2015

Clustering in VANETs under realistic channel conditions

Craig Stephen Cooper
University of Wollongong

Clustering in VANETs under realistic channel conditions

Craig Stephen Cooper

Supervisors:

Dr. Montserrat Ros, Prof. Farzad Safaei, & Dr. Daniel Franklin

This thesis is presented as required for the conferral of the degree:

Doctor of Philosophy (Telecommunications)

The University of Wollongong

School of School of Electrical, Computer, and Telecommunications

November, 2015

Declaration

I, Craig Stephen Cooper, declare that this thesis submitted in fulfilment of the requirements for the conferral of the degree Doctor of Philosophy (Telecommunications), from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Craig Stephen Cooper

November 2, 2015

Abstract

A Vehicular Ad hoc Network (VANET) differs from conventional wireless networks by its lack of infrastructure. Cars equipped with wireless devices must harmonise their transmissions in order to achieve the objectives of the network. Clustering serves as a means of doing so, by grouping vehicles according to spatial distribution and relative velocity. These groups, led by an elected cluster head, can serve as the foundation of routing, collision detection, and infotainment applications. After more than twenty years of research, clustering has diversified into a wide array of approaches and mechanisms, all of which have the potential for wide-spread application. However, the research has lacked crucial elements. Firstly, published taxonomies tended to compare algorithms and protocols based on their application. There is minimal emphasis on the underlying design choices and their suitability to the problem of VANET communication. Secondly, channel modelling practices have been insufficient to adequately validate new designs. Without these two elements, new avenues for research have gone unnoticed. This thesis aims to address these issues.

This thesis provides a taxonomy of the techniques applied in previous literature to solve the problems of cluster head election, affiliation, and management, identifying new directions in the design of these algorithms. Additionally, this thesis analyses the methodology of validating clustering performance, finding a lack of rigorous channel modelling and identifying a pattern of comparing new approaches to much older protocols that were not designed for the unique nature of vehicular networks. The thesis then emphasises the need for a standard, rigorous testing regime for new protocols, requiring the usage of comprehensive channel models and comparison to more recent and adequate competing algorithms, in order to confirm high-performance clustering techniques, develop new approaches, and push VANET

technology toward deployment.

Next, this thesis details the consequences of inadequate channel modelling. A brief review of the most and least accurate modelling approaches is presented, after which a reductionist approach is suggested. Three independent channel phenomena are identified: path-loss, large-scale fading due to vehicular shadowing, and small-scale fading due to multi-path transmission. Models are identified from existing literature that simulate two of the three phenomena to an acceptable degree of accuracy, while the third is shown to be best described by a Rice Distribution. The thesis then presents a method of approximating the Ricean K-factor based on the instantaneous static environment. The strongest signal propagation paths are resolved in order to determine specular and diffuse powers for approximation. The model is experimentally validated in two different urban areas in New South Wales, Australia. High correlation between the model and experimental data was obtained over short-range communication links, demonstrating the suitability of the model in urban VANETs. The three simulation approaches are combined into an Urban Radio Channel (URC).

Three algorithms were selected for a deeper simulation comparison study. These algorithms were selected based on aspects of their design, such as the use of weighting metrics to determine cluster head eligibility and hand-off mechanisms. The algorithms were simulated under both URC and an ideal model, providing two comparisons. The insight from these comparisons were assessed, showing that an ideal channel not only overestimates protocol performance, but can reward simplicity and arbitrariness of design over complexity and comprehensiveness. A realistic channel model, by contrast, allows comprehensive designs to work to their full potential, while excising unsuitable designs. Thus the stated consequences of inaccurate channel modelling are verified.

Following the demonstrated success of URC, a comprehensive study of the three algorithms is conducted. They are tested in an urban and highway scenario with different parameter configurations, in order to study the influence of road structure and mobility on performance, and identify techniques that form stable and long-lived clusters. To the best of this researcher's knowledge, this is the first work which investigates multiple clustering algorithms over multiple scenarios in VANETs. The

simulation study provides valuable insight into the performance of a diverse range of algorithms, and favours the application of driver intention and channel awareness in forming stable long-lived clusters.

One of the surveyed algorithms is extended to Channel and Route Aware Clustering (CRAC). A route similarity metric attempts to cluster vehicles that will remain near each other for longer, theoretically improving cluster lifetime and stability. The maintenance scheme is augmented, using a channel sensing mechanism to safeguard against cluster re-organisation caused by temporarily poor channel conditions. Simulations were conducted on a highway scenario, showing that the modifications quadruple cluster stability. These novel modifications can be applied to any clustering algorithm in the literature.

The conclusion of the thesis includes motivations for future work in improving performance of the Urban Radio Channel, as well as recommendations for new approaches to the VANET clustering problem.

Acknowledgments

If I were to adequately do justice to my acknowledgements section, it might require an additional thesis on its own. So I'll credit the most obvious contributions.

My supervisors, who over the last few years have offered useful insights and feedback on my work – especially in the final few months.

My parents, who spent so much time as my encouraging and supportive soundboards. My dad, for driving one of the cars in my practical experiments; and my mum, who helped proof-read many of my papers and thesis chapters.

Extra special thanks to my colleague, Abhinay Mukunthan, for his assistance with the experiments in Chapter 3; and to his supervisor, Mehran Abolhasan, for his useful feedback.

A special mention to Steve Tree of Rohde and Schwarz, who provided the spectrum analyser equipment for the experiments.

I would like to extend my gratitude to the examiners of this thesis for their thoughtful and helpful feedback, as well as their praise and encouragement of my work.

Contents

Chapter 1

Introduction

1.1 VANETs: Applications and History

In a Vehicular Ad hoc Network (VANET), vehicles are equipped with wireless devices that allow them to communicate directly with each other in an infrastructureless manner. This promises applications ranging from improved traffic safety and congestion avoidance to in-car infotainment systems.

Serious interest in VANET technology started to develop in the early 1990s, and has greatly increased in recent years. The need for standards in this area become apparent after the emergence of electronic road tolling systems based on a variety of proprietary active RFID transponder systems. Realising that this simple concept could be generalised to support a variety of vehicular communications applications, several major manufacturers of electronic toll collection systems established the Dedicated Short-Range Communications (DSRC) Industry Consortium, which worked towards a common physical layer standard for short-range vehicular communication. This effort led to the development of a physical layer based on the ad hoc mode of IEEE 802.11 operating in the 5.9 GHz band; this is known as the Wireless Access in Vehicular Environments (WAVE) / Dedicated Short Range Communications (DSRC) standard and has been formalised by the IEEE as 802.11p [1, 2]. Building on this physical layer, a range of standards for various other parts of the network stack are currently under development by the IEEE, ISO, ETSI and other bodies – in particular, the IEEE 1609 working group is developing standards for security and applications for VANETs built on top of 802.11p [3–6].

Since then, several approaches to the design of a vehicular communication system have been followed. Many focused on the use of stationary wireless devices known as Road-Side Units (RSU), which manage vehicular communication in a similar fashion to infrastructure networks. Others went in the opposite direction, developing routing algorithms to be robust in the face of variable topology, or adapting existing protocols to the challenges of the vehicular environment. As a middle ground between the two, a method of grouping vehicles according to certain traits of similarity – clustering – was developed. A clustering algorithm would allow vehicles to form infrastructure within the ad hoc network by selecting a single vehicle among themselves to serve as a manager. This node is called the Cluster Head (CH). The other vehicles within range would affiliate with the CH, and become Cluster Members (CM). The infrastructure formed from these algorithms could then lay the foundations for further protocols, such routing, collision avoidance systems, addressing, and data aggregation and dissemination, by managing the mobility of the network and responding to the resultant changes in topology.

Clustering can potentially form the backbone of a VANET communication system. Such an approach would need to adequately respond to changes in topology, driver intention, and the channel itself. Routing, security, and safety applications can be built upon this system with the express goal of making roads safer and more efficient. For instance, a vehicle could alert the driver to a traffic jam ahead, and quickly offer a route around it, before the driver can see it.

1.2 Motivations of this work

VANETs remain largely theoretical despite the work conducted over the last two decades. This is, in part, due to economic and social issues. The envisioned applications and benefits of VANET technology will not become apparent until a sufficient percentage of vehicles have been equipped with the technology [7]. Reaching that level of adoption may prove difficult from a consumer perspective. The consumer may be disinclined to serve as early adopters of the technology, if they cannot perceive any benefit from the added expense of a communications system. This is where VANET systems differ from other vehicular systems such as GPS: the technology

only provides benefits to an individual so long as other drivers use it. Governments may place a requirement for all vehicles to be equipped with the technology, but this would present a difficulty with different municipalities placing different restrictions and requirements on communications systems; car manufacturers based in different countries may follow different standards and practices [8].

Another problem preventing adoption of VANET technology is the inherent difficulties of reliable vehicular communication over a highly variable channel [9]. Each vehicle serves as a mobile scatterer and obstruction, their velocity imposing Doppler effects on the channel. Additionally, movement changes the environment, particularly the presence of buildings that obstruct, reflect, and deflect signal power. This forms a complex and variable communications environment, which is difficult to overcome for delay-intolerant applications like collision warning systems.

A compounding problem is how such protocols are assessed for their efficacy. A proposed communication protocol has two options for verification:

1. Practical Experiment - Cars are physically equipped with wireless hardware running an implementation of the protocol being tested.

2. Simulation - A network and channel simulator coupled with a traffic model runs an implementation of the protocol being tested.

The former approach gives highly accurate results, but mandates an untenable resource expenditure. For reliable results, a sufficient number of vehicles are required to participate in the experiment, which in some cases can be thousands. The latter approach is far more manageable, the only expenditure being computer processing power and time to implement the protocol. The disadvantage, however, is that it is difficult to accurately model the VANET channel, which is influenced by various environmental factors [10–14]. The method by which a researcher models wireless propagation would impact on the measured performance of a proposed network architecture [15].

Much of this work was motivated by an observation made during the literature survey period. A review of the VANET literature on clustering has revealed certain drawbacks to how the VANET channel is modelled, and how new algorithms are benchmarked against competing approaches. With so many promising designs filling

the literature, it is apparent that inaccurate simulation methodologies cripple a researcher's ability to differentiate good designs from ineffective ones. The most accurate approach, known as *Raytracing*, is computationally expensive and resource-hungry, while ideal channel modelling misses environmental phenomena that may influence performance in real life. A trade-off between the two approaches would be beneficial.

From these observations, there are two arguments to be made:

1. The complexities of the VANET channel can be modelled simply by taking a different approach to the simplification. The channel can be broken into multiple parts and modelled separately and independently. In so doing, computational efficiency is maintained without a significant loss in simulation accuracy.

2. Accurate simulation and suitable benchmarking will distinguish innovative and efficacious design choices from unsuited ones.

1.3 Contributions

This thesis' contributions are separated across five chapters.

Chapter 2 contains a comprehensive literature review of clustering. VANET clustering taxonomies have been presented in [16, 17]. The typical approach has been to categorise the surveyed algorithms according to some characteristic, and discuss each algorithm within its own grouping. This work is differentiated with three unique contributions:

- Section 2.2 introduces the applications of clustering. The survey distinguishes *General Purpose* architectures as those proposed without a particular application or those that can be applied to achieve multiple objectives.

- Section 2.3 reviews the ways in which the algorithms approach the key elements of the clustering problem. How the cluster head is elected, how unclustered nodes affiliate with a head, and how heads handle the interaction with other clusters are all important design elements. The survey focuses more on the actual design choices rather than the algorithms themselves.

- Section 2.4 contains an assessment of verification methodologies, a contribution unique to this thesis. The simulator frameworks, channel models, and comparison to competing algorithms are reviewed. To the knowledge of the author, no other clustering taxonomy presents such an analysis.

The content of this chapter can be found in the following publication:

Currently under review:

Craig Cooper, Montserrat Ros, Farzad Safaei, Daniel Franklin, and Mehran Abolhasan. A comparative survey of VANET clustering techniques. *IEEE Surveys and Tutorials*, 2015

Chapter 3 describes a reductionist channel modelling approach that attempts to reduce execution time while maintaining accuracy.

- Section 3.1 states the consequences of inaccurate channel modelling in terms of protocol evaluation.

- Section 3.3 presents the components of the VANET channel as identified by this author. Section 3.4 then follows on with established methods of efficiently modelling these components. An additional gap is identified in how small-scale fading is modelled, and that the current approaches require some significant resource expenditure to experimentally obtain the data necessary.

- The remainder of the chapter is dedicated to presenting a method of approximating small-scale fading parameters from more easily accessible data sources. Details are given on an experiment campaign on streets in Wollongong, Australia; the purpose of which is to validate the proposed approximation method. Finally the three components of the VANET channel are combined into a single model called Urban Radio Channel (URC).

The content of this chapter can be found in the following publications:

C. Cooper, A. Mukunthan, M. Ros, D. Franklin, and M. Abolhasan. Dynamic environmental fading in urban vanets. In *Communications (ICC), 2014 IEEE International Conference on*, pages 5641–5646, June 2014

Craig Cooper, Abhinay Mukunthan, Farzad Safaei, Montserrat Ros, Daniel Franklin, and Mehran Abolhasan. Including general environmental effects in k-factor approximation for rice-distributed VANET channels. *Physical Communication*, 14:32 – 44, 2015. ISSN 1874-4907

Chapter 4 covers the first phase of a detailed simulation study of three clustering algorithms. In this phase, the algorithms were compared in two scenarios: one under a commonly-used ideal channel model and another under URC. The three algorithms were selected as representative of the popular techniques identified in Chapter 2. The three algorithms are described in detail. The comparison of performance under the different channel models is used to validate the consequences of inaccurate models delineated in Section 3.1. In particular, URC reveals design flaws and the prevalence of malfunctions that undermine the applicability of these algorithms to the VANET communication problem. The survey shows that an ideal model would hide these malfunctions by creating a hospitable channel environment that rewards simple designs. The second phase of this simulation survey is covered in Chapter 5. This phase assessed the three algorithms' responses to changes in road structure, across both grid and highway scenarios, and with different parameter configurations. From the results, several useful techniques are identified, while problems with unsuited approaches were raised. In particular, algorithms that work toward hierarchical merging of clusters lead to greater stability than ones that require nearby disjoint clusters to absorb each other; timer-based disconnection detection approaches are fundamentally unsuited to the VANET environment, and a polling approach augmented with channel sensing would provide a more robust solution to the problem. The results will indicate the avenues of improvement in VANET clustering, and propose potential new approaches to the problem.

The content from Chapters 4 and 5 can be found in the following publications:

C. Cooper, M. Ros, F. Safaei, D. Franklin, and M. Abolhasan. Simulation of contrasting clustering paradigms under an experimentally-derived channel model. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–6, Sept 2014

Currently under review:

Craig Cooper, Montserrat Ros, Daniel Franklin, and Mehran Abolhasan. A simulation study of VANET clustering techniques. *IEEE Transactions on Intelligent Transportation Systems*, 2015

Chapter 6 follows from the preceding chapter, augmenting one of the studied algorithms with two mechanisms for improving cluster stability. A route similarity metric is presented, intended to cluster vehicles that will remain near each other for longer periods of time; a channel analysis step is added to prevent transient channel black-outs causing unnecessary reaffiliations. The study shows that the channel analysis step greatly improves the success of the algorithm, while the similarity metric is less helpful. This highlights the second argument made in the previous section. The content from this chapter can be found in the following publication:

Currently under review:

Craig Cooper, Montserrat Ros, and Farzad Safaei. Employing route similarity and channel sensing to improve cluster stability in vanets. In *IEEE Transactions on Intelligent Transportation Systems*, 2015

Chapter 2

State of the Literature

2.1 Introduction

Research into clustering began with Ephremides et al and the DARPA packet-radio network [22]. The idea was to form subnets within a Mobile Ad hoc Network (MANET) to facilitate in the distribution of network resources. This work was built upon by Gerla et al [23], who proposed the popular Lowest ID and Highest Degree (LID/HD) clustering algorithms for MANETs. Mobility Clustering (MOBIC) was later presented in [24], attempting to incorporate mobility considerations into the clustering phase. Several other algorithms including Distributed Group Mobility Adaptive (DGMA) clustering [25], and MobHiD [26] were later proposed, each designed for clustering in MANETs and performing increasingly well under the typically-used mobility models such as Random Way Point.

As VANET research slowly moved away from its MANET origins, the methodologies whereby a cluster was formed diversified. Algorithms were designed with specific applications in mind, including Peer-to-Peer (P2P) file sharing and channel access management. In so doing, cluster head election and member affiliation schemes were adapted to or invented for these applications. This diversity has grown to the point that adequately classifying these algorithms and identifying new avenues for research is difficult. Moreover, methods of validating proposed protocols utilise channel and mobility models that do not realistically reflect the VANET environment, and compare new VANET clustering architectures to old MANET approaches that have not been designed for the characteristics of the VANET scenario.

