that of their neighbours. This is a reasonable assumption, especially so today, since the majority of vehicles have access to satellite positioning systems such as GPS [16].

A number of common approaches to routing in VANETs have been proposed. Cluster-based strategies are appropriate for highways, where vehicular clustering is a natural emergent behaviour. Inter-cluster communication in these networks is managed by an elected cluster head. On the other hand, delay and disruption tolerant protocols are often proposed in rural and military applications, where sparsity is the main problem. For urban scenarios, greedy and relatively stateless

protocols are preferred, since the high density, dynamism and variability of vehicular traffic, and the substantial variation in the signal propagation environment makes the maintenance of routes difficult.

This chapter principally concentrates on protocols designed for urban environments, specifically those in which nodes are assumed to be aware of their own position.

2.1 Taxonomy of Reviewed Protocols

The protocols discussed in this chapter are classified into six broad categories as illustrated in Figure 2.2. The figure also illustrates the relationships between the protocols with connecting lines indicating conceptual relationships. A total of 24 protocols are reviewed in this section, starting with three MANET protocols which have had a significant impact on later VANET protocols.

2.2 MANET Protocols

VANETs were initially considered a subset of MANETs, and the earliest VANET routing protocols were heavily influenced by popular MANET algorithms. The reactive route finding strategy used by the Ad Hoc On-Demand Vector routing protocol (AODV) directly influences four of the reviewed protocols; Dynamic Source Routing's (DSR) source-based routing algorithm directly influences two of the protocols, which subsequently influence an additional nine; Greedy Perimeter Stateless Routing (GPSR), however, has directly or indirectly influenced almost every subsequent VANET protocol. This is due to the position-based and stateless nature of GPSR, which proved an ideal combination for the fast-changing and dynamic nature of VANETs. The following section presents a brief overview of AODV and DSR and a detailed study of GPSR, due to its impact on the field of VANET routing.

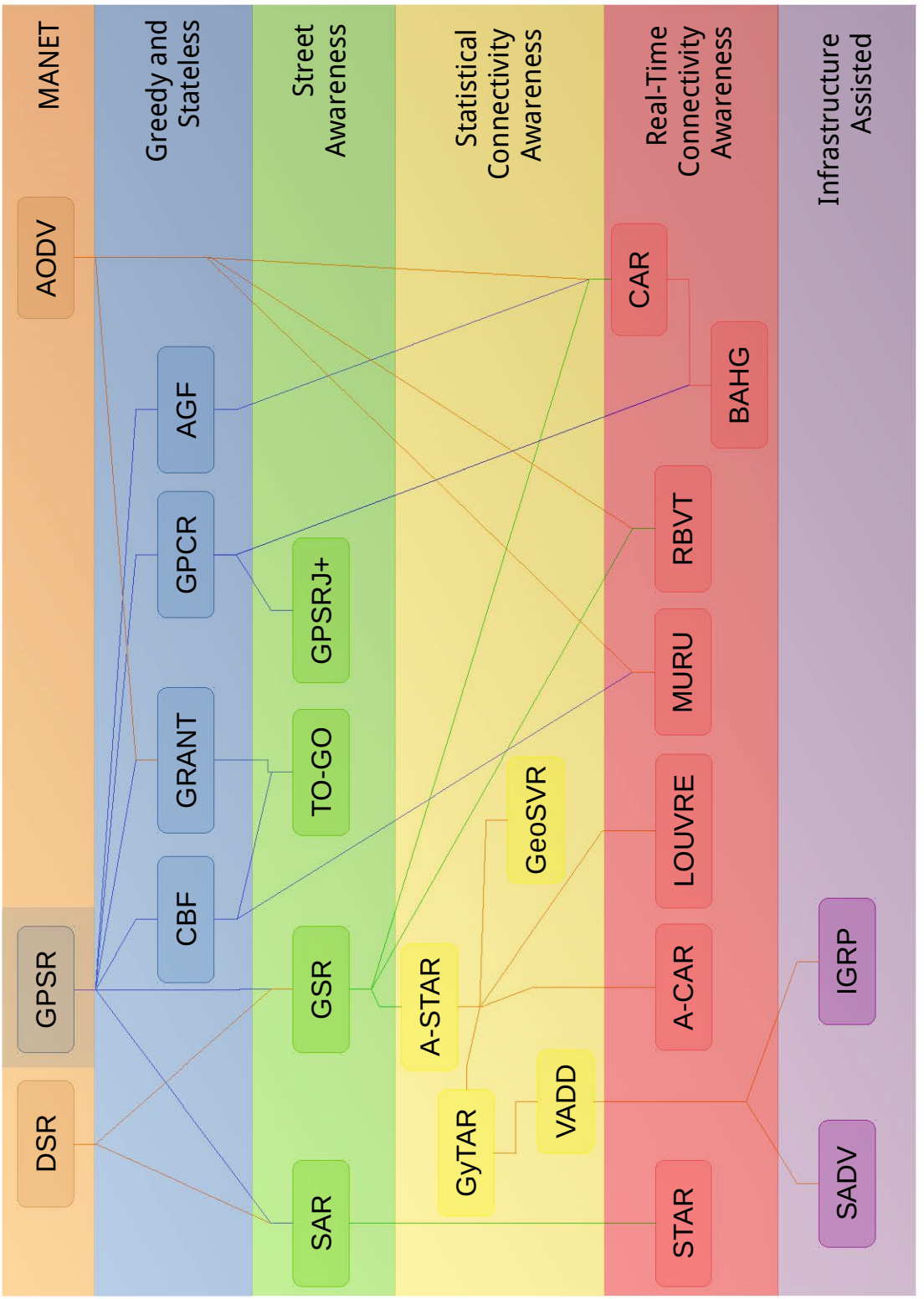


Figure 2.2: Main VANET routing protocols, illustrating conceptual relationships between protocols.

2.2.1 AODV and DSR

AODV and DSR have both had a significant impact on VANET literature [17, 18]. Both protocols use a reactive route finding algorithm, where a node requiring a path to a destination floods a route request (RREQ) to its neighbouring nodes. The neighbours will subsequently rebroadcast the RREQ if they have not previously seen the same request, and it propagates throughout the connected network until the request reaches a node which knows a path to the destination. This node initiates a route reply (RREP), which makes its way back to the source of the request along the path which led it to the destination. Any nodes which detect a break in an established route broadcast a route error (RERR) packet back to the source.

The key difference between the protocols is the use of a 'source route' in DSR, which makes both RREQ and RREP packets carry the complete traversed path in their headers. This is both a loop-avoidance and redundancy-avoidance mechanism, since subsets of the complete route are disseminated to every node that hears a control packet. AODV achieves the same outcome through the use of unique sequence numbers at both the source and destination to prevent loops and maintain route freshness respectively.

Both protocols perform poorly in VANETs due to their design philosophy being oriented to MANETs in which node velocity is relatively low. Maintaining connectivity between a sender and receiver in such a dynamic environment requires consideration of the unique characteristics of VANETs - in particular, the assumption that a particular end-to-end route lifetime can be any longer than a few seconds is not generally valid.

2.2.2 Greedy Perimeter Stateless Routing (GPSR)

GPSR is considered the seminal geographically greedy routing protocol [19]. Although initially designed with MANETs in mind, it has served as the basis for

almost every urban VANET protocol as observed in previous reviews of the VANET literature [20] [21]. GPSR is a stateless protocol because each node in GPSR only maintains a list of its immediate neighbours, which are identified with the help of periodic beacons. This means that it does not attempt to trace or maintain routes between source and destination, allowing each node to independently forward packets to the best neighbour. This reduces the overhead of route discovery and maintenance, and does not waste resources in a futile attempt to maintain stable routes in a dynamic network for an extended period of time. The statelessness of GPSR makes it extremely scalable, and the protocol is well-suited to networks with highly dynamic and ever-changing topologies, such as VANETs. Coupled with the position-based approach, it has been seen by many researchers as an ideal starting point for further research into VANET routing.

GPSR uses greedy forwarding as the main strategy to move a packet from the source to the destination. Since each node knows its immediate neighbourhood, it selects a neighbour that is closer to the destination than itself and forwards the packet to it. If there are multiple neighbours that fulfil this criteria, the neighbour that is closest to the destination is chosen. Figure 2.3 illustrates the concept of geographically greedy forwarding.

Greedy forwarding can fail in scenarios where the destination node is closer to the source node than any of the source's other neighbours, yet lies outside the transmission range of the source node. This is illustrated in Figure 2.4. While the nodes x and y both have a path to D, the greedy algorithm considers them sub-optimal next-hop routers since they are farther away from D then S itself.

If a particular node wishes to forward a packet, but has no suitable neighbours, there exists a topological void between this node and the intended destination. In such cases, GPSR switches to a fall-back mode known as perimeter mode. In perimeter mode, the relative neighbourhood graph (RNG) algorithm is used to *planarise* the immediate network to eliminate link crossovers, and the packet is

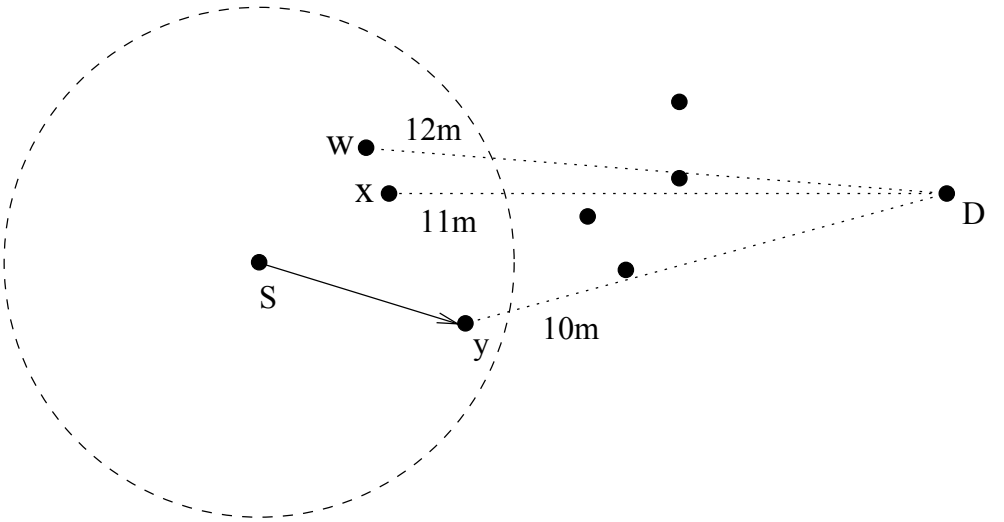


Figure 2.3: Greedy forwarding algorithm.

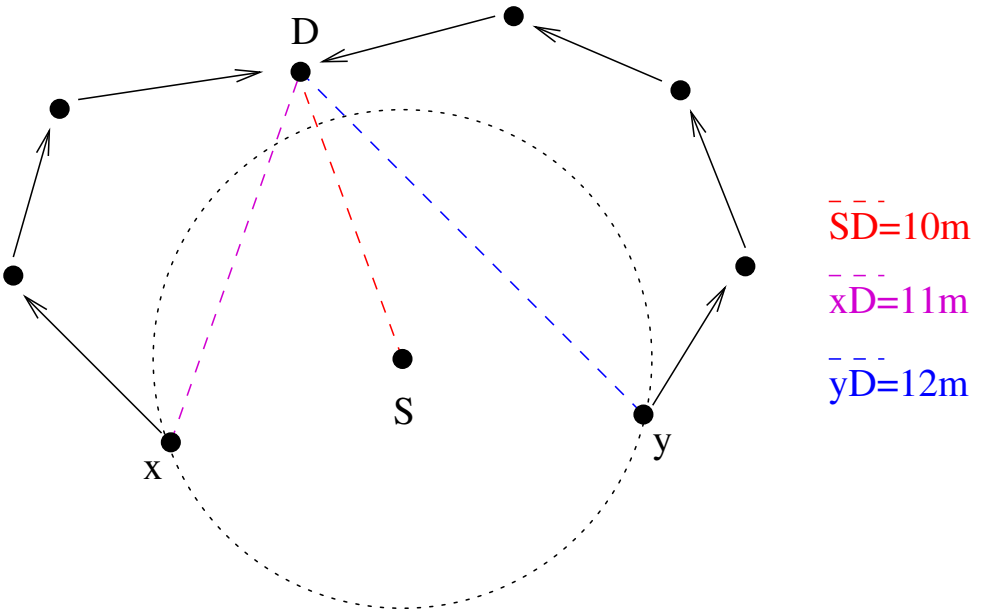


Figure 2.4: Greedy forwarding failure case.

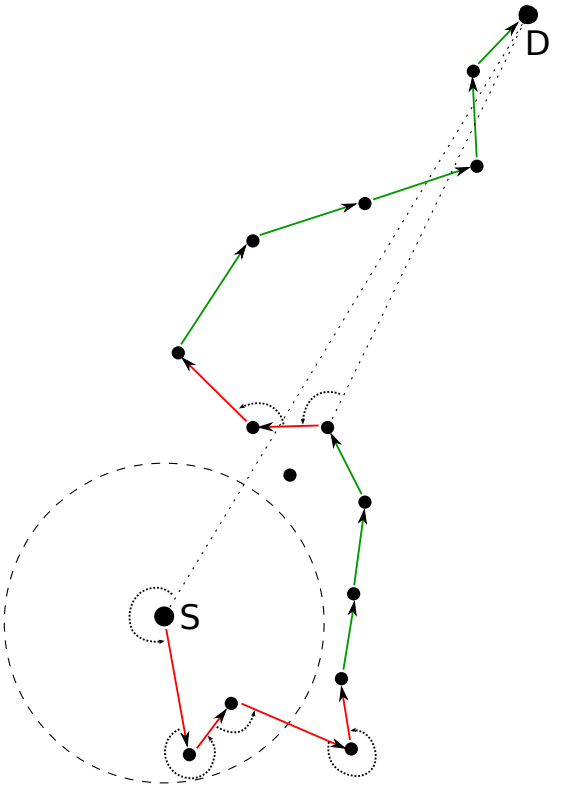


Figure 2.5: Perimeter forwarding.

forwarded using the right hand rule until the void is traversed and the packet can be returned to greedy mode.

The right hand rule for graph traversal is used to traverse a graph without routing loops by exhaustively visiting every node in a specific fashion. The next node selection relies on choosing a node located at the smallest angle counter-clockwise from the angle of entry. This is also known in the literature as face routing [22] and is illustrated in Figure 2.5. The red links are perimeter-mode forwarding decisions, with the node counter-clockwise from the previous hop being chosen according to the right hand rule. When entering perimeter mode, the node counter-clockwise from a line drawn from the entry node to the destination is chosen.

The authors of GPSR compared it to DSR in a simple MANET scenario, where it was found to offer a 1-2% increase in packet delivery ratio while reducing

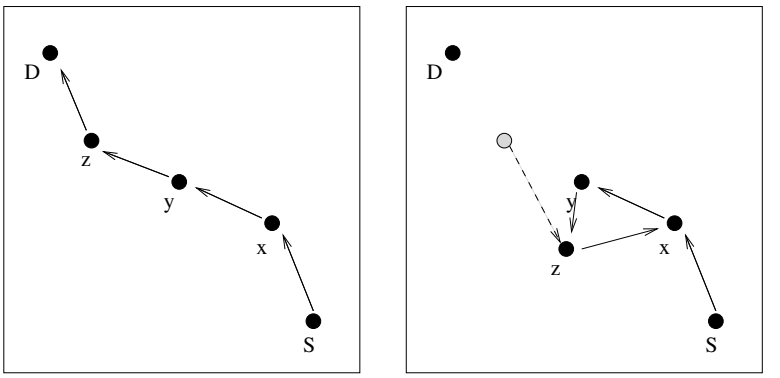


Figure 2.6: Routing loop resulting from the use of perimeter forwarding

protocol overhead by around 50-80% for beaconing intervals in the range of 1-3 seconds.

Problems with GPSR

There exist several problems with GPSR which limits its effectiveness in VANETs. GPSR itself was designed with MANETs in mind, and, like AODV and DSR, works best with slowly-moving nodes in open spaces. Rapid movement of nodes has the potential to induce routing loops when using perimeter forwarding. This is characterised in [20] using the example seen in Figure 2.6. As seen in Figure 2.6, a routing loop forms when node z moves out of range of node D.

Furthermore, perimeter forwarding can potentially cause a significant increase in hop count and hence end-to-end delay, since the packets are forwarded to the node's nearest neighbour instead of the most optimal neighbour. The protocol also cannot exploit or mitigate certain topological scenarios which occur in VANETs, such as highways with stable two-way traffic and dense urban cities with large occluding buildings.

The original paper on GPSR does not specify the means of determining the location of the destination node. It assumes that the destination's location is known and then uses greedy forwarding and perimeter forwarding to route the

packet to the destination. The Grid Location Service (GLS) [23] is mentioned in the GPSR paper, under the related work section and it is assumed to be the means used by GPSR to determine the destination's location. There exist other location determination approaches in the literature as well, some of which are reviewed in [24].

It should be noted that with the increasingly ubiquitous nature of cellular networks in urban environments, it is reasonable to assume that some sort of cellular *control channel* may be used to assist with VANET routing. In these cases, such a control channel may also be responsible for the provision of the destination's coordinates.

Influence of GPSR on VANET Routing Protocols

GPSR's greedy algorithm is often used, at least in a limited fashion, in most routing protocols appearing in the VANET literature. The wide availability of satellite positioning systems means that it is reasonable to assume that all nodes are always aware of their own geographical coordinates. Additionally, a greedy algorithm will, by definition, find the shortest path through which to transmit a packet from one point to another in a uniformly distributed network with no mobility and idealised connectivity. This is because the algorithm chooses the most optimal path at every hop. However, a significant number of VANET researchers have noted that traffic on roads is *not* uniformly distributed [25, 26, 27, 28]. Furthermore, mobility is a significant consideration in VANETs and connectivity is heavily dependent on the terrain, with buildings forming obstructions which may absorb, reflect or diffract radio waves. Therefore, greedy routing, while it is still used, is often restricted to follow street topologies as observed in several protocols further on in this review.

2.3 Greedy and Stateless VANET Routing Protocols

The following protocols build upon GPSR's greedy algorithm. They represent protocols that can be implemented and used with minimal street knowledge, unlike the other protocols in this review, which require detailed knowledge of street topology and/or traffic awareness.

2.3.1 Contention-Based Forwarding (CBF)

Contention-Based Forwarding (CBF) [29] employs similar forwarding mechanisms to greedy forwarding in GPSR. However, the unique aspect of this protocol is beacon-less forwarding. When a node has a packet to forward, it appends its location, the destination's location and a packet ID number to the packet header and broadcasts it. Each node that hears the packet then starts a timer to determine when to forward the packet. The value of the timer is calculated based on the distance gained by the packet towards the destination from the source and the radio range. If the packet is received by a node which is further away from the destination than the source, it is simply dropped.

When the timer expires, the node broadcasts the packet forward, similar to the method described above, unless it hears another node broadcasting the packet first. The authors suggest that in street scenarios, the almost periodic distribution of nodes along a narrow street makes it highly unlikely that two nodes within range of the source, and in a positive forwarding direction, will be unable to hear each other. If a duplicate packet is forwarded, nodes which hear the original and duplicate packet simply disregard the duplicate.

CBF is compared against GPSR, albeit with perimeter mode disabled. The simulation is run on a highway scenario with both an 802.11 MAC and a Null-MAC using the two-ray propagation model and a static transmission range of 250 m. A Null-MAC is a unique MAC protocol which eliminates delay and packet collisions [30] and is useful in modelling the impact of higher layer protocols on

metrics without considering the impact of the MAC layer. It is noted that CBF achieves an almost ideal packet delivery ratio with Null-MAC, while the greedy protocol's ratio falls as the beacon interval increases. No results are presented to highlight the difference in packet delivery with the 802.11 MAC; however, it is observed that the data volume transmitted using beacon-based routing is significantly higher than beaconless routing, regardless of the beacon interval. The authors speculate that this is due to the mobility of vehicles resulting in information that is out of date, especially when a packet needs to be retransmitted due to collisions. As the beaconing frequency increases, the advantage of having up-to-date information is offset by having a busier channel due to the frequent use of beacons.

2.3.2 Greedy Perimeter Coordinator Routing (GPCR)

Greedy Perimeter Coordinator Routing (GPCR) restricts the greedy routing protocol of GPSR to rectilinear streets, meaning that packets are only forwarded between successive junctions [31]. GPCR also introduces the idea of coordinator nodes, which are nodes located at junctions and thus positioned to make better forwarding decisions than nodes which only have access to one road. Nodes forward packets greedily towards a coordinator node, at which point the coordinator node forwards the packet greedily along the street most aligned with the destination. The protocol assumes that nodes on one street are isolated from nodes on another street, leaving coordinator nodes as the only means of entering a different street.

A coordinator node is identified using one of two methods. The first method relies on observing beacons from neighbours. If a node receives a beacon from two neighbours which are within transmission range of each other, but do not list one another as neighbours, it deduces that they are separated by an obstacle and that it is a therefore at a junction. The second method involves calculating a correlation coefficient of all the neighbours of a node. A coefficient close to 1

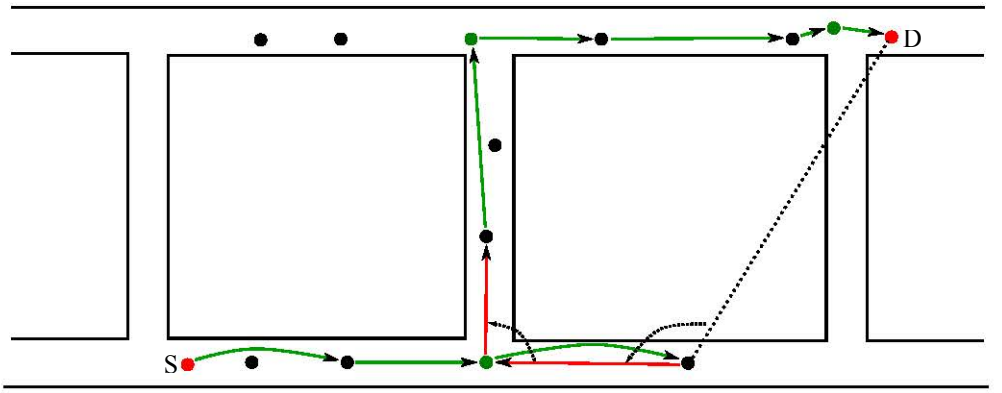


Figure 2.7: Packet forwarding in GPCR

indicates that all neighbours are located linearly with respect to each other while a coefficient close to 0 indicates a high probability that the node is at a junction since it can see two different sets of neighbours.

A recovery strategy is also used in GPCR which is similar to the face routing algorithm, used in GPSR's perimeter mode. However, since the streets represent an already planarised graph, GPCR uses the coordinator nodes as vertices of a planarised graph. Nodes forward packets towards a coordinator node, which then uses the right hand rule to forward packets to adjacent streets until the packet reaches a point where the distance to the destination is less than when it entered the recovery mode. At this point, greedy forwarding resumes towards the destination.

Figure 2.7 illustrates GPCR's forwarding modes. The green lines indicate a restricted greedy forwarding decision and the red lines represent a fall-back mode decision.

Simulations compare GPCR to GPSR and find that while GPCR consistently uses around 33% more hops than GPSR regardless of the distance between source and destination, it also delivers 10-40% more packets than GPSR at source-destination distances ranging from 500 m to 4000 m respectively. It is important to note that the propagation environment is not described in the GPCR paper,

except for a brief mention that the communication range of nodes is set at 500 m. However, it is very likely that the authors used a unit-disk model restricted to line-of-sight (LoS) paths since the protocol depends on nodes on one street not being able to communicate with those on another street.

2.3.3 Advanced Greedy Forwarding (AGF)

Advanced Greedy Forwarding (AGF) is a modification of the GPSR beaconing mechanism which broadcasts both a node's position and its velocity vectors as part of the beacon message [32]. This is stored as two bytes of information, the first being direction and the second being speed. Additionally, each node also considers its processing time when sending a packet and records this in the packet header. The velocity vector is primarily used to determine if a neighbour will be in reach when a packet is to be forwarded, given the last known position and velocity of the neighbour.

When the packet arrives at a node that is a neighbour of the destination, the node determines if the destination will be within range based on the position and velocity information in its table as well as its processing time and forwards the packet if this criteria is met. If not, a route-request (RREQ) is broadcast and either the destination or a node near the destination returns a route reply (RREP), at which point the packet is forwarded accordingly.

AGF is simulated using the probabilistic shadowing channel model, also known as the Log-Normal shadowing model [33], and a realistic micro-mobility based traffic simulator named MMTS which is described in detail in the paper. Two scenarios are tested: a city and a highway. AGF is compared against GPSR, AODV and another protocol introduced in the paper named AODV+PGF. Unlike most previous papers, these simulations found that GPSR performs very poorly, delivering only 5-10% of packets in both the city and highway environments. The authors speculate that this is due to the fact that GPSR often forwards the packet to the furthest possible neighbour in range, which is often

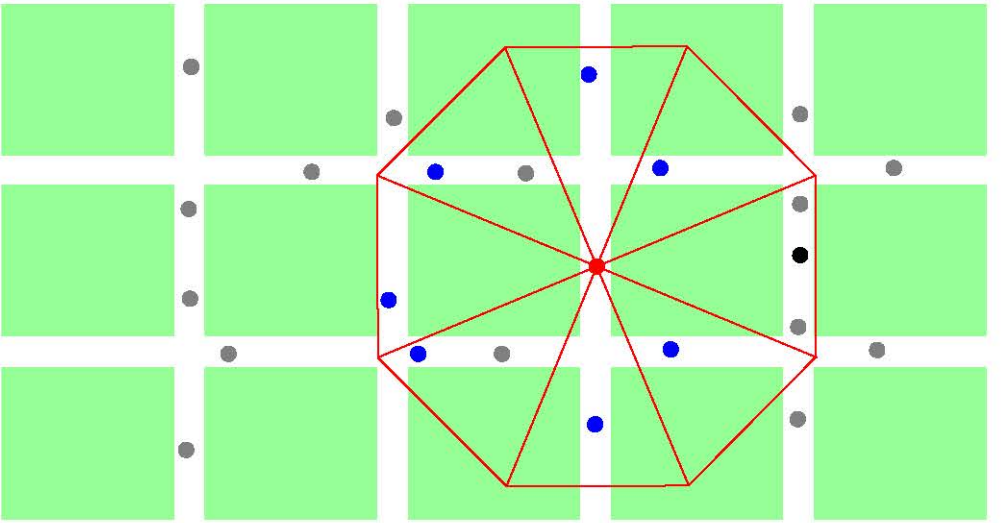


Figure 2.8: Abstract routing table in GRANT.

unavailable due to the use of the probabilistic shadowing model. AGF is shown to deliver between 30-50% of packets in the city and 30-70% of packets in the highway scenario.

2.3.4 Greedy Routing with Abstract Neighbour Table (GRANT)

Greedy Routing with Abstract Neighbour Table (GRANT) is a stateless routing protocol with a stateful fall-back mechanism which combines position based routing with an AODV-inspired recovery algorithm [34]. GRANT also introduces the concept of a multi-hop neighbour table. In their simulations, a 2-hop neighbour table is used. In order to reduce the overhead of transmitting a full neighbour table along with each beacon to maintain a 2-hop knowledge of the neighbourhood, the authors propose the 'Abstract Neighbour Table', which separates the plane around a node into discreet areas, and selects one node to represent each area. This is illustrated in Figure 2.8, where the red node's abstract neighbour table only includes the selected (blue) nodes from each region. This table is then forwarded to all neighbouring nodes.

The recovery strategy is essentially the same as is used in AODV, modified to

implement a restricted flooding algorithm. A node only re-broadcasts the route request if it is further from the destination than the originating node. This minimises flooding of the channel and also ensures that a potential path will route around any voids which may exist due to a greedy routing failure. When a node closer than the source is reached, it sends a route-reply and packets are forwarded along this established path.

The authors do not provide many specifics about the simulation environment. The propagation environment is modelled using the standard approach in VANET research, which assumes that spaces between streets are impenetrable and transmission range in LoS scenarios is static. There is no mobility modelled in the simulations. Instead, a series of nodes is distributed along the streets of a map of Karlsruhe, Germany and the nodes are periodically redistributed. Only the path-finding abilities of the protocols are tested, in terms of the shortest hop-counts, implying that a MAC layer is not simulated. GRANT is tested with varying parameters, with and without recovery enabled, against greedy and face-2 routing algorithms [35], which are similar to the greedy and perimeter modes of GPSR respectively. GRANT is found to consistently discover up to 50% more routes than a simple greedy algorithm.

2.4 Protocols with Street Awareness

This set of routing protocols utilise knowledge of street topology (in the form of predefined street maps) to aid in routing. Specifically, GSR and SAR introduce the concept of source routes anchored along street junctions, which has subsequently been employed by a number of other VANET protocols.

2.4.1 Geographic Source Routing (GSR)

Geographic Source Routing (GSR) is another seminal VANET routing protocol. It was one of the first to use city maps to route traffic along streets instead of

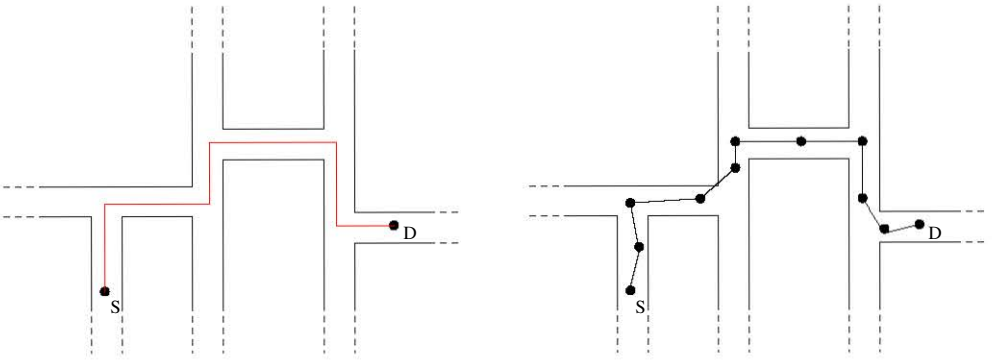


Figure 2.9: Junction-based route generated by GSR (left) and actual route taken by GSR (right)

considering a purely geographical approach [36]. This strategy has been adopted by many VANET routing protocols developed after the introduction of GSR.

In a GSR network, a node wishing to transmit a packet uses Dijkstra's algorithm to determine the shortest path to the destination using the city map [37]. This path, which is defined as an ordered list of street junctions to be traversed between the source and destination, is included in the packet header. Each node attempts to forward the packet to the nearest junction via participating nodes along the pre-determined path using a greedy approach. Nodes at, or near junctions allow the packet to switch streets, at which point the greedy approach is repeated to the next junction in the list.

GSR is considered a stateless algorithm in this thesis, even though packets carry information in the form of an anchored route. This is because the protocol does not attempt to *maintain* an active *network* route between specific mobile nodes in the network, only the static *physical* infrastructure (i.e. the network of roads) on which these nodes move.

The authors of GSR observe that most streets on a road are already planarised. This allows the use of the greedy forwarding approach in a restricted fashion, by forwarding packets to the end of the street in very few hops instead of traversing the street one node at a time. Therefore, with knowledge of street topology,

greedy routing is used in this restricted manner to forward packets towards the street junctions identified in the source routes. This is illustrated in Figure 2.9.

During simulations, the authors were also the first to assume a somewhat realistic radio propagation model for cities, modelling all spaces between streets (i.e. regions normally occupied by buildings) as obstacles. The propagation environment simulated to test GSR only allowed vehicles with direct line-of-sight to communicate. The performance of GSR was compared with AODV and DSR and was found to offer a slightly higher packet delivery ratio (PDR) than AODV and significantly higher PDR than DSR, with the gap widening as the distance between communicating nodes increases. The authors observe that the poor results of DSR are due to its significant signalling overhead in carrying an active source route.

2.4.2 Spatially Aware Routing (SAR)

Published just 3 months after GSR, Spatially Aware Routing (SAR) follows an identical approach to GSR in suggesting the use of geographical source routes [38]. However, it adopts the following recovery strategy in case the source route includes a disconnected segment due to a road on the source route being devoid of traffic:

1. Store the packet in a buffer;
2. Switch to greedy forwarding to the destination or until re-joining a link which is part of the original source route; or
3. Fully recompute the source route.

SAR may be directly compared to GPSR using city maps, although it uses a simple unit-disc radio propagation model (i.e., a constant static transmission range). With this model, it is found that SAR offers improvements in PDR of up to 13% over GPSR at radio ranges from 0-250 m, while significantly reducing delivery delay by an average of 80%. SAR also uses fewer hops than GPSR

at a radio range of 50 m, but the hop counts converge to the same number as the radio range increases to 250 m. This is possibly due to the use of the simplistic propagation model. As the radio range increases, the distribution of nodes becomes less relevant since nodes can easily communicate across streets.

SAR, when using a packet buffer, outperforms GPSR and SAR (without the buffer) in terms of PDR, but as expected, exhibits a significantly larger packet delay and a higher hop count on average. The authors also observe that GPSR requires significantly less routing overhead as radio range increases; this is due to the increasing success of greedy forwarding over perimeter-based forwarding, which requires a much smaller packet header (and also requires fewer hops).

2.4.3 GpsrJ+

GpsrJ+ streamlines GPCR for more efficient performance in grid-like cities [39]. Unlike GPCR, GpsrJ+ assumes that nodes have street-awareness. Each node advertises not only itself, but also a list of its neighbours and the road segments on which they are currently located. Therefore, each node is aware of the topology of its immediate 2-hop neighbourhood. Neighbours of coordinator or junction nodes which have a packet to send may therefore bypass the junction node altogether and forward the packet directly to a neighbour on an adjacent road if the opportunity presents itself. This is because they are aware of the junction node's neighbours, making it possible to anticipate the junction node's forwarding decisions. In most cases, this eliminates the need to add an extra hop at each junction, especially if nodes are forwarding along a straight line.

GpsrJ+ also significantly extends the recovery strategy of GPCR, in particular, exploiting the assumption that cities are arranged in a grid-like layout (i.e., that every junction is the intersection of four streets). In the original paper, a set of six possible scenarios forms the basis of the GpsrJ+ recovery algorithm. The algorithm itself follows the street-based face-routing approach, but uses two-hop knowledge of the network to bypass junction nodes where possible.

GpsrJ+ is compared to GPSR and GPCR in simulations using a fixed transmission range of 371 m, restricted to line-of-sight paths alone. Vehicular mobility is simulated with VanetMobiSim [27] and the simulations are performed on a grid-based map. GpsrJ+ is shown to perform 8-10% better than GPCR and up to 15% better than GPSR in terms of packet delivery ratio. While both GPCR and GpsrJ+ perform better than GPSR in terms of hop counts (as expected), GpsrJ+ is surprisingly shown to have a higher average hop count than GPCR. The authors reason that this is due to the hop-count metric not including failed packets. This results in GPCR showing a lower average hop count since it also delivers a lower fraction of packets than GpsrJ+.

2.4.4 Topology-assisted Geographical Routing (TO-GO)

TOpology-assisted GeOgraphical routing (TO-GO) uses a similar beaconing system to GRANT, in order to provide nodes with a 2-hop knowledge of their neighbourhood [40]. The amount of information in each beacon is limited to the furthest visible node in either direction unless the node is at an intersection. For nodes at an intersection, beacons include information about all the roads on which there is at least one visible neighbour. A target node is then chosen as the next hop, which may either be the node at a junction or the furthest node, within two hops, in the direction of the junction. In order to improve processing efficiency and reduce the size of beacons, all of the aforementioned neighbourhood information is stored and disseminated via a bloom filter [41].

In order to ameliorate the hidden node problem, the authors propose a brute force algorithm to reduce the set of neighbours to those that can hear both the source node and the target node. From this smaller list, the node with the most neighbours is then identified. The algorithm is reiterated to reduce the list of suitable forwarding neighbours to those that can hear the source node, the target node and the node with the most neighbours. The idea behind this approach is to preferentially forward the packet to nodes which are in a region of high node

density, in order to increase the chance of discovering a viable route.

When actually forwarding packets, TO-GO adopts the contention-based forwarding method first proposed in CBF [29]. The set selection process defined above ensures that nodes which receive the packet can hear each other and thus prevent duplicate forwards of the same packet. Packets are forwarded to all the members of the reduced set identified above, and a timer is set as for CBF. The node nearest to the target will therefore forward the packet first, upon which the other nodes which hear the successful transmission will drop the packet.

TO-GO is compared to GPSR, GPCR and GpsrJ+ on a grid map using mobility traces generated by VanetMobiSim and a transmission range of 250 m with a log-normal shadowing model, which is effectively the addition of a Gaussian-distributed fading channel to a unit-disc (i.e., static transmission range) model. Communication is limited to LoS paths only. The authors find that TO-GO performs as well as GPCR and GpsrJ+ when fading is ignored, but performs 50-70% better than GpsrJ+ as the effects of fading become more apparent. One key conclusion from this analysis is that routing protocols in general can be very sensitive to the channel propagation model and the fading model in particular.

2.5 Protocols with Statistical Connectivity Awareness

The following set of routing protocols introduce the idea of using statistical traffic information to find routes with a greater probability of connectivity. Individual streets on maps are rated based on historical vehicular traffic data or may be updated periodically by a traffic service.

2.5.1 Anchor based Street- and Traffic-Aware Routing (A-STAR)

Anchor-based Street- and Traffic-Aware Routing (A-STAR) expands on the basic idea of GSR by assigning weights to streets based on the likelihood of encountering vehicular traffic on them [42]. The protocol uses 'rated' maps which may

be either statistically or dynamically generated. The statistical maps are rated based on the presence of bus-routes, since it is assumed that bus-routes indicate major streets, which would attract more vehicles.

A-STAR also introduces a recovery strategy in the event of a node encountering a disconnected street. The street is marked as 'out-of-service' for a period of time and this information is appended to the packet header. The packet's source route is then recomputed and the outage information disseminated to any nodes which hear the packet on its new route to the destination, so that they can update their street maps appropriately.

In their simulations, the authors propose a mobility model named *M-Grid*, which simulates vehicles and buses on a regular grid. Vehicles tend to prefer streets with bus-routes. The authors also propose a similar radio propagation model to that assumed by GSR. A fixed transmission range is used, however, inter-vehicle communication is limited to direct line-of-sight paths. All spaces between roads are assumed to be filled with radio-opaque obstacles. The performance of A-STAR is compared to both GPSR and GSR. The authors found that GSR and GPSR did not perform significantly differently, and it is suggested that this was because of the regular grid-like arrangement, which was optimal for GPSR since there were no fork-junctions where GPSR could select the wrong path. Two versions of A-STAR were tested: A-STAR-SR, which used statistically rated maps, and A-STAR-DR, which used dynamically rated maps. The dynamically rated maps are generated instantaneously, and assume the presence of a sophisticated pervasive traffic monitoring and information dissemination service which can be accessed by participating nodes.

When compared to GSR, both A-STAR-SR and A-STAR-DR offer an improved PDR of between 10 and 40%, with the improvement in PDR increasing in proportion to node density. This number increases to a maximum of roughly 60% when a recovery strategy is used for disconnected roads. However, both A-STAR protocols exhibit higher end-to-end latency since they choose paths optimised

for packet delivery ratio instead of hop count. This delay increases as expected when recovery strategies are used, from roughly 10% to 50% in excess of GSR. A-STAR-DR delivers slightly more packets than A-STAR-SR, although the difference is minimal when recovery strategies are used.

2.5.2 Improved Greedy Traffic Aware Routing (GyTAR)

Improved Greedy Traffic-Aware Routing (GyTAR) is a traffic aware routing protocol belonging to the same family as A-STAR and STAR [43]. Similarly to A-STAR, with dynamically rated maps, GyTAR assumes the ready availability of real-time traffic information from a centralised traffic authority. Additionally, each road is rated based on both the traffic on it as well as the Manhattan distance to the destination, with the aim of selecting shorter and fully-connected paths. GyTAR also forwards packets one junction at a time instead of sending the entire source route in the packet header. This ensures that routing decisions are made at each junction using relatively up-to-date locally-gathered real-time traffic information. This approach is termed an *intersection-based* routing approach and differs from the anchor based approaches of GSR and A-STAR, since it opportunistically recomputes the route to the destination at each intersection, picking the most optimal street at that instant. Like AGF, GyTAR also uses velocity and position vectors to make better predictions about the connectivity of neighbours. GyTAR uses buffers to recover from situations where the node carrying the packet is the closest to the nearest junction without actually being at the junction.

GyTAR is simulated with and without its recovery strategy on a street map with a street-constrained random-waypoint mobility simulation. The propagation model is a simple unit-disc model, with a radio range of 266 m. The results show that GyTAR performs roughly 13% better than GSR at high node densities and roughly the same at lower node densities. Interestingly, without the recovery strategy, GyTAR performs much worse than GSR at low node densities and only

slightly better at high node densities.

2.5.3 Vehicle Assisted Data Delivery (VADD)

Vehicle-Assisted Data Delivery (VADD) is unique amongst the routing protocols reviewed in this chapter in that it is specifically designed for sparse networks [44]. While protocols such as SAR and CAR suggest buffering as recovery possibilities, VADD is intended for delay-tolerant networks, where buffering is a necessity. A significant amount of the protocol is therefore designed with this requirement in mind. Like A-STAR, VADD relies on statistically rated maps. VADD does not compute a route from source to destination directly, instead relying on an intersection-based approach like GyTAR.

The protocol operates in three separate modes: Straightway, Intersection and Destination. Straightway mode is used when a node is located on a street, and greedy forwarding is used in this mode to move the packet towards an intersection. When a node is within range of an intersection, it switches to intersection mode, where decisions are made about which street the packet should be forwarded onto next. When a packet is within range of its destination, it switches to destination mode, where greedy forwarding is once again used to get the packet to its destination.

Streets are rated statistically based on historic traffic information. Each street has an expected delay associated with it based on its length, vehicle density and average vehicle velocity. When a node is in intersection mode, the expected delay to the destination is computed for every path leading out of the intersection taking into account a series of probabilistic choices made at each subsequent intersection up to the destination. In order to manage the complexity of computing every possible path, a boundary is created encompassing the source and destination and an n x n linear equation is solved to determine the path with least delay, where n is the number of roads within the boundary. Once a delay is estimated for each possible path out of the intersection, they are prioritised

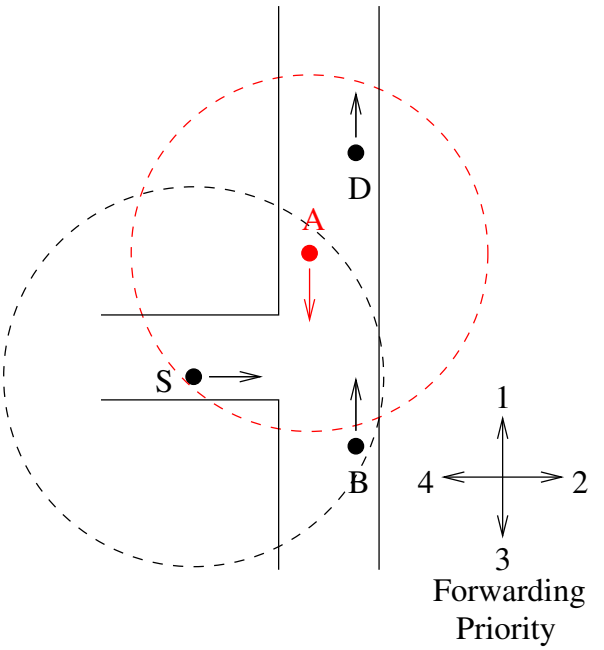


Figure 2.10: Intersection forwarding priority in VADD

based on the expected delay.

Two separate approaches for forwarding packets at an intersection are proposed. The first, location-first, only considers the location of a node when forwarding the packet. If a node is located on a road segment with the highest priority, the sending node forwards the packet to it. However, this may lead to routing loops, since a packet may be forwarded to a node leaving the street, thus carrying it away from the street with the highest priority. This node may then recognise the sender as being the only available node and on a road with a higher priority than its own, forwarding the packet back to the sender. To eliminate this problem, the authors also propose a direction-first approach. In this approach, the packet is forwarded to a node which is moving in the direction with the highest priority. These two approaches are named L-VADD and D-VADD respectively.

This is illustrated in Figure 2.10, where node S has two options when forwarding a packet at the intersection. L-VADD would dictate forwarding the packet to node A, while D-VADD would dictate forwarding it to node B. A loop would

occur if node D was out of range of A, at which point A would have to buffer the packet and forward it to S after crossing the junction. A hybrid approach (H-VADD) is also proposed in the paper, which uses loop-detection to switch from L-VADD to D-VADD if a loop is detected.

The authors compare VADD to GPSR (with a buffer) and DSR on a sparsely populated grid-like city map. The propagation environment is a simple static transmission range limited to 200 m. A street-restricted waypoint mobility model is used in which each vehicle chooses the path with the least delay to its destination, thus mirroring the approach of the routing protocol. With the exception of L-VADD when it encounters loops, the VADD protocols outperform GPSR by between 25 and 40% at low node densities. At higher node densities, there is little difference between the loop-free VADD protocols and GPSR. All the protocols outperform DSR by at least 500%. While VADD doesn't exhibit significantly different delay to a buffered GPSR protocol at lower node densities, the source-destination latency is reduced by about 50% at higher node densities.