This chapter delineates the major threads of clustering research in terms of design aspects, with emphasis placed on how the surveyed algorithms seek to solve the facets of the clustering problem. Protocol validation methodologies are discussed, focusing on inadequacies of channel modelling and benchmarking. Suggestions of potential new avenues of clustering research are presented. The chapter concludes with a call for standardised verification methodologies.

2.1.1 Stages of a Clustering Algorithm

A clustering process can be divided into four clear phases:

1. Neighbourhood Discovery – In this phase, a node collects information about its neighbouring nodes. This information differs between algorithms, but in all cases it contains at least the neighbour's ID (e.g. MAC or IP address).

2. Cluster Head Election – The data collected is used here to calculate a node's suitability to be Cluster Head. The content of this data and its products is referred to as the *Election Criteria*. One of the defining attributes of an algorithm is how the election criteria is interpreted to determine the neighbour best suited to be a Cluster Head.

3. Affiliation – When a neighbour with the best suitability is identified, a node will then try to join that neighbour's cluster. Algorithms also differ in how this phase is carried out.

4. Maintenance – This phase begins when a Cluster Head obtains its first member. It periodically updates data on its members, and performs the cluster's primary function – its *Application*. When a node's connection to a cluster fails, it returns to the first phase to find a new Cluster Head.

This process flow is displayed graphically in Figure 2.1.

2.1.2 Structure of the Taxonomy

The survey begins with an introduction of all studied algorithms, presented in terms of their applications. Each application comes with its own requirements and complications, which are delineated in each subsection and highlight the need for ap-

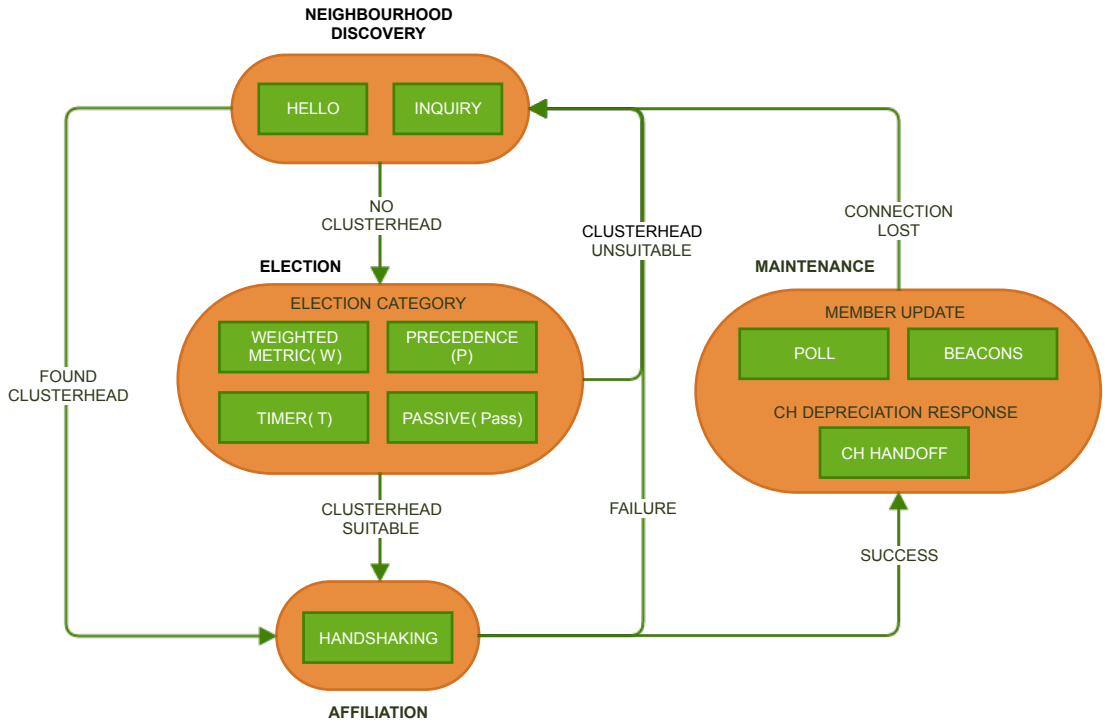


Figure 2.1: Components common to algorithms in this taxonomy. This is used in the survey as logic flow chart to categorise the surveyed algorithms.

propriate assessment of design choices. The survey then moves on to present the taxonomy.

The taxonomy is divided based on the common components of the algorithms presented, with a section dedicated to each component described in Figure 2.1. The survey begins with Cluster Head Election, which can follow one of weighted-metric, precedence, timer, and passive approaches. The survey goes on to analyse and compare the criteria used to assess Cluster Head suitability, followed by the methods used by nodes to affiliate with detected Cluster Heads. Strategies to disseminate data for assessing Cluster Head suitability (referred to as Neighbourhood Discovery) are then discussed. An additional discussion is presented on the election of gateways, which are Cluster Members typically used for inter-cluster communication, RSU affiliation, and cellular integration. The phases described heretofore comprise the steps in establishing the cluster; the next section presents describes the techniques used in maintenance of a cluster once it is established, which includes methods for updating member data in the Cluster Head's neighbour table, and responses to CH depreciation.

This chapter presents a further analysis of the verification strategies of the sur-

veyed algorithms. The functioning of the channel model and the nature of algorithms used as benchmarks is discussed, and drawbacks of these approaches is detailed.

2.2 Applications of Clustering

The design of a clustering algorithm is heavily influenced by the application for which it is intended to be used. Several application-specific algorithms have been proposed in the literature, while others aim to satisfy the requirements of a very broad and varied range of applications. Table 2.1 summarises the algorithms presented in the survey, grouped by their primary application (if defined).

2.2.1 Applications

The following subsections describe the application classes as shown in Table 2.1.

General-Purpose Algorithm

A significant subset of proposed clustering algorithms can be classified as general purpose, in that they were designed for the purpose of grouping vehicles without any particular comfort or safety application in mind. General purpose algorithms tend to be designed with a high emphasis on robustness in the face of vehicular mobility. They can be seen as generic platforms upon which other schemes can be built. Gerla's LID/HD is an example of a general purpose MANET clustering algorithm [23]. Later work by Chiang et al. extended Gerla's algorithm with support for routing [27], and work by Fan et al. applies it to channel access management [28].

Most general purpose protocols in this survey were based on MANET algorithms, and had been adapted to vehicular mobility. An example of this is the Distributed Clustering Algorithm (DCA) [29]. The authors based their work on LID/HD, but replaced the use of ID or Degree with a weighted sum of metrics such as internode distance. This is a simple message-driven method, designed on the premise that nodes are stationary during the clustering phase. The concept was then extended to account for mobility metrics, resulting in the Distributed Mobility-Aware Clustering (DMAC) algorithm. This incorporated a mobility-aware

Table 2.1: List of clustering algorithms in this survey.

Application	**Algorithm Name**	**Acronym**	**Citations**
General Purpose	Distributed Mobility Aware Clustering	DMAC	[29]
	Generalised Distributed Mobility Aware Clustering	GDMAC	[30]
	Modified Distributed Mobility Aware Clustering	MDMAC	[31]
	Density Based Clustering	DBC	[32, 33]
	Criticality-based Clustering Algorithm	CCA	[34]
	K-hop	K-hop	[35]
	Aggregate Local Mobility	ALM	[36]
	Spring Clustering	Sp-Cl	[37]
	Enhanced Spring Clustering	E-Sp-Cl	[38]
	Position-based Prioritised Clustering	PPC	[39]
	Utility Function	UF	[40]
	Threshold-Based Clustering	TBC	[41, 42]
	Fuzzy-Logic-Based Algorithm	FLBA	[43]
	User Oriented Fuzzy-logic-based Clustering	UOFC	[44]
	Mean Connection Time Clustering	MCTC	[45]

Table 2.1 – *continued from previous page*

Application	**Algorithm Name**	**Acronym**	**Citations**
Routing	Cluster-Based Location Routing	CBLR	[46, 47]
	Robust Mobility Adaptive Clustering	RMAC	[48]
	Mobile Infrastructure in VANET	MI-VANET	[49]
	Passive Clustering	PC	[50]
	Vehicular Passive Clustering	VPC	[51]
	Cluster Formation for IVC	CF-IVC	[52]
	Trust-dependant Ant Colony Routing	TACR	[53]
	Dynamic Backbone-Assisted MAC	DBA-MAC	[54]
	Cellular Automata Clustering	CAC	[55]
	Multihoming Clustering Algorithm for VANETs	MCA-VANET	[56]
Channel Access Management	Clustering based on Direction in Vehicular Environments	C-DRIVE	[57–59]
	Adaptable Mobility-Aware Clustering Algorithm based on Destination	AMACAD	[60, 61]
	Clustering with Scalability for Broadcast Performance	CSBP	[28]
	Affinity Propagation in Vehicular Environments	APROVE	[62, 63]
	Cluster-Based MAC	CBMAC	[46, 47]

Table 2.1 – *continued from previous page*

Application	**Algorithm Name**	**Acronym**	**Citations**
	Hierarchical Clustering Algorithm	HCA	[64, 65]
	TDMA Cluster-based MAC	TC-MAC	[66–68]
Security	Vehicular Weighted Clustering Algorithm	VWCA	[69]
	Cluster-Based Public Key Infrastructure	CBPKI	[70]
	Agent Learning-based Clustering Algorithm	ALCA	[71]
QoS Assurance	Cluster Configuration Protocol	CCP	[72, 73]
	Stability Based Clustering Algorithm	SBCA	[74]
	Multi-Channel Cooperative Clustering-based MAC	MCC-MAC	[75]
Traffic Safety	Cluster-based Risk-Aware Cooperative Collision Avoidance	C-RACCA	[76]
Topology Discovery	Clustering Protocol for Topology Discovery	CPTD	[77]
Combination with cellular infrastructure	Cluster-based Multimetric adaptive Gateway Management mechanism	CMGM	[78, 79]

maintenance phase, in which member nodes detected new and failed links, adjusting the cluster structure as necessary.

It should be stressed that DMAC was specifically designed for ad hoc networks undergoing mobility, yet that does not imply the kind of mobility patterns of vehic-

ular environments. The authors continued development in [30], first by analysing DMAC's performance under three different mobility models: Random-Way Point (RWP), Brownian Motion (BM), and Manhattan Grid (MG). The authors found that DMAC's clusters lasted up to 100 seconds. Clusters lasted less than a minute under BM, though average node speed had little impact. Cluster member stability, represented by cluster residence times, was similarly poor, with high rates of reaffiliation under all mobility models. The authors presented Generalised DMAC (G-DMAC), which incorporated additional mobility checks to the cluster head election scheme, which improved efficiency and stability of the process.

General purpose algorithms show the most focus on considerations of vehicular mobility. Research has expanded to consider mobility behaviour such as destination, turning direction at the next intersection, and varied speed of vehicles. Section 2.3 gives a detailed discussion of the approaches to clustering in the face of vehicular mobility.

Routing

Clusters serve as a mechanism for dynamically building a hierarchical infrastructure overlay on top of the underlying ad hoc network, which can be used to route packets. Most often, the cluster head performs the routing between its members and neighbouring clusters. As a result, algorithms tend to form hierarchies in one of three ways. Firstly, one cluster head can also be a member of another cluster, creating a tree-like structure. A classic example of this approach is Robust Mobility-Aware Clustering (RMAC) [48]. In the second major structure type, a dynamic backbone structure is formed, where all cluster heads exchange routing messages with or via each other; Dynamic Backbone-Assisted MAC (DBA-MAC) is a representative example of this method [54]. Finally, a hierarchy can be constructed using gateway nodes, which are members of multiple clusters. Cluster-Based Location Routing (CBLR) is an example of such a scheme [80, 81].

A subset of the routing-oriented clustering algorithms in Table 2.1 include Passive Clustering (PC) [50] and its derivatives. These are built into Ad hoc On-demand Distance Vector (AODV) [82], piggy-backing their control data on existing traffic. This passive approach is discussed further in Section 2.3.2.

Channel Access Management

An advantage of clustering is that it compartmentalises a VANET in a similar way to subnetworks in an infrastructure network. This allows a cluster head to coordinate activities amongst its members, in particular, by controlling channel access scheduling. Algorithms intended for this application typically utilise a TDMA-based approach, whereby the cluster head assigns timeslots to its members (for example, [67]). Such clustering algorithms allow QoS assurance, and eliminate the need for RTS/CTS frames, which have been shown to be ineffective in VANETs due to the hidden node problem [9].

Security

The need for nodes to affiliate with cluster heads in order to participate in the network provides a potential VANET security mechanism. The surveyed algorithms of this kind employ multi-state authentication mechanisms in the affiliation process to protect against malicious vehicles, and often include degree-of-trust parameters in the computation of cluster head weights.

Topology Discovery

Multi-hop clustering structures provide a useful means of analysing the topology of the road network. Cluster heads collect information from affiliated nodes and neighbouring clusters in order to build maps of the network. This data can then be passed – via routing traffic or dedicated information dissemination techniques – to other parts of the network to be used in traffic management solutions.

Traffic Safety

A VANET can use a clustering structure to anticipate vehicular collisions. Cluster-based Risk-Aware Cooperative Collision Avoidance (C-RACCA) incorporates breaking time and collision-avoidance capability into its cluster head election scheme, ensuring that each node has sufficient following distance to break and avoid hitting the car it is following [76]. When an emergency situation is declared, a warning is disseminated through the cluster and "the recipient vehicle, at once, reacts to it based on the event type included in the warning message" (sic).

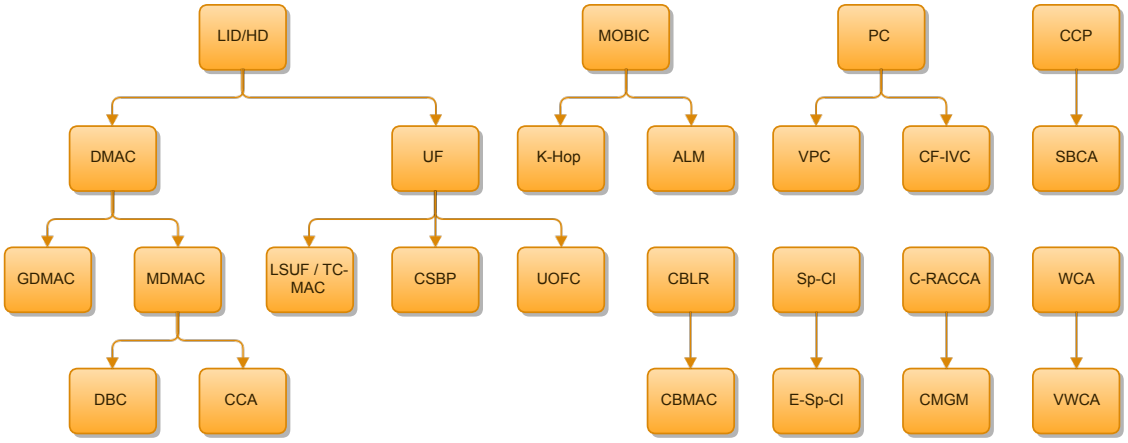


Figure 2.2: Lineage of VANET clustering algorithms. LID/HD [23] is noted as the earliest significant VANET clustering algorithm, as can be seen from its influence on subsequent designs

Combination with UMTS infrastructure

In the literature, VANETs are often assumed to be integrated with cellular infrastructure to support access to the Internet. Cluster-based Multimetric adaptive Gateway Management (CMGM) uses a clustering structure to facilitate integration while mitigating load on the cellular network [78, 79]. The algorithm selects a node equipped with cellular connectivity as an Internet gateway for clusters. The selection is made based on cellular received signal strength, and potentially removes the need for dedicated stationary road-side units.

2.2.2 Discussion

A trend which is evident in this survey is the evolution of early MANET algorithms to progressively account for more aspects of the VANET operational environment. For example, LID/HD has been extensively modified and improved upon, resulting in a diverse set of descendant algorithms resembling a biological clade. C-RACCA [76] has been co-opted for integration with cellular networks [78], while routing algorithms, such as CBLR [46], have been extended to channel access applications [47]. Drawbacks of MOBIC's cluster head election metric were identified in further research, which proposed solutions that are more robust against VANET mobility (Details are given in Section 2.3.3). This indicates research is seeking improvement and solution to the problems imposed by VANET scenarios. Figure 2.2 is an inheritance diagram, showing how clustering architectures have diversified in recent years.

In particular, this diagram shows the high influence of Gerla's LID/HD approach on recently-proposed VANET clustering protocols.

Basing new algorithms on earlier approaches presents a problem, in that certain concepts or assumptions of the original may be retained. For example, the surveyed passive approaches all stem from the original PC proposal [50], which was based on AODV. There are competing routing protocols that may benefit from the passive clustering approach, and potentially offer greater performance than AODV. The scalar metric approach of LID/HD, and the later-developed weighed metric sums, can cause the clustering decisions of one node to influence the decisions of unrelated nodes, leading to instability of cluster affiliations (See Section 2.3.1 for further discussion). The dependence on modifying past architectures can keep researchers from investigating new avenues that were not considered by the authors of the early MANET algorithms, such as the precedence scheme of RMAC [48] or the passive approaches of PC [50].

The surveys in [16] and [17] gave a comprehensive view of the literature, but did so while combining application and design approach in their classification system. While this is a good method of organising the survey, a rigorous assessment of the facets of the clustering problem and the approaches to each has not been conducted. The following Section details a taxonomy on the techniques designed to solve the clustering problem across all applications.

2.3 Taxonomy of Clustering Techniques

Establishing a clustering structure is a multifaceted problem, due to the unique mobility and channel dynamics of the VANET environment. First a node must obtain information about its neighbouring vehicles, in order to identify potential cluster heads. The criteria for head selection includes mobility, signal quality, and capability to provide specific traffic and infotainment functions. The means whereby this criteria is expressed also varies: an algorithm may specify criteria in the form of a scalar indicating its fitness to be a cluster head, or it may compute a countdown timer after which it announces its self-appointment of the role. There are also multiple methods whereby a node joins a cluster, and how a cluster is managed

after formation.

This section presents a taxonomy of the approaches to each of these problems. Each method is presented with an analysis of their advantages and disadvantages with respect to the VANET environment. Additional discussion is given on how disadvantageous approaches have been adapted or modified to improve their applicability to vehicular networks.

2.3.1 Cluster Head Election Strategy

The method whereby a cluster head is elected, and the parameters used in the decision, are described in this section. Three main assessment mechanisms are identified in the literature, based on self-reported weighted network metrics, precedence, and timers respectively. These each use various parameters of the network in an attempt to find the node most suitable to be cluster head.

Weighted Network Metrics

The most common approach for cluster head election requires each node to calculate an index quantifying its fitness to act as a cluster head for its neighbours, and to advertise this index within its 1-hop neighbourhood. Nodes wishing to affiliate with a cluster head will then rank all neighbours in their neighbour table and request association with the most highly-ranked candidate node (which may be the node itself). The index is typically a weighted sum of various network metrics, such as the degree of connectivity, link stability, node uptime etc., with the weights reflecting the relative importance of the selected metrics for this network. Twenty-six of the surveyed algorithms use this approach, making it by far the most popular election mechanism.

The necessary information is distributed to neighbouring nodes via one of the mechanisms discussed in Section 2.3.5. The criteria defining "best score" may mean the lowest among the neighbours (e.g. Adaptable Mobility-Aware Clustering Algorithm based on Destination (AMACAD) [60] and Vehicular Weighted Clustering Algorithm (VWCA) [69]) or highest (e.g. Mean Connection Time Clustering (MCTC) [45] and Multihoming Clustering Algorithm for VANETs (MCA-VANET) [56]). The common metrics for the selection index are discussed in Section 2.3.3.

Table 2.2: Analysis of the algorithms in this survey. The column headings refer to the corresponding components detailed in Figure 2.1. The **Cat.** header refers to the Election strategy, where *Weighted Metric* is represented by "W", *Precedence* by "P", *Timer* by "T", and *Passive* by "Pass."

Name	Cat.	Election Metric(s)	Neighbour Discovery	Affiliation Handshake	CH Handoff
DMAC	W	ID, Degree	Hello	No	No
GDMAC	W	ID, Degree	Hello	No	No
MDMAC	W	ID, Degree, LET	Hello	No	No
DBC	W	SNR, Distance, Velocity	Inquiry	Yes	No
CCA	W	Network Criticality	Hello	No	No
CBLR	T	Static Timeout	Inquiry	Yes	No
CBMAC	T	Static Timeout; Multiple heads decide with Mean Distance and Velocity	Hello	Yes	No
RMAC	P	Cluster head status, Distance, Velocity, Size	Inquiry	Yes	No
HCA	T	Random Timeout	Hello	Yes	No
C-DRIVE	P	Front of platoon, Direction of travel	Hello	No	Yes
K-hop	W	Propagation Delay Ratio	Hello	No	No
ALM	W	Distance	Hello	No	No
C-RACCA	P	Relative velocity, Deceleration capability, In front or behind	Inquiry	Yes	Yes

Continued on next page

Table 2.2 –continued from previous page

Name	Cat.	Election Metric(s)	Neighbour Discovery	Affiliation Handshake	CH Handoff
CMGM	P	Relative velocity, Deceleration capability, In front or behind	Inquiry	Yes	No
MI-VANET	W	Vehicle Class	Hello	Yes	No
Sp-Cl	W	Distance and Relative Velocity, forming weight analogous to Coulomb's Law	Hello	Yes	No
E-Sp-Cl	W	Distance, Relative Velocity, and Vehicle Height, forming weight analogous to Coulomb's Law	Hello	Yes	No
PPC	W	ID, Travel Time, Average Relative Velocity	Hello	No	No
AMACAD	W	Distance, Relative Velocity, Destination	Hello	Yes	Yes
UF	W	Distance, Relative Velocity, ID, Degree	Hello	No	No
CSBP	W	Distance, Relative Velocity, ID, Degree	Hello	No	No
TC-MAC	W	Distance, Relative Velocity, Turning direction	Hello	No	No
APROVE	W	Distance, Relative Velocity	Hello	Yes	No
CCP	T	RSS	Hello	No	No

Continued on next page

Table 2.2 –continued from previous page

Name	Cat.	Election Metric(s)	Neighbour Discovery	Affiliation Handshake	CH Handoff
SBCA	T	RSS, Relative Velocity	Hello	No	Yes
TBC	W	Distance, Relative Velocity	Inquiry	No	No
PC	Pass	First Declaration Wins	None	No	No
VPC	Pass	Degree, Link Quality, Link Expiration Time	None	No	No
CF-IVC	Pass	Speed Grouping	None	No	No
FLBA	W	Relative Velocity	Hello	No	Yes
UOFC	W	Distance, Relative Velocity, Acceleration, Driver Intention	Hello	No	No
VWCA	W	Direction, Degree, Distrust Level, Velocity	Hello	No	No
TACR	W	Distance, Relative Velocity, Traffic Obedience, Packet Forwarding Reputation	Inquiry	No	No
DBA-MAC	T	Link Expiration Time	Hello	Yes	No
CPTD	P	ID, Degree, Link Expiration Time	Hello	Yes	No
MCC-MAC	P	Relative Velocity	Hello	Yes	No
MCTC	W	Mean Connection Time	Hello	No	No
CBPKI	W	Trust Level	Hello	Yes	No
CAC	W	Driver Interest	Unclear	Unclear	No

Continued on next page

Table 2.2 –continued from previous page

Name	Cat.	Election Metric(s)	Neighbour Discovery	Affiliation Handshake	CH Handoff
ALCA	W	Velocity	Unclear	Unclear	No
MCA-VANET	W	Consecutive Hello message reception	Hello	No	No

The advantage of this approach is its simplicity and its ability to guarantee that only a single cluster head exists within a node's communication range. An additional advantage is that the relative importance of each network metric can be tuned for a specific network in the calculation of the weighted fitness index by adjusting the value of the weight coefficients.

Cluster head election strategies based on weighted network metrics need to contend with the dynamism inherent in the VANET environment, which can cause the selection metrics and hence the optimal choice of cluster head to change over time. Often the changes are only temporary; therefore, several authors have applied low-pass filters to the cluster head fitness indices to prevent transient changes from causing unnecessary reclustering [33, 43]. In [71], a learning mechanism is used to elect cluster heads based on the relative velocity of vehicles. Agents are deployed at junctions to determine the velocity and behaviour of nodes entering/leaving the intersection. The learning algorithm rewards positive results (such as selection of a cluster head which exhibits a long lifespan and high packet delivery ratio for routed packets) by an increase in a learning factor, while penalising negative results (a poorly-performing cluster head) with a decrease. This progressively results in better cluster heads being selected, while resisting short-lived changes in cluster head selection metrics. In Criticality-based Clustering Algorithm (CCA) [34] and G-DMAC [30], a hysteresis threshold is used to prevent thrashing of the cluster structure; a cluster member will only change to a new cluster head if the the proposed new cluster head's selection metric exceeds that of the current cluster head by a set threshold. Aggregate Local Mobility (ALM) [36] employs a contention timer to prevent reclustering when two clusters move within communication range of each other, and to avoid the formation of single-member clusters. Position-based Prioritised Clustering (PPC) [39] specifies a *dismiss threshold*, which prevents the merging

of clusters until they are physically closer than the specified threshold distance.

Several possible issues arise with this cluster head selection methodology. Since the index representing a node's suitability to lead the entire cluster is based on its relationship with its entire neighbourhood, a node may be identified as the best candidate as a cluster head for a particular node even when its connectivity is quite marginal, because it has good connectivity with the other nodes in the cluster. Additionally, different nodes may be visible to a different subset of the entire network, and small movements may result in a very different assessment of the best node to select as cluster head. Depending on the distribution, density and behaviour of traffic, this may result in poor clustering performance for some nodes, particularly at the margins of the network.

Precedence

Unlike the weighted-metric fitness index approach, in which nodes advertise their own fitness to be a cluster head, in precedence-based schemes, a node seeking a cluster head will make its own assessment of the suitability of potential cluster heads, again based on a weighted sum of metrics. As the assessment is made by the cluster-head-seeking node, it can act selfishly and choose a cluster head which suits its requirements rather than the aggregate of the other nodes in the network. An example of this so-called precedence-based approach is RMAC [48], which sorts its neighbours in order of relative distance, relative velocity, and whether or not a neighbour is already a cluster head; another is MCC-MAC [75], an 802.11p extension in which vehicles use the CCH to search for cluster heads with the lowest velocity relative to their own. Precedence-based algorithms allow for higher cluster member stability, particularly for marginal nodes, by allowing each node to choose the cluster head that best suits them. Precedence-based algorithms are thus best suited to forming hierarchical structures.

This higher stability comes at the expense of greater control overhead, as a larger number of cluster heads result from the selfish approach. Additionally, if cluster members have the capability to host other nodes (thereby forming a hierarchical structure), there may be a tendency to form chains rather than trees (See Figure 2.3). While this can be advantageous in certain scenarios, such as platoon formation

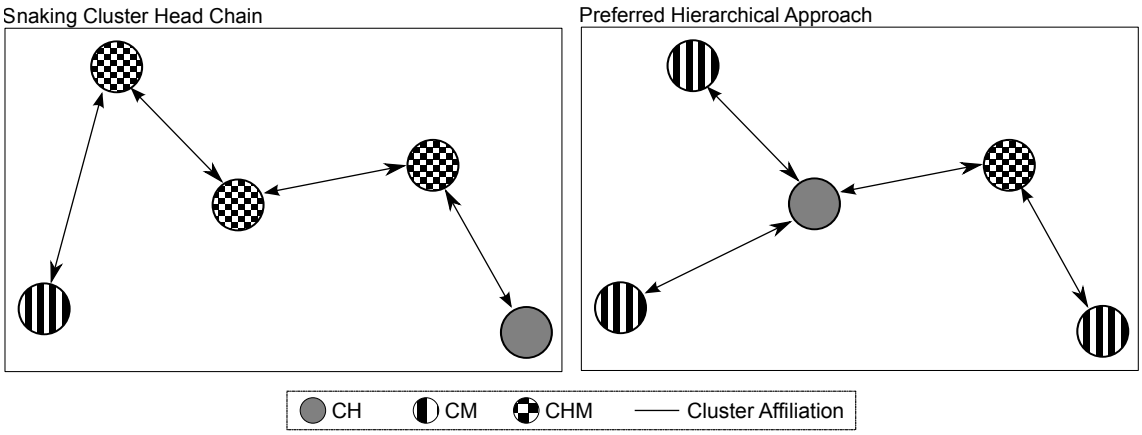


Figure 2.3: An illustration of the Snaking Cluster Head Chain. The Precedence approach may result in chains of subclusters, in which cluster heads are closely grouped together, as seen on the left. The boundary nodes are four hops apart. On the right is shown a master cluster head, one of whose members is a head of a subcluster. In this scenario, the boundary nodes are only three hops apart. A purely selfish method may not result in the best clustering structure.

on a narrow road, it can result in snaking structures with several cluster heads within a single transmission range. When such an algorithm is used to support VANET routing, boundary nodes can be several more hops apart than necessary, increasing the hop-count and reducing data throughput.

Timer

A timer-based approach has nodes remaining in an unclustered state for a set period of time, waiting for a cluster head to announce itself. If a node detects the presence of a cluster head within the timeout period, it affiliates with it; otherwise, if the timer expires, the node declares itself a cluster head. The waiting period may be set statically (e.g. [46, 81]) or randomly (e.g. [65]). A potential cluster head may be further evaluated by nodes that overhear its broadcasts to determine its suitability as their cluster head [72]. For example, a node may affiliate with a cluster head if the RSS of the received broadcasts exceeds a set threshold. In [74], relative velocity is also used to evaluate a candidate cluster head, and link expiration time is used in [54]. The principal advantages are the speed and simplicity of the algorithm; a neighbourhood discovery phase is not required to select cluster heads, and the channel bandwidth will only be consumed by cluster head announcement broadcasts.

However, in networks with regions of high node density, this election scheme can fall victim to the well-known hidden node problem. Multiple nodes can reach the

end of their timers simultaneously or near-simultaneously, and broadcast at the same time. Thus, nodes within range of multiple candidate cluster heads will experience packet collisions and their ability to detect nearby cluster heads within their own time period will be impaired.

2.3.2 Passive Clustering

Weighted-metric, precedence and timer-based clustering techniques have the common trait of being *active* approaches. That is, they use a common channel for cluster formation and maintenance together with other network traffic. This can cause contention between clustering traffic and routing traffic, as both compete for finite channel bandwidth, even when clustering is used to facilitate routing.

First proposed in [50], PC addresses this problem by piggy-backing cluster control data on outgoing routing traffic, eliminating the need for separate broadcast phases for clustering. The authors build their approach on AODV [82], utilising a "first declaration wins" approach for cluster head selection. Because the architecture waits for data to be sent from the upper layers, vehicles running active data-generating services are more likely to be chosen as cluster heads. PC was compared with LID-assisted AODV. PC significantly improves packet delivery ratio while reducing end-to-end delay for a given node density (or conversely, maintaining high throughput with low delay for higher node densities) compared to LID-assisted AODV.

VANET Passive Clustering (VPC) is an extension to PC proposed in [51]. The protocol is also built on top of AODV, but discards the "first declaration wins" election approach in favour of cluster head selection based on channel quality measurements, which results in better performance than PC.

The advantage of passive techniques is that there is no contention between clustering and routing traffic, allowing the former to assist in achieving the goal of the latter. An active clustering strategy can significantly interfere with the effectiveness of an ad hoc routing protocol due to the additional control overhead required for cluster formation and reformation. By contrast, passive approaches exploit the synergy between the two systems to achieve cluster formation while avoiding the need for extra traffic. However, existing network traffic is required in order to provide

a carriage service for the clustering data, therefore cluster formation will be inhibited by low levels of network traffic. In practice, network traffic is usually sporadic, meaning updates to neighbour data required for cluster formation and maintenance will also be sporadic; this makes the system more vulnerable to high node mobility. A hybrid (or dynamically-switched) active-passive system could potentially solve this problem, but to date there have been no proposals of this type.

2.3.3 Cluster Head Election Criteria

The criteria used to rank potential cluster heads are usually chosen with the objective of mitigating the adverse effects of mobility on communications reliability and throughput, the performance of some particular application, or both.