2.5.4 Geographic Stateless VANET Routing

Geographic Stateless VANET Routing (GeoSVR) [45] is similar to A-STAR in the use of anchor-based routes and statistical connectivity determination. Instead of A-STAR's reliance on statistically rated maps or assumption of a 'traffic authority', GeoSVR assumes that wider roads would have a higher chance of having traffic since they are designed to carry more traffic. The authors suggest that dynamic connectivity awareness is unnecessary because vehicle density on roads is stable in VANETs, although no justification is provided for this argument. The map is then weighted according to this assumption and paths are chosen using Dijkstra's algorithm.

The validation of GeoSVR is unique in that it is the only protocol in this review which is tested in both simulations and a real environment. The simulations are performed in NS-2 and compare GeoSVR to GPSR and AODV [46]. The

propagation environment is simplistic and employs a static radio range of 250 m. Two mobility models are considered. The first is a simple highway model where vehicles move at constant speeds over a 1.5 km road. The second is a regular grid and simulates 1000 vehicles travelling around the grid. There are minimal details on node density or other specifics of the vehicular simulation.

The authors find that GeoSVR outperforms both GPSR and AODV in terms of packet delivery ratio by between 25% to 100%, with the absolute PDR approaching 100% in the highway scenario, with a latency 20-100% higher than GPSR but around half the latency of AODV in the highway scenario. The difference in performance is much smaller in the city scenario, with GeoSVR only performing marginally better than AODV in most cases.

The real-world experiments were carried out with three cars driving around a city. The value of of this experiment is unclear due to the very limited number of nodes involved. Regardless, the authors find that nearly 100% of the packets are delivered, which is not a surprising result for such a simple network.

2.6 Protocols with Real-Time Traffic Awareness

The following protocols further extend the approaches of the aforementioned protocols by incorporating real-time traffic detection mechanisms into the protocols themselves.

The need to determine connectivity of the route is desirable since the shortest route is not necessarily fully connected. This is illustrated in Figure 2.11. Under protocols such as GSR or SAR, both paths SAD and SBD are more 'ideal' than the eventually traversed path and would be selected by these protocols even though both paths are disconnected. The protocols described in this section are designed to address this issue. Several different strategies are proposed, from simply checking streets for connectivity during the route formation process to maintaining connectivity across an entire route.

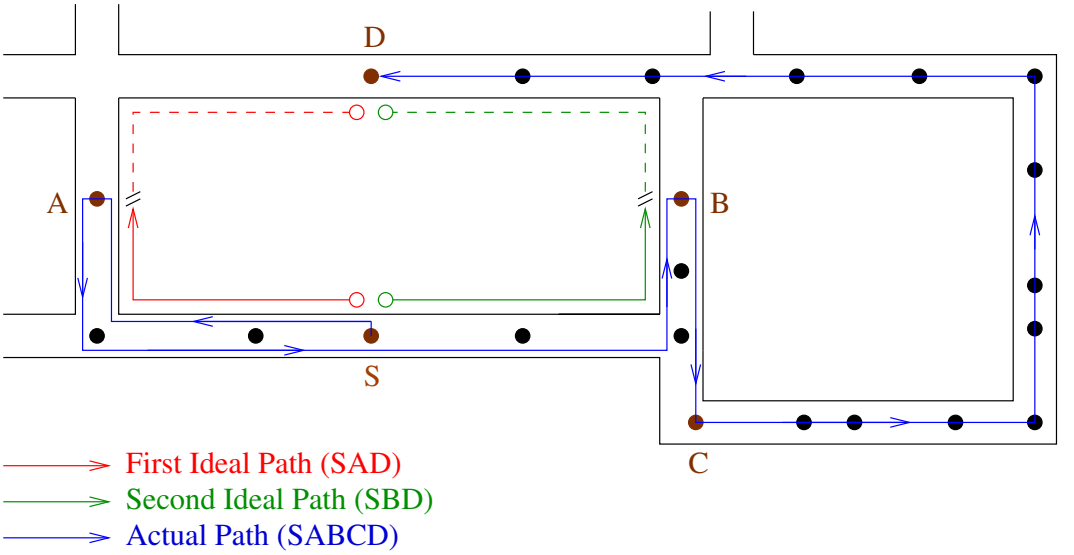


Figure 2.11: Shortest paths vs. connected paths in VANETs

2.6.1 Multi-Hop Routing for Urban VANETs (MURU)

Multi-Hop Routing for Urban VANETs (MURU) is a reactive routing protocol in the style of AODV, with several enhancements to the distance vector algorithm to improve its performance in VANETs [47]. Being a reactive and stateful protocol, MURU automatically takes connectivity into account.

A new metric is introduced called the *expected disconnection degree* (EDD), which uses a combination of path-loss, expected time of contact, expected bit error-rate and the estimates of the node's most probable velocity to provide a metric suitable for VANETs. The expected velocities (speed and direction of motion) are calculated with the use of a Markov chain which assumes that a vehicle may only move in the cardinal directions, and models their turning probabilities accordingly. While this is a simplistic assumption, the authors suggest that it covers the vast majority of cases in VANETs.

An optimisation is also made to the route request algorithm, which can be constrained to a specific area to take advantage of the street topology in reducing control packet overhead. The shortest path to the destination is calculated as

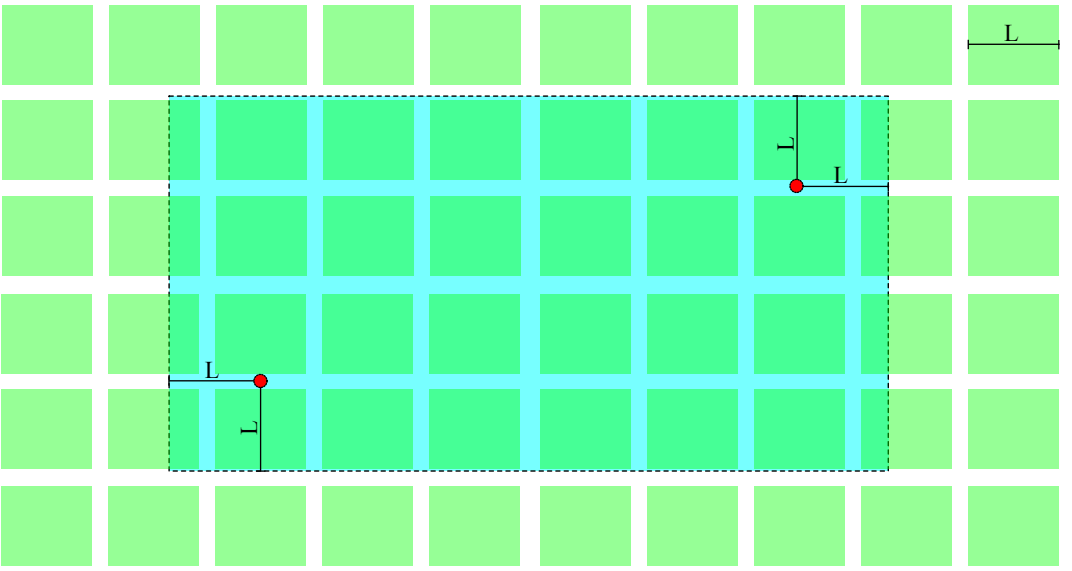


Figure 2.12: Broadcast-range restriction in MURU

a segmented line with turning points. In this case, due to the assumption of a grid like map, a rectangle bound by the source and destination nodes is defined. The retransmission of this broadcast is then restricted to an area defined by this rectangle with each side extended by the length of one block of the grid. This is illustrated in Figure 2.12, where the broadcast range for the source-destination pair (red nodes) is indicated by the dotted rectangle, where L is the length of a single block.

The final optimisation is a contention-based back-off mechanism similar to the one described in CBF [29] which aims to restrict broadcast overhead further. Each node that receives the route request holds the packet for a time which is proportional to the estimated EDD metric. If the node overhears another node broadcast a packet to a neighbour, it overhears the packet and checks the EDD of that link. If it has an EDD that is larger than the one seen on that link, it simply discards the packet.

MURU is simulated in NS-2 on a simple 7×7 grid in which the streets are spaced 100 m apart. A simple mobility model is used where vehicles travel at an average

speed along the street grid. There is no information on the propagation model used, but it is assumed to be a simple static transmission range. Two scenarios are run, one with a static source-destination pair and the other with three mobile source-destination pairs. In each case, MURU is found to outperform (in terms of PDR) GPSR by between 0 and 100%, AODV by 100% to 400% and DSR by a similar margin. None of the results from any of the compared protocols seem to vary significantly with an increase in node density, which is puzzling. However, MURU appears to deliver a significantly higher fraction of packets as the node density increases. MURU also requires a relatively small amount of control overhead, similar to AODV and DSR, while GPSR's control overhead is proportional to the number of nodes in the simulation due to its use of periodic beacons. It should be noted however that the control overhead is measured in terms of the number of packets broadcast and does not account for the size of these packets. Additionally, the packet overhead in AODV, DSR and MURU is low in this paper due to the very low number of transmitting nodes used, and will increase if more contending traffic flows are introduced.

2.6.2 Spatial and Traffic-Aware Routing (STAR)

The Spatial and Traffic-Aware Routing (STAR) protocol proposes a similar anchor-based approach to A-STAR in terms of source routing and traffic awareness [48]. While A-STAR relies on a traffic service for dynamically rated maps, STAR attempts to dynamically ascertain traffic conditions on streets in an ad hoc manner. STAR also limits the number of source-route junctions that are transmitted with the packet to limit control overhead. It is therefore an intermediate strategy between the anchor-based approach of A-STAR and the intersection-based approach of GyTAR. The packet carries this limited source route in its header. A further path is calculated when the packet reaches the last junction on its list.

Traffic awareness in STAR is divided into two situations - a very high number of

vehicles and the total absence of vehicles. The authors note that it is desirable to forward packets to streets with a large number of vehicles due to the higher probability of connectivity. Each node maintains a list of neighbours in each cardinal direction from its present position. This information is disseminated to neighbours during beacon transmissions, with the spread of this information limited by a time to live (TTL), which defines a maximum number of hops that the packet may traverse before being dropped. Stale nodes are removed when their associated timer expires.

Simulations compare STAR to SAR, GPSR and a purely greedy protocol. A static transmission range of 250 m is used for nodes distributed in a regular grid-like environment. A simplified street-constrained random-waypoint model is used to simulate mobility. The results indicate that while STAR outperforms both GPSR and SAR, SAR performs poorly compared to GPSR in this scenario. The authors acknowledge that this is due to the use of an imperfect radio propagation model where communications across streets gives GPSR an advantage over SAR, which forwards packets strictly along streets. The authors note that STAR performs between 0 and 80% better than GPSR in terms of packet delivery ratio, with the performance gap widening significantly as traffic density drops. While STAR successfully delivers more packets to the destination, it also encounters many more collisions at the MAC layer due to its bias towards paths with a higher number of nodes (and therefore higher contention). Furthermore, the average beacon size in STAR is almost 100 bytes due to the carriage of traffic information, which is a 400% increase compared to GPSR and SAR.

2.6.3 Connectivity-Aware Routing (CAR)

Connectivity-Aware Routing (CAR) may be described as a hybrid between a reactive distance-vector protocol such as AODV and a position based routing protocol such as GPSR [49]. It inherits AGF's beaconing mechanism which, includes a node's position and velocity vectors. Additionally, CAR also includes

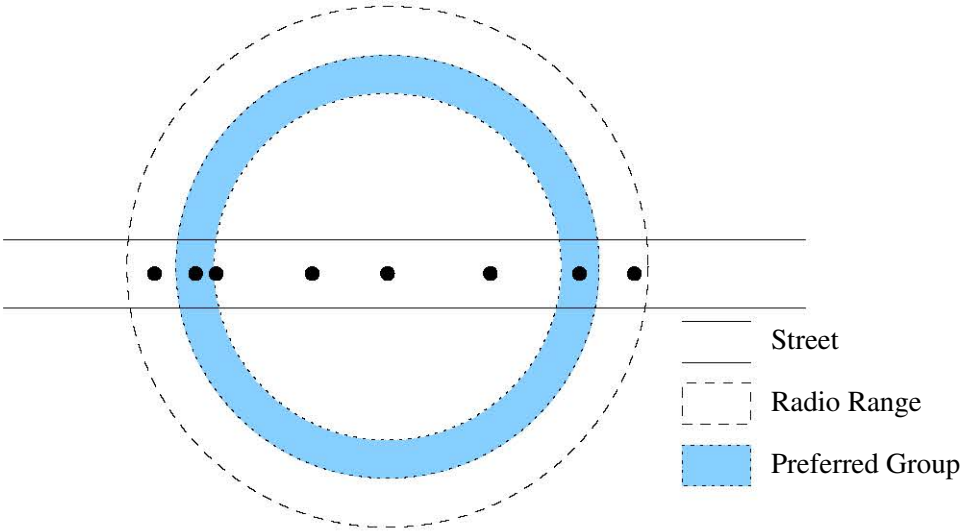


Figure 2.13: Preferred group broadcasting on a road

an adaptive beaconing mechanism which varies the beaconing interval based on the number of neighbours in the vicinity of a node. The greater the number of neighbours in a node's vicinity, the greater the beaconing interval. This is used to reduce the beaconing load on the network in dense areas without affecting the ability to readily reach a neighbour. The authors reason that in a dense area, the need to know the exact positions of neighbours is less important since there will be several to choose from.

CAR is also unique in that it suggests a means of determining the location of the destination using a reactive route finding approach instead of relying on a location service like the other protocols covered thus far in this chapter. It uses preferred group broadcasting (PGB), first described in [32] to find the location of the destination node. PGB is a modification of the AODV route discovery algorithm which is adapted for city scenarios. It restricts route request rebroadcasts to a particular region around a node, known as the 'preferred group'. This is illustrated in Figure 2.13. The region is not at the very edge of the node's radio range in order to mitigate the effects of fading while still re-broadcasting the route request as far from the node as possible.

The connected path to the destination which has thus been computed is represented as an anchor-based source route. An anchor is added to the path if there is a significant change in direction from the previous node. An anchor in CAR is actually composed of two positions corresponding to the positions of the node before and the node after the change in direction. The final path, stored in the packet header, also carries the estimated travel time between anchors.

CAR also introduces the concept of *standing* and *mobile guards*. A guard is a special block of information which is appended to a node's beacon messages. A standing guard is broadcast by a destination node which changes direction, and contains its old and new velocity vectors as well as the coordinates where the guard was activated. This guard is disseminated to the nodes in the vicinity of the node and is also maintained by them. When a node carrying the guard receives a packet, it adds the guard position as an anchor point and recomputes the estimated position of the destination before retransmitting the packet. If a destination node was actively moving away from the routing path before changing direction and is further away from the path than its own radio range, it activates a travelling guard, which 'moves', via beacon messages, towards its old location and forwards packets towards the new location.

Guards are used as a temporary measure until a route maintenance message reaches the source. When a destination node changes direction, it sends a route maintenance message back to the source node with its new position and velocity vectors. Once this message reaches the source node, it recomputes the anchor-based path to take this into account, eliminating the need for guards to maintain the route.

Despite the use of sophisticated mechanisms to maintain a route, a gap may form in the anchored path due to a break in traffic or interference in the channel. The proposed recovery strategy involves first buffering the packet until a preferred neighbour can be contacted or a timer expires. If the timer expires, the node initiates a recovery process which involves using the route discovery algorithm

described above to recompute a route from its present location to the destination, which is similar to the recovery processes of most anchor-based routing protocols surveyed thus far.

CAR is simulated in both a city scenario and a highway scenario with the probabilistic shadowing (log-normal) model, with a maximum transmission range of 400 m in the city and 500 m in highways. However, the communication range of vehicles is not restricted to a line-of-sight path. CAR is compared, with and without its recovery mechanism, to GPSR and AGF and is found to substantially outperform those protocols, both in terms of PDR and latency. The simulations indicate a 300-800% improvement in packet delivery and 40-60% reduction in delay, with the gap widening as node density increases. CAR also significantly reduces routing overhead in dense vehicular scenarios due to the adaptive beaconing mechanism.

2.6.4 Adaptive Connectivity-Aware Routing (ACAR)

The Adaptive Connectivity-Aware Routing (ACAR) protocol combines statistical connectivity prediction for initial geographical route selection with dynamic connectivity awareness for route recomputation and maintenance [50]. Estimation of the connectivity of a particular road is based on the average transmission range of a node and the estimated number of vehicles on the road. The estimated number of vehicles is in turn based on statistical data.

While the initial route is selected based on statistical information, during packet transmission, each node along the route gathers data about the number of neighbours between itself and the next hop. This information is collected for the entire path and appended to the packet header, which is subsequently received at the destination. The destination then informs the source if the difference between the estimated connectivity and the actual connectivity exceeds a particular threshold. Buffering is used to carry packets over any temporary disconnections, and in practice it is expected that the information gathered in successive packet

transmissions will lead to the route selection process converging to an optimal route over several iterations.

While the packet is in transit along the anchored route, the choice of next hop is made based on the expected packet error rate for the rest of the route. This in turn is based on the expected transmission count metric described in [51]. The metric is basically a measure of packet success rate, which is measured as the number of HELLO packets successfully exchanged between neighbours.

ACAR is simulated in NS-2 [46] and its performance is compared with CAR, GSR and GPSR in a realistic city map with traces generated by VanetMobiSim [27]. The propagation model used is a static transmission range of 250 m. The authors find that ACAR outperforms CAR, GSR and GPSR by margins of roughly 20%, 30% and 140% respectively in terms of PDR. Latency is similar to that of CAR, but reduced by 60% and 67% when compared to GSR and GPSR respectively.

2.6.5 Road Based Routing using Vehicular Traffic (RBVT)

Road-Based routing using Vehicular Traffic (RBVT) describes two separate protocols, RBVT-P (proactive) and RBVT-R (reactive) [52]. The terms proactive and reactive describe the approach of the protocols with respect to traffic detection, *not* route discovery (distinct from the use of the terms in ad hoc network literature). The main idea behind RBVT is similar to CAR in that it aims to establish connected paths between source and destination.

RBVT-R uses a very similar location discovery mechanism to CAR, which also doubles as a route discovery mechanism. When a source node has a packet to send, it broadcasts a route discovery (RD) packet which is forwarded on by the node's immediate one-hop neighbours. When there are two nodes on a road segment that both hear the broadcast, the node furthest from the source re-broadcasts the packet while the other node remains silent. This is an effort to reduce the 'broadcast storm' issue and is similar in design to the PGB algorithm

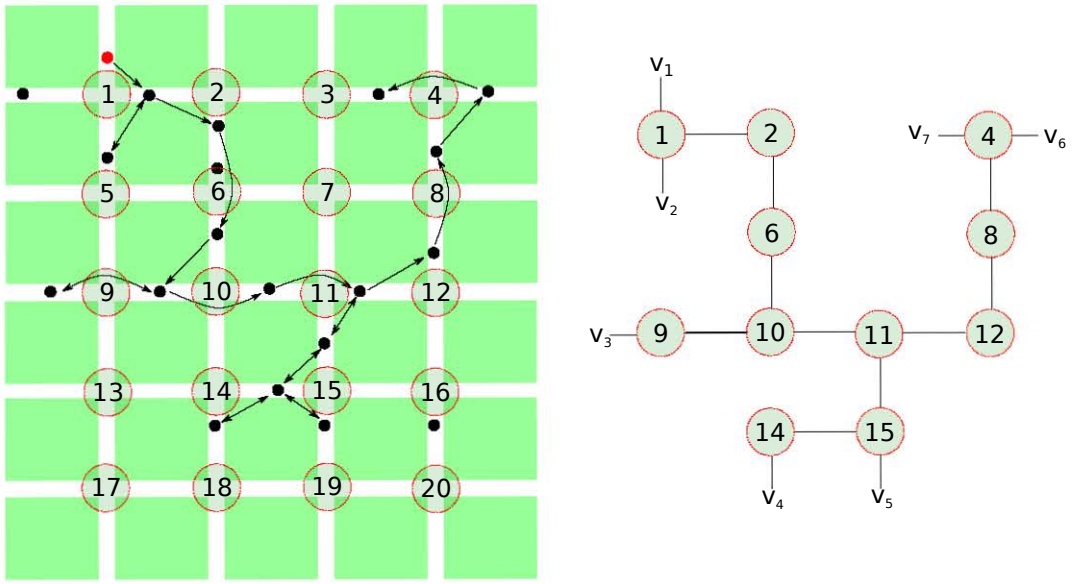


Figure 2.14: Proactive connectivity determination in RBVT-P

used by CAR. As the RD broadcast makes its way around the city network, only the intersections that it traverses are recorded, instead of individual nodes, thus creating an anchor-based route based on the present connectivity of the network. Once again, this approach mirrors CAR, except with only one anchor point maintained per intersection. Upon receiving an RD packet, the destination sends a route-reply (RR) back to the source along with its own position.

Unlike CAR, velocity vectors of nodes are not used to estimate their current position. In order to maintain routes, if either the source or destination move onto a different junction, they send a route update packet to inform the other of the change. In the case of a route error due to a broken link, the node which detects this error simply sends a route-error message back to the source. At this point, the source buffers packets and re-attempts delivery a pre-defined number of times before discarding the route and restarting the RD process.

In summary, RBVT-R is almost identical to CAR, but without the use of velocity vectors for position estimation or guards for route-maintenance.

RBVT-P assumes the availability of a location service for destination location

discovery. In RBVT-P, a 'connected' road-topology is determined using connectivity packets (CP). These packets are transmitted by nodes based on the vehicular density, the historic traffic information in the area and the time since the last CP update. Each CP is allocated a region which it must traverse. A node generates a CP, which traverses the network using an approach similar to a depth-first search [53]. As the packet crosses road intersections, they are added to a stack of 'connected intersections'. When the node encounters a disconnected segment, it returns to the last intersection and continues along a different path, marking a 'virtual intersection' where it encountered the void. Once the CP explores all the defined regions, the node which last received it extracts the information and broadcasts it as part of a route update (RU) packet, which is disseminated to all nodes in the explored region. Figure 2.14 illustrates this information gathering step (left) and resulting connectivity graph (right).

When a packet needs to be sent, only connected paths are considered. A source-routing strategy is then used to find the shortest path to the destination via connected nodes. Each time a node reaches an intersection the source route is recomputed to leverage the more up to date information present at the intersection. In case the routing strategy fails, greedy forwarding is used as a fall-back to move the packet back to another intersection, where a different routing decision may be made.

Performance of both RBVT protocol variants are compared to GPSR, OLSR, AODV and GSR at different packet transmission rates and node densities. RBVT-R was found to deliver up to 28% more packets than GPSR, GSR, RBVT-P and AODV, which all exhibit similar PDRs at low node densities. GSR performs on par with RBVT-P at all node densities, while GPSR and AODV deliver up to 25% fewer packets than GSR at very high node densities.

RBVT-P exhibits an overall lower delay and hop-count than the other protocols evaluated, which potentially indicates better path-finding abilities due to proactive connectivity detection, but at the cost of decreased PDR, which may be a

consequence of collisions resulting from the increased packet overhead. GSR and OLSR exhibit relatively high delays, but RBVT-R is on par with GPSR and AODV at all node densities. Other than RBVT-R, which exhibits a significantly larger hop-count, all the protocols exhibit a similar end-to-end hop-count.

2.6.6 Landmark Overlays for Urban Vehicular Routing Environments (LOUVRE)

Landmark Overlays for Urban Vehicular Routing Environments (LOUVRE) is similar in principle to CAR and RBVT in that it measures connectivity on roads in order to find a connected path between source and destination; however, the way LOUVRE achieves this is quite different [54]. LOUVRE takes a proactive approach to traffic discovery. Each node records the traffic density on the road that it is currently on, and maintains a similar record of all roads that it has visited along with a timestamp indicating the freshness of the data for each road. This is stored as a list with a predefined maximum size in the node's memory. When the list becomes full, the road with the most out of date information is discarded. Additionally, nodes may update information on a particular road when they receive a broadcast with more up to date information.

Traffic density is estimated based on the node's radio range, the number of neighbours in its vicinity that identify themselves as being on the same road, and the length of the road. This information is broadcast as part of the node's periodic beacon messages along with a list of the node's neighbours. Nodes located at junctions do not broadcast neighbour info, but broadcast all the information they have on traffic densities on roads visible to them.

This information is represented as a link-state graph, where each road is a node and the edges between nodes represent connectivity between those two roads. Roads are considered connected once the traffic exceeds a predefined threshold. When a packet is to be forwarded, the route between the source and destination is calculated using Dijkstra's algorithm on the link-state graph. The packet

is subsequently forwarded using an anchor-based approach similar to GSR and A-STAR, where greedy forwarding is used between intersections.

In order to reduce the overhead in each broadcast packet and the amount of information that each node maintains about the network, the network is divided into a grid. Each node in a grid only maintains traffic information about its own grid area, with boundary nodes maintaining information on adjacent grids as well. Additionally, if a node consistently receives the same density information for a particular road, it assumes that a stable state exists and stops broadcasting information on this road until the density falls below a threshold.

LOUVRE suggests two recovery strategies in case a packet encounters a void. The first is to route back to the previous road and attempt to find an alternative path to the destination which avoids the void, and the second is to buffer the packet until a more suitable neighbour is found. Packets are dropped after a timeout if an alternative route is not available.

LOUVRE's performance is evaluated and compared with GPSR and GPCR on a 1 km^2 city map. The propagation environment is a unit-disc model restricted to line-of-sight paths. The specific radio range used in the simulations is not provided. Of the 100 nodes in the simulation, only 20-50% are mobile, with the rest being static and located at intersections. LOUVRE was found to offer a packet delivery ratio 33-100% better than GPCR and 0-40% better than GPSR. It also exhibits significantly lower latency than GPSR and lower average end-to-end hop-counts than GPSR. However, on both latency and hop-count metrics, GPCR still offers superior performance. It should be noted however, that the use of 50-80% of static nodes at intersections would greatly influence protocol performance and seems unrealistic for a real VANET.

2.6.7 Back-Bone Assisted Hop Greedy (BAHG) Routing Protocol

The Back-bone Assisted Hop Greedy (BAHG) routing protocol utilises an algorithm which aims to minimise the hop-count of any particular path, resulting in what the authors term a 'hop-greedy' routing protocol [55]. As with CAR and RBVT-R, instead of relying on an external location service, BAHG has a built-in reactive mechanism, using route requests and replies similar to AODV for determining the position of a destination. Once the position is determined, a route is computed through road intersections and maintained by 'back-hone' nodes which are a superset of the coordinator nodes first described in GPCR.

BAHG defines four categories of back-bone nodes: stable, primary, secondary and 'void-guard' nodes. Stable, primary and secondary back-bone nodes are nodes which are located at or very close to intersections. A stationary node closest to the intersection identifies itself as a stable node. Primary and secondary nodes are nodes that are travelling through an intersection. Primary nodes are closer to the intersection and nominate a secondary node which is further behind to take over its role when it leaves the intersection. This process is repeated, with the secondary node becoming the primary node when reaching the intersection and nominating another node, located further back, to take its place.

Additionally, nodes located at either ends of a void on a street, i.e., an area between two nodes on a street that is greater than the maximum connection distance, can declare themselves as 'void-guard' nodes. A 'void-guard' node informs neighbouring back-bone nodes about the presence of a void. Furthermore, BAHG also uses a reactive location finder like the one proposed in CAR, except applying the hop-greedy algorithm to destination reply messages, since the location of both the source and destination are known at this point. The destination tracks its own movement and sends a route update message to the source if it moves significantly from its previous position - behaviour which is also part of the CAR protocol. Since the destination carries out the function of route maintenance, BAHG is considered to be a semi-stateless protocol.

When packets are forwarded, back-bone nodes are preferred targets since they maintain communication history and act as packet buffers. A mobile forwarding node prefers a stable back-bone node while a stable forwarding node prefers a mobile back-bone node. The stable node acts as a packet buffer in the absence of a suitable forwarding node away from the intersection. Packets are forwarded across a connected graph, where both the hop-count and the degree of connectivity of any edge are used to determine an optimal path to the destination, anchored by street junctions and maintained by back-bone nodes.

BAHG is simulated on a realistic city map and tested against GPCR and GyTAR. The propagation model used for BAHG takes non-line of sight paths into account, characterising them as undergoing a constant attenuation of 5 dB on top of an equivalent line-of-sight path-loss value. BAHG consistently demonstrates up to 70% less end-to-end delay than GyTAR and 85% less delay than GPCR. BAHG exhibits significantly improved PDR compared to GPCR and GyTAR, especially as the distance between source and destination increases or the packet sending rate increases. BAHG delivers up to 50% more packets than GPCR and 30% more packets than GyTAR.

2.7 Infrastructure Assisted Routing Protocols

The two protocols discussed in this section use road-side infrastructure to assist with the routing process. Despite the use of infrastructure, they are still considered ad hoc routing protocols since the infrastructure is not involved in network formation or maintenance, but rather only routing assistance.

2.7.1 Intersection-Based Geographical Routing (IGRP)

The Intersection-based Geographical Routing Protocol (IGRP) is designed to route packets through a VANET to the nearest Internet gateway while maintaining strict Quality of Service (QoS) constraints [56]. The QoS metrics used in

IGRP include bit error rate (BER), hop-count and delay. All routing decisions in IGRP are made by the gateway which considers the connectivity of paths. Each road segment is statistically rated with a 'connectivity probability' using the average speed of nodes and average node density on each lane. This probability is then used to create anchored routes between the source and the destination through a series of intersections.

The routing gateways in IGRP also track the positions of all the mobile nodes, with nodes sending gateways an update packet if they are more than one maximum transmission range away from their previously reported position. The gateways then proactively compute multiple routes to each mobile node, anchored at street intersections. When there is a packet to be sent, the mobile node consults the nearest gateway, which then provides it with the statistically most optimal path. Greedy forwarding is used between successive intersections.

Additionally, the gateway uses the node density in a particular street to adapt the transmission range of nodes in the area with the aim of optimising between connectivity range and bit-error-rate, both of which increase with transmission range. IGRP also uses loose clustering to ensure that multiple mobile nodes can reuse the same pre-determined route, especially if they are sending packets between a nearby source and destination. Each mobile node first queries its neighbours for a route to the destination and the query is relayed to the gateway only if the neighbours do not have the required information.

IGRP is tested in Matlab against GPSR and GPCR, both with and without adaptive transmission range enabled. A simple unit-disc propagation model is used and IGRP is found to exhibit a 30-60% lower hop count compared to GPSR or OLSR [57], and 15-40% lower hop-count compared to GPCR. Interestingly, without the advantage of an adaptive transmission range, IGRP exhibits slightly elevated hop counts compared to GPCR. IGRP is also found to decrease end-to-end latency by 95-98% over OLSR, GPSR and GPCR with a low number of vehicles, though this advantage diminishes as the number of vehicles is in-

creased. Once again, the use of an adaptive transmission range accounts for a 92% reduction in delay, making it the distinguishing feature of the protocol.

The benefits of IGRP appear to result from the use of external infrastructure to make both routing decisions and adapt the transmission range of nodes. It is therefore unclear how, if at all, the comparison to purely ad-hoc routing algorithms such as GPSR or GPCR is useful.

2.7.2 Static-node Assisted Data-Dissemination Protocol for Vehicular Networks (SADV)

Static-node Assisted Data-dissemination protocol for Vehicular networks (SADV) is a modification of the VADD protocol, where static nodes or road-side units are added at intersections to deal with conditions where node density is very low [58]. SADV consists of three separate modules: Static Node Assisted Routing (SNAR), Link Delay Update (LDU) and Multi-Path Data Dissemination (MPDD).

SNAR is very similar to the VADD protocol in that it consists of a greedy routing component *between* intersections and a decision making component *at* intersections. Each intersection has a static node which makes packet forwarding decisions based on the estimated delay between adjacent nodes, which is calculated using the average speed and density of nodes and the length of the road. When a packet arrives at a static node, it forwards it onto the optimal path if there's a node available. If there is no node, it buffers the packet. As the buffer fills up, the authors propose an algorithm called 'least delay increase', which ranks packets according to the paths available, with packets with more optimal paths to their destination available being ranked higher. For example, if the optimal path for a packet is unavailable, but the second most optimal is, the packet is ranked '2'. The highest-ranked packets are then forwarded and removed from the buffer, with packets only being dropped if there is no forwarding path available.

SADV obtains real time traffic statistics, specifically the delay associated with a

link, using the Link-Delay Update (LDU) module. Each packet, when leaving a static node towards another one, carries the departure time in its header. This way, the receiving node has a real-time delay estimate for a particular path. Once an updated delay estimate is available, the information is broadcast and carried by nearby nodes to other static nodes. Tight timing synchronisation is assumed (e.g. from GPS).

The Multi-Path Data Dissemination module simply sends data along multiple paths between static nodes. If delay information on a particular link is out of date, packets may be sent on more than one link in order to optimise for delivery. The mobile nodes simply act as greedy carriers between static nodes.

The performance of SADV is evaluated and compared to VADD on a realistic map with a similar mobility model to that used in the original VADD paper. Each node has a static maximum transmission range of 200 m. Packet delivery delay is found to be significantly lower (30-60%) across the different classes of SADV (with SNAR, LDU and MPDD). The authors find that even deploying a partial set of static nodes results in a significant reduction in packet delivery delay.

2.8 Summary of Reviewed Protocols

Table 2.1 summarises the aforementioned protocols, listing their routing strategies and other characteristics. The routing strategies are described as either greedy, anchor-based, intersection-based or stateful.

Both anchor-based and intersection-based protocols use the greedy algorithm between intersections, but differ slightly in their approaches. While an anchor-based protocol computes a path from the source to the destination anchored at intersections, an intersection-based protocol simply forwards a packet to the nearest intersection and makes forwarding decisions at the intersection.

Most intersection-based protocols make opportunistic routing decisions, taking

into account the traffic on a street and the location of the destination when determining the direction in which to forward the packet beyond an intersection, although some may use a path-cost minimisation algorithm such as Dijkstra's algorithm [37] or the A* search algorithm [59] on successive intersections.

Most anchor-based protocols employ one of the path-finding algorithms described above to pre-compute the street-level route from source to destination. However, there are stateful protocols such as CAR which use an anchor-based approach along with a reactive location discovery mechanism, where the reactive mechanism is directly responsible for finding a connected anchor-based path. These algorithms are denoted 'reactive anchor' algorithms.

Table 2.2 provides a short summary of the simulation parameters of the aforementioned protocols.

2.9 Observations and Conclusions

As seen in Table 2.1, each of the reviewed VANET protocols (excluding AODV and DSR) relies at least partially on greedy routing, with 22 of the 23 relying on greedy routing between intersections using an anchor-based or intersection-based approach. Most of the protocols deal with routing failures by either recomputing the route (8) or buffering the packet (8), with many protocols suggesting multiple possible strategies.

Additionally, 19 of the protocols are completely stateless, with only 3 protocols - MURU, CAR and RBVT-R - performing both route discovery and maintenance between source and destination. BAHG is considered partially stateless since destination nodes in BAHG maintain the end-point of the route by sending location update packets to the source.

Table 4.2 illustrates the testing methodology of the reviewed protocols, including the type of propagation and mobility models used. The aforementioned table also illustrates the key protocols in this review, which characterise specific milestones

Table 2.1: Summary of Routing Strategies and Characteristics of Reviewed Protocols

Protocol	Stateless	Routing Strategy	Recovery Strategy	Connectivity Awareness	Route Discovery	Route Maintenance	Unique Features
AODV [17]	No	Stateful	None	Reactive	Reactive	Reactive	None
DSR [18]	No	Stateful	None	Reactive	Reactive	Reactive	None
GPSR [19]	Yes	Greedy	Perimeter Routing	None	None	None	None
GSR [36]	Yes	Anchor-Based	Reroute	None	Dijkstra	None	None
SAR [38]	Yes	Anchor-Based	Buffer/Reroute/Greedy	None	Dijkstra	None	None
CBF [29]	Yes	Greedy	None	None	None	None	Beaconless Routing
A-STAR [42]	Yes	Anchor-Based	Reroute	Statistical	Dijkstra	None	None
STAR [48]	Yes	Anchor-Based	Reroute	Proactive	Dijkstra	None	None
GPCR [31]	Yes	Intersection-Based	Face Routing	None	None	None	None
AGF [32]	Yes	Greedy	None	None	None	None	Velocity Vectors
MURU [47]	No	Stateful	None	Reactive	Reactive	Reactive	None
GpsrJ+ [39]	Yes	Intersection-Based	Face Routing	None	None	None	Two-Hop Neighbours
GyTAR [43]	Yes	Intersection-Based	Buffer	Idealistic	Opportunistic	None	None
CAR [49]	No	Reactive-Anchor	Buffer/Reroute	Reactive	Reactive [32]	Reactive	Reactive Location Finder
GRANT [34]	Yes	Greedy	Dist-Vector	None	None	None	Two-Hop Neighbours
VADD [44]	Yes	Intersection-Based	Buffer	Statistical	Opportunistic	None	None
ACAR [50]	Yes	Intersection-Based	Buffer	Hybrid	Opportunistic	None	None
SADV [58]	Yes	Intersection-Based	Buffer	Proactive	Opportunistic	None	Multi-path routing
RBVT-R [52]	No	Reactive-Anchor	Reroute	Reactive	Distance-Vector	Reactive	Reactive Location Finder
RBVT-P [52]	Yes	Anchor-Based	Greedy	Proactive	Dijkstra	None	None
LOUVRE [54]	Yes	Anchor-Based	Reroute/Buffer	Proactive	Dijkstra	None	None
TO-GO [40]	Yes	Intersection-Based	None	None	Opportunistic	None	Two-Hop Neighbours
IGRP [56]	Yes	Anchor-Based	None	Infrastructure	Infrastructure	Infrastructure	None
GeoSvr [45]	Yes	Anchor-Based	None	Statistical	Dijkstra	None	None
BAHG [55]	Yes	Reactive-Anchor	Buffer	Proactive	Dijkstra	Partial	Reactive Location Finder

Table 2.2: Comparison of Simulation Parameters of Reviewed VANET Publications

Protocol	Mobility	Propagation	MAC	App.	Sim.	Compared to
GPSR [19]	RWP	STR [a]	802.11b	CBR	NS-2 [46]	[18]
GSR [36]	Street-RWP [b]	STR-LOS	802.11b	CBR	NS-2	[18] [17]
SAR [38]	Street-RWP	STR	802.11b	CBR	NS-2	[19]
CBF [29]	Highway	Two-Ray	802.11b	CBR	NS-2	[19]
A-STAR [42]	Street-RWP	STR-LOS	802.11b	CBR	NS-2	[19] [36]
STAR [48]	Street-RWP	STR	Unspef.	CBR	NS-2	[19] [38]
GPCR [31]	Street-RWP	STR-LOS	802.11b	CBR	NS-2	[19]
AGF [32]	Realistic [32] [c]	PS [d]	802.11b	CBR	NS-2	[19] [17]
MURU [47]	Street-RWP	Unspef.	802.11b	CBR	NS-2	[19] [17] [18]
GpsrJ+ [39]	Realistic [27]	STR-LOS	802.11b	CBR	Qualnet [1]	[19] [31]
GyTAR [43]	Street-RWP	STR	802.11b	CBR	Qualnet	[36] [60]
CAR [49]	Realistic [32]	PS	802.11b	CBR	NS-2	[19] [32]
GRANT [34]	None	STR-LOS	Unspef.	Unspef.	Unspef.	[19] [e]
VADD [44]	Street-RWP [25]	STR	802.11b	CBR	NS-2	[19] [18]
ACAR [50]	Realistic [27]	STR	Unspef.	CBR	NS-2	[19] [36] [49]
SADV [58]	Street-RWP [25]	STR	Unspef.	Unspef.	Unspef.	[44]
RBVT [52]	Realistic [28]	PS-NLOS [f]	802.11b	CBR	NS-2	[19] [36] [17] [57]
LOUVRE [54]	Realistic+SN [27]	STR-LOS	Unspef.	CBR	Qualnet	[19] [31]
TO-GO [40]	Realistic [27]	PS-LOS [g]	802.11b	CBR	Qualnet	[19] [31] [39]
IGRP [56]	Realistic [28]	S/VTR	Unspef.	Unspef.	Matlab [61]	[19] [31] [57]
GeoSVR [45]	Realistic	STR	802.11b	CBR	NS-2	[19] [17]
BAHG [55]	Realistic [28]	STR-NLOS [h]	802.11b	CBR	NS-2	[31] [43]

[a] Unit-Disc Propagation with Static Transmission Range

[b] Street-RWP: Random Waypoint constrained on Simple Street Topology

[c] Micro-simulation of vehicles with traffic-lights and other considerations

[d] Probabilistic Shadowing Model, also known as the Log-Normal Shadowing Model

[e] Tested separately against greedy and face routing algorithms which are the basis of GPSR

[f] Randomly varying attenuation along NLOS paths

[g] Probabilistic Shadowing model with only LOS paths available

[h] 5dB attenuation along NLOS paths

in VANET routing protocols. GPSR, as noted in the review has influenced almost every other protocol. GSR was the first position based protocol to restrict packet transmission paths to those defined by streets. A-STAR first introduced the idea of using vehicular traffic and selecting paths based on the probability of success. GPCR introduces the idea of coordinator nodes and a street-aware fall-back mechanism. Finally, CAR introduces several concepts used by connectivity-aware protocols, including an AODV-inspired route discovery mechanism and a means of maintain routes.

When considering the testing methodology of the protocols, only 9 are tested in a propagation model that considers any sort of obstacles. Of these 9, only 2, RBVT and BAHG, consider non-line of sight (NLOS) paths. This reveals a significant gap in the literature, especially given the existence of studies such as [9, 10, 62, 63] which experimentally demonstrate that the transmission of signals across non-line of sight paths in cities is quite common. This casts some doubt on the purported performance benefits of the protocols which have only been demonstrated in a simplistic signal propagation environment and raises questions on the impact of the propagation environment on routing performance.

The most commonly used propagation models in this review are discussed in the following chapter along with the CORNER propagation model [5], which uses the formulae from the aforementioned paper to create a deterministic propagation model for urban scenarios. Their impact on routing protocol performance is considered. A set of experiments performed in the real-world to verify and improve the CORNER propagation model are also presented.

Chapter 3

PROPAGATION IN URBAN ENVIRONMENTS

The previous chapter summarised a number of position-based routing protocols in the literature, and identified the critical link between routing protocol performance and the choice of radio propagation model. This chapter examines existing propagation models, specifically focusing on those widely cited in the VANET literature and/or designed for urban environments. An overview of channel fading and its role in propagation modelling is also presented. The CORNER propagation model is studied in detail and validated through a series of physical experiments carried out in two representative urban areas. A number of shortcomings in CORNER are identified, and several small but significant improvements to the model are proposed and implemented. Simulation results are presented and compared with the experimental measurements, and shown to be in good agreement for the scenarios evaluated in the study.

Section 3.1 briefly summarises the propagation models identified in the previous chapter, with a particular emphasis on the models designed for urban environments. Section 3.2 describes a simulation-based study of the impact of the propagation model on standard routing protocol performance metrics such as packet delivery ratios and mean end-to-end latency. Section 3.3 provides a detailed description and analysis of the CORNER propagation model and 3.4 presents the results of a series of experiments performed in the cities of Sydney and Wollongong in New South Wales, Australia to validate the CORNER propagation model using physical channel measurements. Section 3.5 discusses specific improvements made to the CORNER propagation model and Section 3.6 concludes the Chapter.

3.1 A Brief Review of Propagation Models in the Literature

The most accurate means of predicting path-loss in simulations is using a three dimensional *ray-tracing* technique combined with accurate models of the geometry and RF properties of the environment. Numerous studies have established the reliability of ray-tracing in accurately predicting radio frequency (RF) field strength and path loss in a physical environment [64, 65, 66, 67]. Ray-tracing determines the total path loss between two communicating nodes by modelling a continuous radio wave as a discrete number of individual rays and tracing the path of each ray from source to destination in three dimensions. This includes modelling the interaction of individual rays with both obstacles (via specular and diffuse reflection, absorption, refraction, diffraction and scattering) and with each other (via interference between different rays).

The accuracy of the ray-tracer depends both on the number of rays simulated and on the availability of accurate information about the environment. In an urban environment, this information includes characteristics such as the permittivity, permeability, reflectivity, diffractivity and so on of building materials and roads, as well as that of other obstacles such as vehicles. For a given wavelength and resolution of geometry and material properties, the accuracy of the simulation asymptotically increases with an increased number of rays.

While ray-tracing is the most accurate means of simulating RF propagation, it is often computationally infeasible - particularly for large-scale simulations. In 2007. Sridhara et al. estimated that the simulation of a 1 km $\times$ 1 km urban grid will take between 50 and 70 CPU-days depending on the accuracy of the environmental model [65]. It should be noted that this is merely to generate a path-loss matrix for an empty city and does not consider the impact of mobility on propagation. Although such a path-loss matrix is useful (particularly for modelling cellular networks), it is an unreasonable simplification for VANETs due to the well-known significant impact of a fast-moving transmitter and/or receiver

on path-loss [68]. Furthermore, the presence of additional reflectors and diffractors in the form of vehicles limits the value of a pre-computed path-loss matrix for VANET work, even with stationary transmitters and/or receivers. For ray tracing to be useful in these scenarios, a real-time ray-tracing approach would be required, which at present is not computationally feasible and is unlikely to be so for many years. Despite these shortcomings, several ray-tracing implementations are available for use with simulators, including the UDel model which is often used for validating propagation models in the absence of real-life measurement data [69].

Due to the resource-intensive nature of ray-tracing, practical propagation models often make several simplifying assumptions about the nature of the environment. Propagation models typically divide the problem into two distinct components: static *path-loss* and time-varying *fading*. The largest component is that of *path-loss*, which is used to compute the power received at a node, taking into account the total power resulting from the sum of the *static* direct and reflected paths between the transmitter and the receiver. The second component, *fading*, models the time-varying interaction of the signal with its environment as a stochastic process with a set of statistical parameters. Fading accounts for effects such as time-varying interference from signals reflected off and diffracted around/through moving objects; it has a significantly smaller impact on received signal power compared to path-loss.

Figure 3.1 is a conceptual illustration showing the relationship between received power and distance, demonstrating the relationship between path-loss and fading. The solid red line represents the predicted received signal power from a deterministic path-loss model, while the dotted blue line represents the effects of a stochastic fading process applied on top of this path-loss model.

Over half of the protocols surveyed in the literature assume that the path-loss is characterised purely by a static maximum transmission range. Figure 3.2 illustrates this range in a conceptual plot of power versus distance. The transmission

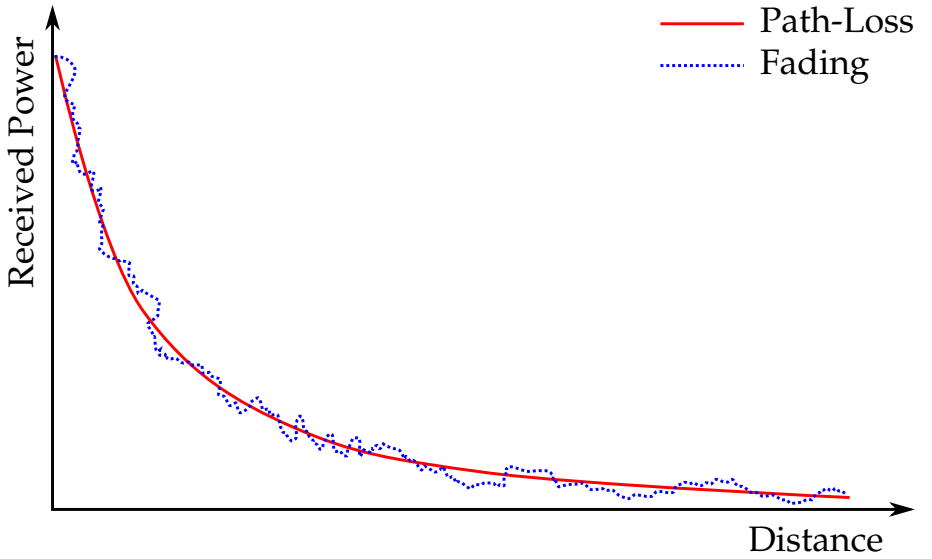


Figure 3.1: Conceptual diagram illustrating the relationship between path-loss and fading

range is usually determined by a simple deterministic free-space path-loss model, which calculates received signal power as inversely proportional to the distance between the transmitter and the receiver (as shown in the graph). The transmitter is assumed to transmit at a fixed power level, and the receiver has a fixed reception threshold below which it will not receive signals. Both the transmission power and the (effective) reception threshold are altered by the gains of the antennas used to transmit and receive signals. Since these path-loss models result in a simple circular transmission range, they are often referred to as 'unit-disc' models.

There also exist more sophisticated models which consider many other factors such as the presence of a reflective ground or an average degree of environmental roughness, terrain and/or the presence of obstructions. The next section looks at a number of common models in more detail.

In all of the models described in this section, the path-loss is the loss experienced by the signal as it travels from the source to the destination through the channel. In order to calculate the power at the receiver, P_r for a path-loss of L_{PL}, the

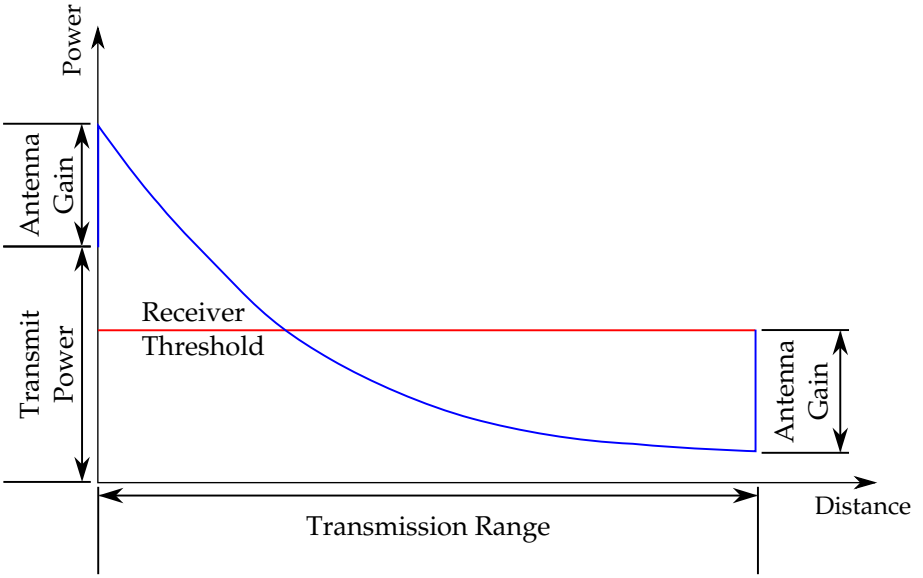


Figure 3.2: Power versus distance, assuming unit-disc propagation model

total gains and losses of the system are considered and represented as follows:

$$P_r = \frac{P_t G_t G_r}{L_{PL} L_{other}} \ , \tag{3.1}$$

where P_t, G_t, G_r and L_{other} are the transmission power, transmitter antenna gain, receiver antenna gain and other system losses respectively.

3.1.1 Free-Space Path-Loss Model

The free-space path-loss (FSPL) model is one of the most commonly used simple propagation models, and is based on the inverse square law [33]. The loss for a distance d from an isotropic radiating element and receiver with no other losses is calculated using

$$L_{fs} = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{3.2}$$

where λ represents the wavelength of the signal. The power delivered by the receiving antenna is then given by the Friis equation, which incorporates transmitter and receiver antenna gains (and potentially other losses L_{other} as well):

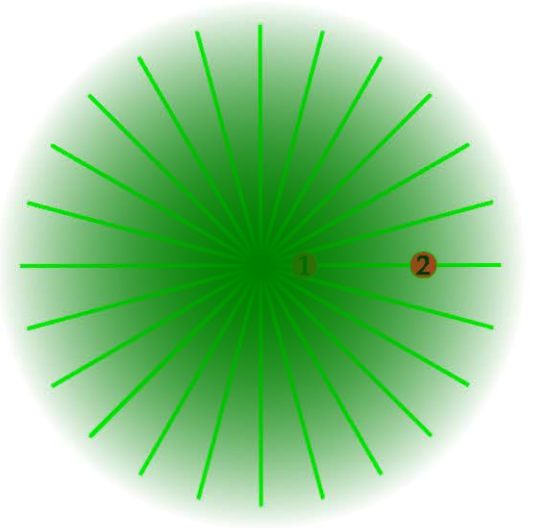


Figure 3.3: Illustration of Free-Space Path-Loss

$$P_r = \left(\frac{P_t G_t G_r}{L_{other}}\right)\left(\frac{\lambda}{4\pi d}\right)^2 \tag{3.3}$$

Figure 3.3 illustrates the free-space path-loss model represented as a discrete number of signals emanating from the source node T. The first receiver, which is closer to T receives a larger number of signals than the second receiver. In reality, signals are not discrete and this is represented by the background, which fades proportionally with the square of the distance and represents the intensity of the signal at various distances from the transmitter.

3.1.2 Two-Ray Ground Reflection Model

Another commonly-used deterministic model is the Two-Ray ground reflection model, which considers the interaction of a ground-reflected signal with a non-reflected direct signal. The received power, at a distance d from the transmitter under the Two-Ray ground reflection model is

$$P_r = \left(\frac{P_t G_t G_r h_t^2 h_r^2}{L_{other} d^4}\right) \tag{3.4}$$

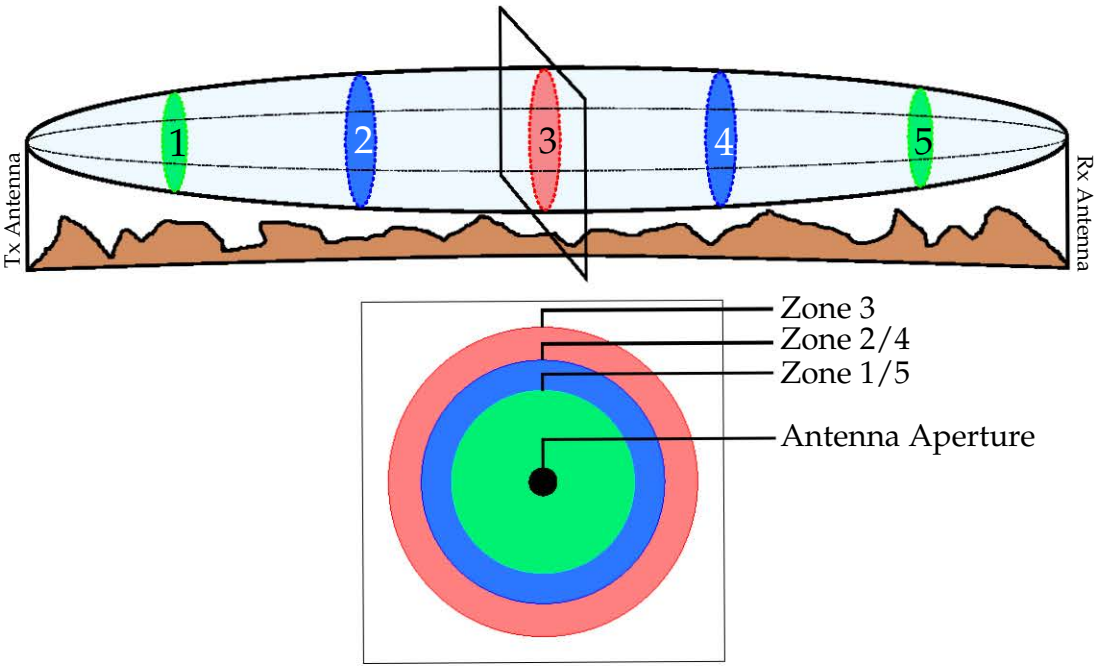


Figure 3.4: Fresnel zones between a transmitter and receiver

where h_t and h_r are the heights of the transmitting and receiving antennas respectively.

It is important to note that the Two-Ray propagation model is prone to significant oscillations until the first Fresnel zone, also known as the critical distance, d_c, which is represented as

$$d_c = \frac{4h_t h_r}{\lambda}. \tag{3.5}$$

Most implementations of the two-ray ground reflection model use predictions from FSPL up to the critical distance and the two-ray model afterwards.

Fresnel zones, as illustrated in Figure 3.4, are a series of concentric regions between a transmitter and a receiver. They occur due to the diffraction of a signal emitted from a circular aperture. Fresnel zones represent successive regions in the ellipse where reflected signals traverse a distance which is $\lambda/2$ greater than d, which is the shortest line-of-sight distance between the transmitter and the re-

ceiver. Signals may reach the receiver through both a direct and reflected paths. Reflected signals cause destructive interference in odd Fresnel zones and constructive interference in even zones. These reflected signals are termed 'diffuse' signals, while the signal travelling along the direct path is termed the 'specular' signal.

3.1.3 The Log-Distance Path-Loss Model

The Two-Ray ground reflection model is often treated as a special case of the Log-Distance path-loss model [33]. The signal power in this model drops logarithmically with an increase in distance. The general expression for received power (in dB) with the the Log-Distance path-loss model is

$$PL_{db} = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right), \tag{3.6}$$

where d_0 is a close-in reference distance determined from measurements close to the transmitter and d is the distance between the transmitter and the receiver. n is the *path-loss exponent* and determines how rapidly the received power drops with distance. For example, The Two-Ray propagation model has a path-loss exponent of 4, leading to the expression:

$$PL_{db} = 40\log(d) - 10\log(Gh_t^2h_r^2), \tag{3.7}$$

where G is a combination of antenna gains and system losses.

The path loss exponent for free space is 2; for a tunnel (or potentially a street between rows of tall buildings) it may be *less* than 2, while for environments such as building interiors which are rich in metallic and high-density structures of varying dimensions, it may be 6 or higher.

3.1.4 Longley-Rice and Edwards-Durkin Models

The Longley-Rice model [70] is also known as the irregular terrain model. Path-loss in the Longley-Rice model is calculated as a function of distance, radio frequency, antenna heights and degree of terrain irregularity. Path-loss is initially predicted by the Two-Ray ground reflection model, after which several factors such as diffraction by the terrain and atmospheric refractivity are taken into account to predict the final received signal power.

Importantly, the Longley-Rice model has been modified to better predict path-loss in urban mobile radio environments with the addition of the 'urban factor' a frequency- and distance-dependent term [71]. This factor was derived from experiments performed by Okumura in 1968 [72].

It is important to note that while the Longley-Rice model provides a median estimate of path-loss in an irregular urban environment, it does not differentiate, between radial and cross streets, nor does it model specific vehicle to vehicle channels. Therefore, it is not particularly useful in studying the behaviour of VANET routing protocols in scenarios that model node-to-node communications across a specific street topology, since it would result in identical distance-dependent path-loss estimates regardless of the underlying urban topology.

The Edwards-Durkin model is similar to the Longley-Rice model in that it accounts for the effects of diffraction across a terrain [73]. Terrain is represented as a 2-dimensional image map. where individual pixels represent the height of the terrain. The Edwards-Durkin model characterises paths as either one of:

- Line-of-sight (LOS) with no obstructions in the first Fresnel-zone;
- LOS with insufficient first Fresnel-zone clearance; or
- Non-LOS (NLOS).

Path-loss in the LOS scenarios corresponds to FSPL. In LOS scenarios with insufficient Fresnel-zone clearance or in NLOS scenarios, both free-space and

two-ray based path-loss estimates are calculated and the smaller estimate is used as a starting point. This is consistent with most implementations of the Two-Ray ground reflection model as detailed in section 3.1.2.

In situations with insufficient Fresnel-zone clearance or NLOS scenarios, the number of diffraction points are calculated and classified according to a pre-defined profile of 1, 2, 3 or 3+ diffraction points. Each profile has an associated set of path-loss calculations based on knife-edge diffraction, and the results are added to the two-ray based path-loss estimates to provide an overall figure for path-loss.

The Edwards-Durkin model has been shown to perform well in irregular terrain [73]. However, it is not intended for use in urban environments with monolithic artificial structures or foliage. Furthermore, neither the Edwards-Durkin nor the Longley-Rice models account for the effects of fading in urban environments that can occur as a result of multi-path interference from reflections off buildings and other sources of interference, instead only providing median situation-agnostic path-loss estimates.

3.1.5 Okumura and Hata Models

The Okumura model is a widely used deterministic path-loss model based on a series of experiments detailed in [72]. The experiments were used to determine median attenuation curves for frequencies in the range of 150 MHz to 1.92 GHz at distances between 1 km and 100 km in urban environments. The Hata [4] model provides an empirical formulation of the median attenuation curves developed by Okumura.

The Hata path-loss formula (in dB) is

$$PL_{db} = 69.55 + 26.16\log(f) - 13.82\log h_t - a(h_r) + (44.9 - 6.55\log(h_t))\log(d), \quad (3.8)$$

where f is the transmission frequency in the range of 150 MHz to 1.5 GHz, h_t

is the transmitter antenna height (between 30 m and 200 m), h_r is the receiver antenna height (between 1 m and 10 m) and d is the distance between the transmitter and receiver in km. $a(h_r)$ is a correction factor for mobile antenna height which varies according to the dimensional scale of the propagation environment. For a small to medium sized city,

$$a(h_r) = (1.1\log f - 0.7)h_r - (1.56\log f - 0.8) \text{ dB} \tag{3.9}$$

Like the Longley-Rice and Edwards-Durkin models, the Okumura and Hata models provide median path-loss estimates and are therefore unsuitable for studying site-specific effects of radio propagation on VANET routing.

Furthermore, due to the distinct transmitter and receiver antenna height ranges assumed in the Okumura-Hata models, they are principally applicable to estimating path-loss in infrastructure-based networks which rely on communication between base stations / access points and mobile nodes; they are therefore of limited use use in node to node communications in VANETs.

3.1.6 The Corner Urban Propagation Model

The CORNER Urban propagation model was proposed by Giordano et al. in [5] and described in greater detail in [11]. It is a light-weight urban propagation model, which aims to accurately reflect the interaction of rays reflected by buildings and diffracted around street corners. It is based on a series of analytical formulae for predicting path-loss in urban environments, previously published in [9]. The model was validated by the authors using packet-delivery-ratio measurements between a stationary and a mobile node pair at different positions around a residential block in Los Angeles.

CORNER is built on three fundamental path-loss scenarios as illustrated in Figure 3.5. They are:

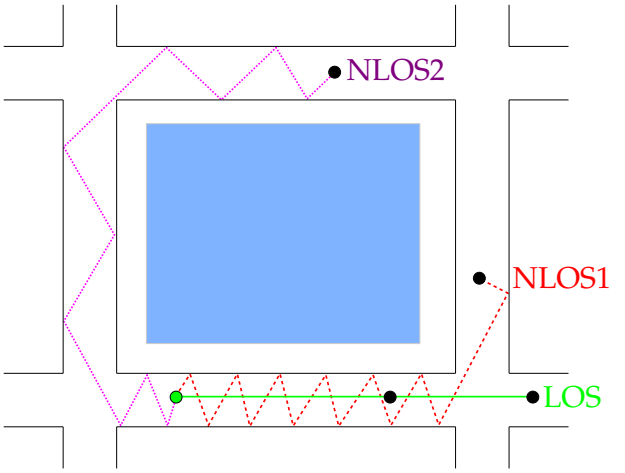


Figure 3.5: Propagation scenarios in CORNER

- Line of Sight (LOS);
- Non-Line of Sight around one corner (NLOS1); and
- Non-Line of Sight around two corners (NLOS2).

The positions of the transmitter and receiver on specific streets are determined with the aid of reverse geo-coding, which is the process of mapping a coordinate onto the nearest street. Once the positions and street-locations of both nodes are determined, the channel is classified into one of the aforementioned path-loss scenarios. The free-space path-loss model is applied in the LOS case, whereas NLOS scenarios are modelled as a sum of reflected and diffracted signals. Specifically, NLOS1 scenarios are modelled as the sum of a reflected path and a diffracted path around a corner and NLOS2 scenarios are modelled as a sum of:

- A reflected path around both corners;
- A reflected path around a corner followed by a diffracted path around the second corner;
- A diffracted path around a corner followed by a reflected path around the second corner; and

- A diffracted path around both corners.

While the CORNER propagation model showed promising results in a real-world experimental study, there were a number of shortcomings in the experimental methodology. The authors performed an experiment where MAC-layer packets were exchanged between a sender and a receiver around a residential block. The resulting packet delivery ratios were then compared to an identical simulation performed using the CORNER model, and the results were found to show a strong correlation. The use of a MAC-layer metric means that the experimental measurements are susceptible to interference from wireless access points and other mobile nodes transmitting at the same frequency. It is therefore desirable to repeat the experiments using physical layer metrics.

3.1.7 Fading In Urban Environments

The previous sections presented an overview of significant path-loss models in the literature, specifically those often used for the simulation of VANET routing protocols or designed for urban environments. *Fading* is the complex and time-varying interaction of signals due to the effects of multi-path interference and diffraction. While fading is negligible in relatively flat environments due to the lack of obstacles to reflect or diffract waves, it has been shown to be significant in urban environments [33, 74, 75].

The Log-Distance path-loss model, described in section 3.1.2 is often written with an additional fading component in the form of a Gaussian distributed random variable, X_g, with zero mean. This modifies the expression for received power to the following equation:

$$PL_{db} = PL(d_0) + 10n\log\left(\frac{d}{d_0}\right) + X_g \tag{3.10}$$

The combined model is also known as the 'Probabilistic shadowing' model, which

is used by CAR, RBVT and TO-GO, which were reviewed in the previous chapter.

Fading is an often ignored in VANET simulations, however, it can significantly affect the performance of higher layer protocols due to the degree of unpredictability it introduces in the assumption of a fixed transmission range. In particular, this can severely impact TCP performance, which attempts to statistically estimate the mean and variance of packet round trip time, which can be greatly affected by sudden changes in source-destination routes and MAC layer retransmissions.

3.2 Impact of the Propagation Model on Routing

In order to demonstrate the impact of the choice of propagation model on the performance of routing protocols, this section quantitatively evaluates the performance of the well-known VANET routing protocol GPSR with four propagation models. Simulations conducted with the Qualnet network simulator, and performance is quantified using the mean packet delivery ratio (PDR). The simulation is performed on a 1.5 km^2 map based on a region in Manhattan. Node mobility patterns were generated using the SUMO vehicular traffic simulator, using random waypoints and with movement constrained to valid vehicular paths through the city. SUMO also models the interaction of vehicles with each other and with traffic lights and junctions. 64-bit CBR packets are exchanged between 20% of randomly selected nodes for a period of 10 seconds and these nodes are varied continuously over the course of each simulation run. 20 runs per propagation model were used, with each run using a different random vehicular traffic trace and a different set of CBR traffic flows. The same set of 20 random traces and CBR traffic flows are used for all propagation models. The results are presented with error bars representing a 95% confidence interval. The tested propagation models include:

1. A Unit-Disc Model (Free-Space with no fading);

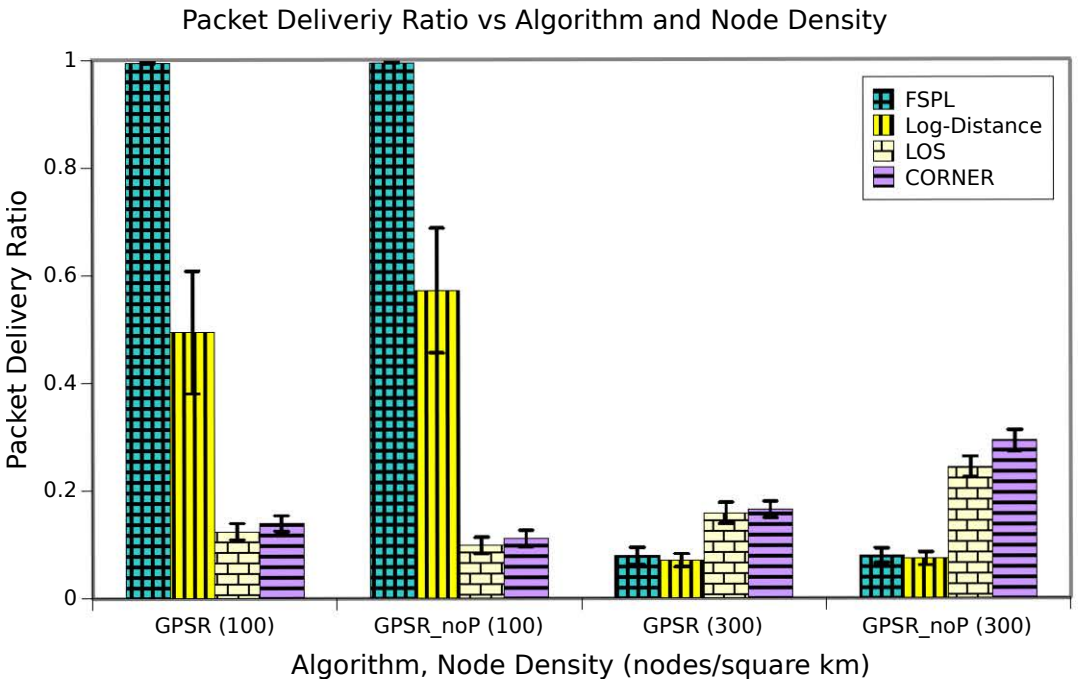


Figure 3.6: GPSR mean packet delivery ratio evaluated under four different propagation models and node density. Confidence intervals are 95%.

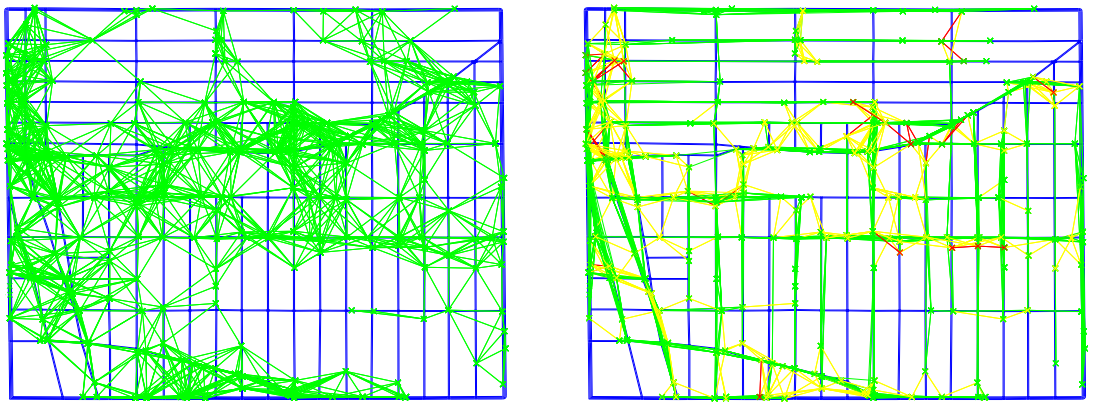
2. The Probabilistic Shadowing Model (Two-Ray with a Gaussian-distributed fading component);

3. A Unit-Disc model restricted to LOS paths; and

4. The CORNER Propagation Model.

The first three models were chosen due to their popularity in the review presented in the previous chapter. A unit disc model was cited by 9 of the 22 reviewed protocols, the probabilistic shadowing model by 4 and the LOS model by 7. The CORNER propagation model is also tested due to its claimed accuracy in modelling an urban environment.

The graph in Figure 3.6 compares the performance of GPSR in terms of mean PDR, both with and without its perimeter mode fall-back algorithm, under different propagation models. The simulations run without GPSR's fall-back mode



(a) Two-Ray (b) CORNER

Figure 3.7: Inter-node connectivity on the Manhattan map

are labelled 'GPSR_noP' As illustrated in the graph, simplistic propagation models show extremely optimistic performance at low node densities, due to the high degree of connectivity assumed by ignoring physical obstructions. Conversely, they show extremely pessimistic performance at higher node densities due to the excessive degree of node visibility resulting in a very large number of packet collisions. This is evident from the connectivity map shown in Figure 3.7, which compares the connectivity at one particular instant in time between the simulation under the two-ray model and under CORNER.

Significantly, by looking at the results from the simplistic propagation models, one could erroneously conclude that GPSR's fall-back mechanism has minimal impact on performance. However, adopting a model which is more appropriate for urban environments, such as the LOS or CORNER models, leads to a different conclusion. Perimeter mode is seen to improve the performance in low node density environments but significantly degrade performance in higher density environments. Furthermore, the performance difference between low and high node density environments is significantly smaller compared to that observed when a simplistic path-loss model is used. This clearly demonstrates the importance of choosing a more realistic propagation model when evaluating or designing

VANET routing protocols.

3.3 A Detailed Examination of the CORNER Urban Propagation Model

CORNER works by determining the losses between two nodes due to the propagation of electromagnetic waves according to one of the three scenarios previously described (LOS, NLOS1 and NLOS2). In order to determine the correct scenario, the position of the nodes on a city map must first be established. In CORNER, a city map is represented as a series of road-segments. The position of nodes on these segments is determined using reverse geocoding, where the actual coordinates of the node are mapped onto the nearest road-segment. This method is used in most GPS receivers to correct for minor errors in positioning.

Each road-segment is assigned a width, RW which is represented as

$$RW = (NoL \times LW) + 10 \tag{3.11}$$

where NoL is the number of lanes on the road and LW is the width of each lane. The constant 10 is added to account for the presence of a pavement or sidewalk (assumed to be $5m$ on each side). Any area of the map not populated by road-segments is assumed to be an obstruction (for example, a building). These parameters can and should be adjusted for different sites as appropriate.

For the sake of simplicity, the term describing the frequency-dependence of received power, which is used in several of the path-loss formulae, is denoted L_2:

$$L_2 = \left(\frac{c}{4\pi f}\right)^2 = \left(\frac{\lambda}{4\pi}\right)^2, \tag{3.12}$$

where $\lambda = \frac{c}{f}$ is the wavelength of the signal of frequency f.

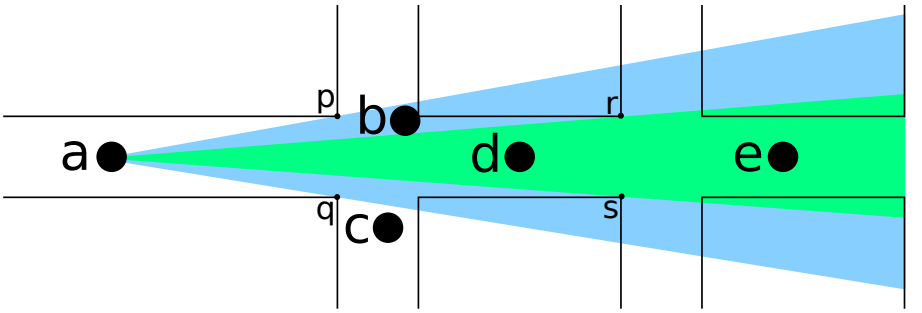


Figure 3.8: Angular View of a Node in CORNER

3.3.1 Path-loss in Line of Sight

It is assumed that in line-of sight scenarios, path-loss follows the well established free-space path-loss model. This assumption is made because the signal should degrade in accordance with the inverse square law in the absence of any obstacles. The path-loss at a distance d from the transmitter is therefore represented as:

$$PL_{LOS} = \frac{L_2}{d^2} \tag{3.13}$$

where d is the distance between the transmitter and the receiver.

Nodes are judged as being in line of sight if they are on the same road segment. In order to determine if two vehicles on different road segments are in line of sight, the 'angular view' of a node is considered. The angular view is defined as the 'portion of a plane that the vehicle can see from the opening afforded by a cross-road' [11]. Figure 3.8 illustrates this concept. The portion of the visible plane from the first intersection is defined by the blue area and the portion visible from the second intersection is defined by the green area. It should be noted that the following method is only valid for nodes spaced within two intersections of each other; however, no theoretical or empirical basis is offered for this heuristic in the original paper.

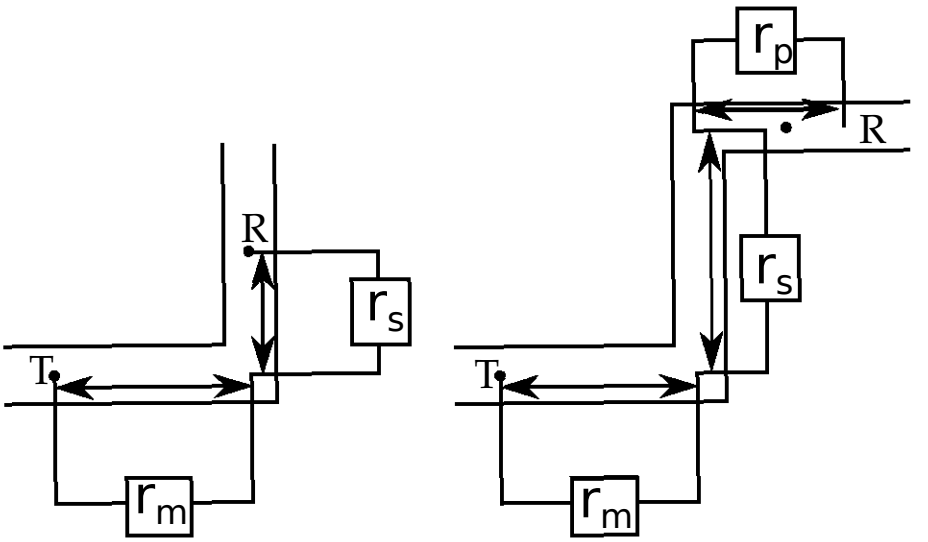


Figure 3.9: Variables Used in Path-loss Around Corners

3.3.2 Definitions for Path-loss around corners

Figure 3.9 illustrates the variables used in the NLOS1 and NLOS2 scenarios. In NLOS1 scenarios, nodes are separated by a main and side street of distances r_m and r_s as illustrated in the figure. The width of these streets are denoted W_m and W_s respectively.

In NLOS2 scenarios, nodes are separated by a main, side and parallel streets, of distances r_m, r_s and r_p as illustrated in the figure. The width of these streets are represented as W_m, W_s and W_p respectively. All the path-loss formulae in this chapter are as originally presented in [9] and are simplified by the substitution of equation 3.12.

3.3.3 Path-loss around one corner

In scenarios where nodes communicate across one corner, the total path-loss is defined by the sum of the path-loss due to reflections off buildings (PL_R) and diffraction around the corner (PL_D). A signal is considered to undergo a minimum number of reflections, based on the length and width of the streets

through which it travels. This number, N_{min} (defined in equation (5) of [9]) is calculated as

$$N_{min} = \left\lfloor 2\sqrt{\frac{r_m r_s}{W_m W_s}} \right\rfloor . \tag{3.14}$$

Each reflection is also assumed to undergo a constant loss. The portion of the signal that is reflected is represented as R_0. R_0 varies by environment, and should ideally be determined empirically for different urban environments.

The total path-loss in NLOS1 scenarios is described in equations (6) and (12) of [9] as

$$PL_{NLOS1} = PL_D + PL_R, \tag{3.15}$$

where

$$PL_R = 10\log\left(\frac{\lambda}{4\pi(r_m + r_s)}\right)^2 + 10\log(R_0)^{2N_{min}} \text{ dB}$$

and

$$PL_D = 10\log\left(\frac{\lambda}{4\pi r_{\underset{s}{m}}}\right)^2 + 10log\left(\frac{\lambda r_{\underset{m}{s}}}{4r_{\underset{m}{s}}^2}\right) \text{dB},\ r_{\underset{s}{m}} < r_{\underset{m}{s}}.$$

Expressed in absolute terms and substituting in 3.12,

$$PL_R = \frac{L_2 R_0^{2N_{min}}}{(r_m + r_s)^2}, \tag{3.16}$$

and

$$PL_D = \begin{cases} \frac{L_2\lambda}{4r_s^2 r_m} & \text{if } r_m < r_s \\ \frac{L_2\lambda}{4r_m^2 r_s} & \text{otherwise} \end{cases}. \tag{3.17}$$

3.3.4 Path-loss around two Corners

Around two corners, the path-loss is significantly more complex. The path-loss is described as the sum of reflected (two corners), diffracted (two corners), reflected (1^{st} corner)-diffracted (2^{nd} corners) and diffracted (1^{st} corner)-reflected (2^{nd} corner) components.

The minimum number of reflections is also significantly more complex due to the presence of an extra street and is defined in equation (15) of [9] as:

$$N_{min} = \left\lfloor \frac{r_m}{W_m}\sqrt{\frac{r_s W_m W_p}{W_s(r_m W_p + r_p W_m)}} + \frac{r_s}{W_s}\sqrt{\frac{W_s(r_m W_p + r_p W_m)}{r_s W_m W_p}} + \frac{r_p}{W_p}\sqrt{\frac{r_s W_m W_p}{W_s(r_m W_p + r_p W_m)}} \right\rfloor \tag{3.18}$$

To improve computational efficiency, the calculation of N_{min} can be simplified by substituting the following variable:

$$t = \sqrt{\frac{r_s W_m W_p}{W_s(r_m W_p + r_p W_m)}}, \tag{3.19}$$

resulting in the simpler expression

$$N_{min} = \left\lfloor \frac{r_m t}{W_m} + \frac{r_s}{W_s t} + \frac{r_p t}{W_p} \right\rfloor \tag{3.20}$$

The various path-losses for the NLOS2 scenario from (4.8) are defined in equations (16), (18), (20), (21) and (22) of [9]. The path-loss across two corners is defined as

$$PL_{NLOS2} = PL_{RR} + PL_{DD} + PL_{RD} + PL_{DR}, \tag{3.21}$$

where, in absolute terms,

$$PL_{RR} = \frac{L_2 R_0^{2N_{min}}}{(r_s + r_m + r_p)^2}, \tag{3.22}$$

$$PL_{DD} = \begin{cases} \frac{L_2 \lambda^2}{16 r_m r_s r_p^2} & \text{if } r_m < r_s \\ \frac{L_2 \lambda^2}{16 r_m^2 r_s r_p} & \text{otherwise} \end{cases}, \tag{3.23}$$

$$PL_{RD} = \begin{cases} \frac{L_2 R_0^{2N_{min}} \lambda r_s}{4(r_s + r_m)^2 r_p^2} & \text{if } r_s < r_p \\ \frac{L_2 R_0^{2N_{min}} \lambda}{4(r_s + r_m)^2 r_p} & \text{otherwise} \end{cases}, \tag{3.24}$$

and

$$PL_{DR} = \begin{cases} \frac{R_0^{2\left\lfloor \frac{r_p r_s}{W_p W_s} \right\rfloor} L_2 \lambda}{4 r_m (r_{s+p})^2} & \text{if } r_m < (r_s + r_p) \\ \frac{R_0^{2\left\lfloor \frac{r_p r_s}{W_p W_s} \right\rfloor} L_2 \lambda}{4 r_{s+p} r_m^2} & \text{otherwise} \end{cases} \tag{3.25}$$

3.4 Experimental Validation of CORNER

The authors of the original CORNER model validated their model with experiments conducted around a city block in Los Angeles. Two vehicles were used, one carrying a transmitter and one carrying a receiver. 64-byte packets were sent from transmitter to receiver at a rate of 10 packets per second. The connectivity between each transmitter-receiver location pair was defined by the packet delivery ratio, i.e., the ratio of the number of received packets to the number of transmitted packets. The authors found that their experimental results closely matched simulation results, in particular correctly modelling the 'void' created by presence of buildings.

The experiments did not explicitly attempt to validate the assumption that path-loss in LOS scenarios follows the free-space model. This is especially critical, since the two-ray ground reflection path-loss model is generally preferred to the free-space model when modelling signal propagation in terrestrial environments in the

absence of buildings, and there does not seem to be a generally accepted value for the path loss exponent in urban vehicle-to-vehicle scenarios. Additionally, the MAC-layer packet-delivery ratio may not be, in isolation, the best metric to evaluate the accuracy of a simulation.

In order to evaluate the choice of path-loss model for the LOS scenario and generally perform an independent validation of the CORNER channel model, a set of vehicle-to-vehicle channel characterisation and connectivity estimation experiments were performed at a number of different urban sites. In addition, the implicit assumptions employed in the the NLOS1 and NLOS2 cases would be tested using physical layer measurements in addition to the MAC-layer PDR used by the original CORNER paper.

3.4.1 Experimental Apparatus and Methodology

The apparatus used for the experiment comprised:

- 2 PC Engines ALIX 3D3 Systems running Voyage Linux 0.8;
- 2 Ubiquiti SR-2 400 mW 802.11b/g cards using the ath5k drivers;
- 1 Rohde and Schwarz FSH-8 spectrum analyser; and
- 3 generic 5 dBi gain omnidirectional antennas.

All experiments described in this thesis were performed using identical hardware. The experimental configuration used for the experiments can be seen in Figure 3.10.

Software was developed for both the transmitter and receiver using the Linux raw-sockets API. All experiments were performed on Channel 1, which operates at a frequency of 2.412 Ghz. This was observed to be the most radio-quiet channel during pre-experimental surveys of the experiment sites. Once a test is started, the transmitter continuously broadcasts a series of test sequences; each consisting of 200 packets of 250 bytes each. Each packet in a test has a

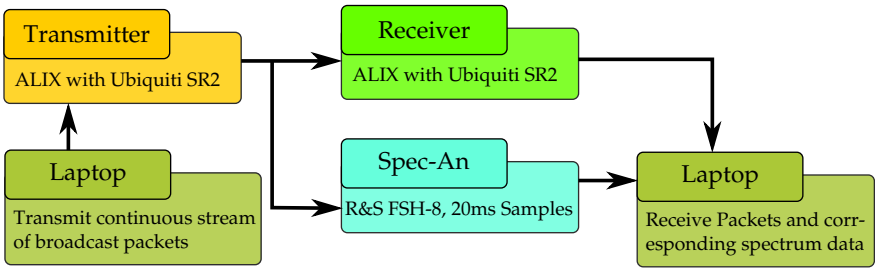


Figure 3.10: Configuration of experimental apparatus

unique sequence number and checksum, and each test has a unique test number. The receiver listens continuously, and upon receiving a packet, computes the checksum to verify the packet's origin. The receiver also notes the number of packets received as part of each test to calculate the packet delivery ratio (PDR).

The receiver also continuously saves snapshots of the spectrum, with a corresponding time-stamp, from the spectrum analyser. To ensure that interference from other sources is minimised, only snapshots of the spectrum corresponding to a received packet (determined using the time-stamp) are used in the analysis. This ensures that a transient peak in the spectrum from nearby access points and residential Wi-Fi devices is not accidentally registered.

Characterising System Gains and Losses

The total gains/losses associated with the system are not precisely known due to the variable characteristics of antennas and wireless cards at different frequencies. Once an appropriate frequency was chosen for the experiments, it was necessary to characterise the total gains and losses associated with the system. In order to do this, an experiment was carried out in an open-field to determine a value for

$$\tau = \frac{P_t G_t G_r}{L}, \tag{3.26}$$

where P_t is the transmit power,G_t and G_r are the transmitter and receiver an-

Figure 3.11: Open-Field Experiment Site

tenna gains respectively and L is the sum of system losses.

A series of measurements were performed in a large open field at the University of Wollongong. The transmitter was kept stationary and the receiver was incrementally moved in discrete intervals along the field. This is illustrated in Figure 3.11. The results were fitted against the two-ray ground reflection model. The antennas were vertically polarised and at distances of 1.06 m and 0.94 m off the ground respectively, giving a free-space to two-ray cross-over distance of 100.65 m [33]. The antennas used for these experiments were roughly 0.17 m in length. The free-space path and two-ray path-loss formulae are only valid in the far-field of the antenna. The distance separating the near-field from the far-field is the Fraunhofer distance, d_f. Since $d_f >> D$, where D is the largest physical dimension of the antenna [33], the Fraunhofer distance for this experiment is 1.7 m. The first sample point for the field experiment is thus 2 m.

Figure 3.12 illustrates the results from the open-field experiment (with error bars

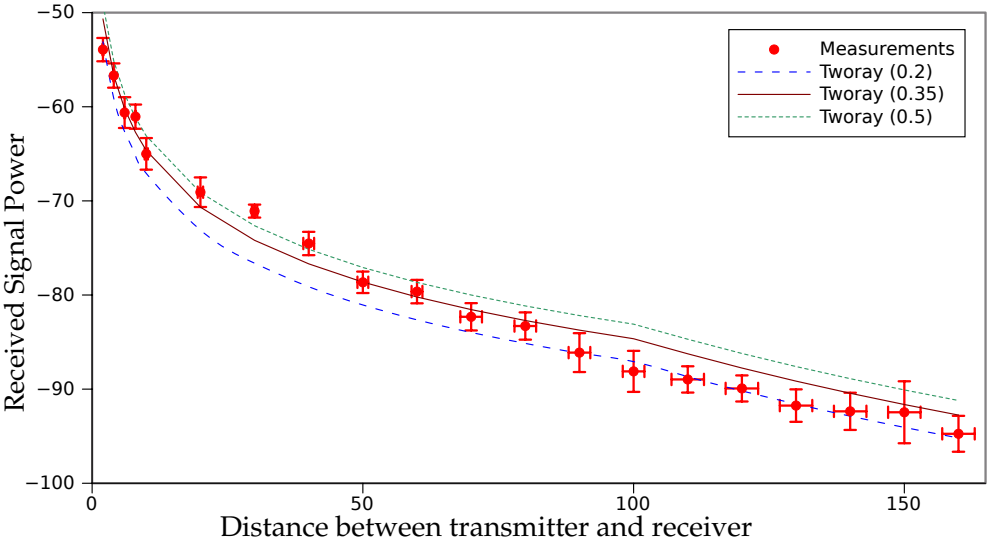


Figure 3.12: Path-loss measurements for the open-field experiment

showing 95% confidence intervals), as well as the analytical curves for different values of τ. From these measurements, the value of τ for this experimental set up is estimated to be 0.35 ± 0.15. It is interesting to note that beyond 80 m, τ is almost consistently 0.20. It is likely that weak reflections from the foliage or nearby structures may be providing constructive interference at shorter distances, artificially increasing the value of τ in this particular situation.

Experimental Path-loss Measurements and Analysis

Three sites were chosen in New South Wales (NSW), Australia to obtain these measurements. The sites are:

- Miranda, NSW: A suburb of Sydney characteristic of an urban environment to test the NLOS1/2 scenarios;
- Montague Street, Wollongong, NSW: A sparsely populated, less dense environment to test the NLOS1/2 scenarios; and
- Burelli Street, Wollongong, NSW: A 800 m long stretch of road to test the LOS scenario.

Figure 3.13: Receiver position and transmission path for the straight-line experiment

3.4.2 Line-of-Sight Experiment

The line-of-sight experiment was performed on Burelli Street in Wollongong. The receiver was kept stationary and the transmitter was moved down the length of the street, a distance of approximately 800 m, broadcasting an average of 3 tests per second. Each test comprised of 200 packets of 250 bytes each, for a throughput of roughly 1.2 Mbps. Beyond approximately 800 m, the transmitter moves out of direct line of sight of the receiver as demonstrated by the red dashed line in Figure 3.13. The transmitting vehicle drove down the road and back a total of ten times, resulting in several discrete measurements for each point on the road. The experiment was performed both during the day and at night to determine the impact of traffic. Burelli Street carried moderate to heavy vehicular traffic during the daytime experiments, including several buses along the route, and almost no traffic at night.