Early VANET clustering schemes were extensions of MANET algorithms, such as LID/HD [23], DMAC [29], or MOBIC [24]. Of these protocols, the former two used some unique node identification number (e.g., the MAC address) or the degree of the node (i.e. number of neighbours within range) as a cluster head election parameter, while the latter attempted to model the mobility of nodes by calculating the ratio of RSS measurements from consecutive broadcasts. Node ID is by far the least suitable metric for VANET clustering as it is entirely arbitrary and offers no way to account for mobility – it only guarantees that one node will have the lowest ID number, preventing a tie in the election process. The VANET channel is known to be subject to highly variable fading behaviour [12, 13, 83], meaning that a single pair of consecutive RSS measurements, upon which MOBIC's cluster head election mechanism is based, may give a quite unreliable estimate of the relative movement of nodes. The degree of connectivity provides a more useful election metric, as it allows nodes to infer which potential cluster head will be able to serve the largest number of vehicles. However, *quantity* does not necessarily imply *quality*, especially when the network topology changes frequently.

VANET research is slowly moving toward innovative cluster head election criteria, making clusters more robust to topology changes. Several recently-proposed strategies utilise node mobility parameters, signal quality indicators, link expiration time, platoon leadership, vehicle class, trustworthiness, and driver intention.

Mobility and Proximity

Relative mobility between a node and its potential cluster heads is a logical choice as a cluster head election metric, since nodes which are moving together (with small relative velocities) are more likely to be able to maintain a stable communications link. Similarly, potential cluster heads which are in close proximity to a node are more likely to be able to support a reliable link (at least in the short term) compared to more distant nodes due to the inverse square law of electromagnetic signal propagation. Therefore, many clustering algorithms use one or both of these metrics to rank potential cluster heads.

Proposed in [36] as an extension of MOBIC, ALM modifies MOBIC's election metric to utilise GPS location data distributed in *Hello* messages, rather than the limited-accuracy method of using consecutive RSS measurements.

RMAC [48] prefers potential cluster heads with more members, lower speed relative to the joining node, smaller distance to the joining node, and higher link expiration time. This strategy is quite successful in identifying cluster heads which are well-connected to a group of nodes moving in a coordinated fashion, resulting in large cluster sizes and long average affiliation times. The preference for nodes with good proximity to many similarly-behaving nodes is very important, since it increases the probability of successful packet reception.

Utility Function (UF) [40] uses a weighted sum of ID or degree of connectivity and either average relative velocity or average relative distance of neighbours. This algorithm performs better than plain LID and HD, providing longer lived and more stable clusters than either.

In PPC [39], the weight is calculated by hashing the system time and a node's ID, and XOR-ing the result with an eligibility function. The eligibility of a node is determined by travel time and the difference between the node's speed and the average speed of its neighbours. In this way, cars having a longer route and moving with the bulk of the vehicle group are regarded as better choices to be the cluster head. The hash function allows the assignment of unique weight values to prevent cluster head collisions.

Affinity Propagation in Vehicular Environments (APROVE) [62] is unique as it applies a statistical analysis method called *affinity propagation* to the clustering

problem. Vehicles are treated grouped according to proximity and velocity relative to neighbours. The node that is most representative of its neighbours' location and behaviour is elected cluster head. The algorithm is multi-pass and must be executed periodically, with the outcome of previous clustering phases retained in order to give preference to current cluster heads and prevent unnecessary reclustering. However, the algorithm is also asynchronous, meaning that received availability and responsibility messages will be at least one period old (i.e. will correspond to the previous iteration). The age of these messages may be even greater if a channel is lossy.

Quantitative comparisons of the performance of MOBIC, ALM, PPC and APROVE with more recent and VANET-specific clustering algorithms are provided in [84]. Performance under lossy channels was evaluated as well, using a uniform error model with increasing channel error probabilities. The study showed that MOBIC, ALM, and PPC's clusters are much shorter-lived with packet loss rates of greater than 10%, while APROVE's clusters lasted up to a minute with loss rates of up to 30%; reaffiliations were also much less frequent when compared against the competing approaches. APROVE's performance began to drop significantly beyond a loss rate of 40%.

A passive clustering algorithm called Cluster Formation for IVC (CF-IVC) is introduced in [52]. It uses a speed-based grouping approach to segment the network. Each vehicle knows the speed group to which it belongs, and affiliates with a CH in that grouping; this speed grouping then becomes part of the addressing scheme. A speed threshold is used to detect when a vehicle has changed speed groups and whether it should change clusters. A CDMA scheme assigns codes to cluster members to uniquely identify them and avoid packet collisions.

Threshold-Based Clustering (TBC) [41, 42] is specifically designed for highway mobility, in which the slowest vehicle amongst local neighbours initiates the cluster formation process. Neighbours within range of this node will affiliate with it if the relative velocity between them is less than a specified threshold. The cluster head election process is similar to HD, with consideration of proximity to mean position and velocity – the cluster head need not necessarily be the vehicle that initiated the clustering process. This results in clusters which tend to form in long trains.

A unique approach to cluster head election is taken by Spring Clustering (Sp-Cl),

which uses an election metric inspired by Coulomb's Law [37]. Cars are considered as point charges which exert forces on each other in proportion to the inverse square of inter-vehicular distance. In place of Coulomb's constant is a function of the relative mobility between a given vehicular node and each of its neighbours; the charge of each node is determined by the sign of the number returned by this function. The net force exerted on a vehicular node by each of its neighbours is then calculated, and the node with the most positive net force is elected as cluster head. The metric intrinsically reduces the score of a node whose neighbours are moving away from it, giving a higher preference for potential cluster heads around which other vehicles tend to group.

Signal Quality

The quality of a communications link is principally determined by the signal to noise ratio (SNR) of the received signal. If the noise floor is assumed to be a particular level or can be otherwise estimated, the received signal strength (RSS) is a reasonable proxy for direct measurements of the SNR. An estimate of RSS is available upon the receipt of any packet at a given node, while noise levels may either be measured during gaps in transmission, inferred from bit error rate and signal transmission parameters, or assumed to be constant (as is the case with many off-the-shelf 802.11 transceivers). Due to the time-varying nature of the channel (with fast and deep fading due to signal reflections from buildings and other vehicles) and transient interference from other localised sources of electromagnetic radiation, RSS or SNR measurements may vary significantly from packet to packet; therefore it is often necessary to perform some sort of low-pass filtering on SNR and RSS metrics to avoid undesirable thrashing of the network topology.

If relative velocity and/or position are also known, these can be combined with RSS or SNR measurements to compute an overall cluster head election metric, with the relative weights adjusted as desired. Density Based Clustering (DBC) [32, 33] is one algorithm that takes this approach. It incorporates SNR measurements into a weighted sum of Euclidean distance and relative velocity, in assessment of the link quality between nodes. This mitigates the aforementioned difficulties with signal quality measurements. SNR may also be used in initial assessment of cluster head

suitability, such as in the Cluster Configuration Protocol (CCP) [72, 73], which requires that a node accepts a cluster head advertisement message only if the RSS exceeds a threshold. This ensures cluster head affiliation will only occur with a sufficiently stable connection. Stability Based Clustering Algorithm (SBCA) [74] extends CCP, complementing the RSS metric with relative velocity to improve cluster head lifetime. Channel metrics other than SNR or RSS may be used. K-hop clustering [35] uses propagation delay in the calculation of the cluster head election weight. Changes in propagation delay indicate that a potential cluster head is either moving towards or away from a node.

VPC [51] discards the "first declaration wins" election approach used by PC in favour of vehicular mobility metrics. The authors tested their approach using three different metrics at both low and high vehicle speeds: degree, link quality, and link expiration time. It was found that using degree as the election metric allowed the highest packet delivery ratios when node density was low; however, using the link quality metric resulted in the best PDR at higher node densities. A similar pattern was observed at both low and high vehicle speeds. In all cases, application of the vehicular mobility metrics resulted in an improvement compared to the original PC algorithm. However, since the choice of metric which resulted in the maximum improvement changed depending on vehicle density, overall performance could be further improved by adding context-sensitivity, such that a different set of metrics is selected depending on the current vehicular traffic density.

Link Expiration Time

Modified Distributed Mobility Aware Clustering (MDMAC) [31] extends the LID/HD ranking scheme used by DMAC [29] through the addition of an estimate of expected link lifespan or *link expiration time* between a cluster-head-seeking node and a potential cluster head. To be considered for cluster head status, a node's cluster head suitability score must be better than the current cluster head, and the link expiration time must exceed a specified threshold. This prevents fast-moving vehicles from taking over the cluster head role before disappearing and causing another reclustering event. It also prevents vehicles travelling in the opposite direction from momentarily taking over the role of cluster head. Link expiration time is estimated

based on relative position and velocity of the node and the potential cluster head, and an assumed maximum communication range. Similarly, Clustering Protocol for Topology Discovery (CPTD) [77] uses an estimate of expected link lifespan to avoid excessive reclustering with temporarily local cluster heads.

In DBA-MAC [54], a vehicle waits a set time before declaring itself a Backbone Member (BM) and broadcasting a *Hello* message containing its mobility metrics. Vehicles overhearing this declaration before their own timers expire then compute the expected period for which they expect to be within range of the first-to-declare BM node. If this period exceeds a set threshold, the node calculates and waits for a randomly-generated contention window, at the end of which it transmits a candidature frame to the BM. The BM declares the first node to transmit a candidature frame to be the next BM node in the chain, and the process repeats until all nodes have joined a chain.

In MCTC [45], the expected connection lifetime between nodes i and j is calculated using a different method to the previously discussed protocols; the difference between the assumed communication range and the inter-node distance is divided by the difference in speed between nodes i and j. If node i is travelling at a greater absolute velocity than node j, then j is ignored in the calculation. This mechanism ensures that nodes are properly separated into groups based on node velocity. Nodes with the greatest link expiration time are elected cluster head.

Platoon Leadership

Another clustering metric uniquely available to VANETs is the position of the node along the road relative to its neighbours. Clustering based on Direction in Vehicular Environments (C-DRIVE) [57–59] elects the front-most car in a platoon as the cluster head, with cluster formation events triggered when traffic is stopped at intersections. Additionally, vehicles attempt to affiliate with cluster heads that intend to turn in the same direction as themselves (See Section 2.3.3).

C-RACCA [76] checks whether cluster head announcements come from behind or in front, as well as whether the distance between it and the responding node exceeds a threshold calculated as a function of difference in velocity and deceleration capability. It then assesses whether it should join the responding cluster or assume

the role of cluster head itself.

Vehicle Class

The concept of selecting a cluster head based on vehicle class is considered with Mobile Infrastructure in VANET (MI-VANET) [49]. In this architecture, private cars never act as cluster heads – this function is performed only by buses. Cars then affiliate with the nearest bus they can locate based on *Hello* received from buses. This algorithm is unique in that there is no distributed election system; therefore, the architecture of MI-VANET shares some characteristics with a traditional infrastructure network, albeit one with mobile access points.

Sp-Cl is extended to differentiate nodes using a height-based metric in [38]. In the resultant Enhanced Spring Clustering (E-Sp-Cl) protocol, taller vehicles are assigned a higher pseudocharge value than shorter ones, ensuring that the tallest vehicle in the vicinity would be elected cluster head. The taller vehicle can then route packets between nodes whose Line-of-Sight path they may otherwise obstruct. Using a shadowing model to account for vehicular obstructions, the authors compared the performance of E-Sp-Cl to Sp-Cl, and showed that performance improved as a result of the increased network coverage resulting from its preference for taller cluster heads. This approach implies cluster head election based on vehicle class, as buses and trucks are taller than cars.

Driver Intention

In [66], an election metric which extends the approach used by UF is presented, called Lane-Sense Utility Function (LSUF). It aims to improve cluster lifespan by selecting a cluster head from a set of vehicles travelling in the same direction, which are expected to continue to take the same path at the next intersection. The overall cluster head election score for a candidate cluster head is a function of relative position and velocity, number of connected neighbours, and the lane the vehicle is currently occupying. It is assumed that the lane determines whether the vehicle will turn left, right, or continue straight at the next intersection, and a weight is assigned to each traffic flow based on the number of lanes belonging to each flow. LSUF functions better than LID, HD, and UF in creating stable clusters on

a highway scenario with multiple highway exits. LSUF is used to construct clusters in a TDMA Cluster-based MAC (TC-MAC) [67, 68].

User Oriented Fuzzy-logic-based Clustering (UOFC) [44] uses a fuzzy classifier to combine vehicular velocity and position information with a measure of driver intention, with the latter inferred by observing passenger interest in certain goods and services accessed via an in-car entertainment system. Vehicles which appear to have a common likely intended destination are then clustered together. The authors' simulation results indicate that clusters last significantly longer than with regular UF, and are more stable. This study also considers the number of lanes and vehicle speed in analysis, with results showing the emergence of fewer clusters with higher stability on roads with more lanes. While an interesting and unique approach to the problem of predicting vehicle behaviour, most passengers would now tend to use their own mobile devices, limiting the amount of information from which these inferences can be made. The extent to which passenger interaction with in-car entertainment systems accurately reflects driver intention is also highly debateable; for many journeys, passengers (and children in particular) may be more interested in consuming media unrelated to their intended destination.

Fuzzy-Logic-Based Algorithm (FLBA) [43] uses relative velocity between vehicles as a cluster head election metric. As in UOFC, a fuzzy-logic algorithm assesses driver intention in order to detect long-term changes in cluster head eligibility.

AMACAD [60] computes an election metric as a weighted sum of inter-node distance, relative velocity, and Euclidean distance between vehicle destinations. This allows the algorithm to cluster vehicles which are taking similar routes.

Cellular Automata Clustering (CAC) [55] groups vehicles together according to the level of interest in nearby services, as indicated by the driver. This forms interest groups that can be used to route data relevant to those interests between the vehicles. Each interest has a channel allotted to it, with an additional channel set aside for emergency data. Unfortunately, little information is presented on how the cluster head is elected or how neighbourhood data is collected.

Security Reputation

Security and trust in VANETs is essential in order to prevent malicious agents sabotaging road safety systems built upon the VANET framework, potentially causing serious disruption to traffic flows or safety hazards. Several authors have proposed cluster head metrics which can assist in identifying malicious vehicles and mitigating their impact by denying them access to cluster resources.

VWCA [69] computes a cluster head selection metric based on vehicle direction, degree of connectivity, an entropy value calculated from the mobility of nodes in the network, and a *distrust level* based on the reliability of a node's packet relaying. Vehicles are assigned *verifiers*, which are neighbours with a lower distrust value. Verifiers monitor the network behaviour of a vehicle, and confirm whether it is routing packets and advertising mobility and traffic information that is consistent with the verifier's own view of the neighbourhood. The distrust value for nodes which behave abnormally is then automatically increased, while it is decreased for nodes which perform reliably. In this way, the trustworthiness of a node is accounted for by the cluster head election process.

Trust-dependant Ant Colony Routing (TACR) [53] supplements position and velocity with Certificate Authority trust metrics to compute a weighted cluster head selection metric. These metrics include *traffic obedience* and *reputation* for correctly forwarding data packets. This has a dual purpose: it firstly ensures that the cluster head is capable of reliably functioning in this role, and secondly creates a layer of security around clusters, preventing malicious vehicles from joining a cluster and becoming cluster head.

Cluster-Based Public Key Infrastructure (CBPKI) [70] elects the most trustworthy vehicle in a neighbourhood as its cluster head. Affiliation is handled by a cluster head's 1-hop neighbours. The authors called this neighbourhood the VANET Dynamic Demilitarised Zone (VDDZ), and intended it to prevent malicious vehicles from accessing the cluster head directly. A cluster member can only directly communicate with the cluster head when its trust value exceeds a set threshold.

Other Criteria

PC's [50] "first declaration wins" approach is unique in that nodes with data to transmit are more likely to become cluster heads; PC's cluster head election metric is therefore dependant on the behaviour of the application layer. Various factors can affect the behaviour of the application layer, such as driver intention and passenger interests. However, this does not necessarily imply the node will be able to perform the cluster head role adequately.

CCA [34] defines a metric denoted *network criticality* as "the normalised random walk betweeness of a node on the network" (sic). That is, how often a node is visited when traversing the network from a source to destination. Through experimental analysis, local network criticality is determined using the MP-inverse of the Laplacian Matrix of local network topology. The election criteria is then the average criticality between a node and its neighbours. Using this metric for cluster head selection ensures that a cluster head is chosen through which most nodes will naturally have to forward packets in order to deliver them to the intended network destination; essentially this approach identifies natural high-level packet relays in a dynamic hierarchical network infrastructure.