The average received power is plotted against distance in Figure 3.14. Distances are quantised to 5 metre intervals. The error due to quantisation is represented in the horizontal error bars and is negligible. The corresponding power measure-

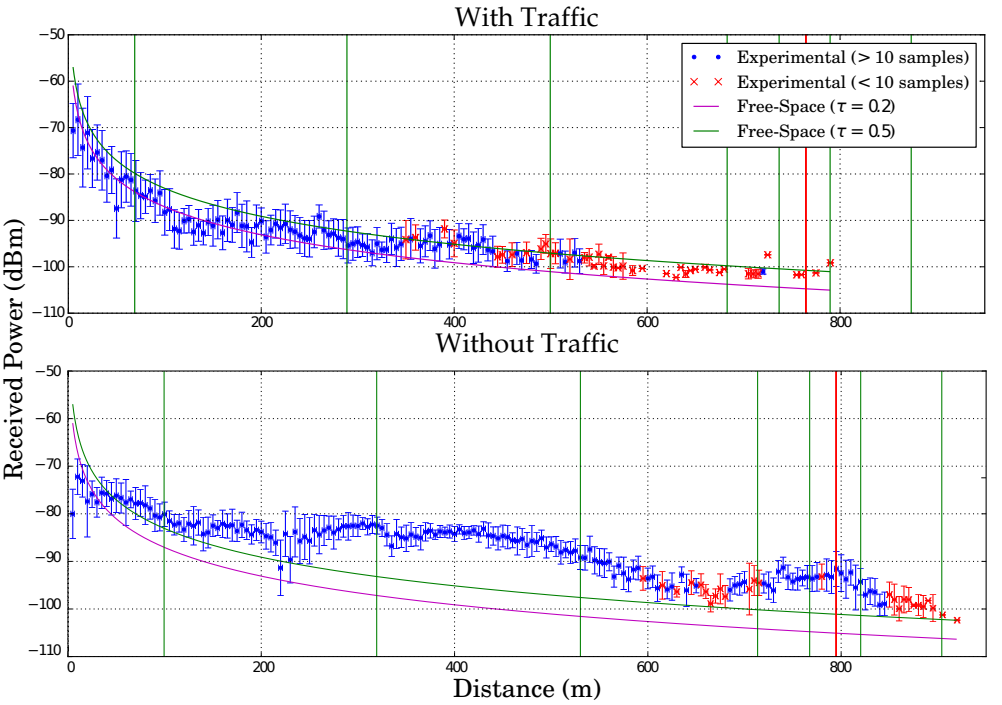


Figure 3.14: Distance vs received power for the LOS experiments

ments obtained from the spectrum analyser are plotted on the graph. The error bars represent 95% confidence intervals.

It is interesting to note that in the absence of vehicles, the observations demonstrate that the received signal strength is consistently higher than that predicted by the free-space path-loss model. This is due to the presence of buildings along the length of the road, which focus the signal down the road, similar to the inside of a waveguide [76]. A similar result was observed by Abbas et al. [77], who observed that an obstructed line-of-sight path, i.e., where one or more vehicles obstruct the direct communication path between two vehicles, results in up to a 10dB loss in signal. The same authors also observe that the waveguiding effect of the buildings in an 'urban canyon' plays a significant role in boosting the received signal strength.

The presence of vehicles significantly degrades the channel, with measured received signal strength decreasing to very closely follow the free-space model. Con-

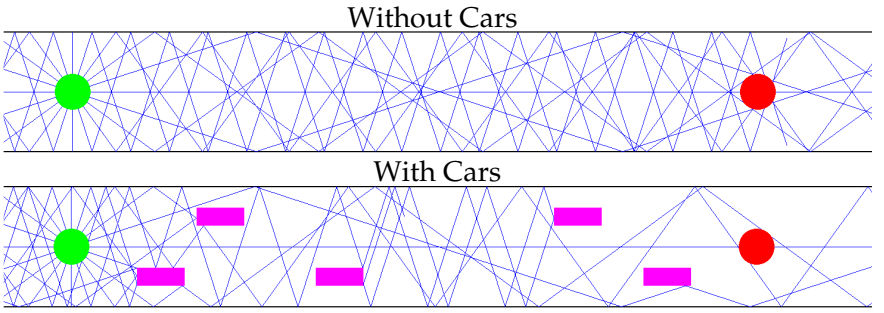


Figure 3.15: Ray-launcher analysis of LOS situations with vehicles

sequently, the maximum transmission range is also considerably reduced, and is closer to 600 m, although packet reception starts to become erratic shortly after 500 m. The free-space like behaviour is an interesting result, but one which can be explained by modelling the cars as additional scatterers. The scenario was recreated in a two-dimensional software ray-launcher environment, in which a continuous signal is modelled as a series of discrete rays. For example, a 400 mW signal, modelled as 8 discrete rays, would result in each ray carrying 50 mW of power. Every time a ray undergoes reflection, it is attenuated due to the effects of scattering. The results, seen in Figure 3.15, illustrate the effect of vehicles as additional scatterers. Dominant line of sight rays remain unaffected (leading to free-space path-loss), while the reflected rays are further scattered by the vehicles in the scenario.

The line-of-sight observations in the presence of traffic are the most significant in this experiment, since they validate the choice of the free-space path-loss model for these scenarios as proposed in the original CORNER paper.

3.4.3 Non-Line of Sight Experiments

The Miranda site is densely packed with buildings and is both substantially planar and regular. The experiments at Miranda were carried out at night, when a minimal number of moving vehicles were present. However, there were

(a) Miranda (b) Montague Street

Figure 3.16: Experimental sites chosen for the non-line-of-sight experiments

a significant number of parked cars along the sides of the roads, due to the residential nature of the site, as well as significant amounts of cultivated foliage along the roads.

The site on Montague Street is less dense and less regular that the one at Miranda. As with the Miranda site, the experiments were carried out at night. There were several large trucks with steel containers parked along each street.

Figure 3.16a and 3.16b show the two sites where the experiments were carried out. The transmitting vehicle paths are highlighted in red.

The NLOS experiments were performed with a stationary receiver which is placed at several sample points and a mobile transmitter which circumnavigates the block ten times for each receiver point. The main aim of these experiments was to illustrate connectivity between various points around buildings. The experiment is similar to one previously described by Eugenio et al., except with packet delivery ratio (PDR) measurements as well as power spectral measurements [5].

Figure 3.17 shows the connectivity map for the Miranda, showing both the a simulated connectivity map generated by the CORNER model and experimental

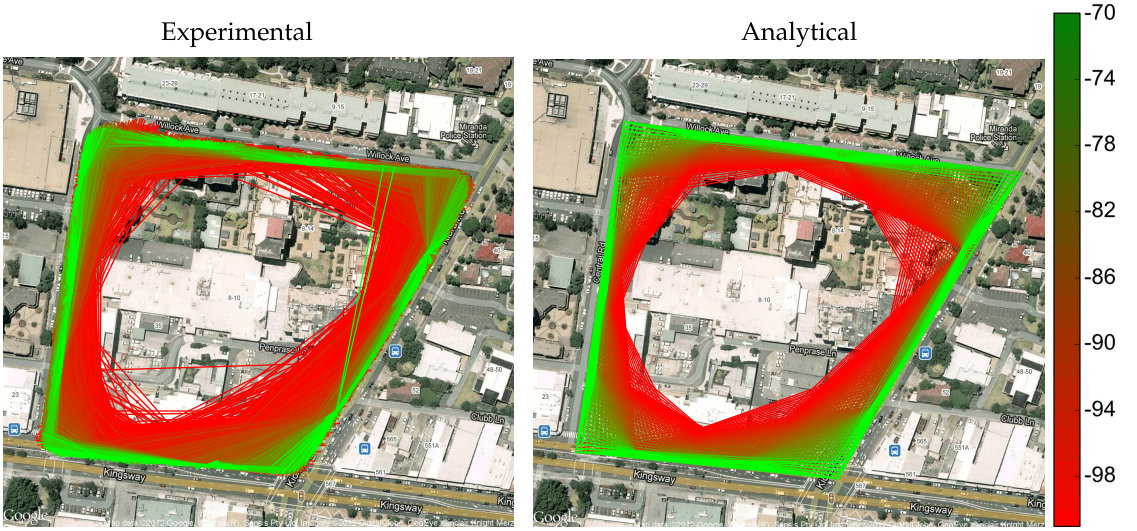


Figure 3.17: Connectivity map of the Miranda experiment site

connectivity measurements. The colours represent the average signal strength of the received signal at that point in relation to the receiver position; the minimum signal strength for which packets were still received by the wireless card was -104 dBm and the measured noise floor was -110 dBm (both as measured by the spectrum analyser). Connectivity is considered to exist in the experimental model if any packets at all are recorded for a given position; for CORNER, it is assumed that if the predicted signal strength in a given location exceeds the receiver threshold for the duration of a packet transmission (-104 dBm, i.e. the same as for the experimental results).

As seen in the figure, the simulated connectivity map produced by CORNER closely resembles the experimental-determined connectivity map. However, there are several minor differences. These include the strength of the signal in LOS conditions, with the simulation results indicating a much higher signal strength at close distances. For NLOS conditions, the CORNER model assumes that rays are both reflected and diffracted around corners. The reflections are modelled with a constant loss per reflection; for this simulation, a loss of 10% per reflection was used.

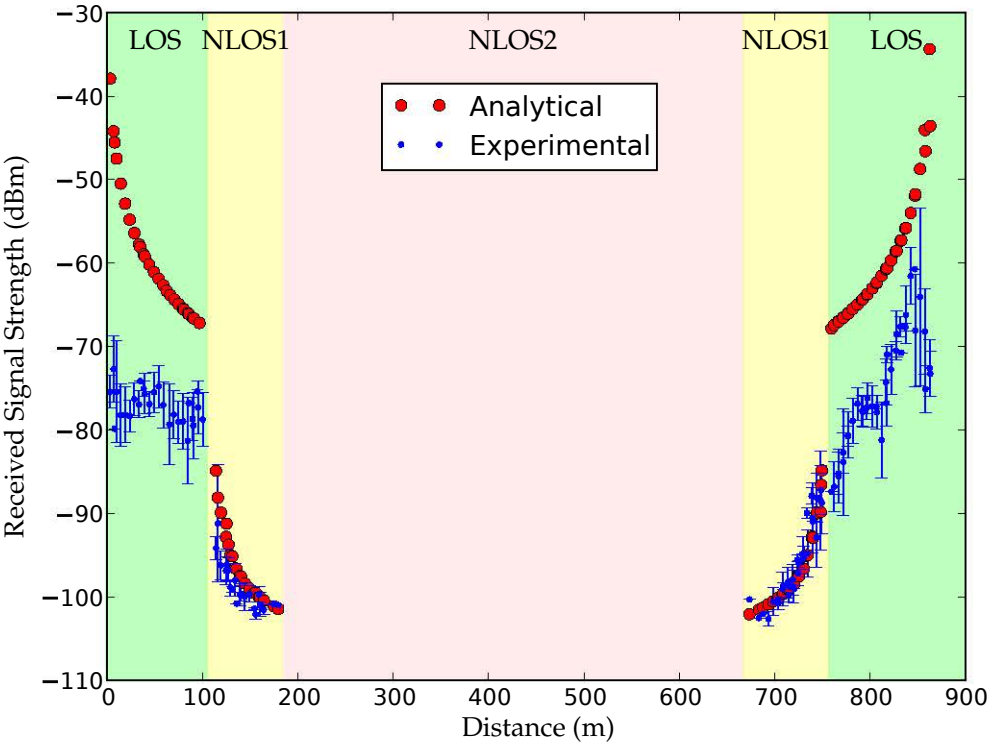


Figure 3.18: Received signal power as a function of distance for Miranda experiment site

Figure 3.18 shows the simulated signal power values predicted by the CORNER model and the experimental measurements plotted as a function of the distance between the transmitter and receiver (measured along the road as the transmitter moves away in LOS, turns a corner (NLOS1), a second corner (NLOS2), a third corner (NLOS1) and back onto LOS till it reaches the receiver). No experimental NLOS2 values were recorded and analytical NLOS2 values are absent since they are below the noise floor of -110 dBm. This figure also shows a significant difference between the analytical and measured values in LOS conditions. While this may be due to the foliage on the streets, it is likely that it is an error in the model that warrants further experimental study in different LOS scenarios. However, the predicted values closely match the measured values in NLOS1 conditions.

Figure 3.19 shows the connectivity map of the Montague Street site. The site is more sparsely built than the Miranda site. There were several large trucks parked

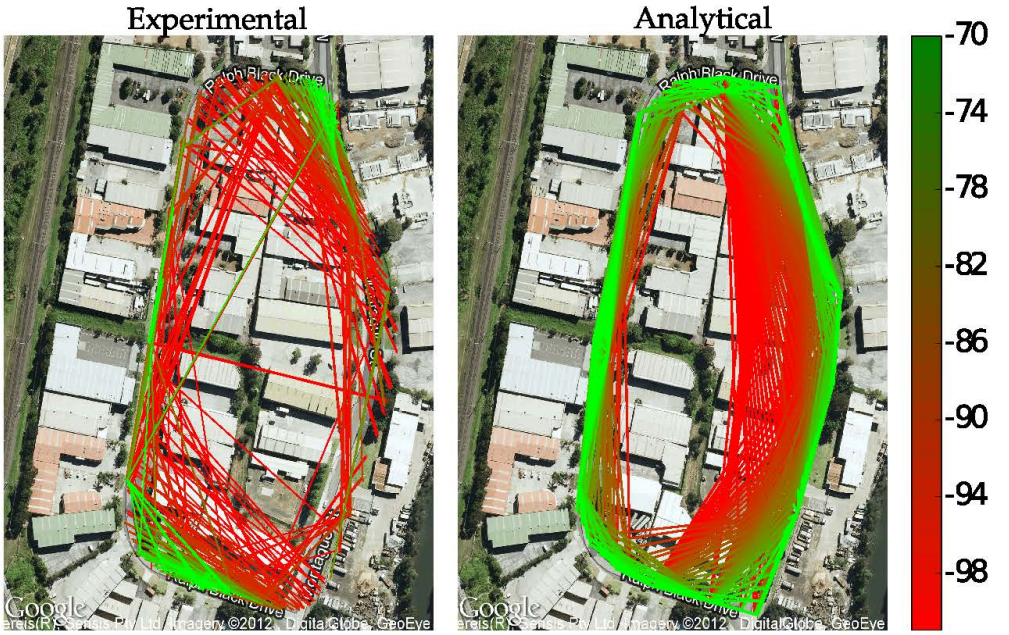


Figure 3.19: Connectivity map of Montague Street

along the road the night of the test, and this resulted in a number of 'holes' in the network. However, it is still possible to draw conclusions from this experiment. For this area, experimentally-measured connectivity is quite different to that predicted by CORNER. Due to the large spaces between buildings, there is a significant amount of connectivity across the road circuit. Furthermore, due to the curved shape of the road, the classifier finds it difficult to accurately estimate the appropriate scenarios across the bottom left corner of the map. This highlights a major shortcoming of CORNER, showing that it is only suited to grid like networks. Curved roads are modelled as several discrete links, which can lead to poor classification around a curved corner or an irregular road. The relatively high number of transmissions which penetrate buildings is due to the fact that they are relatively small and well spaced out.

3.4.4 Analysis of Fading from Experimental Measurements

It has been well established that fading occurs in urban environments due to the presence of buildings and foliage as well as vehicles and a plurality of other obstacles. Many experiments have previously been performed to establish the presence of such fading effects between a large antenna (such as a cellular tower) and nodes at ground level [33, 78]. However, there is very little literature describing fading in direct transmissions between multiple stationary or mobile nodes at ground level - which is the usual case for vehicle-to-vehicle (V2V) communication.

A significant number of samples were collected in the experiments performed in the previous section, which can be used to provide a statistical analysis of the channel to see the distribution of individual samples. Analysis of this statistical data found a heavy correlation with a Rician distribution. The Rician distribution requires a parameter, known as the K-factor, which is defined as the ratio of the total power received from specular rays to the total power received from the reflected rays. This is represented mathematically as

$$K = \frac{A^2}{2\sigma^2} \tag{3.27}$$

where $A^2/2$ is the total specular power received and σ^2 is the total reflected power received [33]. The Rician K-factor for the experimental measurements in this section is determined using the moment-based estimator described in [79].

While there several other K-factor estimators are described in the literature [80, 81, 82, 83, 84, 85], the Greenstein method was chosen since it was well cited and straightforward to implement. The Greenstein estimator was also tested against known data sets to determine its accuracy. In order to determine the number of samples required to obtain adequate statistical certainty, a series of samples were generated from a Gaussian distribution using Clarke's model, and the Greenstein estimator was used to estimate the K-factor of these samples. The results are illustrated in Figure 3.20 and show that the error in the value of K drops to ± 1.28

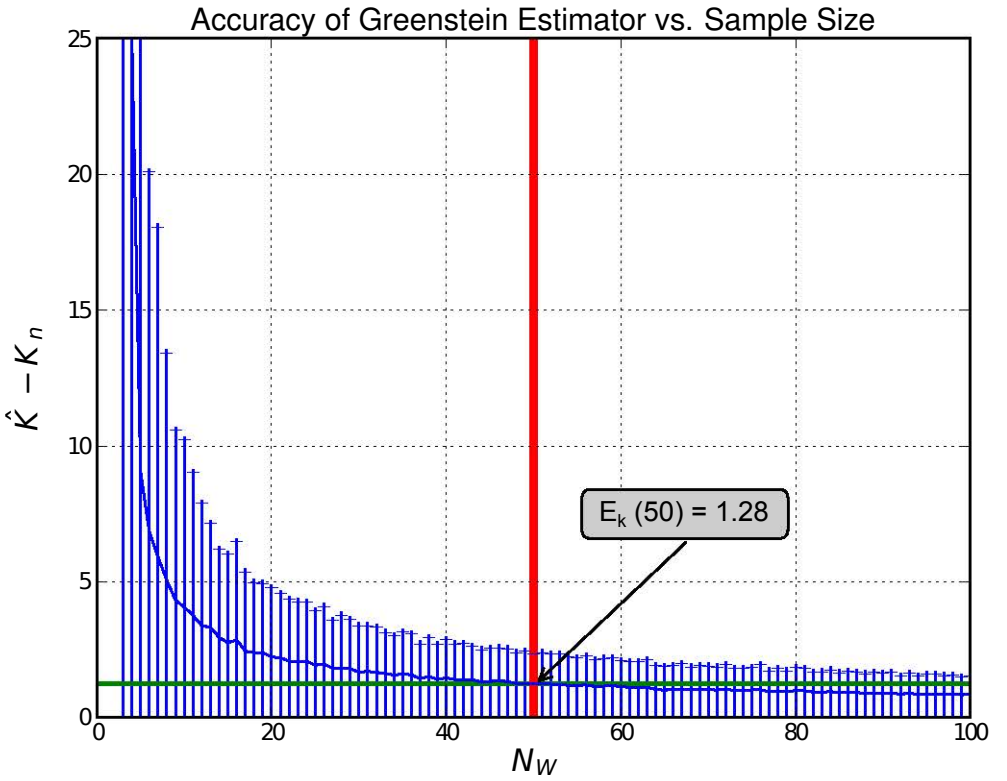


Figure 3.20: Error in the Greenstein estimator for different sample sizes

with 50 samples. A similar method of calculating an acceptable error threshold is also adopted in [13]. Based on this simple experiment, only locations at which more than 50 samples were collected are considered in the following analysis.

For a series of signal strength samples, the K-Factor is determined using the Greenstein estimator. Using this value as a starting point, unconstrained nonlinear optimisation, implemented using Matlab's `fminunc()` function, is then used to fit a value for the K-factor to the corpus of experimental data sets.

Two sets of experiments were performed to measure the fading parameters. The first was a stationary received power measurement between the transmitter and receiver at a distance of 220 m. Approximately 1000 samples were collected during these tests. A cumulative distribution function (CDF) of these measurements is presented in Figure 3.21. The measurements taken in the presence of vehicular traffic closely conform to an ideal Rician distribution; however, in the absence

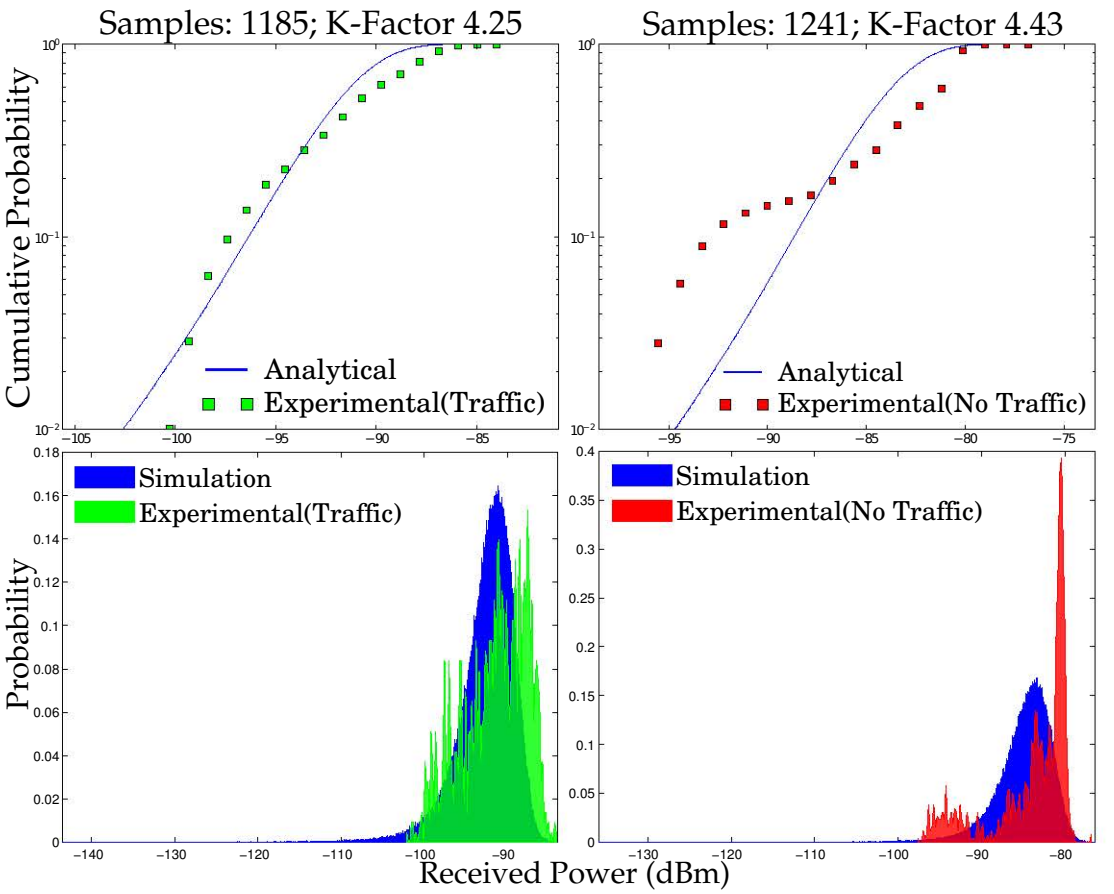


Figure 3.21: CDF plots for a stationary transmitter and receiver at 220 m

of vehicular traffic, it is clear that the CDF is significantly non-Rician. In fact, fading appears to be negligible in this situation. Due to the absence of moving vehicles, the only changes in the channel are in the atmosphere and the movement of smaller objects such as foliage, resulting in much less fading. This can be seen in the corresponding histogram, where the majority of the measurements are around -80 dBm.

The second experiment is an analysis of the signal strength measurements while the vehicles were in motion. A selection of these results is seen in Figure 3.22. These measurements clearly follow a Rician distribution.

Figure 3.23 shows the K-factor obtained in the presence and absence of traffic as a function of distance. The plot only shows positions with more than 30

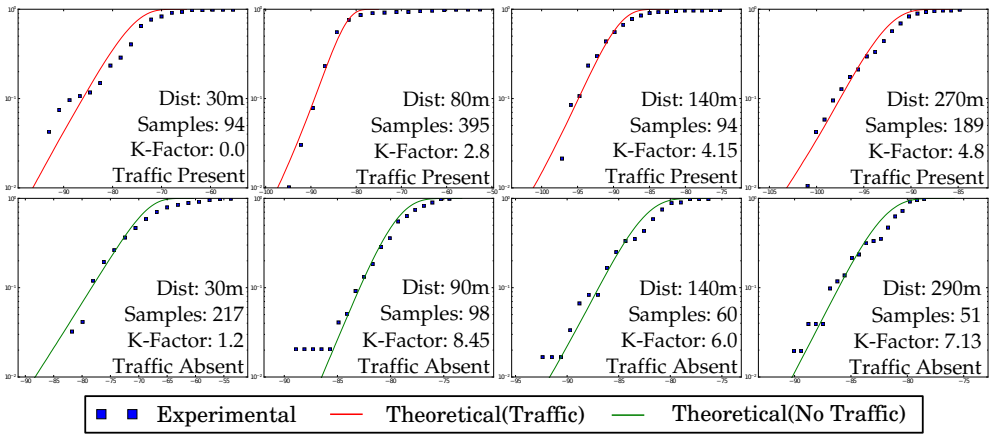


Figure 3.22: Selected CDF plots for the mobile LOS measurements

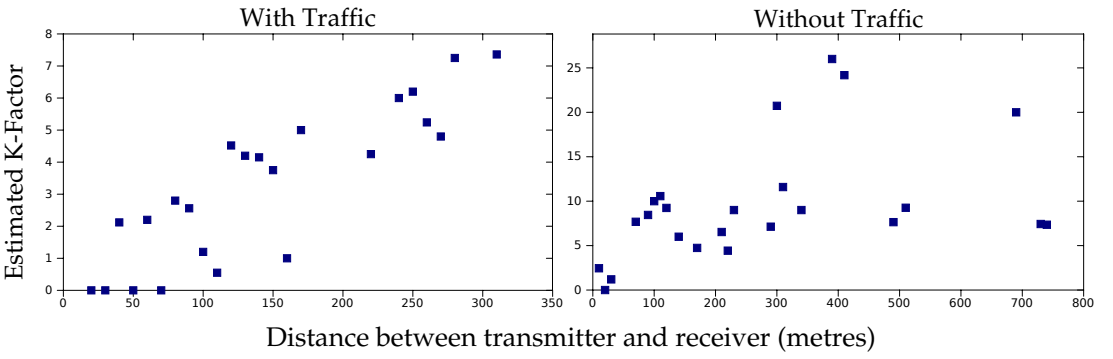


Figure 3.23: Scatter plot of the K-factor vs. distance for a sample size > 50

measurements. In the experiment with traffic, only one of the 18 positions failed to conform to a Rician distribution and only two out of 33 positions failed to conform in the experiment without traffic.

The experiment carried out in moderate traffic shows a clear upward trend in K-factor as the transmitter-receiver distance increases. This is expected due to the increasing relative strength of the specular ray compared to the reflected rays as the distance increases.

Unlike the aforementioned experiment, there is no consistent upward trend in K-factor in the experiment carried out in the absence of traffic. The waveguide-like effect of the buildings is clearly visible here, with the K-factor oscillating between a high and low value between 200 m and 500 m. This potentially represents alternating areas of constructive and destructive interference of the reflected component of the signal. This graph also exhibits a much larger range of results, due to the increased transmission range in the absence of vehicles.

3.4.5 Conclusions from Urban Experiment

The urban experiment validates the CORNER propagation model proposed in [5] using signal strength measurements. It was found to very accurately predict signal strength in dense urban areas. In more sparse areas, the model was less accurate, especially as the density of buildings decreased sufficiently to allow signals to propagate between them. Measurements taken on a long, straight street also established that the received signal strength conformed substantially to predictions from the free-space path-loss model in the presence of traffic, while the distribution changes to Rician in the presence of traffic. In the absence of traffic, a 'waveguide' like effect was observed where signal strength was boosted significantly due to constructive interference from reflections off buildings. Further experiments are required to study this effect in more detail.

3.5 Improvements and Additions to CORNER

The most critical shortcoming with the CORNER propagation model is the existing classifier, which does not consider nodes further than two intersections apart to be within line of sight. Giordano et al. do not specifically justify this heuristic in the original paper. Furthermore, the results in the Burelli street experiments clearly show that this assumption is invalid, as the signal can certainly propagate well past two intersections, reaching up to 4 in the presence of traffic and well beyond 6 intersections in the absence of traffic. Obviously, in practice, the range of signal propagation is a function of many factors, including transmission power and antenna gain. The transmission range greatly influences routing protocol behaviour, since it affects both connectivity and collision rates. Therefore, improvements to this element of the design of CORNER present an opportunity to significantly enhance its accuracy and utility in the simulation of VANETs and the evaluation of routing protocols in particular.

The Rician fading characteristic of the VANET channel was previously demonstrated in the experiments detailed in the previous section. Therefore, a Rician fading estimator is also added to CORNER to simulate the effects of a fading channel.

The new model, which incorporates all of the aforementioned improvements, is named CORNER++.

3.5.1 The new CORNER Profiler

The new profiler permits connectivity with nodes separated by more than two intersections. The algorithm is shown in Algorithm 1. It is a simple recursive procedure which is executed before the simulation to generate a list of 'classifications' relating any two links in the network. The algorithm recursively performs a depth-first search to find all links from an originating link. Each time a new link is reached, the classification relating the original link and the new link is stored.

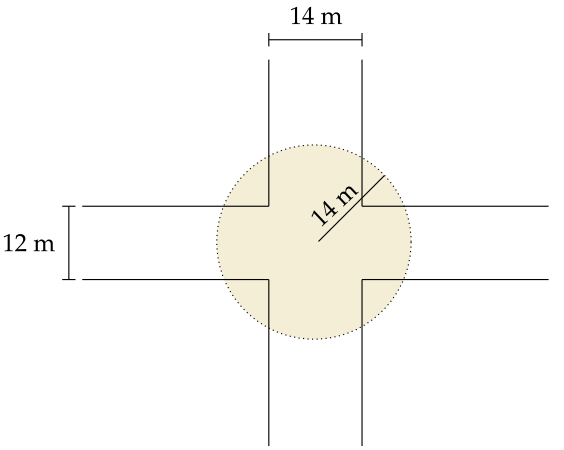


Figure 3.24: Intersection radius

When a link is reached past NLOS2, the classifier terminates that particular trace.

An angular threshold, θ, is defined as the NLOS angle. During a trace, if θ is exceeded on a particular link, that link is considered to be in NLOS with respect to the starting link. The same process is used to determine if a link is two corners away from the starting link. The information is stored in a classification file, which is loaded by Qualnet when the simulation is run. During simulations, the source and destination coordinates are mapped onto their respective links and the classification is obtained from a look-up table. This also greatly improves the speed of the simulation since almost no computation is required during the classification stage.

The disadvantage of this approach is the assumption that two nodes on a pair of links share a common classification regardless of their location on the link. This can lead to situations where two nodes very close to an intersection are incorrectly listed as NLOS1, despite being in direct line of sight. To mitigate these situations, the radius of the intersection is calculated as the maximum width of the streets connected to it. This is illustrated in Figure 3.24. Any nodes which fall within this radius are considered as being on either of the links connected to it, with

Algorithm 1: Classifier($\vec{C}, P, R, E$)

Data: $\vec{C}$ (Current Link), P (Current Path), $\vec{P}$ (Last link in P), R (Current Reduced Path), $\vec{R}$ (Last link in R), E (Entry Node of a Link), X (Exit Node of a Link), θ (LOS Angle), ζ (Current classification)

begin

$P \longleftarrow \vec{C}$;

if $min(|arg(C) - arg(P)|, |arg(C) - arg(R)|) > \theta$ **then**

$R \longleftarrow \vec{P}$

ζ = Number of Links in R

if $\zeta > 2$ **then**

Return Unclassified

Add (ζ, R) to classification list

for $\vec{l} \in \vec{C}.X$ *and* $\notin P$ **do**

Classifier($\vec{l}, \vec{P}, \vec{R}, \vec{C}.X$)

End This Classification

end

the classifier choosing the best link. This method is a compromise between performance and accuracy and works best on regular grid-like networks. A more accurate method would be to pre-compute the classification between quantised points of a specified road network by extrapolating the location of buildings and using a ray-tracing approach. This is however computationally intensive and is left as possible future work.

3.5.2 Addition of Rayleigh and Rician Fading

As demonstrated in section 3.4, Rician fading is apparent in LOS scenarios in the presence of traffic. Further analysis of the results, presented in [14] also shows that the K-Factor value is often close to zero in NLOS situations, leading to Rayleigh fading in these cases. It is therefore desirable to simulate this behaviour as part of the propagation model. The implementation of Rayleigh and Rician fading in this thesis uses an approximation of Clarke's model [86, 87], which is based on a sampled normal (Gaussian) distribution.

Clarke's model for Rayleigh fading is

$$g(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{j(\omega_d t \cos\theta_n + \phi_n)}, \tag{3.28}$$

where N is the number of reflections, ω_d is the maximum Doppler frequency (in radians per unit time) and θ_n and ϕ_n are the angle of arrival and initial phase of the n^{th} propagation path.

This can be expressed in terms of its real and imaginary components

$$g_c(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \cos(\omega_d t \cos\alpha_n + \phi_n) \tag{3.29}$$

$$\text{and} g_s(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \sin(\omega_d t \cos\alpha_n + \phi_n) \tag{3.30}$$

respectively. The path-loss due to Rayleigh fading in dB [33] is calculated as:

$$Rayleigh_{dB} = 10\log_{10}\left(\frac{[g_c(t)]^2 + [g_s(t)]^2}{2}\right). \tag{3.31}$$

The Rician fading model is obtained by adding a dominant signal to Clarke's model. The ratio of the power of the dominant signal to that of the sum of the reflected signals is represented by the Rician parameter K. The path-loss due to Rician fading in dB is calculated as [87]

$$Rician_{dB} = 10\log_{10}\left(\frac{[g_c(t) + \sqrt{2K}]^2 + [g_s(t)]^2}{2(K+1)}\right). \tag{3.32}$$

It is assumed that N is large, i.e. there are many reflected signals in an urban environment. Thus, from the central limit theorem, both $g_c(t)$ and $g_s(t)$ can be approximated as Gaussian-distributed random variables.

Rayleigh fading is applied to NLOS situations and Rician fading is applied to LOS situations in CORNER++.

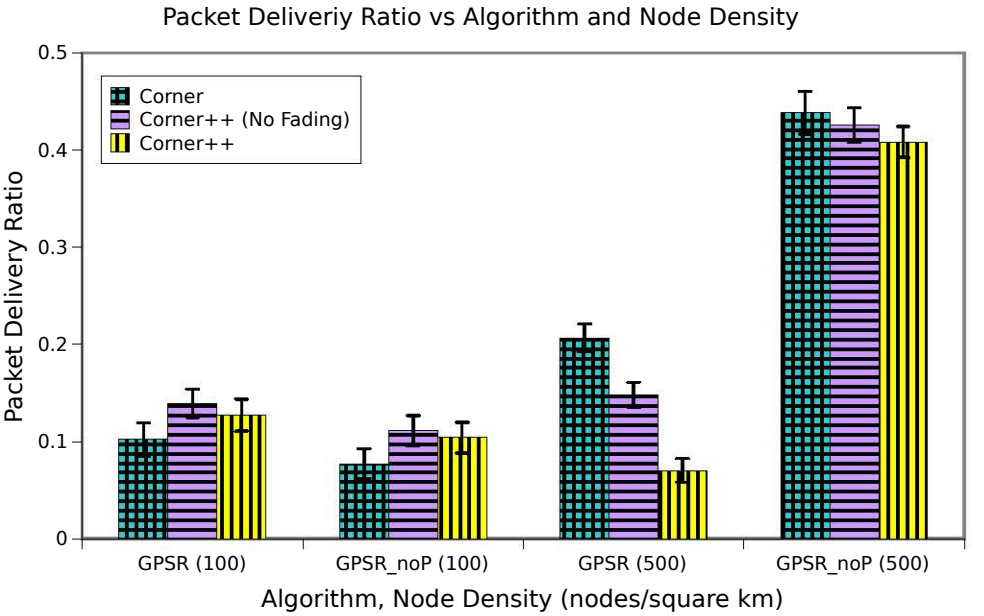


Figure 3.25: Comparison of GPSR's packet delivery ratios under CORNER and CORNER++

3.5.3 Comparison of Results between CORNER and CORNER++

Figure 3.25 illustrates the effect of the new classifier and also the impact of the addition of the fading model by comparing CORNER++ (with and without fading) to CORNER under the GPSR routing protocol. The metric presented in the graph is that of packet delivery ratio.

At low node densities, the 2-link restriction imposed by CORNER results in fewer packets being delivered since fewer nodes are visible. At higher node densities, the 2-link restriction aids in network segmentation, thereby resulting in fewer packet collisions and hence a higher packet delivery ratio.

This is an especially useful result because the old model would have led to the mistaken belief that increasing the node density by a factor of five would lead GPSR to deliver twice as many packets, resulting in an inaccurate representation of its scalability. In fact, the apparent scalability of GPSR is due to the

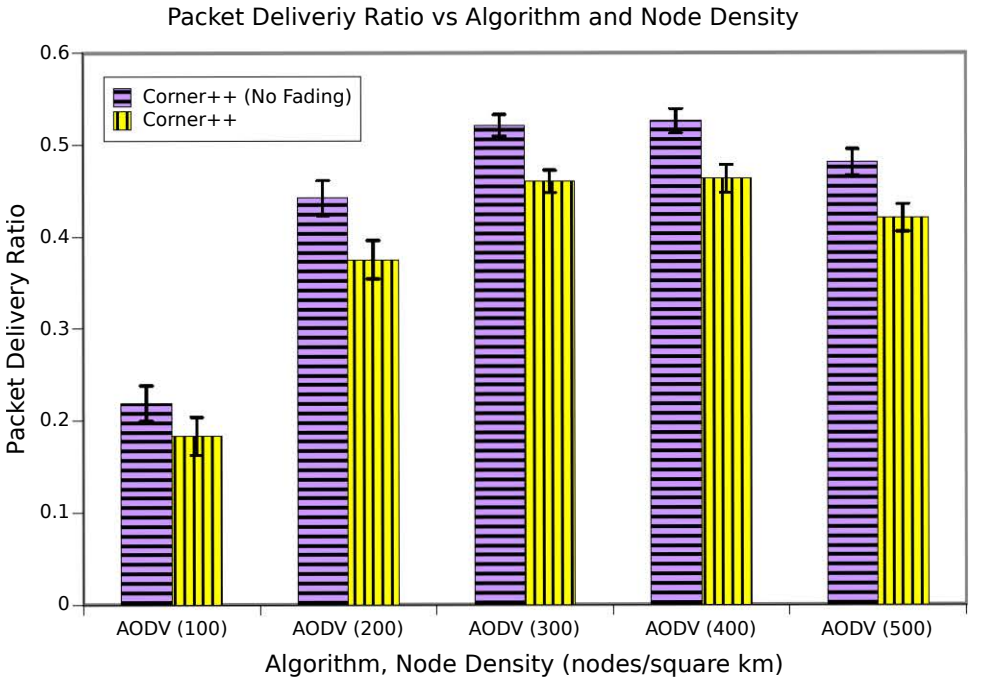


Figure 3.26: Comparison of AODV's packet delivery ratios with and without fading under CORNER++

effective segmentation of the network caused by the assumptions intrinsic to the CORNER model. By contrast, CORNER++ shows that the positive impact of more connectivity is significantly offset by the negative impact of the increased collision rate.

The addition of fading also slightly reduces packet delivery ratios. Fading has a similar impact on AODV, as seen in Figure 3.26, resulting in an almost constant drop in packet delivery ratios across different node densities. This is due to the decreased probability of reaching neighbours at the very edge of a node's radio range due to the random time-varying nature of fading.

3.6 Conclusions

This Chapter presented an overview of various propagation models either designed for urban environments or used in the validation of routing protocols

designed for urban environments. The impact of the propagation model on the routing protocol metric of packet delivery ratios was considered and found to be significant.

The CORNER propagation model was studied in detail and validated independently in experiments. The urban vehicle to vehicle signal strength measurements gathered during the experiment showed very good agreement with the predictions from the model in the NLOS scenarios and also revealed a deficiency in the classifier, which arbitrarily restricts path-loss predictions to 2 intersections.

A new classifier was proposed to address this deficiency. Furthermore, a fading simulator was added to the model after experiments revealed the presence of a Rice-distributed fading channel. The new model was named CORNER++ and was compared with the old CORNER model and found to have a significant impact on routing protocol performance, particularly on protocol scalability.

Chapter 4

A GREEDY AND STATELESS ROUTING ALGORITHM FOR URBAN VANETS

The previous two chapters presented a review of VANET protocols designed for urban environments and a detailed study of urban propagation models for VANETs. This Chapter proposes a new approach to routing in urban environments, which accounts for the propagation of a signal when making routing decisions. Urban environments, as discussed in Chapter 3 can be approximated with reasonable accuracy by the CORNER++ propagation model. This model is therefore used to make better routing decisions.

4.1 Analysis of Existing Routing Methodologies

This section discusses the advantages and shortcomings of the common approaches to routing in the VANET literature. A number of common approaches exist among the protocols reviewed in Chapter 2.

Of the 23 VANET protocols reviewed, only 3 (CAR, RBVT-R and MURU) use a stateful approach, i.e., they discover and *maintain* routes from source to destination. While protocols which use real-time connectivity awareness need some level of information about the city around them, the actual forwarding of packets from source to destination relies on a stateless approach in all except the three aforementioned protocols.

In this thesis, a protocol is considered stateless if it does not attempt to form and maintain routes from source to destination, relying instead on forwarding packets to a succession of neighbours until the packet reaches the destination.

The stateless approach is preferred in VANETs since a combination of street topology, radio obstructions and vehicular speed make it difficult to maintain a route between two communicating nodes. Instead, most protocols rely on maintaining 'anchor-based' routes, where the path from source to destination is defined by a series of markers such as intersections, which are static and may themselves not actually have (or require) any connectivity.

Twenty-two of the reviewed protocols use some form of greedy routing, with MURU being the only exception. Additionally, all but six of the protocols suggest a recovery strategy in case of routing failures. One other interesting feature of the reviewed protocols is that all of them leverage the knowledge of a node's geographical position. This is considered to be a reasonable assumption in VANETs, given the increasing prevalence of satellite navigation devices in vehicles.

4.1.1 Greedy and Stateless Routing

As noted above, greedy and stateless routing protocols are the most common in the literature. Greedy routing refers to the selection of the most optimal node out of a set of neighbours as the next packet relay. In most protocols, this often involves the selection of the node closest to a desired location, which may be a street intersection or the destination. This approach works well in flat geographic topologies, where signals propagate between a transmitter and receiver unimpeded, because the received signal power is inversely proportional to the square of distance. This makes Cartesian distance a convenient metric for selecting the optimal relay, since the neighbour which is closest to the destination should be the neighbour with the best probability of being able to communicate with the destination.

With the exception of GPSR, CBF, AGF and GRANT, in all protocols compared in Chapter2, greedy routing is only used between successive anchor points, which are usually street intersections. This is illustrated in Figure 4.1, where

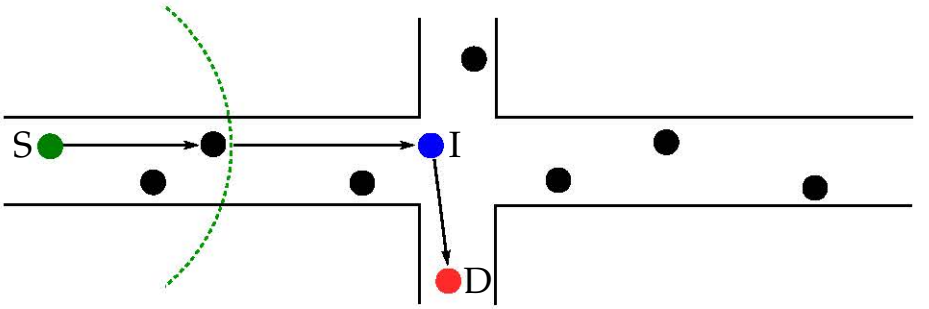


Figure 4.1: Anchor-based greedy forwarding

packets are forwarded to a node at the intersection before being forwarded down a different street. This design is due, in part, to the assumptions made about the propagation environment. When tested in simulations, most of the protocols assume either a very short radio range (50-200 m), or consider that signals only propagate along direct line-of-sight paths along a street. Both of these approaches effectively prevent vehicles from communicating around a street corner, thus modelling buildings as completely radio-opaque structures. In these situations, the most optimal approach is indeed to forward a packet towards a street intersection, which allows it access to other streets. A decision on the most suitable street on which to forward the packet may then be made at the intersection.

As noted in Chapter 3, a growing body of research has shown the assumption of limited propagation range to be a false assumption. The experiments carried out in Chapter 3 also support this conclusion. Radio frequency signals in urban environments are rarely restricted to direct line of sight (LOS) paths such as streets, with reflections and diffraction allowing them to propagate around corners. Only three of the aforementioned protocols (RBVT-R, RBVT-P and BAHG) have been evaluated in a propagation environment that considers any non-line-of-sight (NLOS) paths at all. However, in all of these cases, NLOS propagation is not modelled accurately as it is either assumed to be a fixed attenuation of 5 dB (BAHG) or a randomly varying attenuation (RBVT). Despite the use of

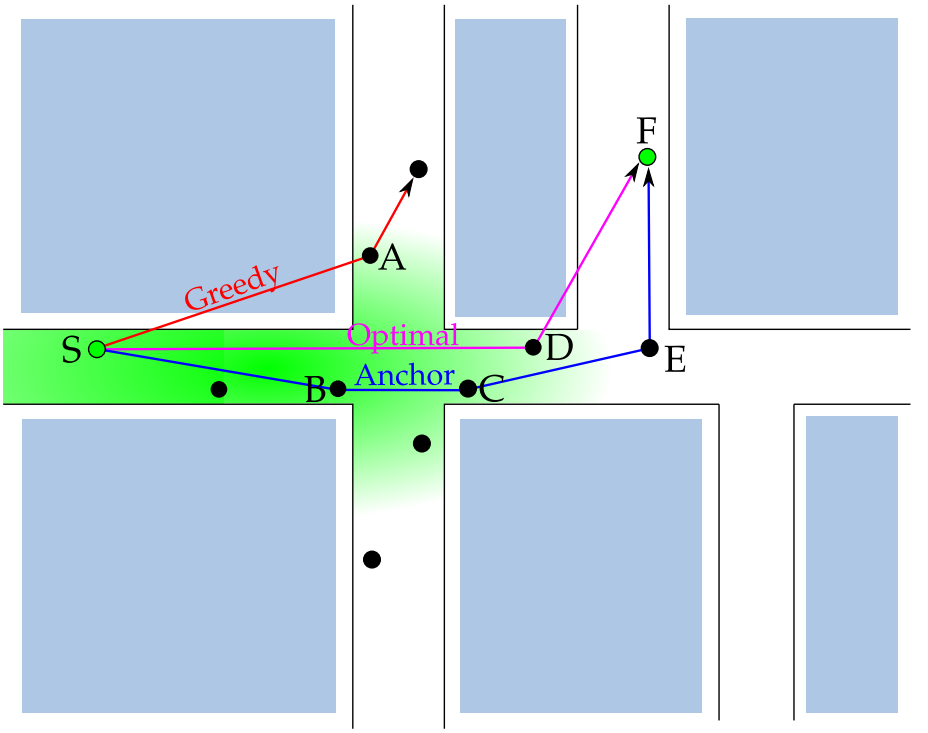


Figure 4.2: Illustration of paths taken by different greedy algorithms

a more realistic propagation model, these protocols adopt the prevailing design decisions in the literature by employing anchor-based routing approaches.

Consider Figure 4.2, which illustrates the routes potentially taken by both a greedy and an anchor-based routing protocol. The greedy protocol will not be able to reach the destination without a fall-back mechanism since the packet is routed down a parallel street and connectivity is completely obscured by a building. The anchor based approach forwards the packet through nodes B, C and E since they represent nodes closest to a junction on the shortest street path. Neither approach is able to take the optimal path in this particular vehicular environment.

Therefore, there is significant potential for improvement in existing greedy and stateless routing approaches.

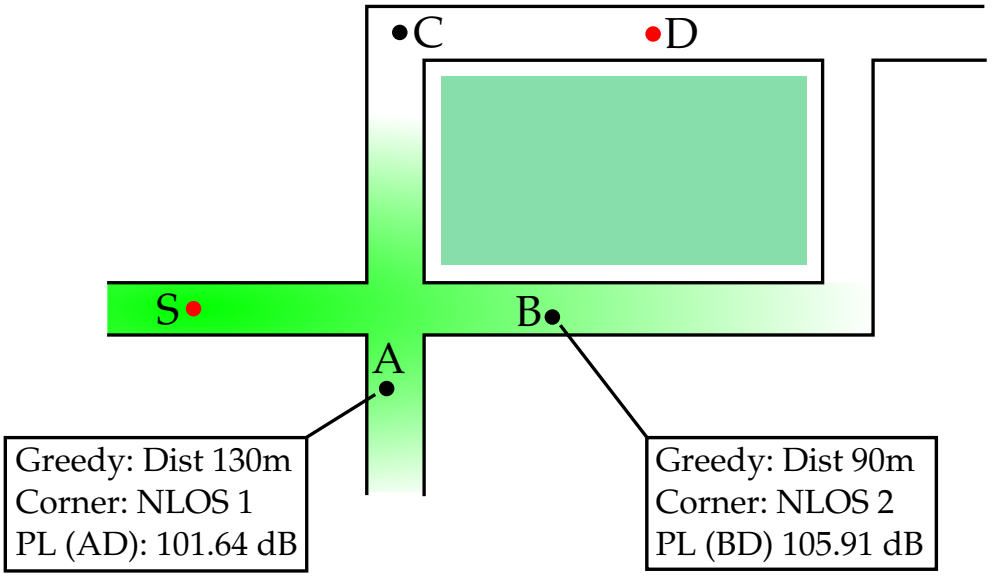


Figure 4.3: Corner and greedy based path-loss predictions

4.2 Proposed Algorithm

The main aim of the proposed algorithm is to select nodes that have the best probability of successfully communicating with the destination in urban environments. In order to achieve this, the previously described CORNER propagation model is used to estimate the path-loss between the neighbours of a node and the destination. The proposed routing algorithm is named Corner Propagation based Stateless Routing (CPSR) and is a greedy algorithm in the vein of GPSR.

For example, Figure 4.3 illustrates the difference between the greedy decision making process and the CORNER-based process. Node S, needing to transmit a packet to node D, needs to choose between nodes A and B for the next hop. Node A, while further away in distance, would exhibit a lower path-loss due to it being in NLOS1 from node D. Even though A cannot directly see node D, it is better placed to communicate with node D than node B in a city since LOS range is significantly longer than NLOS range, as demonstrated in Section 3.4.

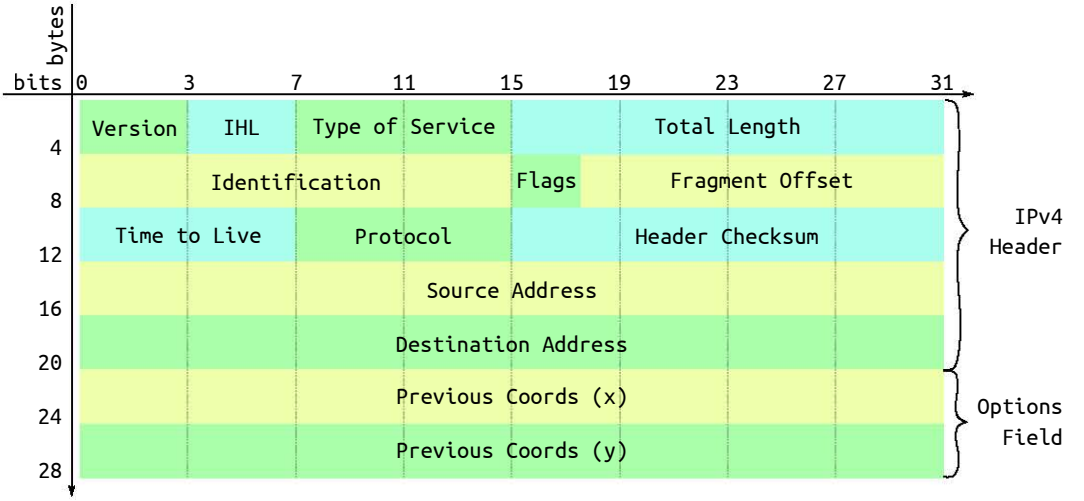


Figure 4.4: CPSR IPv4 Header with coordinates appended to `options` field

4.2.1 Protocol Design

Like GPSR, each node advertises its presence and location using beacons, which are transmitted at user-defined intervals. The beacons carry a node's present coordinates, represented as two 4-byte floating point values: the node's Cartesian coordinates in x (horizontal or East-West) and y (vertical or North-South) respectively.

Nodes also append their coordinates to any other packet that is transmitted and defer from sending a beacon during an interval that any such packet has been transmitted, reducing the beaconing traffic overhead on the network. The coordinate information is added to the IPv4 `options` field [88], as illustrated in Figure 4.4. When a node hears a beacon or promiscuously hears a unicast packet, the neighbour's address and location are extracted and added to the local neighbourhood table.

When this node subsequently needs to transmit a packet, it estimates its own path-loss to the destination and then estimates the path-loss of each of its neighbours to the destination. The aim of the greedy protocol is to forward the packet to a neighbour that has a highest probability of reaching the destination, which

in this instance is the neighbour with the lowest path-loss to the destination. The path-loss is determined according to the method described in section 3.5. The street-location of a node and its neighbours is determined through reverse geo-coding, which is the mapping of a set of coordinates onto the nearest road. Reverse geo-coding is used by GPS devices to determine a vehicle's most likely position on a particular road, given the coordinates obtained from the GPS receiver and its knowledge of the area's street maps.

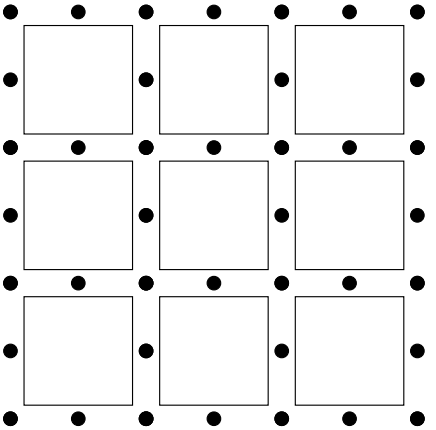
This thesis does not address the discovery of the destination node's position and it is assumed that the location is readily available. There are multiple works which deal with this issue such as the Grid Location Service [23]. Das et al [24] also review other location discovery services. It is envisioned that this information may also be obtained in the future through a cellular network which can be used as a low bitrate control channel for VANETs.

As with the vast majority of the VANET literature surveyed in Chapter 2, destination location discovery is not addressed directly in the algorithm. The presence of a location service is assumed. Several location discovery algorithms are surveyed in [24].

If the most optimal neighbour has a greater path-loss to the destination than the source node, a failure condition is reached since the node itself is a more greedy option than its neighbours. In these instances, the node reverts to a geographically greedy decision making process in order to move the packet closer to the destination, where a more suitable neighbour may be found. Further discussion on recovery mechanisms can be found in Chapter 5.

4.3 Studying the Path-Finding Abilities of the New Algorithm

A rigorous analysis was performed using a simplified grid-like scenario to study the path-finding abilities of the proposed greedy routing algorithm. The analysis is presented in this section and aims to model the impact of the greedy



(a) Grid Illustration

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

(b) Matrix Representation

Figure 4.5: 7x7 Grid with evenly spaced transceivers

algorithm on the purely routing-layer metric of hop-counts without interaction with the other layers of the networking stack. The algorithm is compared to a geographically greedy algorithm and to a source routing algorithm such as GSR. The scenario presented in this section is a hypothetical best case where routes always exist between source and destination.

Figure 4.5 illustrates a 3×3 grid of buildings separated by streets, with three nodes (transceivers) spanning the beginning, middle and end of each building respectively. The same figure also illustrates how this grid may be represented as a matrix. Similarly, for an $n \times n$ grid of city blocks, the matrix representation would have a dimension of $(2n+1) \times (2n+1)$. All zeroes in the matrix represent buildings while the remainder represent nodes.

In order to assess the routing performance of the aforementioned algorithms, a series of routes is determined between every possible node-pair in the network for a range of different network sizes. Only routes with more than one hop are considered. The rest of this section details this model, presents the results and discusses their significance.

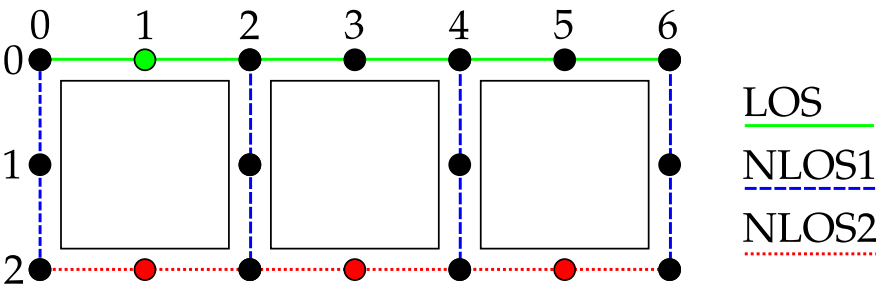


Figure 4.6: Numbered grid with NLOS1 paths illustrated

4.3.1 Simplifying CORNER Classifications for a regular Grid

As described in 3.3, CORNER classifies node-to-node connectivity (if it exists) into one of three classes:

1. Line-of-sight (LOS);
2. Non-line-of-sight around one corner (NLOS1); or
3. Non-line-of-sight around two corners (NLOS2).

The calculation of LOS/NLOS scenarios is simplified in this model due to the regularity of the grid. The algorithm for determining the appropriate CORNER scenario in this grid model is presented in algorithm 2.

Algorithm 2: Classification Algorithm for the Grid Scenario

Data: Coordinates a(x,y) and b(x,y)

if *(a.x = b.x and a.x is not odd) or (a.y = b.y and a.y is not odd)* **then**
  Classification is LOS;

else if *(a.x and b.x are odd) or (a.y and b.y are odd)* **then**
  Classification is NLOS2;

else
  Classification is NLOS1;

The three scenarios are illustrated in Figure 4.6, where LOS, NLOS1 and NLOS2 paths are illustrated with respect to node $(1, 0)$.

Two nodes are in LOS in the aforementioned grid scenario if they share either their x-coordinate or their y-coordinate and there is no building obstructing the link. Since buildings only occur on odd-numbered rows and columns, nodes are in LOS as long as their shared coordinate isn't odd.

Similarly, NLOS2 scenarios only occur if both nodes share an odd-valued coordinate. With reference to Figure 4.6, two nodes on odd-numbered columns can only ever be in NLOS2 due to the need to circumscribe two buildings.

Once LOS and NLOS2 scenarios are eliminated, only NLOS1 scenarios remain.

4.3.2 Simplifying the CORNER Path-loss Formulae for Uniform Grids

The path-loss formulae presented in this section are adapted and simplified for the grid scenario from the formulae originally derived in [9], upon which the CORNER propagation model is based. In the present analytical scenario, the assumptions about a uniform street width and building size allow the expression for path loss to be significantly simplified.

Once again, following term, which represents the dependence of received signal power on frequency (or wavelength), is represented by the variable L_2:

$$L_2 = \left(\frac{c}{4\pi f}\right)^2 = \left(\frac{\lambda}{4\pi}\right)^2, \tag{4.1}$$

where f and λ are the frequency and wavelength of the signal, respectively.

Additionally, for this grid scenario, unity gains, losses and transmit power ($G = 1$) are assumed. Therefore, the total received signal power S for a signal with total path-loss PL_{total} is

$$\begin{aligned} S &= G \times PL_{total} \\ &= PL_{total} \end{aligned} \tag{4.2}$$

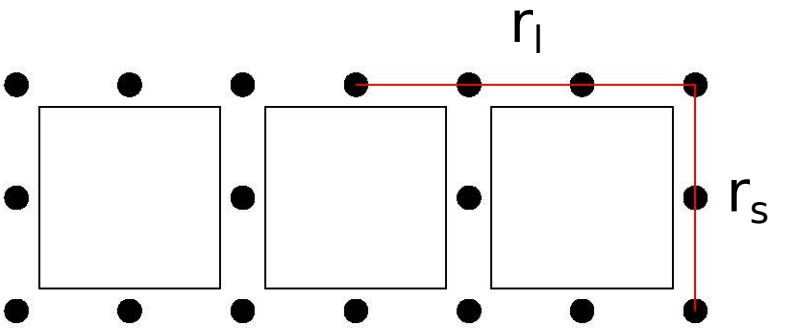


Figure 4.7: NLOS1 Scenario with main and side roads indicated

The following variables used in the model:

- Road width: W
- Building length/width: B
- Distance between adjacent nodes in LOS path: $D = {}^{(W+B)}/_{2}$

Line-Of-Sight (LOS)

The signal received in LOS scenarios is represented by the well-known free-space path-loss formula as stated in equation 3.13.

$$S_{LOS} = \frac{L_2}{d^2} \tag{4.3}$$

Non-Line of Sight Around One Corner (NLOS1)

The signal power received in NLOS1 scenarios is defined as the sum of the signal power received from the reflected and diffracted paths, i.e.,

$$S_{NLOS1} = PL_R + PL_D \tag{4.4}$$

In the NLOS1 case, the minimum number of reflections for the reflected signal,

N_{min}, defined in equation 3.14 can be simplified due to the identical street widths in the grid model as follows:

$$\begin{aligned} N_{min} &= \left\lfloor 2\sqrt{\frac{r_s r_m}{W_s W_m}} \right\rfloor \\ &= \left\lfloor \frac{2}{W}\sqrt{r_s r_m} \right\rfloor, \end{aligned} \tag{4.5}$$

In this scenario, the variables r_m and r_s represent the main (longer) road and the side (shorter) road of the NLOS1 path. as illustrated in Figure 4.7. The variables W_m and W_s represent the respective widths of the main and side roads respectively, which in this scenario are identical. R_0 represents the ratio of the signal carried through each reflection, i.e., an R_0 value of 0.9 would mean that 10% of the signal is lost with each reflection.

The path-loss of a signal arriving through reflected paths, PL_R, defined in equation 3.16 is:

$$PL_R = \frac{L R_0^{2N_{min}}}{(r_m + r_s)^2}. \tag{4.6}$$

Similarly, the path-loss of a signal arriving through diffracted paths, PL_D is defined in equation 3.17. In the grid model, r_m is defined as the longer street and r_s as the shorter street, as illustrated in Figure 4.7. If the street lengths are the same, the values of r_m and r_s are interchangeable. Therefore, given that $r_m \geq r_s$:

$$PL_D = \frac{L_2 \lambda}{4 r_m^2 r_s} \tag{4.7}$$

Non-Line of Sight Around Two Corners (NLOS2)

Recall that the path-loss in NLOS2 scenarios was defined in section 3.3 as the sum of

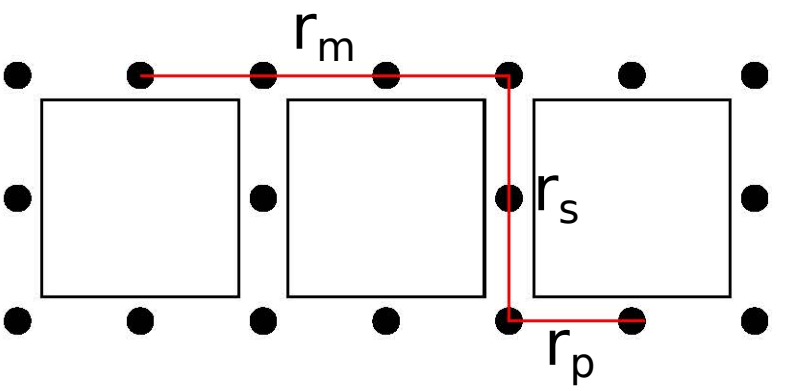


Figure 4.8: NLOS2 Scenario with main, side and parallel roads indicated

1. Path-loss from reflection around both corners;
2. Path-loss from a reflection around 1 corner and diffraction around the other corner;
3. Path-loss from a diffraction around 1 corner and reflection around the other corner; and
4. Path-loss from diffraction around both corners.

Mathematically, due to the assumption of an overall unity gain in the system, the received signal in NLOS2 scenarios is represented as

$$S_{NLOS2} = PL_R + PL_{RD} + PL_{DR} + PL_{DD} \tag{4.8}$$

In this scenario, the variables r_m, r_s and r_p represent the lengths of the main road, the side road and the parallel road of the NLOS2 path as illustrated in Figure 4.8. The variables W_m, W_s and W_p represent their widths respectively. Note that due to the use of a grid scenario, r_m and r_p are interchangeable in the following equations.

The minimum number of reflections in an NLOS2 scenario is defined in equation 3.18 as:

$$N_{min} = \left\lfloor \frac{r_m}{W_m}\sqrt{\frac{r_s W_m W_p}{W_s(r_m W_p + r_p W_m)}} + \frac{r_s}{W_s}\sqrt{\frac{W_s(r_m W_p + r_p W_m)}{r_s W_m W_p}} + \frac{r_p}{W_p}\sqrt{\frac{r_s W_m W_p}{W_s(r_m W_p + r_p W_m)}} \right\rfloor \tag{4.9}$$

This equation includes a common factor, defined in equation 3.19 as:

$$t = \sqrt{\frac{r_s W_m W_p}{W_s(r_m W_p + r_p W_m)}},$$

Due to the assumption of common street widths,

$$W = W_m = W_s = W_p \tag{4.10}$$

Therefore, by substituting in (3.19) and (4.10) into (4.9), it can be simplified as follows:

$$\begin{aligned} N_{min} &= \left\lfloor \frac{r_m t}{W} + \frac{r_s}{Wt} + \frac{r_p t}{W} \right\rfloor \\ &= \left\lfloor \sqrt{\frac{r_s}{r_m + r_p}}\frac{r_m + r_p}{W} + \sqrt{\frac{r_m + r_p}{r_s}}\frac{r_s}{W} \right\rfloor \\ &= \left\lfloor \frac{2}{W}\sqrt{r_s(r_m + r_p)} \right\rfloor \end{aligned} \tag{4.11}$$

As before, the formulae presented in equations 3.22, 3.24, 3.25, and 3.23 are simplified for a grid environment and presented below:

$$PL_R = \frac{L R_0^{2N_{min}}}{(r_s + r_m + r_p)^2}, \tag{4.12}$$

$$PL_{DD} = \begin{cases} \frac{L\lambda^2}{16 r_m r_s r_p^2} & r_m < r_s \\ \frac{L\lambda^2}{16 r_m^2 r_p r_s} & r_s \leq r_m \end{cases}, \tag{4.13}$$

$$PL_{RD} = \frac{LR_0^{2N_{min}}\lambda}{4(r_s + r_m)^2 r_p}, \text{ and} \tag{4.14}$$

$$PL_{DR} = \begin{cases} \frac{R_0^{2N} L\lambda}{4r_m(r_s+r_p)^2} & r_m < r_s + r_p \\ \frac{R_0^{2N} L\lambda}{4(r_s+r_p)r_m^2} & r_s + r_p \leq r_m \end{cases}, \tag{4.15}$$

where N , which is a common factor removed from equation 3.25 indicating the minimum number of reflections of the reflected component following diffraction, is defined as

$$\left\lfloor \frac{r_p r_s}{W^2} \right\rfloor. \tag{4.16}$$

The difference between the expressions for PL_{RD} and PL_{DR} is due to the simplification of the CORNER model for the grid scenario. NLOS2 paths are always calculated with a unit-length parallel street as illustrated in Figure 4.8. This results in the simplification of the expression for the PL_{RD} case since $r_s + r_m$ is always greater than r_p.

4.3.3 Analytical Studies

In order to study the differences between GPSR's geographically greedy algorithm, the source-routing algorithm which is used in GSR, and the CORNER based greedy algorithm, the grid and classification system described previously was implemented in Python [89]. The GSR routing algorithm effectively restricts the greedy algorithm to line-of-sight paths only. Under the ideal-connectivity grid scenario under which this study is performed, this should result in identical route-selections to GPCR and several other of the anchor-based protocols discussed in Chapter 2 which impose the same restrictions as GSR.

The grid represents a regular street scenario, with idealised connectivity. The number of hops required for every source-destination pair is computed using the parameters shown in Table 4.1.

Grid size	$[5, 10, 15, 20, 25, 30]$ buildings
Wavelength (λ)	0.125 m
Reflection transmission ratio (R_0)	0.9
Building size (B)	80 m
Street width (W)	20 m
LOS range	$[100 - 1000]$ m

Table 4.1: Parameters for the analytical scenario

The LOS range is used to calculate the reception threshold of the system under free-space path-loss. This reception threshold is then used to compute which points are visible under NLOS conditions. Recall the free-space received-power equation (3.3):

$$P_r = \left(\frac{P_t G_t G_r}{L}\right)\left(\frac{\lambda}{4\pi d}\right)^2 \tag{4.17}$$

As stated in section 4.3.2, an overall unity gain is assumed for the system along with a unity transmit power. Therefore, for an LOS range of 100 m to 1000 m, this leads to a corresponding minimum signal reception threshold of -80.40 dBm to -100.04 dBm. Since the reception threshold does not vary in real life, it is useful to represent the LOS range as a function of transmission power. Therefore, normalising the reception threshold at -110 dBm leads to a transmission power of -30 dBm to -10 dBm at a corresponding LOS range of 100 m to 1000 m. It is important to note that the results are still presented as a function of LOS range, and the transmission power value provided above is purely for illustrative purposes. A sample connectivity map at an LOS range of 600m is shown in Figure 4.9.

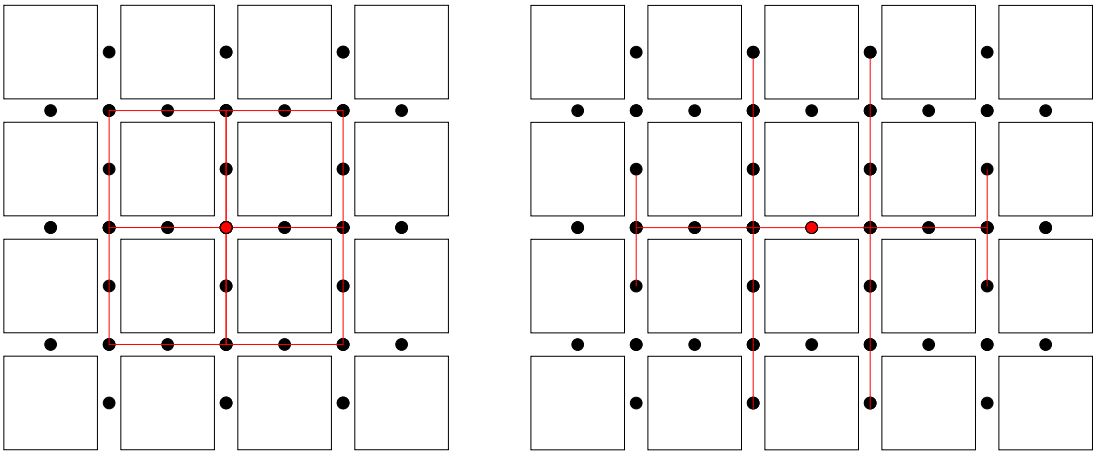


Figure 4.9: Connectivity in NLOS scenarios on the grid map at an LOS range of 600 m

4.3.4 Simulation Results Using the Proposed Model

The model does not attempt to characterise the MAC layer interactions such as packet collisions and instead attempts to purely characterise the routing layer metric of hop-counts. In effect, the algorithms are judged by their abilities to find the shortest paths in a scenario with ideal connectivity. The impact of transmission power, represented as the LOS range and the size of the grid are both studied.

Path-Finding Example

An example is shown in Figure 4.10, illustrating paths found by CORNER, Greedy and Source routing algorithms respectively. In this figure, a path is traced from node $(0, 18)$ to $(19, 0)$. The specific parameters for this figure include an LOS range of 600 m and a grid size of 30 m. The rest of the parameters are identical to those stated in Table 4.1. In this Figure, the greedy algorithm is seen to take 5 hops to reach the destination, with the source routing algorithm taking 4 and the CORNER algorithm taking 3. While it is possible for the source routing algorithm to follow the same path as the CORNER model, the actual

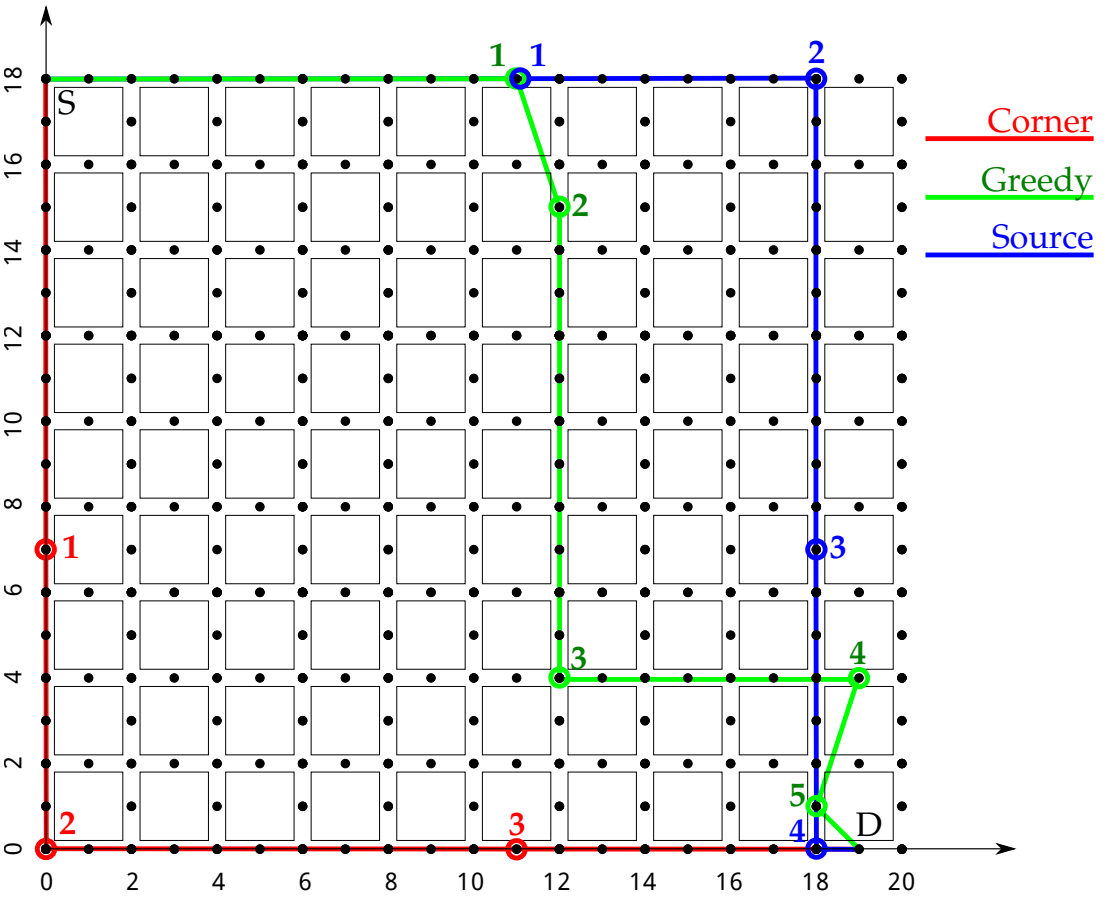


Figure 4.10: Example route determined by each of the three algorithms

path taken has the same 'Manhattan-distance', which is the total street distance covered by the path. This results in both paths being considered equally optimal under the source routing algorithm, and makes the initial direction selection significant. Since the decision is made in a blind manner, both paths are equally probable.

Hop Counts and Distribution at an LOS Range of 600 m

Figure 4.11 shows the average hop-count for the various routing algorithms at an LOS range of 600 m. As seen in the Figure, the CORNER-based greedy algorithm consistently finds shorter paths than either a purely greedy algorithm or the source-routing algorithm characteristic of GSR. In order to further analyse

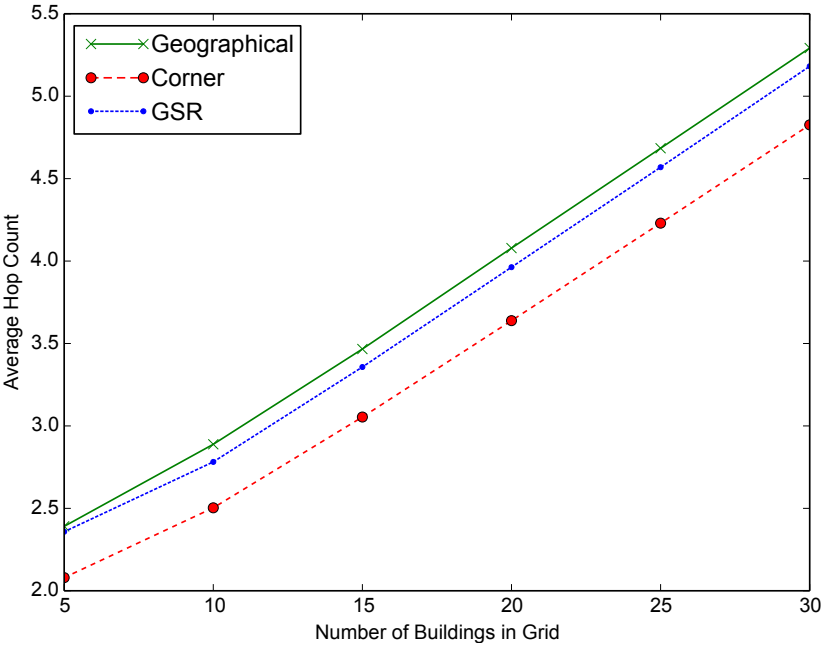


Figure 4.11: Average hop-count against different grid sizes at 600 m LOS range

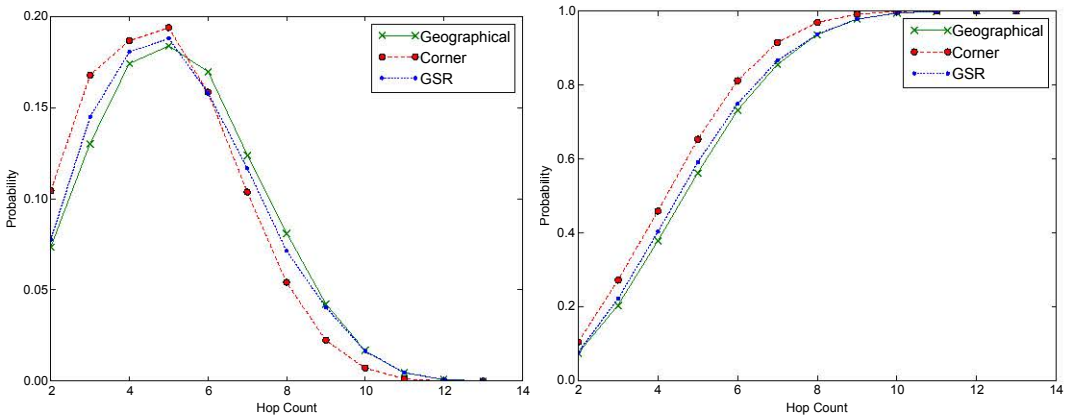
this result, the probability and cumulative distributions of hop-counts at the same LOS range, with a grid size of 30 m are shown in Figure 4.12.

As illustrated in the Figure, the CORNER-based greedy algorithm finds shorter paths with greater frequency than the other algorithms, resulting in more paths at less than five hops.

Impact of LOS Range on Hop Counts

Figure 4.13 shows the impact of LOS range on average hop counts. As expected, when the range is extremely short, there is almost no difference between the average hop counts obtained using the different routing algorithms, since the connectivity limits all paths to very short LOS paths. Each algorithm would therefore select the same path from source to destination. However, as the range increases, the differences become more apparent.

Figure 4.14 shows the difference in hop count for the three routing protocols at



(a) Probability distribution (b) Cumulative distribution

Figure 4.12: Distribution of hop counts for a 30 m×30 m grid at an LOS range of 600 m

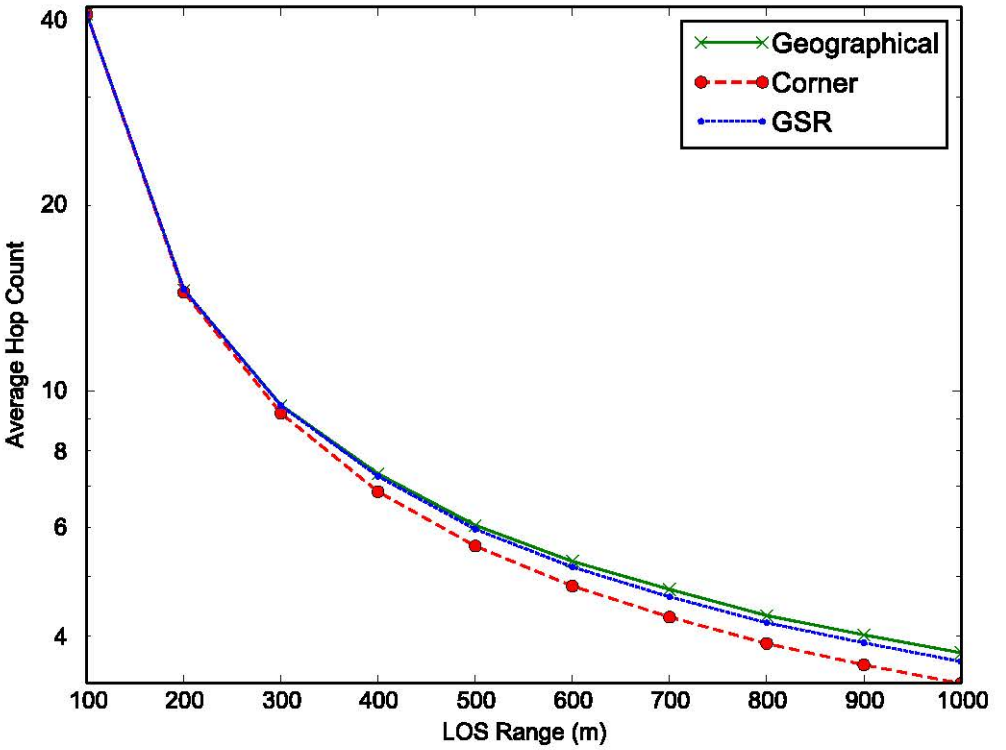


Figure 4.13: Impact of LOS transmission range on average hop count for a 30 m×30 m Grid

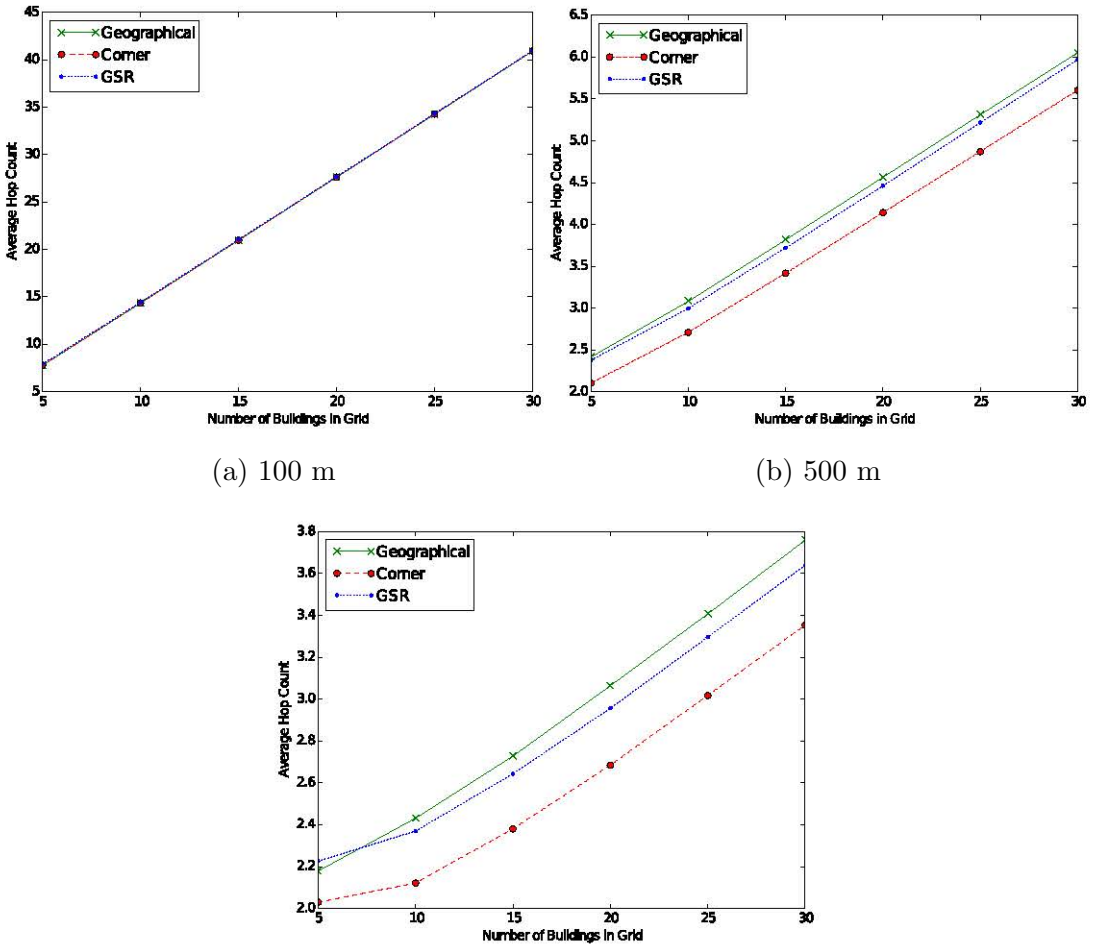
three specific distances. It is interesting to note that at a large LOS range, GSR performs worse than greedy routing when the grid size is small. This is due to the restriction of GSR to LOS paths, forcing it to take a longer path than greedy routing when several NLOS paths are available. However, GSR continues to outperform greedy routing when the grid size is larger than 10 buildings.

4.3.5 Summary of Analytical Model

This section established the theoretical path-finding benefits of using CORNER in a regular grid-like scenario. The CORNER based path-finding algorithm consistently found shorter paths than both the greedy and source routing approaches. The next section attempts to verify if this represents a practical improvement in routing performance with simulations on a realistic city map.

4.4 Simulation Based Analysis of the new Routing Algorithm

CPSR was implemented and simulated in the Qualnet Network Simulator (v5.1) [1]. GPSR was also implemented and tested in Qualnet to provide a basis for comparison, based on the description of the protocol from [19]. GPSR was chosen as a basis of comparison since it is extremely well documented and provides a valuable baseline performance indicator as most recently-published VANET routing protocols have been tested against it. The detailed implementation of GPSR is further described in Appendix A. Vehicular traces were generated using the SUMO traffic generator [28] on a section of a map of Manhattan extracted from OpenStreetMap [90]. Each set of simulation parameters is run with 20 unique seeds to produce a different set of both constant-bit-rate (CBR) flows and vehicular traffic for each parameter set. At the beginning of the simulation, 10% of nodes select a random node and transmit a stream of CBR packets to this node for a period of 10 s; a new set is selected every 10 s for the duration of the simulation, ensuring the participation of most nodes in the simulation in either transmission or reception at some stage of the simulation. Simulation



(a) 100 m

(b) 500 m

(c) 1000 m

Figure 4.14: Average hop count vs. grid size for different LOS ranges

Parameter	**Value**
Channel Frequency	2.4 Ghz
PHY/MAC	802.11b
R_{max}	560 m
Map Size	1.6 x 1.6 km
Node Count	[100, 200, 300, 400, 500]
Application (CBR)	64 bytes, [1, 3] Pkts/s
Transmitters	10% of all Nodes
GPSR/CPSR Beacon Interval	U[250, 750] ms
Simulation Time	300 s

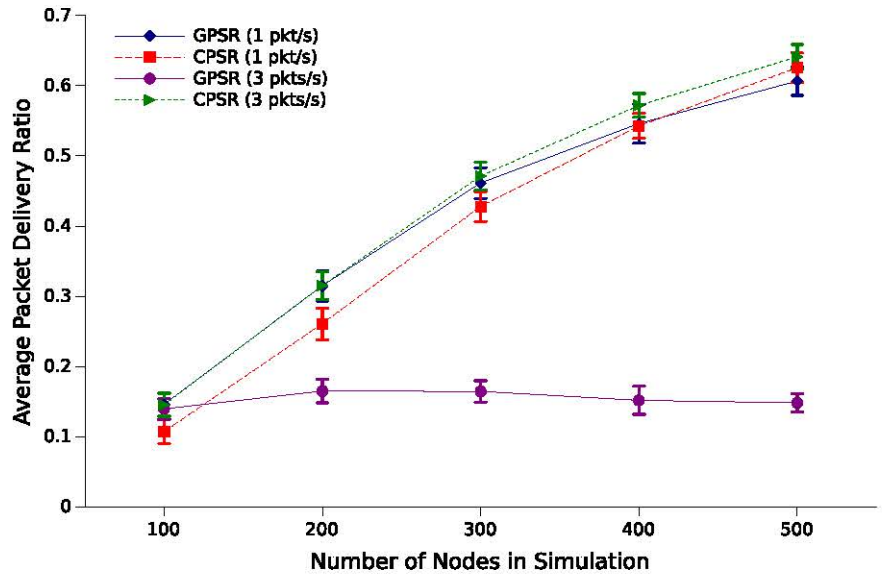
Table 4.2: Simulation Parameters

parameters are summarised in table 4.2. Results are presented with error bars that represent a 95% confidence interval.