MCA-VANET [56] utilises an unusual metric for cluster head selection. Rather than vehicles joining the network in an initially unclustered state and then electing a cluster head, all nodes start in the head role, and subsequently decide whether to step back to the role of a normal member, while selecting another node as its cluster head. The decision is based on an 8-bit counter maintained by each node for each of its neighbours. The counter is incremented for every *Hello* message received from the neighbour per beaconing period, and halved every time a period expires without reception of a message. As a result, nodes decide the eligibility of their neighbours to be cluster heads independently from other nodes, with mutual eligibility slowly increasing between cars that remain close for a sufficient period of time, then rapidly decreasing as beacons are missed, indicating that the nodes are no longer in proximity. This method is unique in that it does not need GPS data (which can be inaccurate in built-up areas in any case), but instead estimates proximity from a connectivity standpoint. Additionally, nodes can have access to multiple cluster heads as a form of route redundancy. A node backs down to cluster

member status if there are more than $numCH$ cluster head nodes with an eligibility level above a set threshold.

2.3.4 Cluster Member Affiliation Strategy

Once a node identifies its preferred cluster head, it will send a join request message to the prospective cluster head (unless it has decided to take on the role itself). When it receives the message, if the cluster head accepts the request, it will insert the requesting node's identification token (e.g. its MAC address) into the cluster table. Most algorithms surveyed will stop there, while others require a confirmation frame to be returned to the cluster member to inform it of success (or failure). This *handshaking* is an important distinction between algorithms, and it results in a number of specific advantages and disadvantages, which are discussed below.

Cluster Control Capability

Using handshaking, a cluster head can exercise control on the topology and composition of the cluster by deciding whether or not to admit a prospective cluster member. In RMAC [48], a cluster head will deny access to a cluster if it has reached a predefined maximum size. This method would be particularly useful in channel access schemes such as CBMAC [46] and TC-MAC [67], where the number of channel slots limits the number of members that a cluster head can serve. In AMACAD [60], the handshake allows a cluster head time to calculate the new member's suitability to be a new head, and transfer leadership to the new node if neccessary (see Section 2.3.7).

Handshaking also allows the development of multi-state affiliation strategies. Nodes may enter an intermediate cluster membership state, allowing them to temporarily join clusters in order to gather more information about the environment before deciding whether or not to join. This approach is used in DBC [32, 33] and CCA [34]. Through this 'trial membership' period, the cluster head has the opportunity to acquire more information about a cluster member, such as obtaining statistics for routing reliability and trust metrics by observing its behaviour in the cluster over time. This approach is used in CBPKI [70].

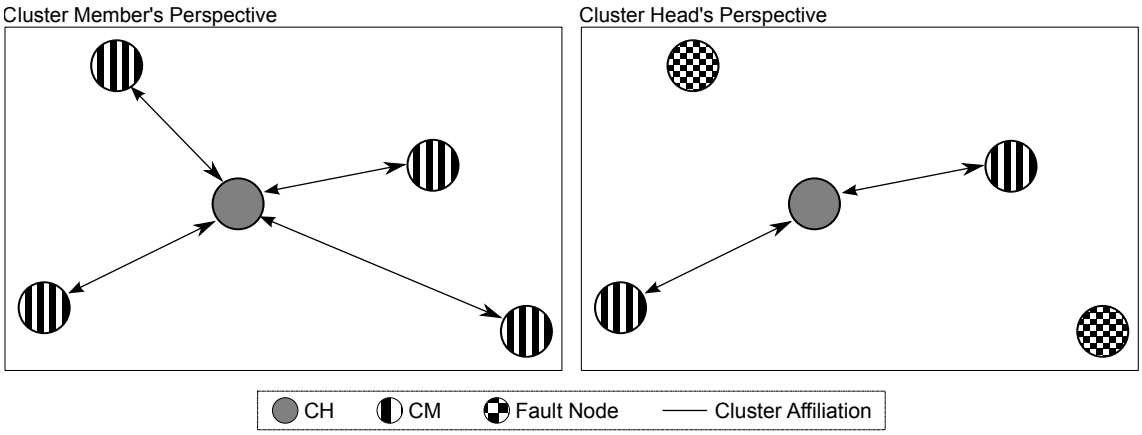


Figure 2.4: An illustration of the faulty affiliation malfunction. Join messages lost due to packet collisions or fading cannot be detected by an affiliating node, unless the join requires an acknowledgement on the part of the cluster head. In some cases, fault nodes become isolated from the network, as broadcast-based maintenance mechanisms prevent the fault node from disconnecting and seeking out other clusters to join. This makes algorithms such as MDMAC [31] unsuitable for the various envisioned applications of clustering.

Improved Integrity

The VANET channel has a high potential for packet loss due to vehicular obstructions and multi-path fading. An affiliation message can be lost or dropped by the cluster head, and without confirmation requirement a cluster member will erroneously believe that its affiliation was successful. This is called a *faulty affiliation*, and is a serious problem for clustering algorithms intended for use in routing and security applications. An instance of faulty affiliation is shown in Figure 2.4. Beacon-based maintenance mechanisms (see Section 2.3.5) isolate faulty nodes from the network, as the node may continue to hear a cluster head's signal for up to a minute, during which it will not restart the clustering process. If such an algorithm were used as a foundation for data dissemination or routing, a faulty affiliation will isolate potential members from the network and they will be unable to disseminate or receive data. Therefore, this method is not well-suited for real-time traffic collision warning applications in which a CH distributes collision warnings. Fault nodes will not receive the warning data, as the CH is unaware of them, and appropriate avoidance action will not be taken.

Increased Overhead

The disadvantage of protocols with handshaking is the increased data overhead, which consumes some fraction of bandwidth. As shown from the results in the analysis of PC [50], clustering overhead can interfere with both routing control traffic and other data traffic, and the use of handshaking will cause even greater reduction in performance. A passive clustering approach in which handshakes are piggybacked onto existing traffic could potentially solve this problem, provided that a sufficient level of background traffic exists in the network.

2.3.5 Neighbourhood Discovery

There are two neighbourhood discovery mechanisms used in VANET cluster design. *Hello* messages are periodic beacons that are either broadcast by all nodes, as in MDMAC [31], or only by those that have declared themselves to be a cluster head, as with timer based approaches such as CBLR [81]. The alternative is *Inquiry* messages, which are polls of the local network topology – a node sends out an *Inquiry* frame, and waits for responses from nearby nodes. Thus neighbour discovery is either proactive (via *Hello* broadcasts) or reactive (via *Inquiry* requests).

Hello Messages

Hello messages are a simple method for local topology discovery, with the added benefit that missed beacons may be used to detect failed or failing connections to cluster members. However, if a multi-hop algorithm forwards *Hello* messages it receives – as with MDMAC [31] – these broadcasts consume a non-trivial fraction of available bandwidth in high node density scenarios, while increasing collision rate due to increased medium contention. This reduces the reliability and speed of information dissemination, which may compromise vehicle safety applications. Furthermore, the use of *Hello* messages without handshaking for affiliation can result in isolated nodes suffering from faulty affiliation.

Inquiry Messages

These are broadcast by nodes as data solicitation messages. Nodes that receive an *Inquiry* message will respond with a unicast acknowledgement that contains their data. The transmission of *Inquiry* messages may only be needed when a node enters an unclustered state, as in RMAC [48]; or it may be periodic, as in DBC [32] and CBLR [80]. Algorithms such as RMAC, which only perform the *Inquiry* phase once, can then update their neighbour table opportunistically by receiving other *Inquiry* frames, as well as topology updates from their cluster head.

Inquiry frames allow neighbour information to be disseminated only on demand, without risking a broadcast storm. Response frames can incur a slightly larger overhead than *Hello*. Nodes will send out *Inquiry* frames while responding to other nodes. By contrast, *Hello* messages are advertisements of a node's own information, meaning nodes need only store the data from messages they receive without sending their own data in unicast form.

2.3.6 Gateway Election Metrics

Several of the algorithms discussed in this paper use clustering to assist with routing or integration between cellular networks and VANETs. These algorithms identify a member which is best suited to act as a *gateway* to other clusters (not necessarily the cluster head); the specific election criteria for each algorithm are listed in Table 2.3. The criteria tend to be application specific. Routing algorithms employ cluster membership as a requirement for gateway status, while algorithms targeted at security-sensitive applications additionally require gateway nodes to have a sufficient trust level to guard against attack from malicious nodes.

2.3.7 Cluster Maintenance

After a cluster has been formed, the cluster head is responsible for coordinating communication between members while maintaining a stable cluster topology. As identified in [15], road network structure and vehicle density both have a strong influence on cluster stability. In order to adequately maintain the stability of a cluster, the cluster head must actively maintain a database of its members and

Algorithm	Gateway Election Metric(s)
CBLR	Cluster Membership
CBMAC	Cluster Membership
CMGM	3G RSSI
CSBP	Randomised Timeout
CF-IVC	Speed Grouping, RSS w.r.t cluster head
MCTC	Relative Velocity w.r.t cluster head
CBPKI	Trust Level w.r.t both cluster heads

Table 2.3: Surveyed Gateway Election Metrics

respond in a timely way to changes in membership. Two major database update mechanisms are described in the literature; these, along with potential causes of instability and proposed mitigation strategies are presented in the following sections.

Cluster Member Data Updates

Two main mechanisms for updating cluster membership information are described in the literature. The simplest approach is to utilise the *Hello* messages sent by all nodes for neighbour discovery, although this approach is subject to the disadvantages previously discussed in Section 2.3.5. MDMAC [31] and AMACAD [60] employ the *Hello* broadcast in their cluster membership update mechanisms.

An alternative approach is to periodically poll the cluster. A cluster head sends out a broadcast, containing cluster-wide information for all cluster members (e.g. mobility metrics for all nodes). Each member then then responds with a unicast acknowledgement containing its own current state information, such that the cluster head's database can be updated (if necessary) for the next broadcast. Thus members can obtain up-to-date information about their cluster head (and potentially about other nodes in the cluster which are not all necessarily within 1-hop range). Polling also provides another layer of protection against nearby candidate cluster heads that may temporarily have a higher cluster head selection metric. RMAC [48] and DBC [32] are examples of protocols with a poll-based data update mechanism.

Polling shares some traits with *Inquiry* mechanisms, including higher control

overhead. It also makes a cluster vulnerable to transient channel drop-outs (e.g. due to fading) and packet collisions. If a cluster member does not respond to a poll, the cluster head assumes that cluster member has left; similarly, if a cluster member does not hear a poll request from its cluster head within the expected time period, it considers the connection to the cluster head to be dead. It is possible that both nodes are still within range of each other, and either a poll or the corresponding response was lost due to collisions or transmission errors. This results in unnecessary an reaffiliation attempt and decreases the stability of the cluster.

Cluster Head Depreciation

As the cluster members move relative to one another due to normal traffic behaviour, the cluster head selection metrics naturally vary. A cluster head eventually becomes a sub-optimal choice for the role, as it may be unable to communicate with all of its members, leading to the collapse of the cluster structure. In response to this, a cluster head can intentionally disband the cluster, enabling its members to find a new cluster head in a controlled manner. C-DRIVE [58] actively detects when a cluster is approaching an intersection and forces it to disband. This results in a reclustering event which may result in one of the ordinary members of the previous cluster becoming a new cluster head. Alternatively, a secondary or alternative cluster head, which can assume the role of cluster head when required, can be explicitly nominated by an existing cluster head. The secondary cluster head is either chosen on demand (i.e. in response to a decision by the current cluster head to relinquish its role, after which a handoff procedure is initiated and the secondary becomes the new primary), or a standby secondary cluster head is selected immediately after the election of the primary cluster head, potentially changing as decided by the cluster head based on its knowledge of the cluster membership and topology. Five of the surveyed algorithms employ a secondary cluster head handoff mechanism.

AMACAD [60] achieves this by evaluating new cluster member affiliations, determining the cluster head selection metric for the new node with respect to the members of the current cluster. The cluster head decides to relinquish its role if the newly affiliated cluster member has a better selection score. FLBA [43] uses the second-best cluster head candidate as a backup (secondary) cluster head. C-DRIVE

[57–59] passes the cluster head role to overtaking vehicles.

C-RACCA [76] decides whether a node should take over as cluster head based on the inter-vehicular distance between them. The distance between the cluster head and the new cluster member must be such that, in the event of an accident, there is sufficient time for the following vehicle to brake and avoid collision with the vehicle ahead. If there is not, the cluster head role is transferred to the new member.

SBCA [74] selects secondary cluster heads based on a mobility calculation similar to its cluster head election metric, except that it is performed only with respect to the current cluster head. The implication is that a node nearest to the current cluster head would be a good candidate to take the role. However, this does not account for the possibility that the current cluster head has already become a sub-optimal choice, and thus a node with similar mobility patterns would also be equally inappropriate.

Merging Clusters

Due to node and cluster mobility, it is likely that at some point two cluster heads will pass within range of each other. Most clustering algorithms try to form a small number of large clusters; hence when two clusters approach one another, they may merge to form a single large cluster. However, for the case of two clusters with different group velocities (for example, one group of vehicles overtaking another group on a highway), the merger will only be temporary as the faster-moving nodes will subsequently move out of range. This will result in a partial cluster collapse, with both groups of nodes potentially needing to recluster.

To avoid this problem, some algorithms take measures to prevent unnecessary cluster absorption. GDMAC [30] employs two additional checks in its election mechanism, such that members will not reaffiliate unless the new candidate's cluster head selection metric exceeds that of the current cluster head by a set threshold. ALM [36] avoids reclustering when dealing with nearby cluster heads by means of a contention time-out, requiring that the heads be within range for a certain minimum period of time before merging. MDMAC [31] uses link expiration time to excise high-speed cluster head candidates, which also avoids the merging of clusters with significantly different group velocities.

Hierarchical clustering algorithms actively try to merge adjacent clusters as an intended part of their design, with the smaller cluster becoming a sub-cluster of the other. As a result, a backbone or hierarchy is formed, which can then be used for routing and data dissemination protocols. Such algorithms extend the network coverage at the expense of increased overhead, and must avoid the formation cyclical structures and backbones consisting only of cluster heads. RMAC [48], DBA-MAC [54], and CBLR [80, 81] employ this approach.

2.4 Verification Methodology

Rigorous experimental evaluation of proposed VANET protocols and applications in urban environments requires the deployment of hundreds or thousands of wireless vehicular nodes. Even if a research program is able to recruit enough volunteer drivers, the cost and time expenditures become highly prohibitive beyond a few tens of nodes, severely limiting the value of experimental results. For this reason, VANET research is heavily reliant on software-based simulations for protocol and application development and evaluation. The principal advantage of such methods is that a wide range of proposed protocols can be evaluated under identical or equivalent channel and vehicular mobility conditions, with arbitrary vehicle distributions, vehicle behaviour and network traffic loads. As the simulation is not constrained to run in real time, the only limiting factors are the execution time, processing and storage/memory requirements. Despite these limits, very large simulations can be run in a reasonable period of time with a modest amount of commodity computing hardware, with the additional benefit of being able to quickly and easily modify the simulated environment.

Both the choice of channel model and the choice of simulation framework itself have a significant impact on the accuracy of the results of simulations. Several alternatives for each of these parameters are discussed in the following sections.

2.4.1 Channel Models

The method by which the wireless signal propagation environment is modelled has a significant impact on the performance of the protocol or application under evalu-

Simulator	Citation
ns-2	[85]
Unspecified	N/A
Custom Testbed	N/A
JiST/SWANS++	[86]
ns-3	[87]
MATLAB	[88]
VEINS	[89][90]
GloMoSim	[91]
Traffic Simulation 3.0	[92]
VANETMobiSim	[93]
NCTUns	[94]
SIDE/SMURPH	[95]
OPNET	[96]

Table 2.4: List of simulators and their citations.

Table 2.5: Assessment of validation methodologies.