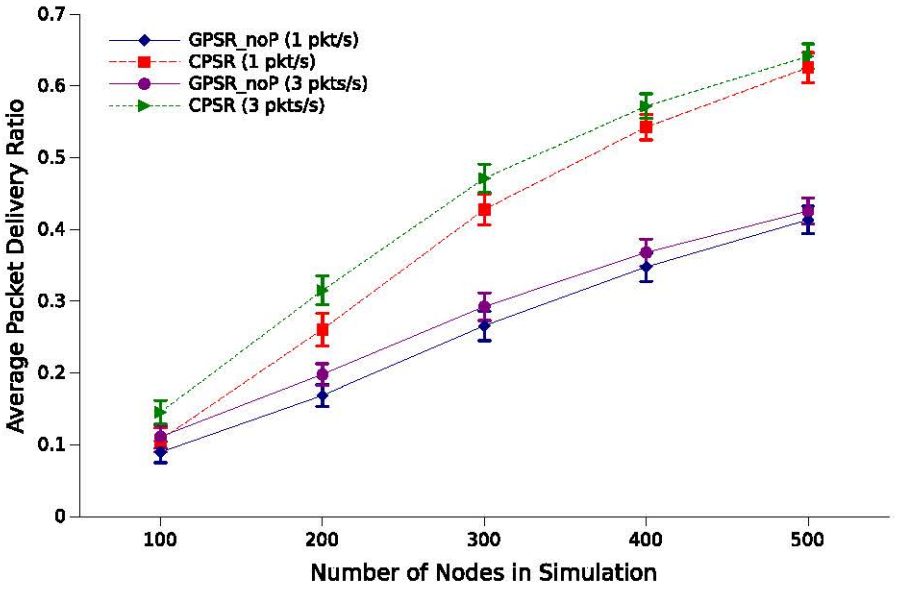
The parameter R_{max} represents the maximum transmission distance in LOS scenarios, which is characterised by free-space path-loss. This distance is used by the simulator to calculate the transmission power, reception threshold and system gains and losses respectively and can thereby be used to work out respective path-losses in both NLOS1 and NLOS2 scenarios. This parameter is set based on the observations in section 3.4.

CPSR was evaluated with the CORNER++ propagation model, with both Rayleigh (NLOS) and Rician fading (LOS) enabled as described in section 3.5. The protocol is compared to GPSR, both with and without the perimeter mode route recovery strategy enabled.

Figure 4.15 compares the packet delivery ratio of CPSR to GPSR for different rates of packet transmission. As seen in the figures, CPSR delivers roughly the same number of packets as GPSR at a low packet rate, but delivers up to 300%



(a) Perimeter mode enabled



(b) Perimeter mode disabled

Figure 4.15: Packet delivery ratio of CPSR compared to GPSR

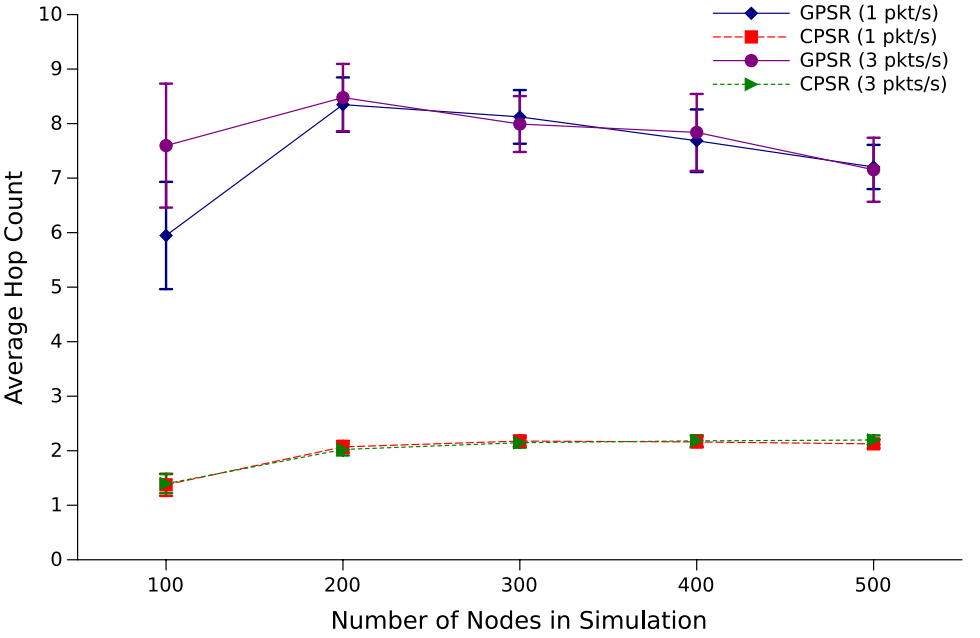
more packets when the packet rate is tripled. This is due to GPSR's fall-back strategy, perimeter mode. Perimeter mode provides a significant improvement in PDR when network traffic load is low. However, as network traffic levels increase, perimeter mode becomes detrimental to PDR. This is due to the planarisation algorithm used by GPSR, which forces a packet to go through several 1-hop links when traversing a street. This is also evident in Figure 4.16a, in which GPSR exhibits a significantly higher hop count than CPSR when run with perimeter mode enabled. When run without perimeter mode, CPSR delivers up to 87% more packets than GPSR at both low and high packet transmission rates.

Figure 4.16 compares the hop counts for successfully delivered packets, while Figure 4.17 shows the average source-destination delay. GPSR (without perimeter mode) shows a lower average hop-count than CPSR for most node-densities. This is due to CPSR delivering packets that GPSR would normally drop due to the lack of a suitable neighbour. Similarly, GPSR shows a marginally lower end-to-end delay. Hop-count and delay are only calculated for successfully delivered packets. By virtue of delivering a significantly larger number of packets than GPSR, CPSR exhibits a slightly larger value for both these metrics. It is interesting to note also that CPSR delivers up to 87% more packets for a mere 10% increase in delay. This increase, as with the hop-count, is due to the delay only being calculated for successfully delivered packets.

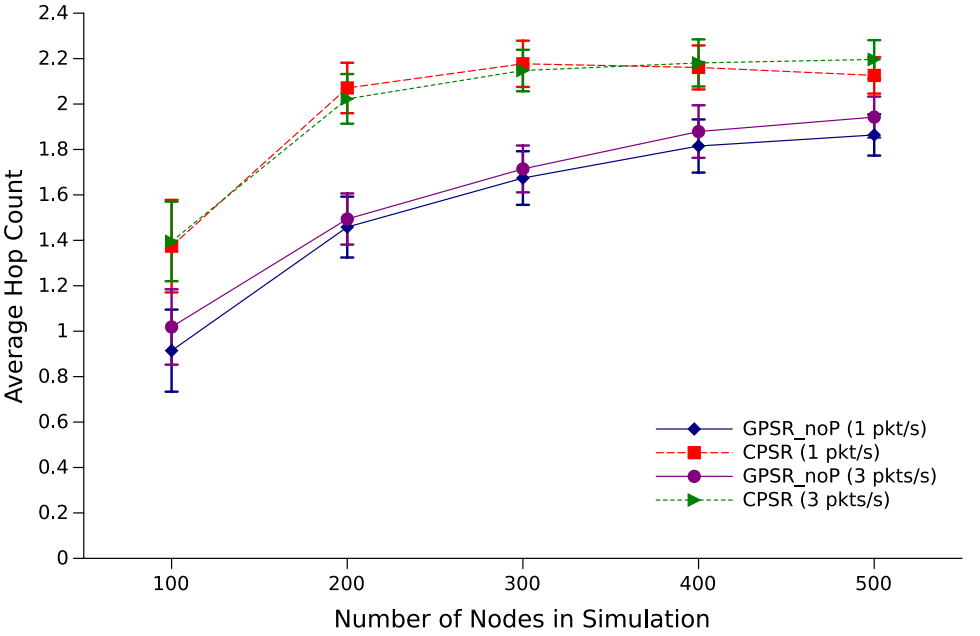
4.5 Conclusions

In this Chapter, a new routing protocol, CPSR, was proposed which implements a greedy algorithm using CORNER++'s propagation estimates instead of simple Cartesian distance. The algorithm's ability to find paths in a grid-line scenario is examined and found to be significantly superior to both GPSR's Cartesian distance-based greedy algorithm and GSR's anchor-based source routing algorithm.

CPSR was then tested against GPSR without perimeter mode and was found to

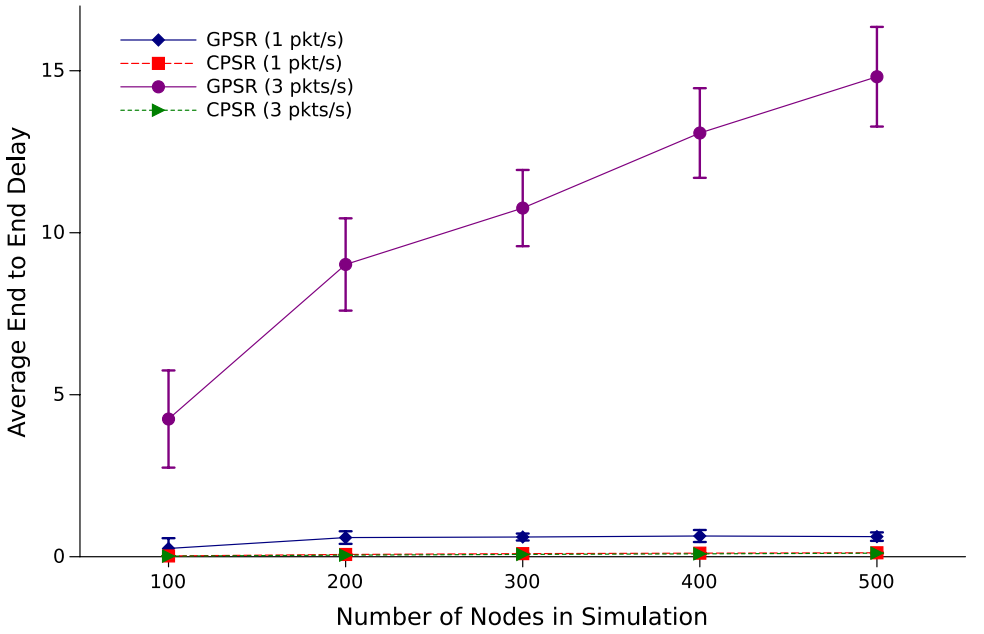


(a) Perimeter mode enabled

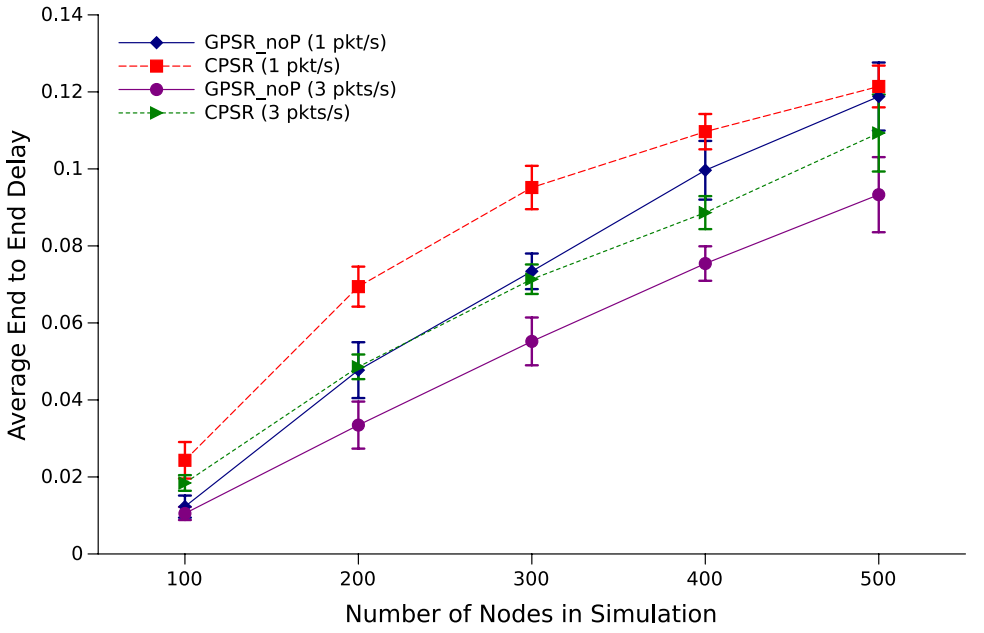


(b) Perimeter mode disabled

Figure 4.16: Average Hop counts for CPSR compared to GPSR



(a) Average Delay with Perimeter mode enabled



(b) Average Delay with Perimeter mode disabled

Figure 4.17: Average Source-Destination delay for CPSR compared to GPSR

produce a significant improvement in packet delivery ratio with minimal increases in delay and hop-count. With perimeter mode enabled, GPSR was found to perform on par with CPSR in terms of packet delivery ratio at low network traffic densities, albeit with increased delay and a significantly increased hop-count. At higher network traffic densities, CPSR outperforms GPSR significantly in all metrics.

While CPSR uses a distance-based greedy routing algorithm as a fall-back when the CORNER-based algorithm fails, there will still be a number of cases where a packet is not forwarded despite the existence of a valid route. It is therefore important to either adapt a fall-back mechanism from the literature or create a suitable fall-back algorithm for CPSR to cover those cases. This is explored in the next Chapter, where a short summary of existing fall-back approaches is presented and a new fall-back mechanism is proposed for CPSR.

Chapter 5

A SEMI-STATEFUL FALL-BACK MECHANISM FOR GREEDY VANET ROUTING ALGORITHMS

In the previous Chapter, a new stateless greedy algorithm, CPSR, was proposed for urban environments. This algorithm was implemented and tested in a network simulator and found to outperform GPSR by a significant margin. However, as with all greedy algorithms, there exists a failure case, which occurs when a node is unable to find a more suitable neighbour than itself to which to forward the packet, reaching a 'local maximum'. While stateless greedy routing is ideal in situations where there is a uniform distribution of nodes from source to destination, it is unlikely that this is always the case in practice. Nodes often cluster towards the ends of streets due to the presence of traffic lights, and a packet carrier may well encounter a situation where a network void makes the carrier a better greedy choice than any of its neighbours. Therefore, to handle situations where greedy routing is unable to provide a next hop, a fall-back algorithm must be developed.

The following section analyses existing approaches in the literature and studies their advantages and disadvantages. Specifically, the disadvantages of a stateless fall-back mechanism is studied and a new semi-stateful approach is presented. The protocol resulting from this addition is called the CORNER Propagation-based Routing (CPR) protocol.

5.1 Analysis of Existing Fall-Back Mechanisms

Most stateless protocols in the literature account for routing failures in their primary routing mechanism. Almost all routing failures in greedy and stateless

protocols stem from the presence of a 'void', which makes it impossible for a node to find a neighbour that is a more optimal forwarding choice than itself using a greedy algorithm. As previously noted, 20 of the 23 protocols discussed in Chapter 2 are stateless.

Anchor-based protocols such as GSR [36] and SAR [91] rely on a precomputed 'source route', i.e. a route defined by street intersections. This route is often computed using a path-finding algorithm such as Dijkstra's algorithm [37]. When a node on this route encounters a void, it simply recomputes the route, marking the encountered street as empty. While this would automatically find the shortest street-based path to the destination, there is no guarantee that it will be connected (i.e. that there actually exist nodes on that street which can relay the packet).

To mitigate this issue, a significant number of protocols attempt to utilise knowledge of the actual network connectivity of specific paths. For example, both A-STAR [42] and VADD [44] rely on a statistical approach, predicting the connectivity of potential paths based on factors such as the historical traffic flow. While not foolproof, this approach is an improvement over completely blind routing. When they encounter voids, A-STAR relies on route recomputation while VADD relies on packet buffering. Buffering is also used by protocols such as GyTAR [43], A-CAR [50] and BAHG [55].

While buffering is a useful means of ensuring that a packet is not immediately dropped for lack of a viable path, it can result in significant delays. The transmission time of a packet through a network is often of the order of tens to hundreds of milliseconds. On the other hand, a vehicle may often take several seconds or even minutes to come in contact with a different neighbour. While buffering is useful in delay-and disruption tolerant networks, it is unsuitable for general-purpose vehicular communication networks in urban environments.

In order to improve knowledge of connectivity even further in stateless networks, protocols such as RBVT-P [52] and BAHG account for traffic flow through proac-

tive approaches which involve dissemination of information throughout a network. For example, RBVT-P constantly transmits connectivity packets around a network, while BAHG creates several classes of nodes which communicate with each other to monitor traffic and aid in routing. Both of these approaches increase protocol overhead significantly, especially since they are proactive approaches and are executed regardless of whether a packet needs to be transmitted.

Stateful routing protocols such as CAR [49] and RBVT-R [52] rely on finding a fully connected path between the source and the destination using a reactive approach. They use a controlled flooding strategy similar to that used in AODV, which allows them to search the network for viable end-to-end paths. However, this implies a significant protocol overhead. Both CAR and RBVT use route discovery and maintenance to trace a path from the source to the destination. This necessitates a series of measures, such as sending route-error packets or maintaining 'guards' which increase beaconing overhead, due to the dynamic nature of VANETs. Both protocols can still suffer from route failures despite these measures, and rely on route recomputation to recover when this occurs.

Of the stateless protocols surveyed in the literature, only one - GRANT [34], uses a stateful *fall-back approach* along with its primary stateless 2-hop greedy routing protocol. GRANT adapts AODV's distance-vector route-finding algorithm to find a path to the destination when greedy routing encounters a local void. The original paper only evaluates the ability of the protocol to determine a route to the destination, and did not show simulation results for mobility. While the route-finding results are positive, it is difficult to assess the performance of the protocol since mobility is not simulated.

GPSR [19], GPCR [31] and GPSRJ+ [39] all use 'face-routing', which is a method of statelessly traversing a network and always finding a path to the destination should one exist. The next section discusses stateless fall-back mechanisms, focusing specifically on face-routing.

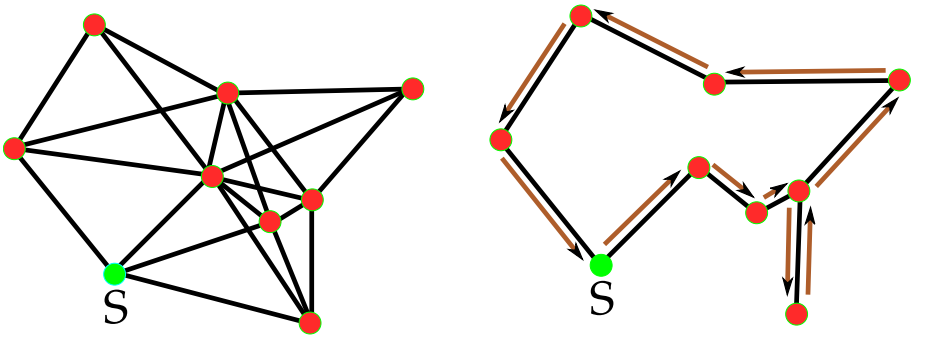


Figure 5.1: Unplanarised (left) and RNG planarised (right) networks

5.1.1 Face Routing with Planarisation

Face Routing [35] is a loop-free method of traversing a graph and finding a path to the destination if one exists. Face routing guarantees that any path to the destination will be found if it exists. In order to avoid loops, the network graph must first be planarised. This involves eliminating all cross-links and is usually performed with either the Relative Neighbourhood Graph (RNG) algorithm or the Gabriel Graph algorithm, both of which are detailed in [19]. An example of a planarised graph is illustrated in Figure 5.1.

Once a graph is planarised, the well known right-hand rule may be used to traverse all the edges in the graph one at a time. This involves always choosing the node counter-clockwise from the vector of entry. In Figure 5.1, the traversal of the planarised graph is indicated as a series of arrows, starting from node S.

GPSR's perimeter mode is essentially an implementation of face routing with the relative neighbourhood graph planarisation algorithm. The perimeter mode algorithm is designed to traverse a void which prevents the greedy discovery of a neighbour. However, in VANETs, the presence of a large number of vehicles on an already largely planar street topology means that perimeter mode may lead to a very hop-intensive path. This is illustrated in Figure 5.2.

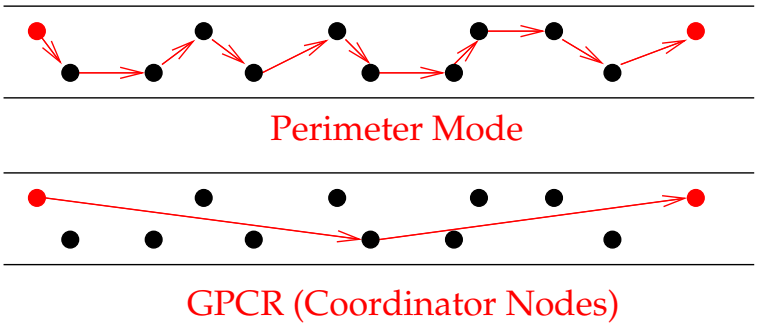


Figure 5.2: Packet transmission with and without network planarisation

5.1.2 Eliminating Planarisation in Cities

GPCR overcomes this problem with the use of 'coordinator nodes', which are nodes at intersections which make routing decisions using the right hand rule, leaving nodes between intersections to make greedy forwarding decisions to an intersection [31]. This essentially represents the street as the edges of a planar graph with the nodes of the graph represented by the coordinator nodes. This approach is also illustrated in Figure 5.2, with the coordinator nodes highlighted in red.

While this approach alleviates the hop-intensive approach of perimeter mode, it still suffers from the use of blind decisions, due to the stateless nature of the algorithm. For example, consider Figure 5.3. Due to the low density of nodes in this scenario, GPSR and GPCR follow an identical path to the destination. The shortest path to the destination is through node A (illustrated as the succession of red nodes). However, the entry condition of GPSR and GPCR dictates that packets be forwarded to the node counter-clockwise from the line SD. The succession of forwarding decisions is illustrated in the diagram as a series of greedy (blue) or perimeter (red) mode decisions.

The biased initial decision of the right-hand rule leads to a much longer and more repetitive path being taken to the destination. While this is an extreme case, it illustrates the issues with a stateless approach to packet forwarding. While the

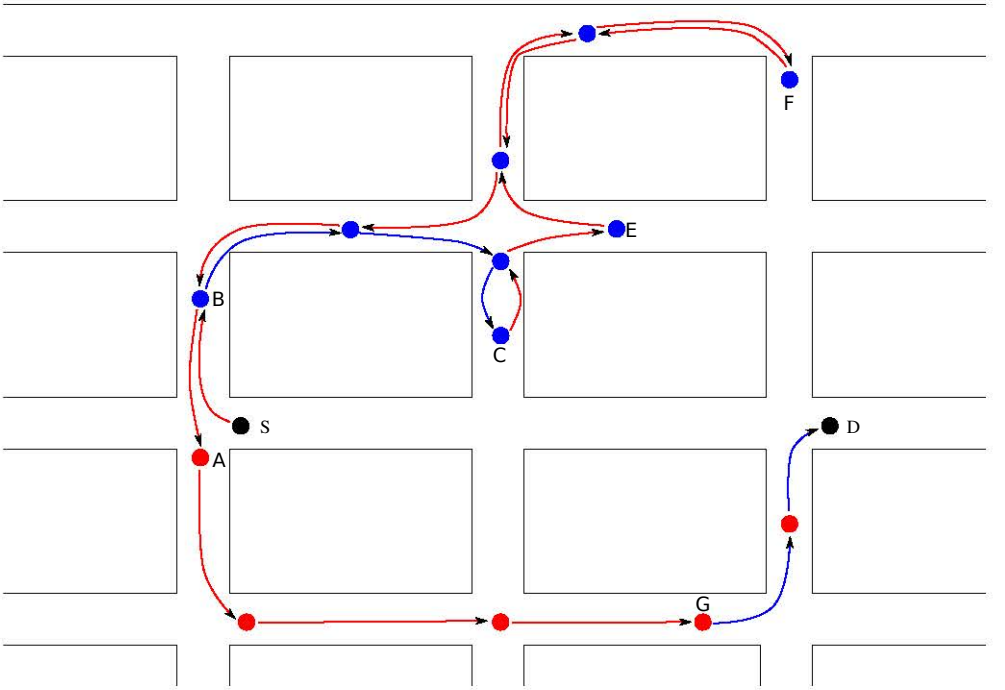


Figure 5.3: Right-Hand Rule based fallback algorithm

packet is forwarded greedily at node B after the first perimeter mode decision is taken, it reaches a dead-end at node C, where it is switched back to perimeter mode. It then follows the right-hand rule, reaching dead ends at node E and F, before doubling back and finding a path via node A, not reverting to greedy mode until it reaches node G, which is closer to the destination than node C.

Other than the exceedingly long paths that are likely to be taken as a result of a stateless recovery mechanism, an increase in hop counts also leads to a corresponding increase in the likelihood of failed delivery due to packet collisions or changes in network topology. While GPCR doesn't rely on network planarisation, it still uses the right-hand rule, thus causing the same route to be taken.

Therefore, it is clear that the use of a stateless fallback mechanism when the primary routing mechanism fails has significant disadvantages. Exploring any path in a completely stateless manner provides no guarantee that it will lead to the destination or that it will be an optimal route to the destination.

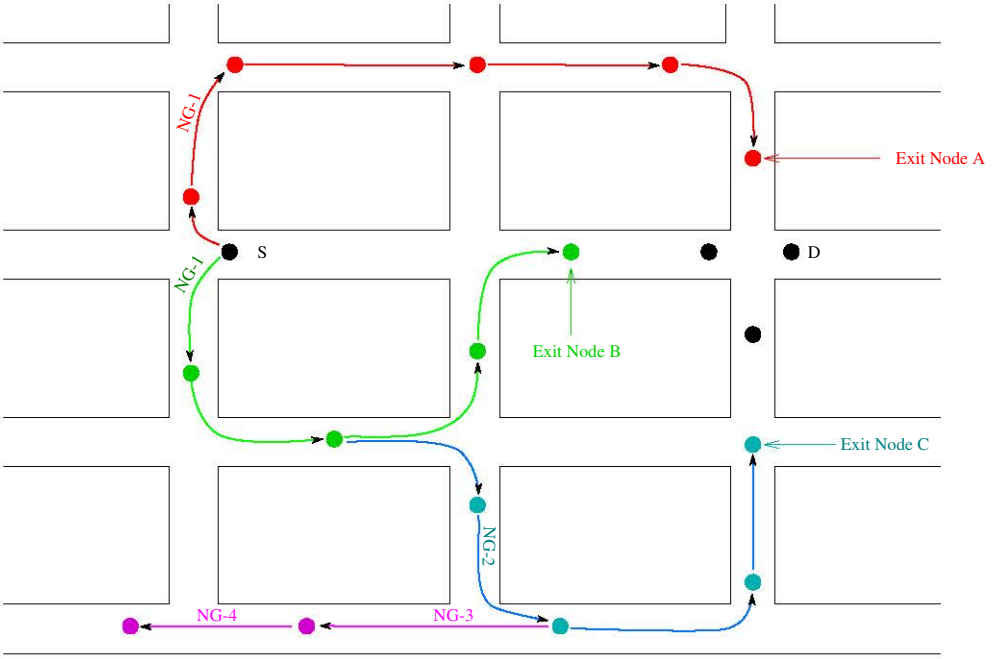


Figure 5.4: Fall-back mechanism used in CPR

5.2 A Semi-Stateful Fallback Approach

The approach proposed in GRANT, while not thoroughly tested, is interesting since it only relies on statefully routing around the void and not finding a route all the way to the destination. It also has the advantage of exploring all possible routes around the void, although it is not in any way optimised to take street topology into account.

The proposed approach adopted in this Thesis is similar in that it uses a reactive route finding algorithm to explore multiple possible ways to traverse the void. When a packet encounters a void, a route-request (RREQ) packet is broadcast by the node holding the packet - known as the entry node - which also notes its path-loss to the destination. The RREQ packet is subsequently re-broadcast by the surrounding nodes until one encounters a neighbour that has a smaller path-loss than the entry node. This node, known as the exit node, then replies with a route-reply (RREP) to the entry node. If multiple paths are found, the

path with the smallest hop-count is selected.

The approach is illustrated in Figure 5.4, which shows two possible paths (red nodes and green nodes) to the destination. Neither of these paths is an automatic selection when greedy forwarding is used, since the first node of each path exhibits greater path-loss than the entry node S. The route selection algorithm traverses both paths until reaching either exit node A or B, both of which have a smaller path-loss to the destination than the source node S. At this point, greedy forwarding is resumed, with the packet having been routed around the void.

As in AODV, sequence numbers are used both to prevent a node from re-broadcasting the same RREQ packet and as a rudimentary time-stamp for RREQ and RREP packets.

The recovery strategy focuses only on getting a packet across an existing void, and *not* all the way to the destination. This minimises the risk of route errors by minimising the length of the path to be maintained. In order to limit the spread of a route request down a path that repeatedly leads it away from the destination, each route request is restricted to a certain number of 'non-greedy forwarding decisions'. Each node that re-broadcasts a route request increments the non-greedy forwarding counter (NGFC) if it only sees neighbours that are less optimal greedy choices than itself. Once the limit is reached, any nodes that receive the route request will cease propagating it further. This is illustrated in Figure 5.4, with a fixed limit of 4, in the paths labelled NG-1 to NG-4.

The limit is computed by each node that rebroadcasts an RREQ packet based on the number of neighbours it has, since the higher the node count, the better the probability of finding a route. The number is represented as a 4-bit unsigned integer in the broadcast packet header, giving it a range of 0-15. It is computed each time a packet is forwarded and is based on the number of neighbours seen by a node, being inversely proportional to the number of nodes seen. If a node sees 15 or more nodes, the limit is set to 1 and if it sees only 1 node, it is set to

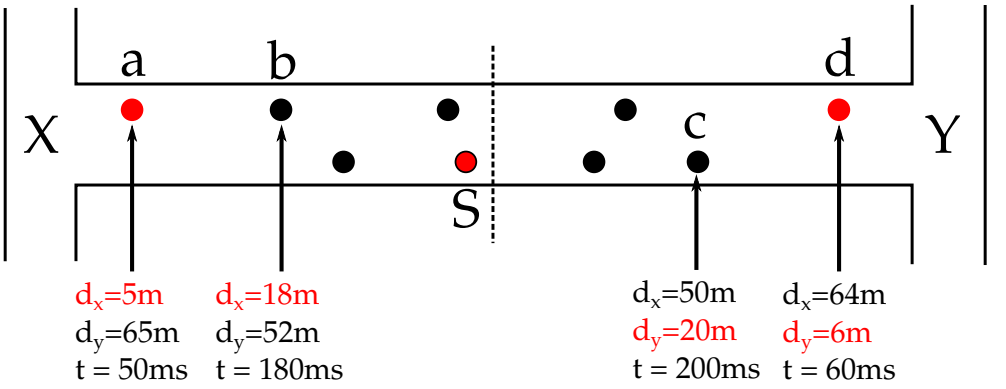


Figure 5.5: Timer-based RREQ broadcast restriction

15.

5.2.1 Utilising Street Knowledge to Further Restrict Flooding

The knowledge of street topology can be used to further reduce the broadcast of RREQ packets. On any particular street segment, there exist a maximum of two intersections. In order for a route request to reach a useful exit node, it needs to exit the street that it is on. This principle applies to every street that the route request propagates on. Therefore, if the re-broadcast of RREQ packets is restricted to the nodes furthest from the source node and also closest to the street junctions, the route request would still have the same chance of moving to another street as before, but with significantly reduced network load.

In order to restrict RREQ re-broadcasts as outlined above, a timer-based approach is employed. While the RREQ may be multicast to nodes selected by the source node, it is susceptible to failure due to outdated neighbour information.

Consider Figure 5.5. When a node hears a route-request, it calculates its position on the street with respect to the nearest intersection. This is illustrated in the Figure, where four nodes are located on a segment bound by intersections **X** and **Y** respectively. Each node calculates its distance from both intersections and selects the shortest distance, noting which of this is nearest. A timer is then set

based on this distance, as illustrated in Figure 5.5. The propagation time of the signal and the processing time at each node are also considered and added to the timer.

When the node's timer counts down to zero, it re-broadcasts the RREQ packet, with its own coordinates attached to the packet header. Any node which hears this RREQ packet calculates the closest intersection to the node which broadcast the packet. In Figure 5.5, node a re-broadcasts the packet first. The packet is subsequently heard by nodes b, c and d. Node b then drops the packet since it is closer to intersection **X** than to **Y**, but a is even closer. Nodes c and d overhear the packet but do not drop it, since they are closer to intersection **Y**. Node d broadcasts the packet next, and node c also backs off upon overhearing this packet. Therefore, the broadcast is only repeated by the nodes closest to the intersection.

The same principle applies each time the route request is re-broadcast. Figure 5.6 illustrates the same method on nodes located both on the same street and on streets in NLOS conditions with respect to the source node's street. Only the nodes highlighted in red carry the request forward in this example.

5.2.2 Handling Errors

Routing errors may occur when a node which is part of an established route drops out of range of its neighbours in the route. A node periodically checks its own routing table to ensure that the 'next-hop' nodes are still in range. The node obtains this information from its neighbour table, which is updated by periodic beacons. Additionally, RERR packets are sent when a packet delivery to a 'next-hop' node fails, indicating that the node has moved away. The route-error packet is once again similar to that sent by AODV, carrying information about an unreachable destination node. The packet is broadcast by the node which detects the broken connection. It is subsequently re-broadcast by nodes which maintain a path to the destination through the node from which they

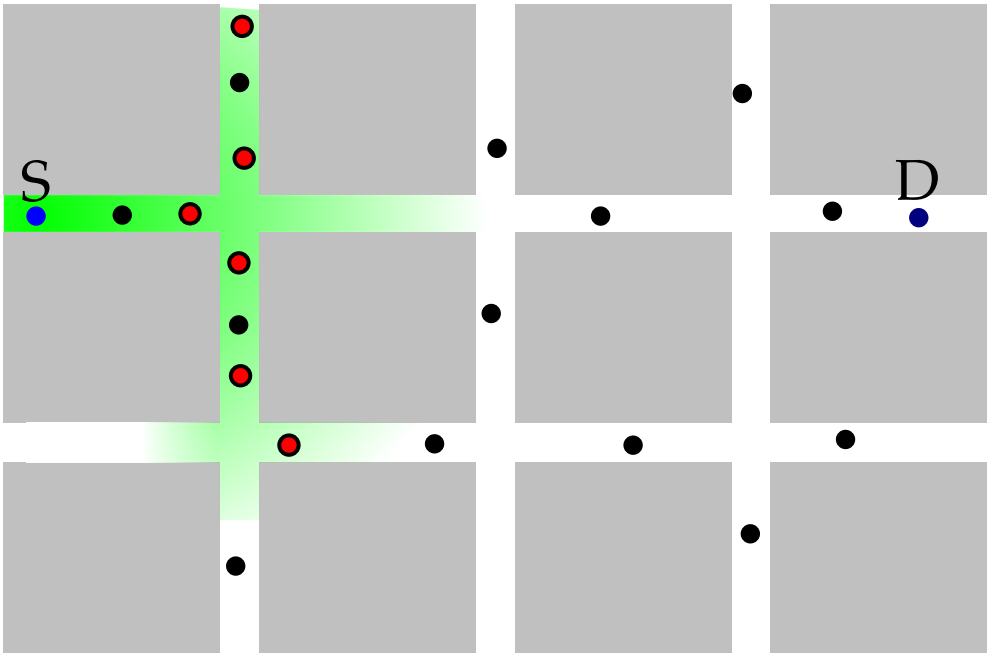


Figure 5.6: Route re-broadcasting on multiple streets

received the RERR packet.

5.2.3 Protocol Design

The new protocol, named *Corner Propagation Routing* (CPR), is an extension to CPSR, described in Chapter 4. The main changes are to the unicast packet header, which needs to accommodate a fall-back mode and the addition of RREP, RREQ and RERR packets.

CPR Packet Header

The IP packet header for CPR is slightly larger than CPSR due to the need for an identifier to distinguish between greedy and fall-back modes. This is represented as a 1-bit identifier. Due to the requirement that IPv4 headers have a round number of 32-bit words, a 31-bit padding is added to this identifier, which is reserved for future use. This header is illustrated in Figure 5.7. The coordinates

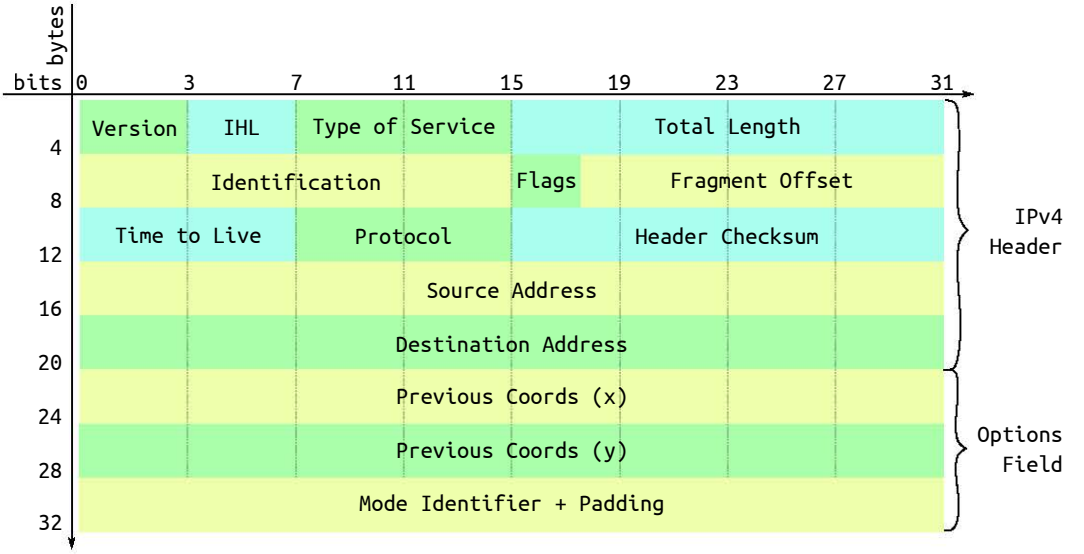


Figure 5.7: CPR IPv4 Header with Mode Identifier

are used, as with CPSR, by nodes which promiscuously overhear unicast packets to advantageously update their neighbourhood tables.

RREQ, RREP and RERR Packets

The structure of RREQ, RREP and RERR packets is illustrated in table 5.1. In order to save packet overhead, none of the packets has a mode identifier. They are first identified using the 'Protocol' field in the IPv4 header and distinguished by reading the 'Total Length' field since each type of packet has unique and static sizes. The padding on RREQ and RREP packets is to make their packet sizes a multiple of 4-bytes as mandated by the IPv4 specification [88].

5.3 Simulation based Study of the Proposed Routing Algorithm

CPR was implemented and simulated in the Qualnet Network Simulator (v5.1). GPSR was also implemented and tested in Qualnet to provide a basis for comparison. The GPSR implementation was written using the algorithms described in [19] and validated against the results from the same paper. AODV was also sim-

RREQ Packet - Size: 40 bytes

Data	Type	Bits
Previous Hop	Coordinates	64
Destination	Coordinates	64
Source IP	IP Address	32
Dest. IP	IP Address	32
Entry Node	IP Address	32
Entry Metric	Float	32
Src. Seq-No.	Integer	32
Hop Count	Integer	12
Non-Opt. Count	Integer	4
Padding		16

RREP Packet - Size: 28 bytes

Data	Type	Bits
Previous Hop	Coordinates	64
Source IP	IP Address	32
Dest. IP	IP Address	32
Dest. Seq-No.	Integer	32
Exit Metric	Float	32
Hop Count	Integer	12
Padding		4

RERR Packet - Size: 12 bytes

Previous Hop	Coordinates	64
Dest. IP	IP Address	32

Table 5.1: RREQ, RREP and RERR Packet Formats in CPR

Figure 5.8: Section of Manhattan used for the simulations

ulated for the same scenarios to provide a baseline comparison to a well-known non-position based routing protocol. Vehicular traces were generated using the SUMO traffic generator and the mobility model as detailed in section 5.3.1. Each set of simulation parameters is run with 20 unique random seeds for the generation of both CBR flows and vehicular traffic. The results are presented with error bars which represent 95% confidence intervals.

5.3.1 Vehicular Mobility Simulation

In order to simulate a realistic set of mobility traces, a dedicated vehicular mobility simulation was created using the SUMO traffic simulator [28]. The simulator uses a section of a real city map exported from the OpenStreetMap project and inserts a pre-determined number of vehicles, with each given an origin and a destination. A vehicle travels from the origin to its destination, obeying traffic rules and interacting with the other vehicles in the scenario. When a vehicle reaches its destination, it either randomly picks another destination or returns to its origin before picking another destination. This simulates a vehicle leaving home, picking a destination and returning home before picking another destination. This pattern is repeated until the end of the simulation. Once simulated in SUMO, the traces are transcribed to a format readable by Qualnet, where the network simulation is carried out.

5.3.2 Simulation Scenarios

In order to study the impact of both node-density and the routing method, simulations were carried out at varying node densities using both GPSR and CPR. Due to the dynamic nature of SUMO, the overall node density of the network is not indicative of the average node density in any specific part of the network. SUMO's route finding algorithm finds the *fastest* path to the destination based on factors such as road width and distance, which often place a large

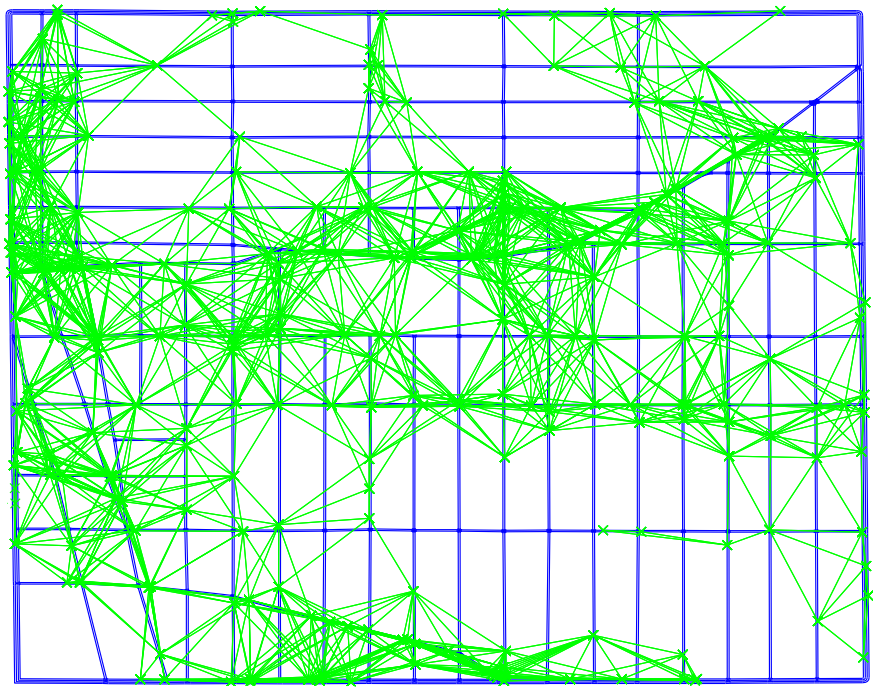


Figure 5.9: Distribution of 250 Nodes along with Two-Ray Connectivity Graph

number of vehicles on a small number of major roads. This therefore leads to a realistic simulation of gridlocks with an increase in overall node density.While the absolute node density is not a useful measure in these experiments, increasing the number of nodes in the network still provides valuable insight into the behaviour of the studied protocols under increasingly stressful conditions due to an increase in traffic and interference.

This is illustrated in Figure 5.9, which highlights a scenario with 250 nodes in the network. The green lines are radio-links between the nodes under the two-ray model and are useful to indicate the clustering of nodes in specific areas. As illustrated, a large proportion of the network is empty, but nodes have clustered around the north-west corner and the central areas of the network, with almost no nodes in the south-east corner.

The simulations are run on a 1.5 km^2 section of Manhattan, shown in Figure 5.8. Between 100 and 500 nodes are included in the simulation, representing networks

ranging from sparsely distributed to extremely high density. In each simulation, 10% of the nodes are initially chosen to transmit packets to another 10% of nodes in the network. Each source-destination pair transmits CBR packets of 64 bytes for 25 seconds, after which another set of nodes are chosen (again, 10% of nodes randomly selected as source nodes for the CBR flows). The start time of each 25 second CBR transmission is varied according to a uniform distribution to prevent synchronisation between multiple transmitters. Packets are transmitted at rates of 1-3 packets per second.

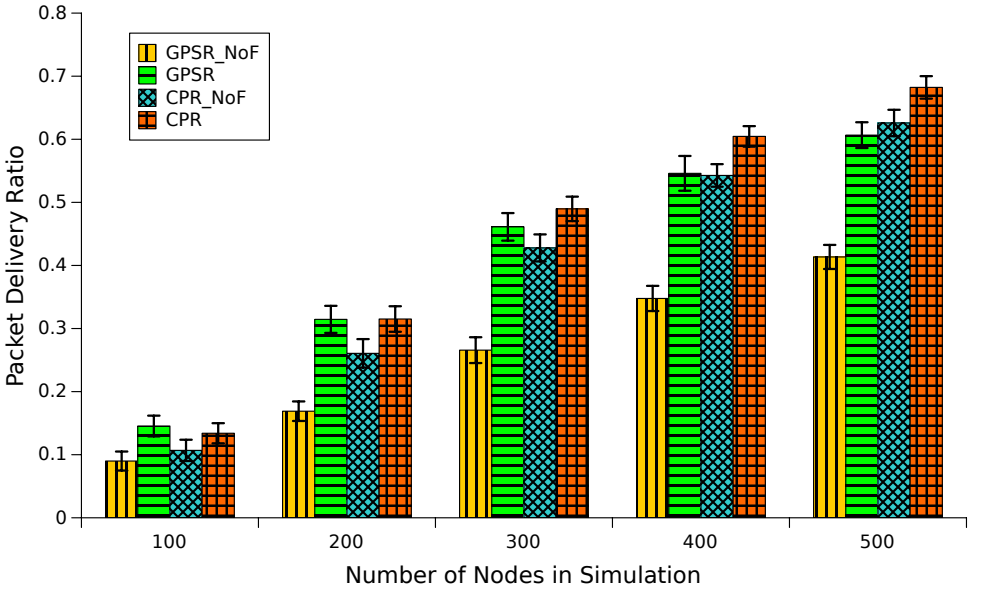
5.3.3 CPR Compared to GPSR

CPR is compared with both GPSR and AODV, since they are typical protocols using the greedy-stateless and stateful routing approaches respectively. Figure 5.10 compares CPR and GPSR at low packet transmission rates (1 packet/s) at node densities ranging from 45-220 nodes/km^2 (represented as 100-500 simulated nodes in the 1.5km^2 grid). Both the packet delivery ratio and the average hop count of successfully delivered packets is shown.

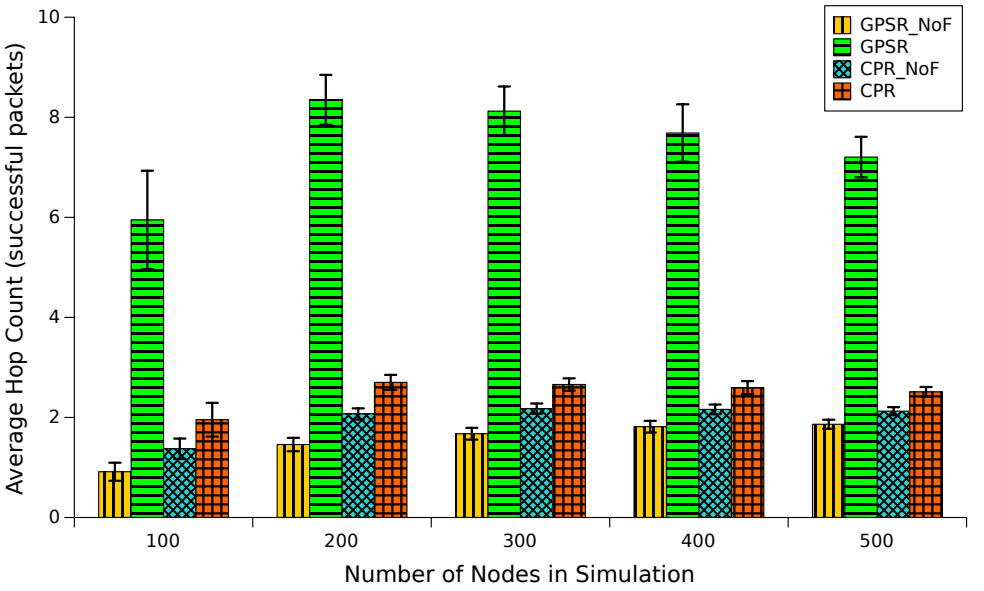
It should be noted that the hop counts are only calculated for successfully delivered packets, and therefore trend higher when more packets are delivered. They are therefore only useful to distinguish differences in orders of magnitude, such as between CPR and GPSR (with perimeter mode). The significant increase in hop-counts when GPSR is run with perimeter mode, as predicted in Section 5.1, is due to the use of graph planarisation. While this does not impact the network when packet density is low (1 pkt/s), it results in a significant performance decrease when the packet transmission rate is tripled.

Effect of Packet Rate

When the packet rate is tripled, the results change significantly as shown in Figure 5.11. The stateful fallback mechanism of CPR still provides an improvement



(a) Packet delivery ratios



(b) Average hop count

Figure 5.10: CPR compared to GPSR at 1 packet/s

at low to medium node densities, but starts to fail at higher node densities. This is due to the broadcast storm created by multiple RREQ packets, resulting in a large number of packet collisions. This also leads to a large number of packet retransmissions, further compounding the problem and showing up as an increased hop-count.

5.3.4 CPR compared to AODV

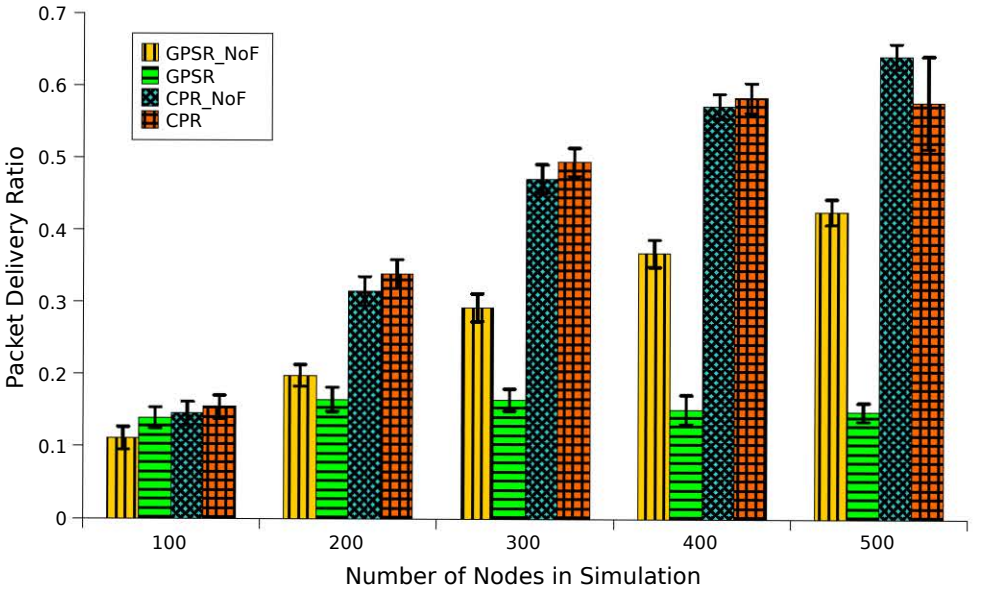
This discrepancy is further evident when CPR is compared to AODV in Figure 5.12, which uses a completely stateful approach. Both CPR and AODV show a drop in packet delivery ratio when the number of nodes is increased from 400 to 500 at 3 packets/s. AODV registers a minor drop even at 1 packet/s, illustrating the disadvantage of a fully stateful approach at extremely high node densities.

5.4 Analysis and Conclusions

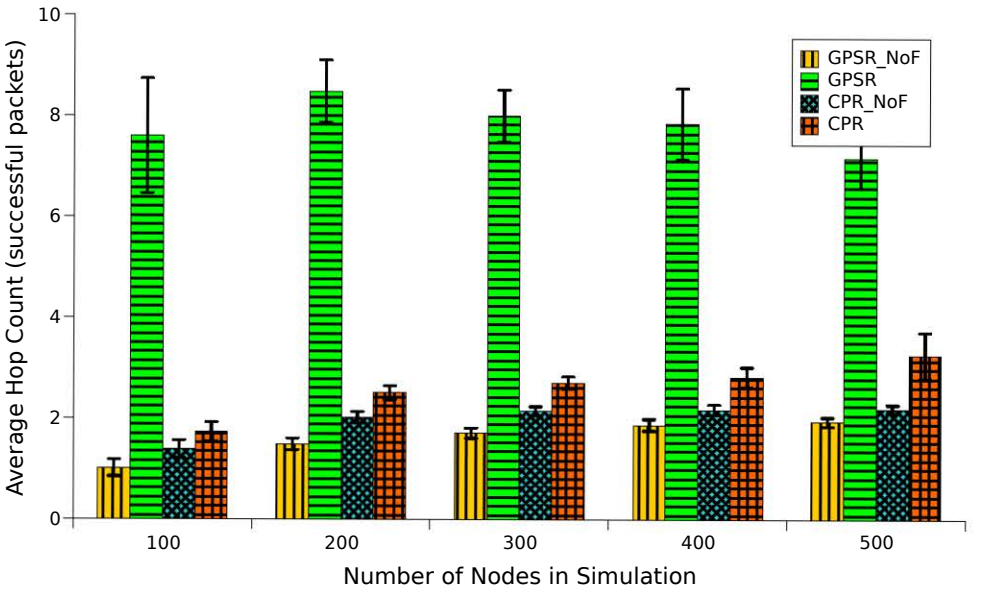
Packet failures often occur at the MAC layer when a forwarding decision by the routing protocol is invalid. This can happen quite often in mobile networks, as a decision based on a slightly outdated neighbour table might result in a selection of a neighbour that is no longer within communications range of the sender. This is an even more severe problem in networks using routing protocols based on greedy forwarding, since the selected neighbours are often located at the very edge of a node's communications range, as they are often the greediest choices.

CPR usually intercepts packets that have exceeded the MAC-layer retransmission limit and reroutes them via a different neighbour to the destination. This is only performed for those packets that are intercepted in greedy mode, since the fallback mode packet would require a further route request and recovery process from the intermediate node. At present, CPR simply drops packets intercepted in fallback mode and sends an RERR packet.

Figure 5.13 illustrates the number of packets that are dropped at the MAC layer.



(a) Packet delivery ratios



(b) Average hop count

Figure 5.11: CPR compared to GPSR at 3 packets/s

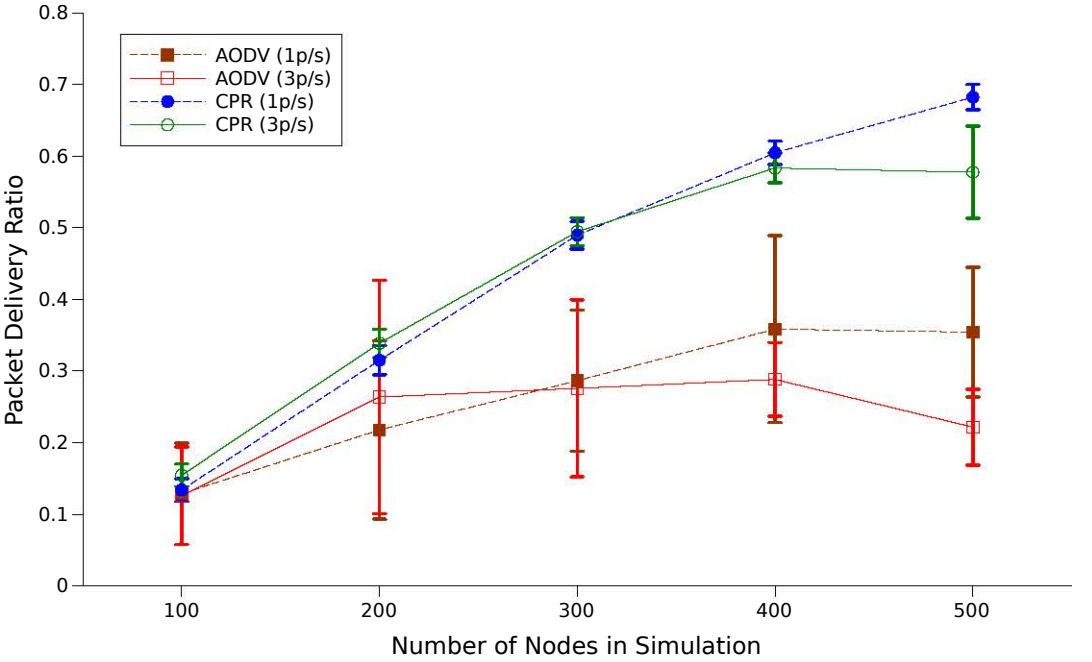


Figure 5.12: CPR compared to AODV under different packet rates

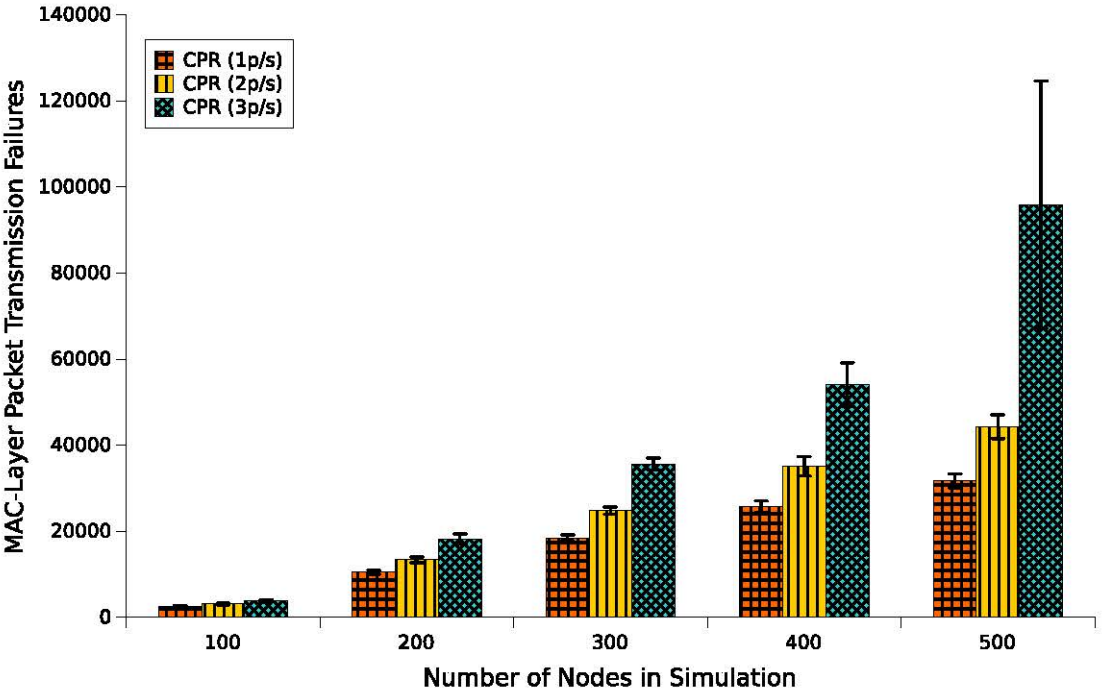


Figure 5.13: Number of MAC-layer dropped packets with CPR

As seen in the Figure, this number increases significantly at higher packet densities, due to an increased rate of packet failures from a combination of both collisions and neighbour unavailability. This results in even more packet retransmissions, which results in even more traffic, further compounding the collision problem. It also illustrates the main reason for the failure of the fall-back mechanism of the protocol at higher node and packet densities; this is something that should be addressed in future versions of the protocol.

A potential solution to this problem lies in reducing the dependence on individual nodes in path maintenance. In the current implementation of the protocol, route maintenance is carried out by each node along a connected path. If one of these nodes were to be unavailable temporarily due to transient events such as a packet collision or fading, the route would collapse. This is seen above in the significantly increased MAC-layer retransmission counts at higher packet densities.

A potential method of alleviating this issue would be to use street intersections as anchor points, as proposed in protocols such as A-STAR. The initial RREQ/RREP phase may be used as a means of selecting and establishing the best connected path, which is then represented as a series of street intersections. Nodes which hear an RREP packet can trace its last known street and select the corresponding intersection as an anchor point. Nodes closest to the intersection can maintain alternative paths. A failure due to lack of connectivity on any street would only require redirection of the packet back to the most recent intersection instead of rebroadcasting an RREQ packet. Additionally, the route is less sensitive to the loss of a particular node and only to the complete absence of vehicles on a street, which is unlikely at higher node densities. This solution should significantly reduce the issue of MAC-layer transmissions, but the implementation of such a solution is left to future work.

5.5 Chapter Summary

This Chapter presented a stateful fallback mechanism for the stateless and greedy routing approach proposed in Chapter 4. The fall-back mechanism was found to improve packet delivery rates at the cost of a minor increase in hop-counts. The CPR protocol was also found to outperform both AODV and GPSR across a variety of node and packet densities. The fallback mechanism starts to fail at higher packet and node densities and this was found to be due to the exponential increase in packets resulting from MAC layer retransmissions, leading to an increase in packet collisions. This makes the maintenance of a routing path anchored by specific nodes difficult, since each error leads to an immediate route recovery phase, which is often unnecessary.

A potential solution to this problem is also suggested and involves the use of street based anchor points instead of mobile nodes for route maintenance. Future work should therefore focus on implementing and testing this solution, and improving the robustness of the fallback mechanism.

Chapter 6

CONCLUSIONS AND FUTURE AVENUES FOR RESEARCH

This Thesis demonstrated the significant impact of the signal propagation model on the performance of wireless VANET routing protocols. An existing time-invariant signal propagation model for VANETs known as CORNER was discussed in detail, and a number of enhancements were proposed to both improve the signal propagation model and model time-varying fluctuations in the channel. The enhanced CORNER, known as CORNER++, was shown to be in good agreement with a number of experimentally-measured VANET channels. A new VANET routing protocol, CSPR, is introduced which utilises knowledge of the underlying physical channel (based on CORNER++) as its primary routing metric. The final evolution of this protocol, CPR, also includes a robust semi-stateful fallback mechanism in the case that the primary routing strategy fails, which avoids some of the shortcomings of simple stateless fallback strategies.

6.1 Research Outcomes

6.1.1 Review of VANET Routing Protocols

A comprehensive review of position-based routing protocols was undertaken to study the existing routing methodologies in VANET routing and determine potential areas for improvement. The review focussed on protocols designed for urban environments; a total of 23 VANET routing protocols were studied. The survey revealed significant similarities in the routing methodologies, with all but one of the protocols using a greedy routing algorithm as a significant part of the

protocol, and all but three relying on a stateless approach. All but six of the protocols further implemented a fallback mechanism when greedy routing was unable to locate a suitable next-hop neighbour.

The effect of the choice of propagation model on the performance of these protocols was also evaluated and found to be extremely significant. Based on the review of protocols and channel models, a number of important areas for potential contribution were identified:

- Experiments to establish the propagation characteristics of a dense urban environment;
- Independent validation of the CORNER propagation model and improvements to the fundamental assumptions to the model, including a new classifier and the addition of a statistical fading model;
- The development of a new greedy routing method for urban environments which uses predicted signal propagation characteristics as the primary routing metric; and
- The development of an improved fall-back mechanism to handle failures in greedy routing protocols, which can be applied to the aforementioned propagation-based routing protocol.

Contributions were made in each of these areas, resulting in significant improvements to the CORNER urban propagation model and a new routing protocol which uses a new greedy routing method designed for urban environments featuring a unique semi-stateful fall-back approach (which can also potentially be applied to other greedy routing protocols).

6.1.2 Contributions in Propagation Modelling

Chapter 3 presented a brief overview of propagation models designed for urban environments or used in the VANET literature. A number of models, such as

free-space path-loss, the two-ray ground reflection model and the probabilistic shadowing model were found to be too simplistic to provide a realistic assessment of VANET routing protocol performance. The Longley-Rice and Edwards-Durkin models were found to be unsuitable for microscopic modelling of path-loss due to their design being focused on providing a median estimate of path-loss. The Okumura-Hata models were designed for base-station to mobile-node communications and are therefore also unsuitable for inter-vehicle communications. Therefore, the main focus was on the CORNER propagation model, which models microscopic interactions between individual vehicles and obstacles by approximating the effects of reflected and diffracted paths.

The original validation method for the CORNER model used packet delivery ratios, calculated by the transmission of unicast packets between two nodes travelling around a city block. This is prone to interference from nearby sources operating at the same frequency. Furthermore, the original model does not consider the possibility of communication between any streets separated by more than two intersections. No justification was provided for this assumption; therefore, it needed to be experimentally tested.

In order to eliminate the effects of interference from nearby sources operating on the same radio channel, the experiment was carried out using signal-strength measurements instead. The results largely validated the CORNER model, while also demonstrating the Rician nature of the vehicular fading channel. Measurements obtained in a largely straight street environment also revealed the propagation of signals well beyond the arbitrary two-intersection limit defined by the CORNER classifier.

A new classifier was therefore proposed to overcome the limitations of the original CORNER classifier. The model was also improved with the addition of a Rician fading channel in line-of-sight scenarios and a Rayleigh fading channel in non-line-of-sight scenarios to form the CORNER++ model. The statistical behaviour of CORNER++ was confirmed to closely match experimental observations on long

line-of-sight paths beyond 2 intersections. The performance of a variety of routing protocols was evaluated using the new model, demonstrating the significant effect of the propagation model choice on various routing protocol performance metrics.

6.1.3 Contributions in VANET Routing

Chapters 4 and 5 present a new greedy routing algorithm and fall-back mechanism respectively. The greedy algorithm is targeted at urban environments and uses CORNER's path-loss estimates as a metric for greedy decisions. The algorithm was tested in an analytical grid scenario and found to consistently identify shorter paths than either a purely greedy approach or an intersection-based routing approach. Simulations also showed a 50% improvement in packet delivery ratios over GPSR without perimeter mode and an almost 500% improvement over GPSR with perimeter mode at higher traffic loads. Furthermore, traffic flows exhibited a significantly lower average hop-count and delay than GPSR.

The protocol was still susceptible to failure due to the presence of network voids, so a fall-back strategy was proposed to overcome these situations. Existing fall-back strategies were studied and found to either suffer from significantly increased average hop-count and path-lengths due to statelessness or from significantly increased delay due to buffering. Therefore, a semi-stateful fall-back strategy was proposed which attempts to find routes around the void using route request and reply packets. The strategy was found to improve packet delivery ratio across a variety of node densities at lower packet transmission rates, although it resulted in a significant drop in PDR at high packet transmission rates and high node densities. This was a result of a 'broadcast-storm' from multiple route requests flooding the network. Although this issue is not addressed in this dissertation, a significant opportunity is identified for future improvement.

6.2 Future Research Directions

There is significant potential for improvements in both major contributions of this thesis.

The Rician fading channel, which was found to be present in practical experiments in vehicular environments, requires further study. The experiments revealed a varying Rician K-Factor in the results, which needs to be studied and modelled appropriately. Work is already underway in this field, with [13] and [14] presenting a ray-launcher based method of K-Factor approximation. The method shows significant correlation with the observed results in the first 100 metres of signal propagation. Further refinements are possible to this technique to both improve the approximation and extend the range of the model beyond 100 metres.

The proposed greedy routing algorithm works well in dense urban environments due to its ability to approximate realistic signal propagation behaviour. A different environment may conform to a different propagation model, and therefore a means of correctly identifying the appropriate model to use for a given scenario would be valuable in creating a universal solution. For example, an algorithm that switches between flat, rural and urban modes would be useful in VANETs where cars may travel across a variety of areas.

A failure point of the proposed fall-back mechanism was also identified. The mechanism results in a significantly reduced packet-delivery rate at high node densities and high packet transmission rates due to an increased likelihood of collisions from repeated route-request packets. An idea was proposed to significantly reduce the number of route-request retransmissions by minimising the dependence on mobile nodes to maintain an active route. This would be a significant contribution in itself and would be a useful way of mitigating the 'broadcast storm' problem faced by several routing protocols. Implementation and testing of this idea would be a valuable addition to the literature.

The new protocol was also not tested against a number of protocols in the literature due to the difficulty of implementing complex VANET protocols in a limited amount of time. A performance-based review of multiple protocols in a realistic VANET scenario would therefore also be a useful starting point for identifying beneficial design choices when creating the next generation of VANET routing protocols. Additionally, practical implementation of the proposed protocols would allow experimental validation of the performance in a real vehicular environment and potentially identify further areas for improvement.

Finally, the experiments in propagation and the use of the resulting model in routing would benefit from being repeated under the recently standardised IEEE 802.11p PHY/MAC layers. As noted in Chapter 1, the differences in channel bandwidth and frequency at the PHY layer and timings at the MAC layer mean that different results may be expected in the relative performance of different protocols. Additionally, the use of a higher frequency and a narrower channel may result in a different physical connectivity profile in urban environments, which should be considered when developing new routing protocols.

BIBLIOGRAPHY

[1] "Qualnet Network Simulator." [Online]. Available: http://web.scalable-networks.com/content/qualnet

[2] "ISO/IEC 7498-1:1994 - Open Systems Interconnection - Basic Reference Model: The Basic Model," *ISO/IEC JTC 1 Information Technology*, 1994.

[3] R. Braden, "Requirements for Internet Hosts - Communication Layers," *IETF RFC 1122*, Oct. 1989.

[4] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, Aug. 1980.

[5] E. Giordano, R. Frank, G. Pau, and M. Gerla, "CORNER: a realistic urban propagation model for VANET," in *2010 Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE, Feb. 2010, pp. 57–60.

[6] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, Jul. 2010.

[7] "IEEE SA - 802.11b-1999 - IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band," 1999.

[8] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen, "Analytical Study of the IEEE 802.11p MAC Sublayer in Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 873–886, Jun. 2012.

[9] Q. Sun, S. Y. Tan, and K. C. Teh, "Analytical Formulae for Path Loss Prediction in Urban Street Grid Microcellular Environments," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1251– 1258, Jul. 2005.

[10] X. Zhao, T. Rautiainen, K. Kalliola, and P. Vainikainen, "Path-Loss Models

for Urban Microcells at 5.3 GHz," *IEEE Antennas and Wireless Propagation Letters*, vol. 5, no. 1, pp. 152–154, Dec. 2006.

[11] E. Giordano, R. Frank, G. Pau, and M. Gerla, "CORNER: A Radio Propagation Model for VANETs in Urban Scenarios," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1280–1294, Jul. 2011.

[12] A. Mukunthan, C. Cooper, F. Safaei, D. Franklin, M. Abolhasan, and M. Ros, "Experimental validation of the CORNER urban propagation model based on signal power measurements in a vehicular environment," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013, pp. 2626–2631.

[13] C. Cooper, A. Mukunthan, M. Ros, D. Franklin, and M. Abolhasan, "Dynamic Environmental Fading in Urban VANETs," in *IEEE International Conference on Communications (ICC) 2014*, 2014.

[14] C. Cooper, A. Mukunthan, F. Safaei, M. Ros, D. Franklin, and M. Abolhasan, "Including general environmental effects in K-factor approximation for rice-distributed VANET channels," *Physical Communication*. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874490714001001

[15] A. Mukunthan, C. Cooper, F. Safaei, D. Franklin, and M. Abolhasan, "Studying the Impact of the CORNER Propagation Model on VANET Routing in Urban Environments," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, Sep. 2012, pp. 1 –5.

[16] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, "Global Positioning System. Theory and practice." *Global Positioning System. Theory and practice., by Hofmann-Wellenhof, B.; Lichtenegger, H.; Collins, J.. Springer, Wien (Austria), 1993, 347 p., ISBN 3-211-82477-4, Price DM 79.00. ISBN 0-387-82477-4 (USA).*, vol. -1, 1993.

[17] S. R. Das, E. M. Belding-Royer, and C. E. Perkins, "IETF RFC3561 - Ad hoc On-Demand Distance Vector (AODV) Routing." [Online]. Available: http://tools.ietf.org/html/rfc3561

[18] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, ser. The Kluwer International Series in Engineering and Computer Science, T. Imielinski and H. F. Korth, Eds. Springer US, Jan. 1996, no. 353, pp. 153–181.

[19] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. Boston, Massachusetts, United States: ACM, 2000, pp. 243–254.

[20] J. Bernsen and D. Manivannan, "Unicast routing protocols for vehicular

ad hoc networks: A critical comparison and classification," *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 1–18, Feb. 2009.

[21] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, Jun. 2007.

[22] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," in *IN PROC. 11 TH CANADIAN CONFERENCE ON COMPUTATIONAL GEOMETRY*, 1999, pp. 51–54.

[23] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 120–130.

[24] S. Das, H. Pucha, and Y. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Mar. 2005, pp. 1228–1239 vol. 2.

[25] A. K. Saha and D. B. Johnson, "Modeling Mobility for Vehicular Ad-hoc Networks," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '04. New York, NY, USA: ACM, 2004, pp. 91–92.

[26] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks - VANET '05*, Cologne, Germany, 2005, p. 69.

[27] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: Generating Realistic Mobility Patterns for VANETs," in *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '06. New York, NY, USA: ACM, 2006, pp. 96–97.

[28] D. Krajzewicz, "Traffic Simulation with SUMO – Simulation of Urban Mobility," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. Springer New York, Jan. 2010, no. 145, pp. 269–293.

[29] H. Füßler, H. Hartenstein, M. Mauve, W. Effelsberg, and J. Widmer, "Contention-based forwarding for street scenarios," *1st International Workshop in Intelligent Transportation (WIT 2004)*, 2004.

[30] B. Williams, D. Mehta, T. Camp, and W. Navidi, "Predictive models to rebroadcast in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 295–303, Jul. 2004.

[31] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing

in city scenarios," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 1, pp. 69–72, 2005.

[32] V. Naumov, R. Baumann, and T. Gross, "An Evaluation of Inter-vehicle Ad Hoc Networks Based on Realistic Vehicular Traces," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 108–119.

[33] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, Jan. 2002.

[34] S. Schnaufer and W. Effelsberg, "Position-based unicast routing for city scenarios," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, Jun. 2008, pp. 1–8.

[35] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wirel. Netw.*, vol. 7, no. 6, pp. 609–616, Nov. 2001.

[36] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 2003, pp. 156–161.

[37] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[38] J. Tian, L. Han, and K. Rothermel, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 2, 2003, pp. 1546–1551 vol.2.

[39] K. Lee, J. Haerri, U. Lee, and M. Gerla, "Enhanced Perimeter Routing for Geographic Forwarding Protocols in Urban Vehicular Scenarios," in *2007 IEEE Globecom Workshops*, Nov. 2007, pp. 1–10.

[40] K. Lee, U. Lee, and M. Gerla, "Geo-opportunistic routing for vehicular networks [Topics in Automotive Networking]," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 164–170, May 2010.

[41] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[42] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, "A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications," in *Third International IFIP-TC6 Networking Conference*, 2004, pp. 989–999.

[43] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, "An Improved Vehicular Ad Hoc Routing Protocol for City Environments," in *IEEE*

International Conference on Communications, 2007. ICC '07, Jun. 2007, pp. 3972–3979.

[44] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 3, pp. 1910–1922, 2008.

[45] Y. Xiang, Z. Liu, R. Liu, W. Sun, and W. Wang, "GeoSVR: A map-based stateless VANET routing," *Ad Hoc Networks*, vol. 11, no. 7, pp. 2125–2135, Sep. 2013.

[46] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*. Springer Science & Business Media, Dec. 2011.

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

[48] F. Giudici and E. Pagani, "Spatial and Traffic-Aware Routing (STAR) for Vehicular Systems," in *High Performance Computing and Communcations, First International Conference*, 2005, pp. 77–86.

[49] V. Naumov and T. Gross, "Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 1919–1927.

[50] Q. Yang, A. Lim, S. Li, J. Fang, and P. Agrawal, "ACAR: Adaptive Connectivity Aware Routing Protocol for Vehicular Ad Hoc Networks," in *Proceedings of 17th International Conference on Computer Communications and Networks, 2008. ICCCN '08*, Aug. 2008, pp. 1–6.

[51] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-throughput Path Metric for Multi-hop Wireless Routing," *Wirel. Netw.*, vol. 11, no. 4, pp. 419–434, Jul. 2005.

[52] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "VANET Routing on City Roads Using Real-Time Vehicular Traffic Information," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 7, pp. 3609–3626, 2009.

[53] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, Jun. 1972.

[54] K. Lee, M. Le, J. Harri, and M. Gerla, "Taking the LOUVRE approach," *IEEE Vehicular Technology Magazine*, vol. 4, no. 1, pp. 86–92, Mar. 2009.

[55] P. Sahu, E. Wu, J. Sahoo, and M. Gerla, "BAHG: Back-Bone-Assisted Hop Greedy Routing for VANET's City Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 199–213, Mar. 2013.

[56] H. Saleet, R. Langar, K. Naik, R. Boutaba, A. Nayak, and N. Goel,

"Intersection-Based Geographical Routing Protocol for VANETs: A Proposal and Analysis," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4560–4574, Nov. 2011.

[57] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol (OLSR)," 2003. [Online]. Available: http://hal.inria.fr/inria-00471712

[58] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. Montreal, Quebec, Canada: ACM, 2007, pp. 59–68.

[59] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[60] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wirel. Netw.*, vol. 6, no. 4, pp. 307–321, 2000.

[61] "MATLAB - The Language of Technical Computing - MathWorks." [Online]. Available: http://au.mathworks.com/products/matlab/

[62] T. Iwama and M. Mizuno, "Prediction of propagation characteristics for microcellular land mobile radio," in *Proceedings of the International Symposium on Antennas and Propagation*, vol. 2. Japan: Institute of Electronics, Information and Communications Engineers, 1992.

[63] A. Rustako, N. Amitay, G. Owens, and R. Roman, "Radio propagation measurements at microwave frequencies for microcellular mobile and personal communications," in *'World Prosperity Through Communications', IEEE International Conference on Communications, 1989. ICC '89, BOSTONICC/89. Conference record*, Jun. 1989, pp. 482–486 vol.1.

[64] J. Nuckelt, T. Abbas, F. Tufvesson, C. Mecklenbrauker, L. Bernado, and T. Kurner, "Comparison of Ray Tracing and Channel-Sounder Measurements for Vehicular Communications," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, Jun. 2013, pp. 1–5.

[65] V. Sridhara and S. Bohacek, "Realistic propagation simulation of urban mesh networks," *Computer Networks*, vol. 51, no. 12, pp. 3392–3412, Aug. 2007.

[66] S. Seidel and T. Rappaport, "A ray tracing technique to predict path loss and delay spread inside buildings," in *, IEEE Global Telecommunications Conference, 1992. Conference Record., GLOBECOM '92. Communication for Global Users*, Dec. 1992, pp. 649–653 vol.2.

[67] R. Valenzuela and L. Greenstein, "Performance evaluations for urban line-of-sight microcells at 900 MHz using a multi-ray propagation model," in

Global Telecommunications Conference, 1991. GLOBECOM '91. 'Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services, Dec. 1991, pp. 1947–1952 vol.3.

[68] A. Paier, T. Zemen, L. Bernado, G. Matz, J. Karedal, N. Czink, C. Dumard, F. Tufvesson, A. Molisch, and C. Mecklenbrauker, "Non-WSSUS vehicular channel characterization in highway and urban scenarios at 5.2ghz using the local scattering function," in *International ITG Workshop on Smart Antennas, 2008. WSA 2008*, Feb. 2008, pp. 9–15.

[69] S. Bohacek, V. Sridhara, G. Singh, and A. Ilic, "The UDel Models: MANET Mobility and Path Loss in an Urban/Suburban Environment," University of Delaware, Electrical Engineering Department, Tech Report, 2004.

[70] P. L. Rice, A. G. Longley, K. A. Norton, and A. P. Barsis, "Transmission Loss Predictions for Tropospheric Communication Circuits, Vol. 1," Tech. Rep., Jan. 1967.

[71] A. Longley, "Radio propagation in urban areas," in *28th IEEE Vehicular Technology Conference, 1978*, vol. 28, Mar. 1978, pp. 503–511.

[72] Y. Okumura, E. Ohmori, T. Kawano, and K. Fukuda, "Field strength and its variability in VHF and UHF land-mobile radio service," *Rev. Elec. Commun. Lab*, vol. 16, no. 9, pp. 825–73, 1968.

[73] R. Edwards and J. Durkin, "Computer prediction of service areas for v.h.f. mobile radio networks," *Proceedings of the Institution of Electrical Engineers*, vol. 116, no. 9, p. 1493, 1969.

[74] H. Suzuki, "A Statistical Model for Urban Radio Propogation," *IEEE Transactions on Communications*, vol. 25, no. 7, pp. 673–680, Jul. 1977.

[75] L. Rubio, J. Reig, and N. Cardona, "Evaluation of Nakagami fading behaviour based on measurements in urban scenarios," *AEU - International Journal of Electronics and Communications*, vol. 61, no. 2, pp. 135–138, Feb. 2007.

[76] A. Braga and G. Cavalcante, "Theoretical analysis of a UHF channel in a street city modeled as a deterministic multislit waveguide," in *Microwave and Optoelectronics Conference, 2001. IMOC 2001.Proceedings of the 2001 SBMO/IEEE MTT-S International*, vol. 1, 2001, pp. 421 – 424 vol.1.

[77] T. Abbas, K. Sjöberg, J. Karedal, and F. Tufvesson, "A Measurement Based Shadow Fading Model for Vehicle-to-Vehicle Network Simulations," *International Journal of Antennas and Propagation*, vol. 2015, pp. 1–12, 2015, arXiv: 1203.3370.

[78] T. Rappaport, S. Seidel, and R. Singh, "900-MHz multipath propagation measurements for US digital cellular radiotelephone," *IEEE Transactions on Vehicular Technology*, vol. 39, no. 2, pp. 132 –139, May 1990.

[79] L. J. Greenstein, D. G. Michelson, and V. Erceg, "Moment-method estimation of the Ricean K-factor," *IEEE Communications Letters*, vol. 3, no. 6, pp. 175–176, Jun. 1999.

[80] K. K. Talukdar and W. D. Lawing, "Estimation of the parameters of the Rice distribution," *The Journal of the Acoustical Society of America*, vol. 89, no. 3, pp. 1193–1197, Mar. 1991.

[81] J.-M. Bonny, J.-P. Renou, and M. Zanca, "Optimal Measurement of Magnitude and Phase from MR Data," *Journal of Magnetic Resonance, Series B*, vol. 113, no. 2, pp. 136–144, Nov. 1996.

[82] J. Sijbers, A. den Dekker, P. Scheunders, and D. Van Dyck, "Maximum-likelihood estimation of Rician distribution parameters," *IEEE Transactions on Medical Imaging*, vol. 17, no. 3, pp. 357–361, Jun. 1998.

[83] A. Abdi, C. Tepedelenlioglu, M. Kaveh, and G. Giannakis, "On the estimation of the K parameter for the Rice fading distribution," *IEEE Communications Letters*, vol. 5, no. 3, pp. 92–94, Mar. 2001.

[84] C. G. Koay and P. J. Basser, "Analytically exact correction scheme for signal extraction from noisy magnitude MR signals," *Journal of Magnetic Resonance*, vol. 179, no. 2, pp. 317–322, Apr. 2006.

[85] A. J. den Dekker and J. Sijbers, "Data distributions in magnetic resonance images: A review," *Physica Medica*, vol. 30, no. 7, pp. 725–741, Nov. 2014.

[86] R. Clarke, "A statistical theory of mobile radio reception," *Bell Systems Tech. Journal*, pp. 957–1000, Aug. 1968.

[87] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Statistical simulation models for Rayleigh and Rician fading," in *IEEE International Conference on Communications, 2003. ICC '03*, vol. 5. IEEE, May 2003, pp. 3524– 3529 vol.5.

[88] J. Postel, "IETF RFC 791 - Internet Protocol." [Online]. Available: http://tools.ietf.org/html/rfc791#page-11

[89] "Python - About Python." [Online]. Available: https://www.python.org/about/

[90] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[91] D. Tian, K. Shafiee, and V. C. M. Leung, "Position-Based Directional Vehicular Routing," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 2009, pp. 1–6.

[92] "Valgrind - Callgrind." [Online]. Available: http://valgrind.org/docs/manual/cl-manual.html

[93] “KCachegrind.” [Online]. Available: http://kcachegrind.sourceforge.net/html/Home.html

Appendix A

IMPLEMENTATION OF THE GPSR PROTOCOL

Greedy Perimeter Stateless Routing (GPSR) is the seminal position based routing protocol. Since the Qualnet Network Simulator had no existing implementation of the GPSR routing protocol when commencing this thesis, it was important to implement it as it forms the basis of almost every VANET routing protocol that followed, as noted in Chapter 2.

This appendix describes the implementation of a new routing protocol for Qualnet, focusing specifically on that of the GPSR routing protocol. This description is designed for versions 4.5 or 5.1 of the Qualnet network simulator, but should be equally useful for later versions of Qualnet or Exata.

Qualnet's root directory will be familiar to Linux users since binaries are located in `bin` and headers are in `include`. The main source code of the simulator is stored in the `main` directory and the `libraries` directory consists of various libraries including wireless (IP, Routing Protocols etc...), GSM and UMTS extensions for the simulator. Any user created protocols are usually placed in the `addons` directory, and a `Makefile` should be added appropriately. The `Makefile` in the main directory should be updated to call this file during a `make` command.

A.1 Anatomy of a Qualnet Routing Protocol

A routing protocol in Qualnet needs to be linked to both the layers above and the layers below it. Typically, the protocol receives a packet from a higher layer, such as the application layer, to be routed to a particular destination. This packet is often sent through the Internet Protocol, which can be considered

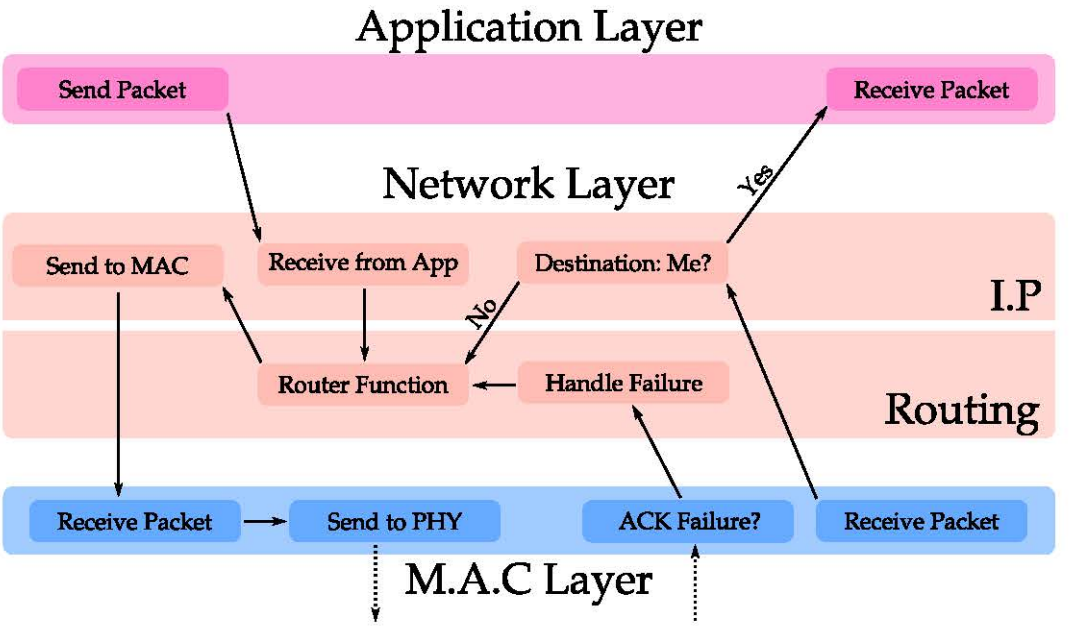


Figure A.1: Inter-Layer Interaction in Qualnet

to encapsulate the routing protocol. The routing protocol then chooses the best next hop and sends the packet to the MAC layer to be forwarded. Both the entry of a packet to a routing protocol and the exit from the protocol are represented as functions in Qualnet's internal architecture. Different functions are used in the implementation of the routing protocol to handle higher-layer packets and control packets (which originate at the routing layer). Control packets are called 'protocol packets' in Qualnet. Figure A.1 illustrates a simplified version of the relationship between the Application, MAC and Network layers, with the specific positions of the Internet Protocol and the routing protocol illustrated within the network layer.

The following sections describe various function calls to a routing protocol, specifically GPSR from the relevant parts of the Qualnet Stack.

A.2 Initialising and Finalising the Routing Protocol

When a simulation starts, every layer of the stack is successively initialised by the kernel. Similarly, every layer has a finalising function that is called when the simulation ends. For routing protocols, these functions are called from `network_ip.cpp` which defines the Internet Protocol segment of the Network layer. The function prototypes and data types for `network_ip.cpp` are found in `network_ip.h`.

The first step to creating a routing protocol is the creation of source and header files in the addons directory. In this case, `routing_gpsr.cpp` and `routing_gpsr.h` are created respectively. All the functions are declared in the header and defined in the source file. The function `GpsrInit(Node, GpsrData, NodeInput, InterfaceIndex, NetworkRoutingProtocolType)` is responsible for initialising GPSR on a specific node and interface, defined by the Interface Index. A node can have multiple interfaces and these are enumerated as a series of integers (`enum`). `GpsrData` is a data structure defined in `routing_gpsr.h`, which holds all the necessary data for each node's implementation of GPSR, including protocol parameters, statistics and other information.

The `GpsrInit` function is called from `network_ip.cpp` in the `IpRoutingInit(Node, NodeInput)` function. The NetworkRoutingProtocolType is an `enum` which identifies the routing protocol as GPSR and is defined in `network.h`, which holds `enums` that identify protocols at every layer of the networking stack. This `enum` is used in `network_ip.cpp` to determine which routing protocol initialisation or finalisation function to call. Similarly, the `GpsrFinalize` function is called from the same file, in the `NetworkIpFinalize` function.

A.2.1 Role of the Initialisation Function

The initialisation function is called for each node individually and is responsible for the following:

1. Initialising the protocol data structure which includes parameters, statistics and node-specific variables such as the neighbourhood table and the random number generator;
2. Reading in protocol parameters from the configuration file (using `IO_ReadInt`, `IO_ReadString`, `IO_ReadTime`, `IO_ReadBool` and associated functions);
3. Initialising the statistics variables (represented as a C++ `struct` which holds statistics such as packets delivered and hop counts;
4. Setting function call-backs for higher layer functions such as the router function, a function to handle MAC layer failures and a function to handle packets overheard in promiscuous mode; and
5. Setting timers for protocol broadcast messages such as beacon messages.

Function call-backs are especially important since this they allow Qualnet to call the relevant functions in the routing protocol. The routing protocol functions in GPSR that are accessed directly by other layers are shown below.

Listing A.1: Function callbacks for GPSR

```
//This function is called when there is a packet to be routed
NetworkIpSetRouterFunction(node, &GpsrRouterFunction, interfaceIndex);
//This function promiscuously hears unicast packets from neighbouring nodes
NetworkIpSetPromiscuousMessagePeekFunction(node, &GpsrPeekFunction,
    interfaceIndex);
//Handles failures at the MAC layer, i.e. when no ACK is received
NetworkIpSetMacLayerStatusEventHandlerFunction(node, &GpsrMacLayerStatusHandler,
     interfaceIndex);
```

A.2.2 Role of the Finalisation Function

The finalisation function is responsible for printing out all the statistics to the statistics file. The `IO_PrintStat` function is used to print the output as shown in below. The `gpsrData` variable should then be deleted to clear the associated memory.