Algorithm	**Simulator**	**Channel Model**	**Comparison**	**Based upon**
DMAC	N/S	N/S	N/S	LID/HD
GDMAC	ns-2	Static R_{TX}	DMAC	DMAC
MDMAC	JiST/SWANS++	Log-Normal	DMAC	DMAC
DBC	JiST/SWANS++	Log-Normal	HD	MDMAC
CCA	Unspecified	Unspecified	MDMAC	MDMAC
CBLR	OPNET	Unspecified	Self	Original
CBMAC	Custom	Custom	Self	CBLR

Table 2.5 – *continued from previous page*

Algorithm	**Simulator**	**Channel Model**	**Comparison**	**Based upon**
RMAC	ns-2	Static R_{TX}	DMAC	Original
HCA	VEINS	Static R_{TX}	k-ConID	Original
C-DRIVE	NCTUns	Static R_{TX}	Self	Original
K-hop	ns-2	Two-Ray	Self	MOBIC
ALM	SIDE/SMURPH	FSPL	GDMAC	MOBIC
C-RACCA	ns-2	Two-Ray	Self	Original
CMGM	ns-2	Two-Ray	Self	C-RACCA
MI-VANET	ns-2	Static R_{TX}	Self	Original
Sp-Cl	Unspecified	Unspecified	LID	Original
E-Sp-Cl	Custom	FSPL Shadowing	Sp-Cl	Sp-Cl
PPC	ns-2	Static R_{TX}	LID, HD	Original
AMACAD	Custom	Static R_{TX}	LID, MOBIC, DGMA, MobHid	Original
UF	Traffic Simulation 3.0	Unspecified	LID, HD	LID/HD
CSBP	JiST/SWANS++	Unspecified	Self	UF
TC-MAC	ns-3	Static R_{TX}	LID, HD, UF	UF

Table 2.5 – *continued from previous page*

Algorithm	**Simulator**	**Channel Model**	**Comparison**	**Based upon**
APROVE	ns-2	Static R_{TX}	MOBIC, ALM, PPC	Original
CCP	Unspecified	Unspecified	Self	Original
SBCA	ns-2	Static R_{TX}	CCP	CCP
TBC	Custom	Unspecified	CBMAC, PPC	Original
PC	GloMoSim	Static R_{TX}	LID	Original
VPC	ns-2	Static R_{TX}	PC	PC
CF-IVC	N/S	N/S	N/S	PC
FBLA	ns-2	Nakagami	APROVE, CCP	Original
UOFC	ns-2	Unspecified	LID, UF	UF
VWCA	MATLAB	Static R_{TX}	LID, HD, WCA	WCA
TACR	Unspecified	Unspecified	Self	Original
DBA-MAC	ns-2	Static R_{TX}	Self	Original
CPTD	ns-2	Static R_{TX}	Self	Original
MCC-MAC	MATLAB	Unspecified	Self	Original
MCTC	MATLAB	Unspecified	HD	Original
CBPKI	VEINS	Static R_{TX}	Self	Original

Table 2.5 – *continued from previous page*

Algorithm	**Simulator**	**Channel Model**	**Comparison**	**Based upon**
CAC	Unspecified	Unspecified	Self	Original
ALCA	VANET MobiSim	Static R_{TX}	Self	Original
MCA-VANET	ns-3	FSPL fading	MOBIC	Original

ation [15]. Numerous authors (e.g. [10–12]) have previously demonstrated the need for accurate modelling of channel dynamics for meaningful evaluation of the performance of VANET protocols and applications. A model that accurately accounts for many known physical channel properties (such as path loss, fading and shadowing) will provide a much better indication of the likely practically achievable performance of a proposed protocol or application; however, despite the known sensitivity of VANETs to channel properties, much of the literature assumes an ideal channel model such as free-space or two-ray path loss, or simple Rayleigh or Rician fading channels. These models do not provide a realistic representation of the characteristics of signal propagation, particularly in urban environments. This can result in misleading relative performance estimates between protocols under test, and makes direct comparison of results from different publications almost impossible. The use of ideal channel models persists despite on-going research and development of more realistic and sophisticated urban channel models [10, 11, 13].

Seven distinct wireless channel models are currently in use in the VANET literature, and thirteen simulators (including very general-purpose platforms such as Matlab). The simulators are listed in Table 2.4. They have been arranged in order of popularity with respect to the papers surveyed in Sections 2.2 and 2.3. A table listing the distribution of the reviewed algorithms between these simulators and models is shown in Table 2.5. The channel models are presented below, listed in order of popularity (as measured by the number of published clustering protocols using that model for performance evaluation).

1. *Static R_{TX}*: A maximum transmission range has been specified such that nodes separated by more than this distance are unable to communicate;

2. *Unspecified*: The authors have not given any details on the channel model;

3. *Two-Ray*: A pathloss model more sophisticated than FSPL, modelling the direct path and a single reflection from the ground;

4. *Log-Normal*: A shadowing/fading model based on the log-normal distribution;

5. *Nakagami-M*: A well-known fading model for communication in urban environments, which assumes one transceiver is at a higher elevation than the others – this model is best suited to cellular networks;

6. *Friis Free-Space Path Loss*: The simplest form of deterministic channel model, with only a direct path between source and destination. Some published algorithms are evaluated using the Friis channel model with statistical extensions to represent fading and/or shadowing; and

7. *Custom Models*: The authors have developed their own channel model, giving a brief description of its principles.

2.4.2 Evaluation of Simulators and Channel Models

Mainstream Models

Among the papers in this survey, the most popular simulator is ns-2 [85]. A review of the source code for ns-2 shows that it functions based on estimated power measurements and drops packets under the condition that the calculated received power is less than a specified threshold. As of this writing, the current stable release of ns-2 is version 2.35, and the basic package offers support for five channel models: Friis Free-Space Path Loss, Two-Ray Ground, Nakagami-M, Log-Normal Shadowing, and Obstacle-based Shadowing. ns-3 [87] also provides these models. Analysis of the code reveals Obstacle-Based Shadowing uses a monochrome bitmap of the simulation environment to determine if there are static objects blocking line-of-sight transmission. Two different configurations to a Log-Normal Shadowing distribution are used based on whether there is a line-of-sight or not. This model follows in

the same line as CORNER [10, 97], attempting to model the channel with environmental considerations. However, in all the cited works using ns-2, none use these more advanced models, choosing to rely only on static transmission ranges, Two-Ray, or Nakagami-M models. It should also be noted that ns2, and other simulators implementing models such as log-normal shadowing, update their coefficients on a per-frame basis. These shadowing variations are more frequent than would happen in real environments, which essentially overestimates the influence of shadowing in the simulation.

The authors of [64, 65] use a simulator called VEINS [90]. This combines the OMNeT++ discrete event simulator [89] with the SUMO traffic simulator [98] via the TraCI communication protocol [99]. As OMNeT++ is not a network simulator but a general purpose simulation framework, the authors combine the MiXiM wireless modelling API [100] to simulate wireless networks. The result is a dedicated bi-directional VANET simulator. The MiXiM framework offers FSPL, Two-Ray, Log-Normal Shadowing, Rayleigh Fading, and Simple Obstacle Shadowing. The authors of [64, 65] claim to incorporate topological data into their channel model, using a different model for different programmed scenarios. A similar model is applied in [15].

NCTUns [94, 101] was used in two papers citing a static R_{TX} value [58, 59]. In [94], the authors describe the application of NCTUns to VANET research, specifying the approach to modelling vehicular mobility and application interfaces. On the other hand, [101] introduces the simulator in a general wireless modelling context. The simulator offers three modelling approaches: the simple R_{TX} channel model, Two-Ray Ground, and a Rayleigh fading model. The simulator source code is available, allowing the implementation of any desired channel model; however, if a different channel model has been implemented, none of the cited papers have given details.

GloMoSim [91] provides a similar suite of channel models to NCTUns and ns-2: FSPL, Log-Normal, and a fading model that incorporates shadowing. The citation of a static transmission range in [50] implies the use of FSPL or other deterministic path-loss model.

MATLAB [88] was used to assess the performance of the algorithm in [69]. The

communication parameters were stated to have been based off the DSRC standard, with a set of constant parameters, though the detail is minimal, and explanation of the phenomena incorporated into the model is lacking.

Custom Channel Models

The authors of [47] use a custom simulator described in [102]. The channel model is a static transmission range approach with support for collision detection and BER, described in the publication as an "easy signal propagation model, which considers reflections and diffractions at corners of buildings within cities" (sic). This appears to resemble CORNER [10, 97] in principle. However, little more explanation is given, and there are no citations to papers explaining the workings of the model in detail.

Unspecified Channel Model Citations

It must be noted, on the subject of reliability of results, that the second most popular channel model specification is *none*. That is, no information on the channel model used in algorithm analysis has been given in the paper. It is then difficult to determine reliability of the authors' results. However, of the reviewed papers lacking information on channel model configuration, ten identify the simulation framework in use, which allows some insight into which models may have been used.

The authors of [41] use a C++ testbed of the author's own design. However, no detail is given regarding the sophistication or phenomena modelled by the simulator.

The authors of [40] cite modification of Traffic Simulator [92], a Java-based Intelligent Driver Model. It is assumed that the authors wrote additions to this simulator to incorporate VANET communication, though details on the modifications have been omitted.

JiST/SWANS++ [86] was used to validate the algorithm presented in [28]. Like OMNeT++, JiST is a Java-based discrete event simulator. SWANS is a wireless network simulator built into JiST. It provides support for Rayleigh or Rician fading, FSPL, and Two-Ray propagation models. The fading and path-loss components can be used in tandem. Availability of source code means that researchers are able to integrate their own models into the simulator. It should be noted that, while fading

is optional, the simulator requires the researcher to specify a path-loss model.

2.4.3 Competing algorithms

Another element of analysis is the comparison with existing VANET clustering algorithms. It is appropriate, when presenting a new algorithm, to provide information on the performance gains of the new approach over an existing and published algorithm. Most of the papers cited include this comparison. Among those that don't (indicated in Table 2.5 as comparing to *Self*), several were compared to routing algorithms or to DSRC performance. As these architectures were designed with specific applications in mind (See Section 2.2) it is sensible to compare them to competing approaches to the same application.

Figure 2.5 shows the comparative relationships between the algorithms in this survey. The algorithm names have been ordered clockwise in order of publication year. The most pertinent trend is that much of the surveyed architectures are compared to MANET approaches. When VANET research rose out of MANET research, it made sense to use these approaches as comparative benchmarks in the early days. But MANET clustering algorithms have not been designed for the mobility characteristics of VANET scenarios. Despite the emergence of new algorithms properly suited to the challenges of VANETs, comparison is still continually made to MANET approaches. If clustering techniques were improving upon recent advancements, Figure 2.5 should show newer algorithms linked to the names of more recent proposals; instead, the diagram shows the opposite. Thus, when new algorithms are presented in the literature, improving performance in comparison with MANET approaches does not imply that performance improvements have been achieved over recent VANET approaches.

2.5 Discussion and Conclusion

Several surveys and taxonomies such as [16, 17] have expressed a need for exploiting more elements of the VANET environment toward the clustering problem. The suggested directions include wider application of learning techniques, use of bi-directional traffic flows in the formation process, and road-side unit involvement.

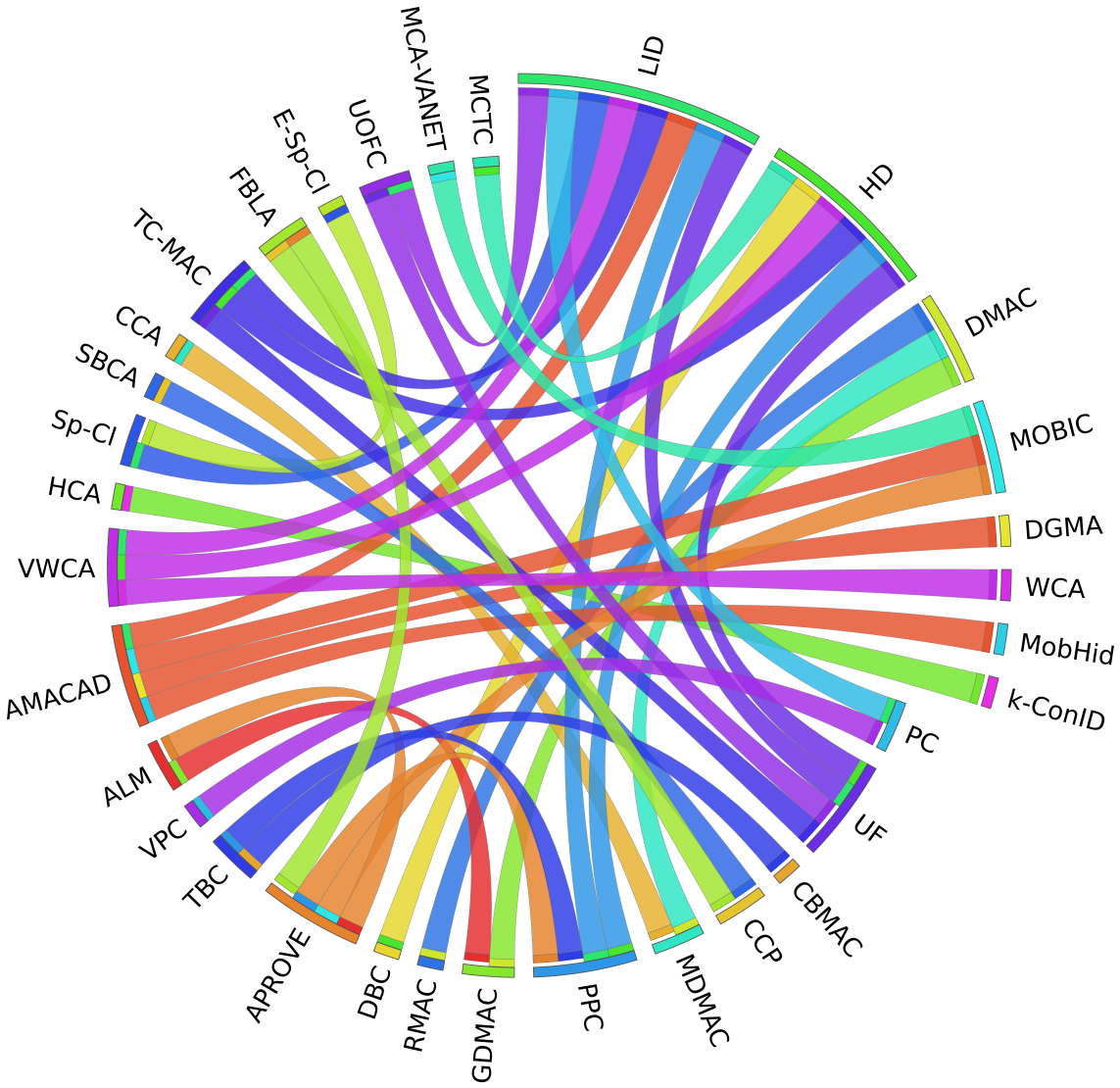


Figure 2.5: Cluster comparisons. Algorithms are sorted clockwise in order of publication year. The colour of the ribbon matches the colour of the algorithm that is being proposed, and the ribbon leads to the algorithm to which it was compared. MANET algorithms jut out from the edge of the circle. Diagram created with circos [103].

This survey conversely finds more pressing matters closer to the crux of VANET clustering research, which overlap with various other fields.

2.5.1 Insufficient use of VANET-specific approaches

As seen in Table 2.2, many VANET clustering algorithms are so similar in their approach to the facets of the clustering problem that they may be considered incremental contributions. Many approaches differ only in their cluster head election metrics. Gateway election metrics and cluster head hand-off strategies are identical

in some cases. The persistence of these practices stems from the reliance on older MANET methods, using weighted metrics or timer-based approaches, and the tendency to consider VANET research a sub-field of MANET, which is doubtful when considering the mobility patterns, channel behaviour, energy capacity, and available processing power. Meanwhile, there are a wealth of possibilities with respect to predictive capability, using road structure data, and road-side unit assistance.

New concepts are available for grouping vehicles and determining cluster head suitability, including similarity of routes and the influence of the road structure on the possible topologies the VANET can form. Cluster robustness and stability could be improved with something as simple as modulating weights applied to cluster head election metrics based on the number of lanes on the road, which have been shown in [15] to constrain vehicles to simple lines. Driver intention has been addressed in the surveyed literature, often with respect to infotainment and comfort applications. However, basic intentions such as changing lanes can alter clustering structure, and using such intention (which can be detected by the on-board unit when the driver turns on the indicator) to predict these changes can accelerate cluster head hand-off mechanisms.

2.5.2 Channel Modelling Disconnect

There is an apparent disconnect between new findings in VANET channel research with respect to environmental dependence and non-stationarity, and the validation methodology of clustering. Many papers in this survey used unsuited channel models – many others don't cite any channel model at all. Accurate channel modelling is the lynch-pin of the entire field, therefore any improvement or advancement in performance and efficacy of clustering algorithms cannot be determined reliably without it.

More accurate modelling, such as the model applied in [15], may reveal problems with certain design choices and enable researchers to improve their algorithms. This will lead to innovative approaches for VANET protocol design and allow researchers to investigate new applications of the technology with greater confidence in their verification methodologies.

2.5.3 Benchmarking and Validation

Any algorithm designed for VANETs should out-perform a MANET algorithm in a vehicular scenario. While it can be said that older but well-understood approaches like LID/HD and MOBIC serve as adequate benchmarks, one can only make this statement if the simulation scenarios are identical between publications, which is not so. Algorithms are validated in highway scenarios with different numbers of exits, in urban scenarios of differing size and node densities; channel model implementations differ between simulator frameworks. Performance metrics differ across publications in form and nomenclature.

As VANET research has progressed, the de-facto research approach has been to present the algorithm succinctly, followed by simulation analysis, preferably in comparison to a competing algorithm. VANET research would benefit significantly from a standardised validation methodology. Additionally, source code availability would enable fellow researchers to benchmark their new proposals to recent algorithms. Care should be taken to adequately cite the channel model, including implementation and configuration, to allow third-party repetition of results. Furthermore, standard performance metrics should be presented, which have been agreed upon by the research community.

2.5.4 Approach to VANET Research

In the early days of the Internet, the nature of the channel – being that of a simple wired connection with a largely frequency- and time-independent impulse response and manageable inductances and capacitances – was conducive to protocol design that focused on the transfer of general-purpose information. Wired networks were built on direct links that required only sender and receiver to negotiate channel access. The limits were the capacity of the copper – and later fibre-optics – to carry the signal, and the processing power of the router to handle incoming and outgoing traffic.

Wireless networks are emphatically different, requiring participants to harmonise their communication and ensure fair distribution of channel use. This makes Internet research and design methodology unsuited for the design of a VANET protocol that serves the purpose of moving data around without concern for its content.

This is reflected in the observations by Kwon et al in the design of PC [50], where a general purpose clustering algorithm with its own control frames interferes with the objective it is intended to achieve. Too much emphasis is placed on sending data without concern for the objective of the network. In this light, the vision of a VANET that forms topology-homeostatic clusters, routes data reliably and quickly, prevents collisions safely, delivers traffic data in time for the driver to respond, and executes comfort and infotainment applications, all the while providing a secure and robust system that protects users from malicious agents becomes difficult to realise.

For most researchers, VANETs, unlike the Internet, have highly specific applications. Thus, protocol design methodology should begin with the intended application. From there, the known and accepted obstacles of vehicular communication should be considered and analysed. The protocol can then be designed around these obstacles; alternatively, the obstacles can be innovatively used to the protocol's advantage. An example of such advantageous use would be employing the broadcast nature of wireless to send data to multiple receivers with a single transmission (as is used in the passive clustering approaches), or utilisation of coding, frequency, or cooperative diversity. Finally, the protocol is then tested with an accurate channel model that accounts for as many known propagation phenomena as possible, with comparison to the most recent competing protocols. An improvement in methodology and analysis of the VANET problem will accelerate the roll-out of this technology and the benefits it is hoped to provide.

Chapter 3

Urban Radio Channel

Chapter 2 detailed the various design approaches to the clustering problem, and emphasised potential flaws in the methodology whereby new designs are validated. As clustering is a higher-layer function, protocol designers give little attention to the channel model used in their simulations, as evidenced by Table 2.5.

This state of affairs is understandable, since mobility of nodes and constantly changing environmental conditions makes the VANET channel highly variable. Designing an algorithm to be robust against changes in mobility while working to achieve the goals of a vehicular network is a complex problem. The more comprehensive channel models are so intricate – often demanding large amounts of information about the surrounding environment – that execution times become an issue. The most accurate channel modelling solution, *Ray-tracing*, requires processing power beyond the reach of most researchers. Unfortunately, there are several consequences to simplifying the simulation, which should be acknowledged.

The question is then, are there ways to simplify the modelling approach such that accuracy is maintained while processing, data, and time requirements are reduced? In this chapter, the impacts of channel simplification are detailed. The VANET channel is then broken into three key environmental components, and the known methods for modelling them are assessed. During this assessment, a gap in modelling stochastic elements of the channel is identified. A method for the approximation of these elements, based on experimental data, is presented.

3.1 Impacts of Channel Simplification

This section details the consequences of over-simplification of channel modelling:

1. **Over/Under estimation of performance:**

   FSPL best models an empty space scenario, depicting a scenario in which there are no other vehicles or buildings in the simulation. Communication over such a channel is very easy, and protocols will perform very well because they do not have to deal with frequent packet loss. Researchers may then receive good performance in simulation, without recognising flaws in their design.

   Conversely, an ideal channel model can cause certain protocol designs to perform poorly in simulation. The study in [104] suggests that environmental factors cause inhomogeneities in a node's transmission range, meaning that a car's transmission may successfully reach a destination in front, but not behind. This may have some benefits in certain scenarios, where two nodes in close proximity have a poor link to each other, but not to their separate neighbours, thereby mitigating the Hidden Node Problem. An ideal channel model would not account for this, which means transmissions are more likely to collide and reduce performance.

2. **Insufficient Comparative Analysis:**

   An ideal model can prevent adequate comparative analysis of competing protocols. Different approaches to the facets of VANET communication will behave differently in response to channel conditions. However, when simple models are used, the playing field is unduly even amongst these competing designs. Therefore, it is difficult to determine which of two approaches achieves the objectives of the VANET better.

It is therefore advisable that higher realism be made a priority in channel modelling. There needs to be more conversation between protocol designers and channel investigators, so that the efficacy of a protocol's design can be fully realised. In so doing, high performance designs can be verified and built upon. Perhaps researchers will identify new innovations that exploit aspects of the VANET channel that are presently seen as hurdles to overcome.

3.2 Ray-tracing: The Gold Standard for Accuracy

Excluding physical communications experimentation, the best solution for a realistic model of the channel is ray-tracing. The placement, geometry, and composition of objects in the environment are analysed, producing a series of reflected and diffracted paths from the transmitter to the receiver. These paths are then used to determine the channel impulse response. This simulated impulse response provides a highly accurate depiction of the transmission conditions in reality.

There are two drawbacks with ray-tracing. First, the data input to the system must be an accurate depiction of the area in which the simulation is taking place. This includes knowledge of the building material and geometry, placement and structure of scatterers such as sign posts and trees, and the position, geometry, and mobility characteristics of the vehicles in the simulation. The greater the required accuracy, the more information needs to be supplied. There are a number of research projects targeted at obtaining and collating this data for a wide variety of locales. Vehicular mobility data can be gathered from a traffic simulator, and specific geometry and material data for various car makes and models can be obtained from car manufacturers. Building geometry sources include government construction records and community-driven databases like Open Street Maps [105]. In fact, work by Nuckelt et al [106] found simulations using data extracted from OSM demonstrated close agreement with results obtained by applying building data supplied by local government, especially in verifying channel sounder measurements with ray-tracing simulations [107].

The second drawback of ray-tracing is the large demand for data storage and processing capability. As researchers pursue more accurate simulations, the required amount of environmental data grows. This is not limited to hard disk space, as the data will be cached in RAM. The frequency with which data is loaded into memory alone increases the processing time of channel analysis. This, coupled with the processing power required for most ray-tracing applications, entails a resource expenditure beyond the financial and logistical reach of some research groups.

Despite these drawbacks, there is work underway to improve performance and ef-

ficiency of ray-tracing, and bring the technique within the capabilities of all VANET researchers. Unfortunately, there is still the bottleneck of dealing with mobility. Ray-tracing is most often applied to wireless networks involving stationary or slow-moving nodes, such that the commonly followed Wide-Sense Stationary and Uncorrelated Scattering (WSSUS) assumption holds over long periods of time. However, Bernardo et al [108] showed that this assumption is invalid for VANETs. Thus, ray-tracing techniques will need frequent recalculation for mobility of nodes, or large amounts of storage to contain the channel impulse response for as many geographic positions and velocities as possible. This, again, poses a high resource expenditure and potentially long simulation times.

Ray-tracing is one extreme of the channel modelling spectrum, with ideal models like FSPL falling at the opposite end. Trade-offs can be made between accuracy and realism by making smaller approximations and assumptions, which can be verified by experimental analysis. Furthermore, the influences of the channel can be isolated into separate components by recognising that the different phenomena comprising the channel have separate causes that can be modelled independently of one another. The following section details these components and methods of simulating them.

3.3 Components of a Channel

Implementations of channel models typically involve multiplying the transmitted signal power by a gain factor computed from the simulated channel's impulse response. This gives the strength of the signal that reaches the receiver. When the units of signal power and gains are decibels (dB), the computation of received signal power P_{RX} typically follows

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} + G_{PL} + G_f^{SS} + G_f^{LS} \tag{3.1}$$

where

P_{TX} is the transmission power,

G_{TX} is the gain of the transmitting antenna,

G_{RX} is the gain of the receiving antenna,

G_{PL} is the path loss factor,

G_f^{SS} is the loss factor due to small-scale fading, and

G_f^{LS} is the loss factor due to large-scale fading.

The first three components contribute to the success of the transmission. These factors are often constant and easily controllable in experiment. The last three factors in the list are negative values that degrade the received signal strength. These are computed from the channel impulse response.

3.3.1 Path Loss

The distance-dependant attenuation of signal strength, commonly known as Path Loss, is often determined using either Friis Free Space Model or the Two-Ray model. Both these models operate over the straight-line distance between transmitter and receiver, and represent the physical attributes thereof. However, wireless signals can be obstructed by objects occluding the destination. In the urban environments where the applications of VANETs are expected to be realised, vehicles and buildings will obstruct direct paths between communicating vehicles.

3.3.2 Large-Scale Fading

During the operation of a wireless network, a channel will experience long-lived transient drops in received signal strength that are not explained by distance-based calculations like path loss. These drops are called Large-Scale Fading, and are typically caused by obstructions to a line of sight path. In a VANET, such obstruction can be caused by buildings or, more prominently, other vehicles [109].

3.3.3 Small-Scale Fading

Small-Scale Fading differs from its Large-Scale counterpart, in that it is characterised by persistent fluctuations in received signal strength due to multi-path interference. As mentioned in Section 3.3.2, building walls reflect and confine signal energy, which interferes with each other at a receiver. In an urban environment, the statistics of Small-Scale Fading follow a Rice Distribution – thus earning the name Rician Fading.

A Rician Faded signal envelope has the following CDF,

$$R(x) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2 + A^2}{2\sigma^2}\right) I_0\left(\frac{xA}{\sigma^2}\right) \tag{3.2}$$

where $A^2/2$ is the power in the specular or line-of-sight component, and σ^2 is the power in the diffuse or multi-path components. These two components are combined in a single fading parameter called the K-factor, which is the ratio of specular to diffuse power.

$$K = \frac{A^2}{2\sigma^2} \Rightarrow K_{dB} = 10 \log_{10}\left(\frac{A^2}{2\sigma^2}\right) \tag{3.3}$$

The use of the K-factor allows a simple way to indicate the level of multi-path interference. When $K = 0$, the diffuse components dominate the signal in a case of Rician fading called Rayleigh. As the K-factor increases, the diffuse components become less significant with respect to the specular signal – that is, there is less interference. An infinite K-factor implies zero fading.

3.4 Simulation Methods

3.4.1 CORNER

Buildings exert another influence on path loss behaviour. Signals are known to reflect off the outer walls and diffract around the corners of buildings. It is therefore possible for signals to reach vehicles that have no direct path due to being on different streets from the transmitter. As shown in Figure 3.1, S has no direct path to D, yet signals reflect and diffract around the obstruction in order to reach D. Therefore, despite the presence of obstructions, it is still possible for S and D to communicate with each other. However, it should be stressed that neither FSPL nor Two-Ray are capable of modelling the path loss behaviour of such a propagation scenario.

Ray-tracing models (e.g. [104, 110]) are the best approach for computing diffraction and reflection in urban environments, despite their high resource usage. Upon recognition of this fact, Sun et al [111] presented analytical formulae to model the path loss behaviour between S and D around building corners. The authors determined the losses due to reflection and diffraction based on geometric optics and diffraction theory, and produced formulae for three different propagation scenarios:

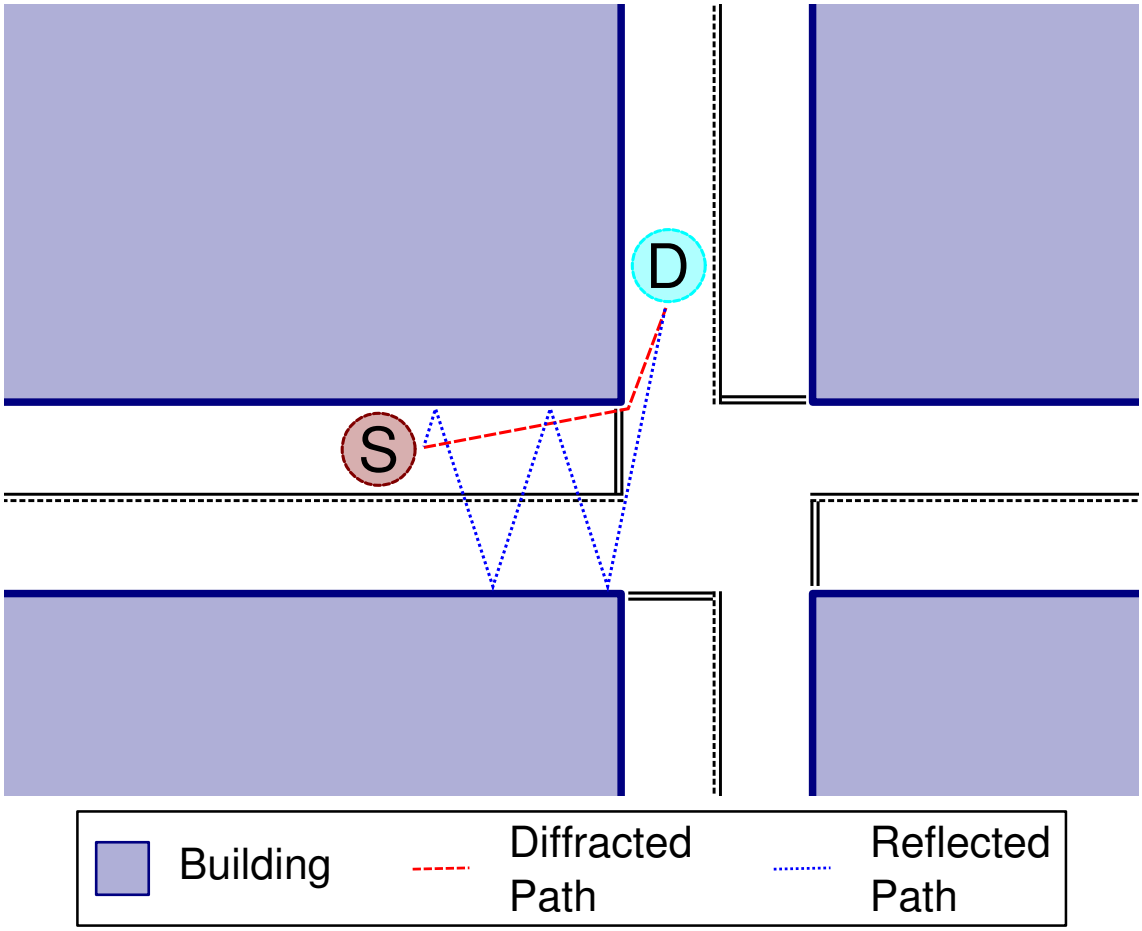


Figure 3.1: Urban communication scenario. Although S and D are obstructed by the building, signals transmitted from S diffract and reflect around the corner of the building to D. However, both FSPL and Two-Ray are insufficient to model the received signal from such a propagation scenario.

1. Line-Of-Sight (LOS): Path between source and destination is unobstructed by buildings. This does not preclude the possibility of vehicular obstruction. In this case, the regular FSPL equation is applied.