Listing A.2: "Snippet of the Finalise Function"

```
char buf[MAX_STRING_LENGTH];
sprintf(buf, "Number of received packets = %u", gpsrData->stats.numReceived);
IO_PrintStat(node, "Network", "GPSR", ANY_DEST, -1, buf);
//print remaining statistics
delete gpsrData;
```

A.3 Sending and Receiving Beacon Packets

Qualnet uses a message passing API, where packets and any other information is passed between layers of the networking stack encapsulated in a message. Messages carrying packets are normally exchanged between different layers to simulate the transmission of packets in a real network. A node can also send itself a message for a variety of reasons, such as an internal beacon time-out. When a timer runs out, messages are received by the `GpsrHandleProtocolEvent` function, which is called from `network_ip.cpp`. The protocol is identified by a pre-processor macro value defined in `network_ip.h`.

When the GPSRInit function is called, it also sets a timer to send out the first beacon. Each time a beacon timer expires, the `GpsrHandleProtocolEvent` function is called, and the event type is identified by an `enum` set in `api.h`. The identifier used in this implementation is `MSG_ROUTING_GPSR_BeaconTimeout`. Each type of protocol packet needs a unique `enum` to identify it in the protocol handler function. The code for sending a beacon is shown below. The random number generator is used to jitter the beacon interval to prevent beacon synchronisation.

Listing A.3: "Setting the Beacon Timer"

```
gpsrData->rng.setDistributionUniform( halfUpdateInterval, gpsrData->parameter.
    beaconUpdateIntervalClk+halfUpdateInterval);
Message* newMsg = MESSAGE_Alloc(node, NETWORK_LAYER, ROUTING_PROTOCOL_GPSR,
    MSG_ROUTING_GPSR_BeaconUpdate);
MESSAGE_Send(node, newMsg, gpsrData->rng.getRandomNumber());
```

A.3.1 Sending Broadcasts

Beacon packets are broadcast by the routing protocol at regular intervals. In order to send a broadcast packet, a new message must be created and passed to the MAC layer. This is shown below.

Listing A.4: "Sending the Beacon Message"

```
void GpsrSendBeaconUpdatePacket(Node * node, GpsrData * gpsrData)
{
        int packetSize = sizeof(Cartesian2D);
        Message *msg = MESSAGE_Alloc(node, MAC_LAYER, 0, MSG_MAC_FromNetwork);
        MESSAGE_PacketAlloc(node, msg, packetSize, TRACE_GPSR);
        Cartesian2D *beaconPkt = (Cartesian2D *) MESSAGE_ReturnPacket(msg);
        MOBILITY_ReturnCoordinates(node, beaconPkt);
        for (int index = 0; index < node->numberInterfaces; index++) {
                NetworkIpSendRawMessage(node, msg, NetworkIpGetInterfaceAddress(
                    node, index), ANY_DEST, index, IPTOS_PREC_INTERNETCONTROL,
                    IPPROTO_GPSR, 1);
        }
        ++gpsrData->stats.numBeaconSentPkt;
        gpsrData->stats.numBeaconSentOH += packetSize;
}
```

The `NetworkIpSendRawMessage()` function is used to send a packet that doesn't have an IP header to the MAC layer. The function appends an IP header and sends it through to the MAC layer to be broadcast. The `TRACE_GPSR enum` is defined in `trace.h` and allows the Qualnet GUI to trace the packet. The `IPPROTO_GPSR enum` is defined in `network_ip.h` and allows the simulator to recognise that the message represents a GPSR control packet. This subsequently allows the IP layer to call the `GpsrHandleProtocolPacket` function.

A.3.2 Handling Beacon Packets

When a beacon packet is received, it is intercepted by the protocol packet handler function, which is shown below.

Listing A.5: Handling Beacon Packets in GPSR

```
void GpsrHandleProtocolPacket(Node * node, Message * msg, NodeAddress srcAddr,
    int interfaceIndex, int ttl)
{
        //get the GPSR data pointer for this node
        GpsrData *gpsrData = (GpsrData *) NetworkIpGetRoutingProtocol(node,
            ROUTING_PROTOCOL_GPSR);
        Cartesian2D *beaconPacket = (Cartesian2D *) MESSAGE_ReturnPacket(msg);

        GpsrAddOrUpdateStateTableEntry(node, gpsrData, &srcAddr, beaconPacket);

        // update beacon statistics
        ++gpsrData->stats.numBeaconReceived;
        MESSAGE_Free(node, msg);
}
```

The `GpsrAddOrUpdateStateTableEntry` function updates the neighbour table with newer information if the neighbour was previously known and adds the information if a new neighbour is encountered. When the information is added, an additional timer is sent to remove the neighbour if no beacon is received from the neighbour for a specific period of time, indicating that the neighbour has moved away. This timer must be cancelled when a neighbour is updated, since a new timer is set from the time of update. This is shown in listing A.6. Additionally, this function also allows for emptying of a packet buffer if GPSR is configured with one. This is explained further in section A.4.1.

Listing A.6: Updating the Neighbour Table

```
void GpsrAddOrUpdateStateTableEntry(Node *node, GpsrData *gpsrData, const
    NodeAddress* neighborAddress, const Cartesian2D* neighborLocation)
{
        Message* newMsg = MESSAGE_Alloc(node, NETWORK_LAYER,
            ROUTING_PROTOCOL_GPSR, MSG_ROUTING_GPSR_BeaconTimeout);
        NodeAddress *neighbor = (NodeAddress*) MESSAGE_AddInfo(node, newMsg,
            sizeof(NodeAddress), INFO_TYPE_GPSRBeaconTimeOut);
        *neighbor = *neighborAddress;
        bool newNeighbor = false;
        //new entry?
```

```
        if (gpsrData->stateTable.empty() || gpsrData->stateTable.find(*
            neighborAddress) == gpsrData->stateTable.end()) {
                GpsrStateTableEntry newEntry;
                gpsrData->stateTable.insert(pair<NodeAddress,
                    GpsrStateTableEntry> (*neighborAddress, newEntry));
                newNeighbor = true;
        }
        if(newNeighbor) {
                gpsrData->stateTable[*neighborAddress].incomingInterfaceIndex =
                    gpsrData->interfaceIndex;
                gpsrData->stateTable[*neighborAddress].linkUsage = true;
        } else {
                //cancel the existing neighbour checking timer if it exists
                if(gpsrData->stateTable[*neighborAddress].beaconTimeoutMsgPtr !=
                     NULL)
                    MESSAGE_CancelSelfMsg(node, gpsrData->stateTable[*
                        neighborAddress].beaconTimeoutMsgPtr);
        }
        gpsrData->stateTable[*neighborAddress].neighborLocation = *
            neighborLocation;
        gpsrData->stateTable[*neighborAddress].timestamp = getSimTime(node);
        gpsrData->stateTable[*neighborAddress].beaconTimeoutMsgPtr = newMsg;
        //set the new timer
        MESSAGE_Send(node, newMsg, gpsrData->parameter.beaconMaxAgeClk);

        //if buffers are used, empty the packet buffer upon adding a node
        if(newNeighbor && !gpsrData->packetBuffer.empty()) {
                GpsrEmptyBuffer(node);
        }

        return;
}
```

Listing A.7: GPSR State Table

```
typedef struct {
        Cartesian2D neighborLocation;
        int incomingInterfaceIndex;
        clocktype timestamp;
        bool linkUsage; //used during planarization to indicate if link can be
            used
        Message* beaconTimeoutMsgPtr;
} GpsrStateTableEntry;

typedef map<NodeAddress, GpsrStateTableEntry> GpsrStateTable;
```

The state table is represented as an `std::map`, keyed by a node's IP address which allows for easy searching. Each entry is a `struct` as shown in listing A.7.

A.4 Protocol Implementation

Figure A.2 illustrates the processes involved in finding a route in GPSR. The pink rectangles represent functions and the green ones represent points of entry or exit. The blue diamonds are binary decision making blocks. `GpsrRouterFunction` is called by Qualnet whenever a node has a packet to send. This can either be when a higher layer protocol on the node generates a packet or if a packet is received from another node. In the former case, the GPSR header needs to be added to the packet, and greedy mode is chosen by default. If a greedy hop isn't found, the packet is switched to perimeter mode and the first hop is computed. The exact algorithms behind the GPSR protocol can be found in [19].

When a packet is received in perimeter mode, the node first checks if it is closer to the destination than the entry node, which is the exit case for perimeter mode. If so, the packet is switched back to greedy mode. If continuing in perimeter mode, the face change algorithm checks if the perimeter face should be changed, and updates the header appropriately.

A.4.1 Addition of a Packet Buffer

In order to suit sparse networks, the authors of [44] expand GPSR with packet buffers. This feature was also added to this GPSR protocol implementation, though the buffer is not used in this thesis. It was found to have minimal impact in the urban environments tested in this thesis, mostly because sparse networks were not a priority in urban environments. While it resulted in a very minor improvement in packet delivery rates, the average packet delay increased significantly.

The buffer is used when a node has no neighbours in its table. From the point of

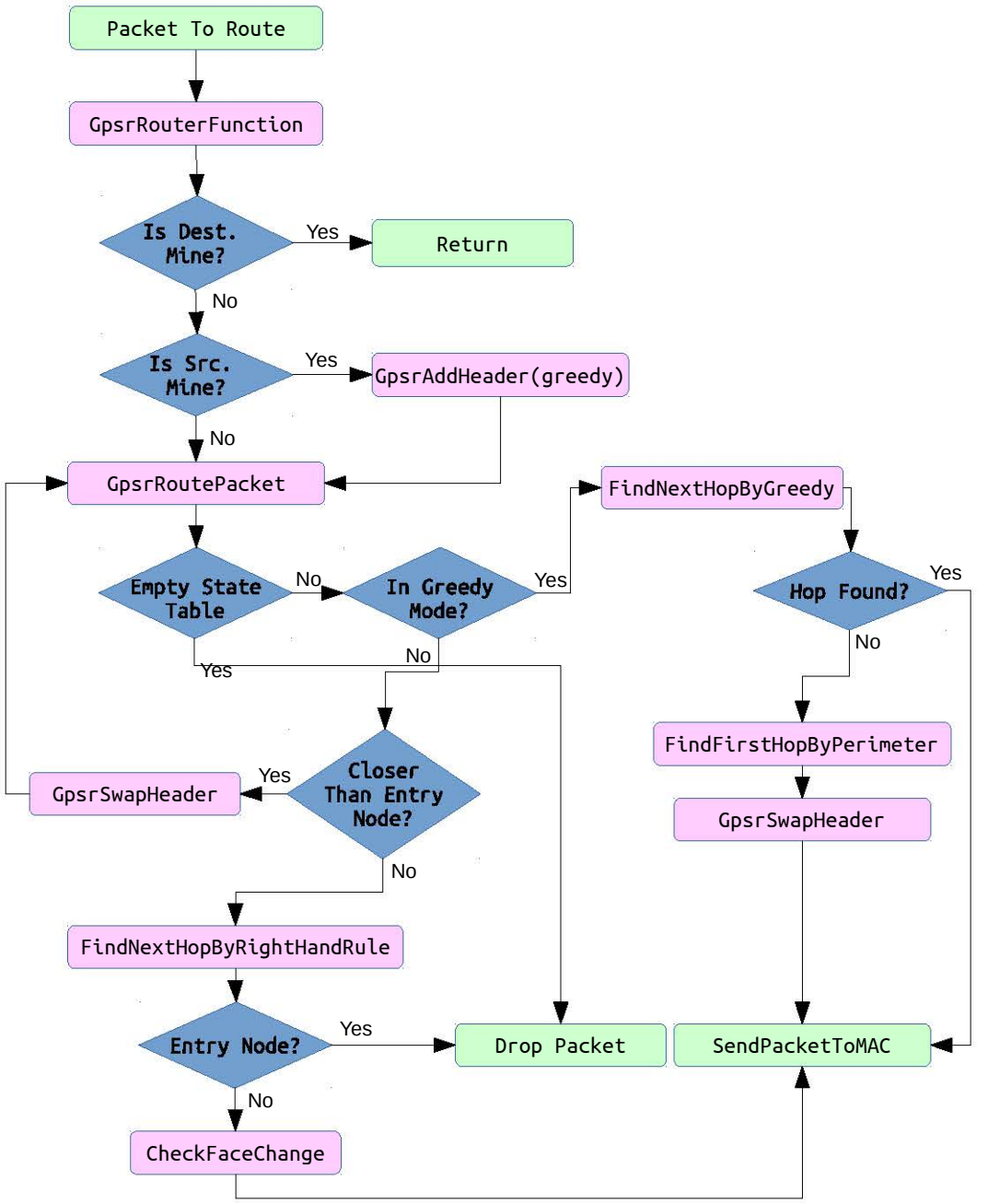


Figure A.2: GPSR Protocol Flow-chart

view of Figure A.2, the 'drop packet' function is replaced with a 'buffer packet' function. It is represented as an `std::list` and a corresponding `struct` as shown in listing A.8.

Listing A.8: GPSR Buffer Data Structure

```
typedef struct {
        Message* msg;
        NodeAddress destAddr;
        NodeAddress previousHopAddr;
        Message* selfMsg;
} GpsrPacketBufferEntry;

typedef list<GpsrPacketBufferEntry> GpsrPacketBuffer;
```

The `*msg` pointer holds the packet and the `*selfMsg` pointer holds a pointer to the time-out message, which is sent each time a packet is added. The time-out message ensures that the buffer entries are cleared after a pre-defined amount of time. A buffer size limit is not enforced in this implementation, though if one were to be enforced, the `deque` structure would be ideal since it would allow access on both ends.

Appendix B

EFFICIENCY IMPROVEMENTS AND OPTIMISATION OF C++ CODE

This section examines a series of design decisions that led to significant improvements in code efficiency. This is mainly examined with the use of the GPSR and CORNER protocol implementations. Code optimisation is of critical importance, since even a minor reduction in the time taken to run one simulation can result in significant time savings when thousands of simulations are run. Valgrind is a well-known debugger and comes with an extremely useful profiler named *Callgrind* [92], which monitors the time taken to run any particular function in an application. Kcachegrind [93] is a useful utility that provides a visual display of the output from Callgrind.

B.1 Using the right Data Structures

The initial implementation of GPSR used a simple array for implementing the GPSR state table. The state table entry and subsequent state table declaration for this list is shown below.

Listing B.1: "Array based State Table for GPSR"

```
typedef struct {
        NodeAddress neighbourAddress;
        Coordinates neighbourLocation;
        int incomingInterfaceIndex;
        clocktype timestamp;
        bool planarisedLinkUsage;
} GpsrStateTableEntry;

GpsrStateTableEntry stateTable[MAX_NEIGHBOURS];
```

While the aforementioned state table is simple and easy to implement, it has disadvantages. When a data structure is being designed, the means of its use should be considered. In the case of the state table, it is often parsed to find the optimal neighbour based on distance to the destination or to check if the destination itself is in the list. The search for the destination is illustrated in the following bit of code, where the state table is iteratively searched till the destination is found. This results in a time complexity of $O(n)$.

Listing B.2: "Destination searching in the GPSR State Table"

```
for (i = 0; i < gpsrData->numNeighbours; i++) {
        if (destAddress == gpsrData->stateTable[i].neighbourAddress) {
                *nextHop = destAddress;
                *outgoingInterfaceIndex = gpsrData->stateTable[i].
                    incomingInterfaceIndex;
                return;
        }
}
```

A better solution would be to replace the neighbour list with an `std::map`. This reduces the aforementioned $O(n)$ complexity algorithm to an $O(logn)$, since the map uses a binary search.Each of the built-in standard-template library functions are optimised for specific purposes and with careful consideration, a significant amount of performance and code-readability enhancements may be derived from their use. The bits of code illustrated in B.1 and B.1 are shown below, implemented as an `std::map`.

```
typedef struct {
        Cartesian2D neighborLocation;
        int incomingInterfaceIndex;
        clocktype timestamp;
        bool linkUsage; //used during planarization to indicate if link can be
            used
        Message* beaconTimeoutMsgPtr;
} GpsrStateTableEntry;

typedef map<NodeAddress, GpsrStateTableEntry> GpsrStateTable;

if (gpsrData->stateTable.find(destAddress) != gpsrData->stateTable.end()) {
```

```
        *nextHop = destAddress;
        *outgoingInterfaceIndex = gpsrData->stateTable[destAddress].
            incomingInterfaceIndex;
        return;
}
```

B.2 Case Study: Cachegrind based efficiency improvements to CORNER code-base

The CORNER propagation model was obtained from the UCLA repositories and was implemented by the authors of [11]. The implementation was integrated into Qualnet 5.1 and run on two separate machines with two different sets of simulation parameters to measure it's performance. The rest of this section demonstrates the improvements in terms of simulation time from very minor optimisations made to the code-base, based on analysis from Callgrind. The Kcachegrind tool is used to create the visualisations of Callgrind data.

The time taken for each run is listed for a 50-node simulation run on a Core-2-Quad 9400 machine and a 350-node simulation run on a Core i7 2600K machine. Each of the simulations is run 5 times and the average time taken is noted.

B.2.1 Initial Run

Figure B.1 illustrates the pre-optimisation run of CORNER. The time taken for this run is **69s** for the 50-node simulation and **2940s** for the 350-node simulation. The figure shows that an extraordinary amount of time is spent in the "BuildLinkId" function, which is shown in listing B.3.

Listing B.3: Old buildLinkId Code

```
std::string CornerData::buildLinkId( string nodeA, string nodeB ) {
        return nodeA + std::string("to") + nodeB;
}
```

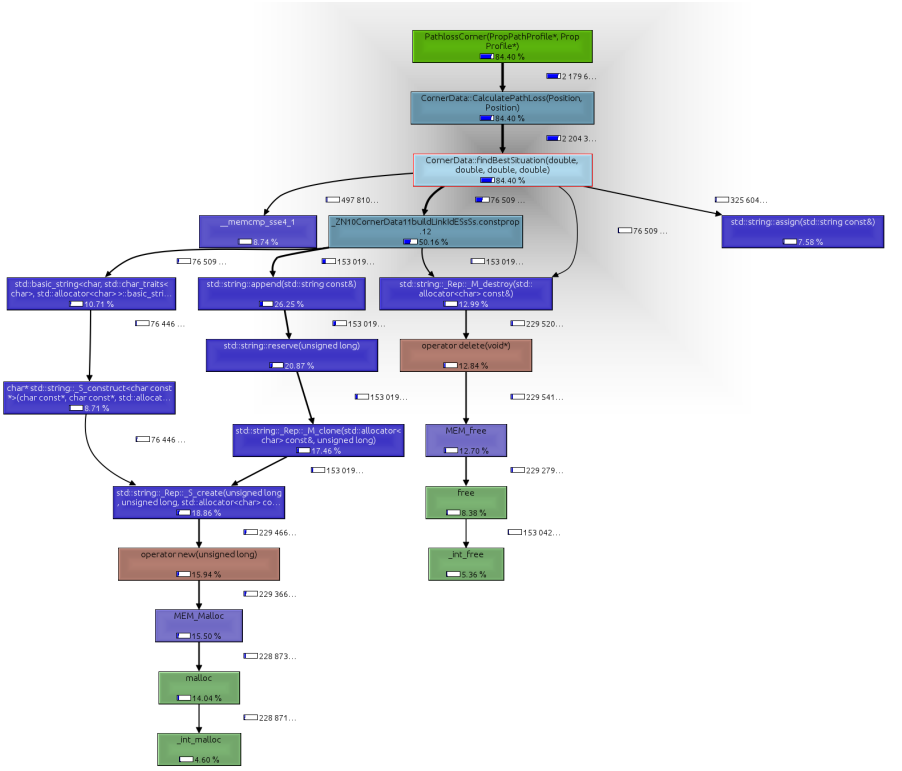


Figure B.1: First Profile run of CORNER Algorithm

B.2.2 Second Run

Further analysis indicates that most of this time is spent appending strings and constructing new ones. The function's purpose is to create a unique link-id by combining two existing link-ids. The addition of the "to" string between the two ids seems redundant and also wastes a lot of time since a new `std::string` needs to be allocated, assigned and an extra append operation takes place. The "to" string is therefore removed. The modified function is shown in listing B.4.

Listing B.4: New buildLinkId Code

```
std::string CornerData::buildLinkId( string nodeA, string nodeB ) {
        return nodeA + nodeB;
}
```

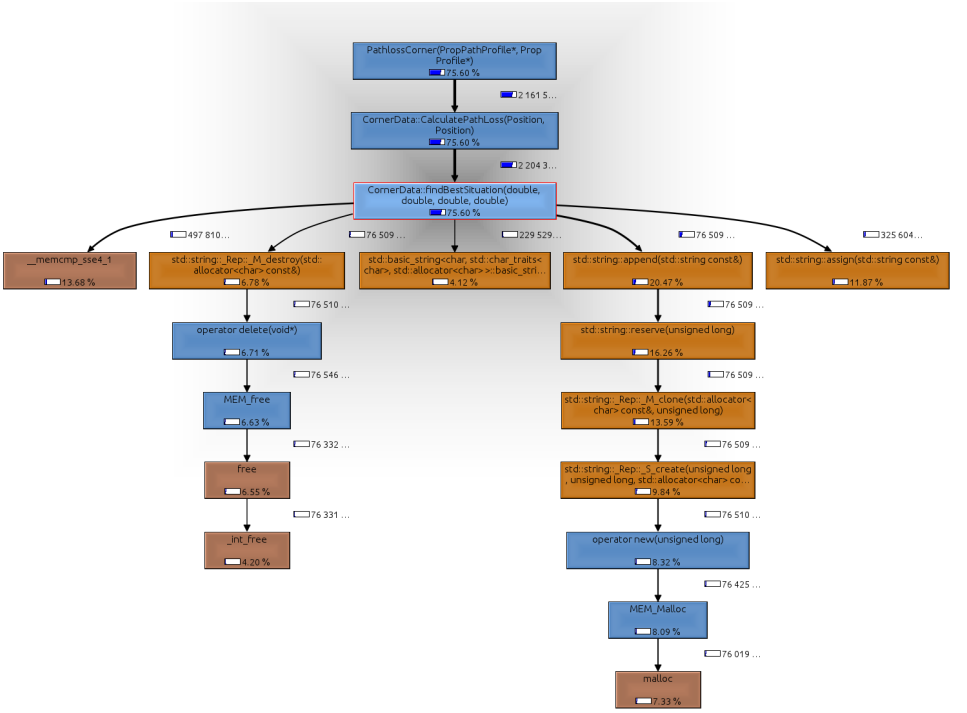


Figure B.2: Second Profile run of CORNER Algorithm

A second run with the aforementioned change illustrates a significant reduction in processing time, from 69s to 45s and 2940s to 2089s for the 50 and 350 node simulations respectively. The outputs of the simulations were also verified against the old outputs and no changes arose as a result of the change.

B.2.3 Third Run

The new code from the second run is once again profiled and the results are illustrated in Figure B.2.

There still appears to be a significant amount of time spent on string operations. The compiler has been able to optimise out the BuildLinkID function to a direct string operation as a result of the rather simple return statement. The string append operation still seems to be taking a significant amount of time, and it would be better if it was completely removed. Analysis of the code reveals

that it is only used twice within the program to compare the "linkIDs", which are essentially "NodeA+NodeB". It might therefore be more efficient to instead compare the individual node IDs rather than comparing the LinkIDs. This allows us to remove the buildLinkId function altogether and also remove a large number of string concatenations.

The code changes for the following run are illustrated in listing B.5.

Listing B.5: Changed Node Comparison Code

```
/*  OLD CODE  */
if ((NLink.linkId == MLink.linkId))
...
possLink[0]=buildLinkId(NLink.nodeA, MLink.nodeA);
possLink[1]=buildLinkId(NLink.nodeA, MLink.nodeB);
...
if (tempLink.linkId == possLink[u])

/*  NEW CODE  */
if ((NLink.nodeA == MLink.nodeA && NLink.nodeB == MLink.nodeB) || (NLink.nodeA
    == MLink.nodeB && NLink.nodeB == MLink.nodeA))
...
possLinkA[0] = NLink.nodeA;
possLinkB[0] = MLink.nodeA;
possLinkA[1] = NLink.nodeA;
possLinkB[1] = MLink.nodeB;
...
if (tempLink.nodeA == possLinkA[u] && tempLink.nodeB == possLinkB[u])
```

While readability is sacrificed slightly, the changes result in another significant decrease in execution time. The execution time for the 50 node simulation drops from 45s to 30s and the 350 node simulation drops from 2089s to 1309s.

B.2.4 Fourth Run

The profiler output of the code from the third run is illustrated in Figure B.3. A significant amount of time is still used by the `string::assign` function. Analysis of the code suggests that it should be possible to replace all these string

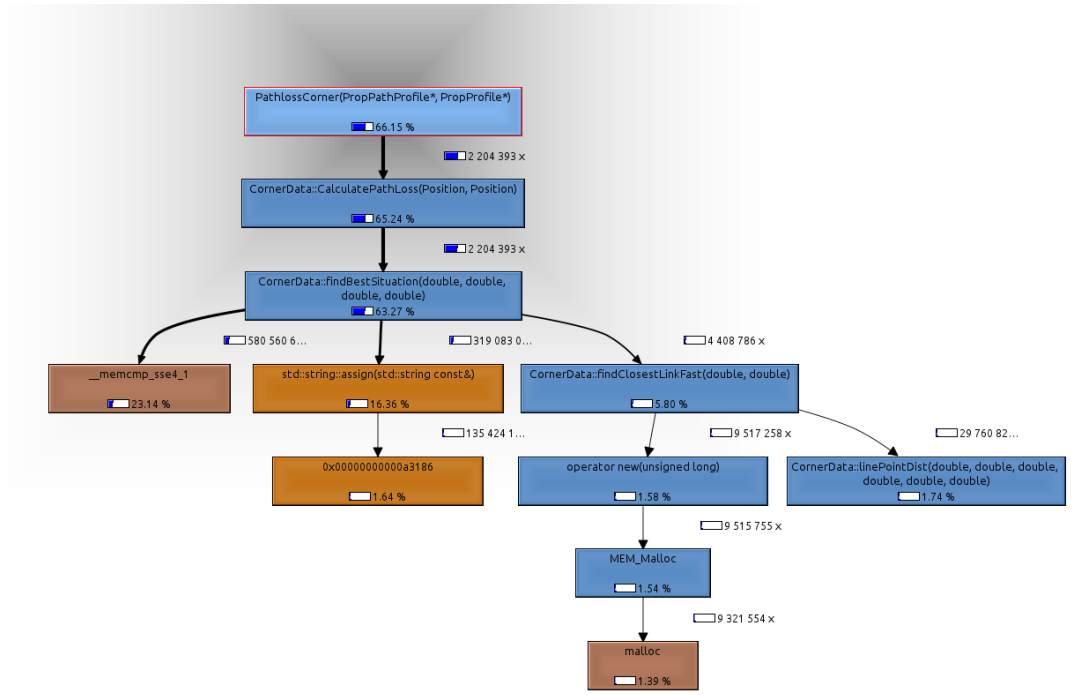


Figure B.3: Third Profile run of CORNER Algorithm

assignments with pointers because they are not modified and are only used for comparison. The changes to the code are illustrated in listing B.6.

Listing B.6: Replacing String Creation with Pointers

```
/* OLD CODE */
string possLinkA[8];
string possLinkB[8];
possLinkA[0] = NLink.nodeA;
possLinkB[0] = MLink.nodeA;
possLinkA[1] = NLink.nodeA;
possLinkB[1] = MLink.nodeB;
...
/* NEW CODE */
string* possLinkA[8];
string* possLinkB[8];
possLinkA[0] = &(NLink.nodeA);
possLinkB[0] = &(MLink.nodeA);
possLinkA[1] = &(NLink.nodeA);
possLinkB[1] = &(MLink.nodeB);
```

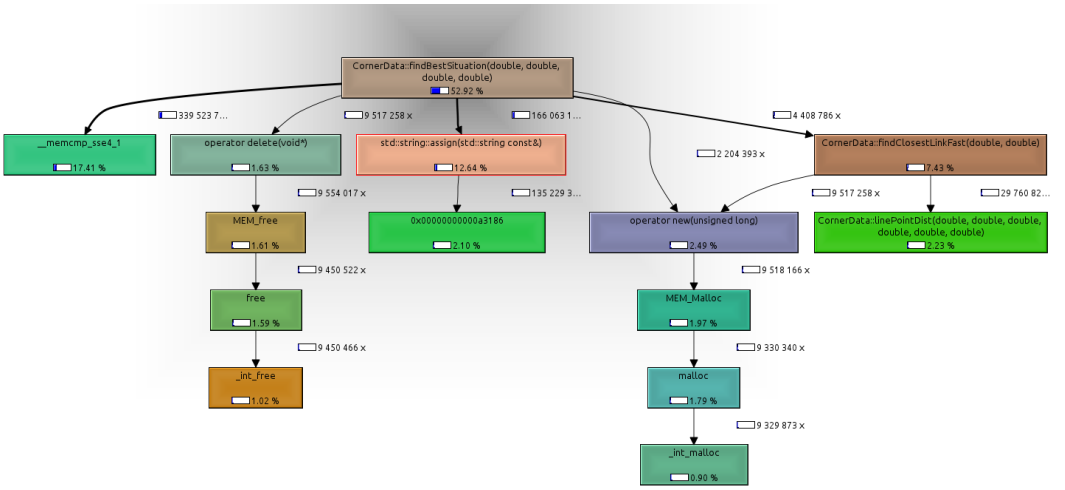


Figure B.4: Fourth Profile run of CORNER Algorithm

The changes result in a comparatively modest performance improvement. The 50-node simulation run drops from 30s to 23s and the 350-node run drops from 1309s to 983s.

B.2.5 Fifth Run

The profiler output of the code from the fourth run is illustrated in Figure B.4.

Replacing all the string assignment operations with pointers has made a significant difference. However, there still appears to be significant amount of time spent in string operations (`memcmp` and string allocation). Further analysis indicates that the only real use of the strings (`CornerLink.nodeA`, `CornerLink.nodeB` and `CornerNode.nodeId` is in cross-linking links to nodes. Since these are represented as arrays, it seems more logical to use the actual index of the arrays, which are integers, to replace all comparison operations.

This involves more of a rewrite, but it fairly straightforward. The run-times for the 50-node simulation drops to 13.2s and for the 350-note simulation drops to 660s. The final Callgrind analysis is shown in Figure B.5.

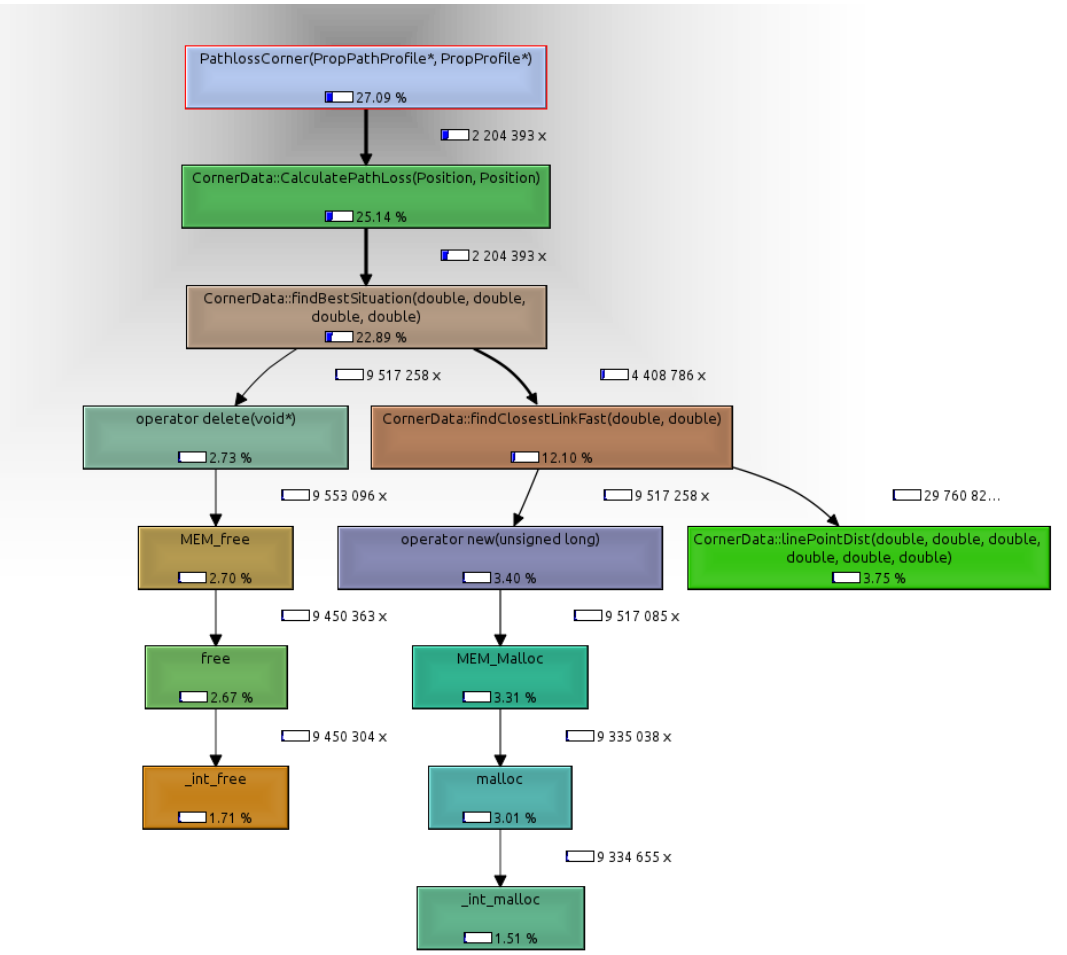


Figure B.5: Final Profile run of CORNER Algorithm

B.3 Summary

The aforementioned section discussed the use of the *Callgrind* tool in Valgrind to help optimise code. As a simple case-study, the CORNER implementation was optimised, through very minor changes to result in a roughly 80% decrease in execution time. While this is significant over a single simulation, it is even more critical when running a large number of simulations for a study. A weeks worth of simulations may only take a day and a half, resulting in more time to draw conclusions and perform actual research.

Appendix C

INTEGRATION OF SUMO, CORNER AND QUALNET

The SUMO traffic simulator runs independently and simulates a series of vehicular traces using microscopic mobility simulation. SUMO runs through a simulation and outputs an `xml` file with a second by second output state of the simulation. This file can be converted to a Qualnet mobility file for input into Qualnet.

CORNER is the propagation model used to determine the path-loss between two nodes on a road network. When the mobility is simulated with SUMO, the SUMO road network and node positions are also converted to a format usable by CORNER. The rest of this appendix looks at the representation of road networks under SUMO and CORNER and studies how the simulation stack between the three entities functions.

C.1 Representation of Road Networks

SUMO and CORNER have slightly different representations of road networks. SUMO represents networks as edges, lanes and nodes. An intersection is represented as a node, with edges connecting adjacent nodes. Each edge is defined as a series of lanes, which are multi-segmented lines.

CORNER relies on a simplified approach with only edges and nodes. Therefore, a Python based script was written to convert a SUMO network into a CORNER map. Figure C.1 illustrates the differences between a small segment of a SUMO and CORNER map. As illustrated, an edge in SUMO can have multiple lanes and multiple segments, while in CORNER, each edge is a single segment defined by

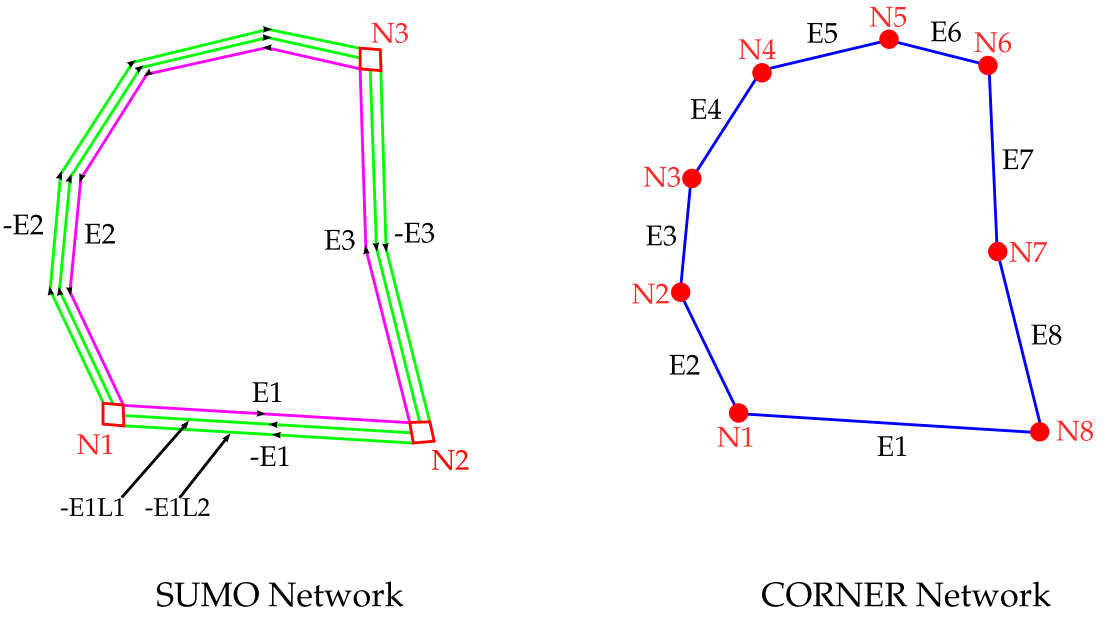


Figure C.1: SUMO and CORNER Networks

the nodes on either edge. Additionally, forward and reverse edges are represented individually in SUMO, while CORNER only considers one edge between nodes. This simplification is carried out to reduce computational complexity.

C.1.1 CORNER File Structure

CORNER represents four separate files, nodes, edges, classifications and mobility. These files are read in by the network simulator when the propagation environment is initialised. Nodes and edges are defined in flat text files as illustrated in Figure C.2. The first line in both files is the number of entries in the file. This is written for ease of memory allocation in C++. The number of lanes is used to calculate the width of the road based on the average lane width. The direction and speed are not used at present, but are defined for possible future use in some aspect of the simulation.

In addition to the above two files, CORNER also defines a classifications file and a mobility file. The mobility file is used to calculate the position of a node on a

```
ID   x-coord     y-coord          |          ID   n1ID  n2ID Lanes Dir  Speed
-----------------------------------|-----------------------------------------
182                               |          329
0    320.57      0.22             |          0    41    47    2    0   13.89
1    285.54      0.0              |          1    17    66    2    0   13.89
2    402.58      1.29             |          2    65    64    2    0   13.89
3    1115.2      5.32             |          3    66    65    2    0   13.89
4    1039.82     4.83             |          4    63    24    2    0   13.89
...                               |          ...
```

Figure C.2: CORNER Nodes (left) and Edges (right) files

```
edge1  edge2  corners.  intNodes  mainWidth  sideWidth  paraWidth  turn1  turn2
-------------------------------------------------------------------------------
48171
63     249    2        9         2          2          2          117    92
135    299    1        4         2          2                     41
63     241    2        5         2          2          2          115    151
98     100    0        6
63     240    2        6         2          2          2          114    121
....
```

Figure C.3: CORNER Classifications File

particular edge and is represented simply as a list consisting of the node-ID, time and edge(s). This file is discussed in further detail in C.2. The classifications file is similarly a text file which lists the classification between any two edges.

An example classifications file is illustrated in Figure C.3. Like the nodes and edges files, both these files list the number of entries at the very top for ease of memory allocation. As illustrated, the LOS scenario only has the number of intermediate nodes between the source and destination edges. The NLOS1 scenario has, in addition to this information, the width (in lanes) of the main

and side streets as well as the node where the turn occurs. This is used by CORNER to compute the distance to the turning point and work out the path-loss between the two nodes. The NLOS2 scenario has extra parameters, i.e., the width of the parallel street and a second turning point.

C.1.2 SUMO to CORNER Conversion

A Python based tool was written to convert a SUMO network/output to a CORNER network/mobility file and a Qualnet mobility file for the purposes of network simulation. The classification algorithm used is detailed in section 3.5. The script to convert the network iteratively runs through each edge of the SUMO network and first averages out the lanes to a single defined lane. Reverse edges, i.e., edges that run in the opposite direction are then merged onto the primary edge. The following code snippet shows the process for converting an edge, defined by a shape that is extracted from the SUMO network file, to a series of edges and intersections for CORNER. As seen in the snippet, each sub-shape is treated as a separate edge and the change-point is treated as a node.

Listing C.1: SUMO to CORNER Edge conversion

```
for coord in shape:
      #if coord is the first point of the shape, then link is [from, 0]
      if coord == shape[0]:
              int_b=self._getSubShapeName(edge_prefix, edge._from._id, edge._to.
                  _id, subshape)
              if int_b not in nodeMap:
                      self._intersections.append(Intersection(shape[1][0],shape
                          [1][1]))
                      nodeMap[int_b]=nodeid
                      nodeid += 1
              allLinks.append(Link(edge._id, iteration, nodeMap[edge._from._id],
                  nodeMap[int_b],edge.getLaneNumber(),edge.getSpeed(),self.
                  _intersections))

      #if coord is the last point of the shape, then link is [3, to]. Also,
          break...
      elif coord == shape[-2]:
```

```
                if edge_type=='f':
                        shp_a = subshape-1
                else:
                        shp_a = subshape+1
                int_a=self._getSubShapeName(edge_prefix, edge._from._id, edge._to.
                    _id, shp_a)
                if int_a not in nodeMap:
                        self._intersections.append(Intersection(shape[-2][0],shape
                            [-2][1]))
                        nodeMap[int_a] = nodeid
                        nodeid += 1
                allLinks.append(Link(edge._id, iteration, nodeMap[int_a],nodeMap[
                    edge._to._id],edge.getLaneNumber(),edge.getSpeed(),self.
                    _intersections))
                break
        #intermediate point. Calculate both intersection a and b, for first iter,
            it is [0, 1]
        #since int_a is from the previously calculated value, it is unnecessary to
             calculate it again
        else:
                if edge_type=='f':
                        shp_a = subshape-1
                else:
                        shp_a = subshape+1
                int_a=self._getSubShapeName(edge_prefix, edge._from._id, edge._to.
                    _id, shp_a)
                int_b=self._getSubShapeName(edge_prefix, edge._from._id, edge._to.
                    _id, subshape)
                if int_b not in nodeMap:
                        self._intersections.append(Intersection(shape[iteration
                            +1][0],shape[iteration+1][1]))
                        nodeMap[int_b] = nodeid
                        nodeid += 1
                allLinks.append(Link(edge._id, iteration, nodeMap[int_a],nodeMap[
                    int_b],edge.getLaneNumber(),edge.getSpeed(),self.
                    _intersections))
        if edge_type == 'f':
                subshape += 1
        else:
                subshape -= 1
        iteration += 1
```

C.2 Creating the Mobility File

Qualnet's mobility file is represented as a plain text file with a list of nodes, times and coordinates for each node-time combination. This file is also created by converting a SUMO output file, which is an `xml` file, using Python. At the same time, a CORNER mobility file is also created, which holds a list of potential edges for each node-time combination. This file is used by CORNER in Qualnet to work out the appropriate classification by means of a look-up table, with the reverse-geocoding step done in pre-processing. This makes the simulation significantly more processor-efficient, by sacrificing a small amount of memory.

If a vehicle is within a certain radius of an intersection, it is considered to be at either one of the edges. The code snippet below illustrates the Python code used to implement this.

Listing C.2: "CORNER Mobility File Creation"

```
posLinks = set([str(edgeMap[(edge, segment)])])
for i in cInts:
      if dist((x_coord, y_coord), cInts[i]) < intRadius[i]:
               posLinks.update(intMap[i])

#output the corner links into the mobility file
corner_mob.write(str(nodeid)+" "+str(timeid)+" "+str(len(posLinks))+" ")
for i in posLinks:
      corner_mob.write(str(i)+" ")
corner_mob.write("\n")
```

Once the mobility file is created, Qualnet loads the CORNER nodes, edges, classifications and mobility files, along with its own mobility file (with coordinates). Subsequent calls to the CORNER path-finder functions then simply use the pre-processed files as look-up tables.

C.3 Generating Traces with Python and TraCI

Throughout this thesis, mobility traces are generated based on the random-waypoint idea, where a node picks a random destination and moves towards it before picking another destination and repeating the process. A Python-based program was written and implemented for this thesis which implements a similar idea but with street and traffic considerations by using SUMO. SUMO has a set of extensions known as TraCI, which allows external applications to communicate with and control the traffic simulator.

TraCI was used, along with a Python based implementation of the A-Star [59] path-finding algorithm to feed SUMO vehicular mobility traces. The following code snippets illustrate the control of TraCI through Python.

Listing C.3: "Adding a Driver to a Simulation"

```
#generate a route for the driver (uses A*)
(startNode,destNode,driverRouteEdges) = self.generateRoute()
#add route to SUMO
traci.route.add(routeID,driverRouteEdges)
#pick a random vehicle type (car, bus etc)
vtypeid = PickRandomVehicleType(self.vehicleDefinitions)
#add the vehicle and route to the simulation
traci.vehicle.add(driverID,routeID,typeID=vtypeid)
```