2. Non-Line-Of-Sight 1 (NLOS1): The destination is on a street perpendicular to that of the source. This scenario is shown in Figure 3.1. The source and destination are separated by a single intersection I, where the source is r_m metres from I and the destination is r_s metres from I. The widths of the source and destination street are given by W_m and W_s, respectively. The minimum number of reflections is calculated as $N_{min} = \left\lfloor 2\sqrt{\frac{r_m r_s}{W_s W_m}} \right\rfloor$. Thus, the path loss is computed as:

$$G_{PL} = \left(\frac{\lambda}{4\pi}\right)^2 (PL_r + PL_d) \tag{3.4}$$

where

$$PL_r = \left(\frac{L_R^{N_{min}}}{r_m + r_s} \right)^2$$

$$PL_d = \frac{\lambda}{4} \begin{cases} \frac{1}{r_m r_s^2}, & \text{if } r_m < r_s \\ \frac{1}{r_s r_m^2}, & \text{otherwise} \end{cases}$$

3. Non-Line-Of-Sight 2 (NLOS2): The destination is on a street parallel to that of the source. In this scenario, the source and destination are separated by two intersections, I_1 and I_2. The source is r_m metres from I_1; the destination is r_p metres from I_2; I_1 and I_2 are r_s metres apart. The widths of the source, destination, and intervening streets are W_m, W_p, and W_s, respectively.

$$G_{PL} = \left(\frac{\lambda}{4\pi} \right)^2 (PL_r + PL_{dd} + PL_{rd} + PL_{dr}) \tag{3.5}$$

where

$$PL_r = \left(\frac{L_R^{N_{min}}}{r_m + r_p + r_s} \right)^2$$

$$PL_{dd} = \frac{\lambda^2}{16 r_s r_m r_p} \begin{cases} \frac{1}{r_p}, & \text{if } r_m < r_s \\ \frac{1}{r_m}, & \text{otherwise} \end{cases}$$

$$PL_{rd} = \frac{\lambda}{r_p} \left(\frac{L_R^{N_{min}}}{2(r_m + r_s)} \right)^2 \begin{cases} \frac{r_s}{r_p}, & \text{if } r_s < r_p \\ 1, & \text{otherwise} \end{cases}$$

$$PL_{dr} = \frac{\lambda L_R^{2N}}{4 r_m (r_p + r_s)} \begin{cases} \frac{1}{r_p + r_s}, & \text{if } r_m < r_p + r_s \\ \frac{1}{r_m}, & \text{otherwise} \end{cases}$$

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

$$T = \sqrt{\frac{r_s W_m W_p}{W_s \left(r_m W_p + r_p W_m \right)}}$$

$$N = \left\lfloor \frac{r_p r_s}{W_p W_s} \right\rfloor$$

L_R is included in all of the above formulae, being the loss coefficient per reflection.

Sun et al [111] tested these formulae based on measurements taken in Tokyo,

Japan, and in Manhattan, USA. The calculated path loss components showed good agreement with the experimental measurements. However, they were not incorporated into a model to be used in simulations, owing to the lack of a mechanism to determine which of the three above states best described a given source-destination pair. The formulae were further tested by Giordano et al [10, 97]. Supplementing this contribution was a means of classifying the LOS state of two nodes, so that a simulation program could determine which of the above formulae to apply in the computation. The authors called their implementation CORNER, and performed an independent wireless experiment in a residential area of Los Angeles. Comparison with simulations that incorporated their classifier showed strong agreement.

A third independent experiment survey was carried out in Miranda, a suburban area of Sydney, Australia, by Mukunthan et al [12]. Several implementation and logical problems with the Giordano LOS classifier were identified, and a new one was implemented that computed the states between each individual street to be loaded into the simulation program at run time. The simulations performed faster and more accurately than Giordano's implementation.

3.4.2 The Knife-Edge Model

Large-Scale Fading can be simulated using the Knife-Edge model. This technique has been investigated extensively by Boban et al [112], Wang et al [11], and many others. It is well understood and has been independently verified with respect to intervehicle communication. The model works by treating each car as a simple geometric object, and calculating the diffraction of signal energy around the shape. Unlike ray-tracing, it requires only the height of the vehicle relative to the transmitter and receiver, and gives a good approximation of the attenuation caused by vehicles.

3.4.3 Environmental Influences on Fading

Often, VANET researchers analyse proposed communication protocols using a constant K-factor, implying a stationary fading scenario [113]. This is an instance of the WSSUS assumption. It has been widely assumed that this principle is also valid in the VANET channel; however, VANETs differ significantly from cellular networks

in terms of antenna heights, frequency range, scattering properties, and degree of mobility [114].

Related experimental work by Bernado et al has established that the WSSUS assumption is not generally valid in VANETs [83, 108], and that the K-factor is variable in space and time [115–118]. Empirical studies have shown that the K-factor is significantly dependent on the surrounding environment [117, 119]. A simulation that accounts for this dependence will therefore provide a more accurate demonstration of protocol behaviour than one which assumes WSSUS.

The authors of [83, 108] used two metrics, spectral divergence and coherence bandwidth, to analyse the channel in both the time and frequency domains, and the VANET channel was shown to violate the WSSUS assumption. Acknowledging this problem, the authors of [120] presented an optimised parameterisation of a local scattering function. They developed an estimator of the scattering behaviour of the environments in which they conducted their measurements, and then configured the parameters of the function based on channel measurements made in those areas.

Experimental measurements have since been used to model the behaviour of the K-factor. The first attempt to develop an analytic approach to K-factor modelling was published by Messier et al [119]. The authors conducted a wireless communication survey in a dense urban area in Canada, applying a moment-based K estimator from [115]. By analysing the statistical distribution of the K-factor measurements, they produced a Gaussian-based stochastic approximator for the K-factor that can be used in simulation. This provided a statistically accurate model of K-factor dynamics.

A similar approach was presented in [117, 118]. Using the results of the measurement campaign from [83, 108], the K-factor dynamics of the experiments were analysed, and found to vary in time, frequency, and space. They also found that the K-factor is dependent upon elements of the surrounding environment, such as the presence of scatterers and reflectors. The authors make an additional contribution in [118], in which they configure the parameters of a bi-modal Gaussian distribution to generate K-factors with the same statistical properties as those derived from their experiments.

3.4.4 Discussion

Experimentally verified solutions have been identified for two components of the channel. CORNER accounts for the Path-loss component, while Knife-Edge shadowing accounts for Large-Scale fading. Means of approximating the statistics of Small-Scale fading exist, in the form of empirical analysis methods developed by Messier [119] and Bernado [117, 118]. However, these methods have disadvantages:

1. They require physical measurement equipment, which can entail a larger resource and time expenditure.

2. They are only valid for the particular site of the original measurement campaign, being sensitive to vehicle density and weather conditions at the time of experiment. A new set of channel measurements must be taken for any area an investigator wishes to simulate, which can be impractical for large cities.

3. While simulations based on these models may give accurate mean results for the entire area in which the original measurements were taken, it may be inaccurate for a specific location within that environment, due to its statistical nature.

The following section details a proposed method of approximating Small-Scale Fading behaviour from easily accessible data sources.

3.5 Approximation of Fading Statistics

Empirical studies are not the only approaches to modelling fading statistics. This section describes a method of approximating K-factors from simplified geometric data. By analysing the signal paths intercepted by the receiver, the K-factor for that source-destination channel may be approximated. Knowledge of the environment – in particular, building geometry, placement, and composition – is required to determine the multi-path components necessary for the approximation.

3.5.1 Background

An alternative approach to channel modelling is applied in [121]. The authors propose a geometry-based stochastic channel model (GSCM) to compute the channel

impulse response based on the distribution of mobile (that is, vehicles) and stationary (that is, roadside trees and lamps) scatterers, and verified the model using a channel measurement campaign in Lund, Sweden. In the model, the scatterers were randomly distributed between lanes of a highway and along it's length. This work also discusses the possibility of using vehicular mobility models (e.g. SUMO [98]) in the computation.

This methodology was extended in [122]. As in [121], the authors stress the use of simplified ray-tracing approaches through 2D representations of the simulation environment to parameterise the channel impulse response. The complete parameterisation of the model is presented for various environments (such as campus roads and highways) for use in V2V scenarios, along with verification of the proposed model compared with GSCM. These models are of particular interest, as they show that the VANET channel may be modelled using simplified geometric components.

3.5.2 Approximation Method

The K-factor is approximated using the 2D simplification approach suggested in [122], quantising transmitted power into rays. The individual power contributions of each ray meeting at the receiver is then computed from the number and nature of reflections it has undergone.

Power radiated from an omnidirectional antenna is quantised (in two dimensions) into N_{ray} rays, each having an initial power level of

$$P_{ray} = \frac{P_{TX} G_{TX}}{N_{ray}} \tag{3.6}$$

The quantised rays are radiated in all directions from the transmitter. A ray is determined to have been intercepted by the receiver if it passes through the effective aperture of its antenna. The effective aperture is given by

$$A_e = \frac{G_{RX} \lambda^2}{4\pi} \tag{3.7}$$

The power contribution of the i'th received ray is determined by the number of reflections it has undergone, N_i^{ref}, and the reflection coefficient Γ of each reflective surface. Γ is dependent on the relative permittivity of the surface material ε_R and

the incident angle θ_I.

$$\Gamma = \frac{\sqrt{\varepsilon_R - \cos^2\theta_I} - \varepsilon_R \sin\theta_I}{\sqrt{\varepsilon_R - \cos^2\theta_I} + \varepsilon_R \sin\theta_I} \tag{3.8}$$

Consecutive reflections are modelled by multiplying their reflection coefficients, such that the combined coefficient of the i'th ray is

$$\Gamma_i^T = \prod_{n=0}^{N_i^{ref}} (\Gamma_n)_i \tag{3.9}$$

and the power of the ith ray is [a]

$$P_i = P_{ray} \left(\Gamma_i^T\right)^2 \tag{3.10}$$

The receiver forms a set Q of rays that pass through its aperture after reflecting through the environment. The specular and diffuse ray sets are then $Q_S = \{i \in Q | N_i^{ref} = 0\}$ and $Q_D = Q \setminus Q_S$.

The K-factor can now be estimated by

$$\hat{K} = \frac{\sum_{i \in Q_S} P_i}{\sum_{j \in Q_D} P_j} \tag{3.11}$$

This formula has the useful property of being scale-invariant. That is, it is independent of the transmitter power or gain. By substituting (3.10) into (3.11)

$$\hat{K} = \frac{\cancel{P_{ray}} \sum_{i \in Q_S} \Gamma_i}{\cancel{P_{ray}} \sum_{j \in Q_D} \Gamma_j} \tag{3.12}$$

Thus, the approximator is purely a function of the communication environment, as was predicted by Messier et al [119]. It should be noted that the formula is not independent of the *receiver* properties – in particular, it depends on the effective aperture of the receiver.

[a] The approximator was first published in IEEE ICC 2014 [13]. There was an error in the derivation that influenced the results published in the paper. The error has been fixed and improved results have been obtained. The authors would like to thank the reviewers of Physical Communications Journal for recognising the error and pursuing the matter.

3.5.3 Ray-launching Algorithm

As a proof-of-concept, a simple geometric intersection algorithm was used to trace rays through a top-down view of the building environment, with the faces of each structure tagged with the permittivity of the material from which that structure is composed. The model is purely two dimensional; it is assumed that there are no overhead reflectors – meaning that any signal power directed skyward can be considered lost (a similar approach to models proposed in [121, 122]). Considering only a two-dimensional plane simplifies construction of the simulation scenario and reduces computational complexity.

The ray-launching approach allows for multiple receivers to determine a K-factor to a single transmitter using a single execution of the algorithm, shown in Figure 3.2. The initial length of each ray, d_{max}, is determined by the FSPL model as

$$d_{max} = \sqrt{\frac{P_{TX} G_{TX} A_e}{4\pi S}} \tag{3.13}$$

where S is the sensitivity of the receiver. This criterion introduces a dependence of the approximated K-factor on transmission power. However, this dependence does not affect the magnitude of the result, but rather the maximum distance over which approximation is valid. When a reflection occurs, the new ray's coefficient is computed from Equation (3.8), and its length is given by

$$\Delta d = \Gamma d_{max} - d \tag{3.14}$$

Once the algorithm has computed the ray distribution for the given number of rays initially launched from the transmitter, each receiver can simply use its location and effective aperture size to calculate the number and power of rays (out of the total launched) reaching the aperture, and hence obtain the total power received at that location. The resultant set of ray paths are used to determine the K-factor as per Equation (3.11).

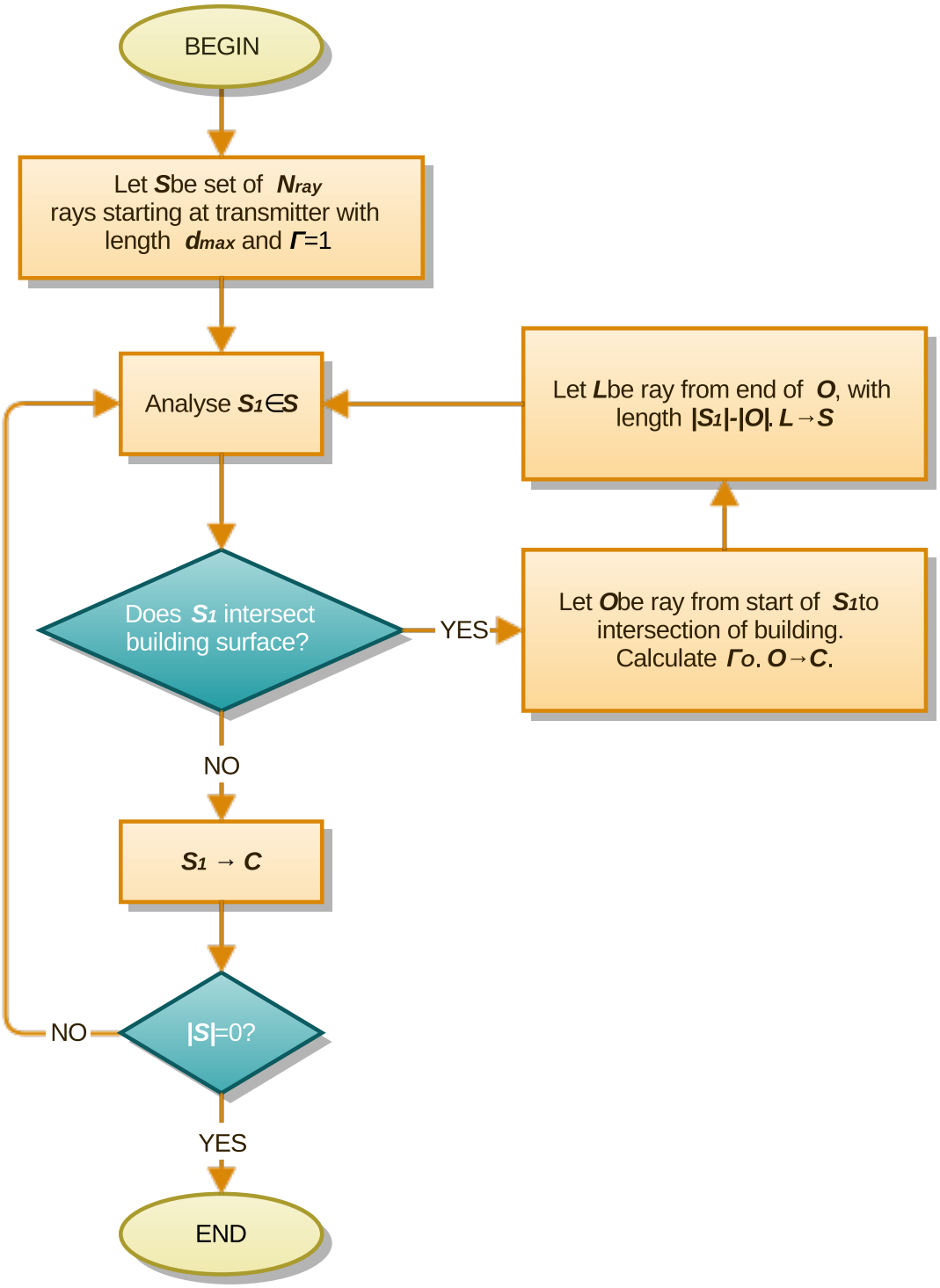


Figure 3.2: The ray-launching algorithm.

Performance Considerations

A multithreaded implementation of the algorithm has a worst-case execution time of the order of

$$\mathcal{O}\left(\frac{N_{ray}^2 \rho_B d_{max}^2 \mu_E}{N_{core}}\right) \tag{3.15}$$

where ρ_B is the density of buildings per unit area, μ_E is the average number of edges per building, and N_{core} is the number of cores used in the simulation. Typical execution time for the ray-launcher is explored in Section 3.7.3.

Building Geometry Data

At present, this model does not take vehicles into account; only static building geometry is considered. The building data is taken from OSM. Its format allows additional attributes to the XML descriptors of building definitions, including the material out of which each building is predominantly composed. The validity of this method has previously been established by Nuckelt et al [106], who found the approach provided results in close agreement with those obtained by applying building data supplied by local government, especially in verifying channel sounder measurements with ray-tracing simulations [107].

Often researchers will use a program to generate an artificial road map to better control the characteristics of road topology and analyse their impact on a protocol's performance. In this case, there is no building geometry available. For this purpose, a program was written called *BuildingSolver*, which constructs building geometry from a SUMO network file. This algorithm is described in Appendix A.

Permittivity of Building Materials

The permittivity of several common building materials at 2.41 GHz is listed in Table 3.1. These parameters are used throughout the simulations.

3.6 Verification of Approximator

Simulation and experiment studies were undertaken to verify the performance of the approximator described in Section 3.5.2. The experiments were conducted in

Material	ε_r	Reference
Glass	2.3	[123]
Stone	4.5	
Concrete	5.4	
Concrete (non-reinforced)	8.721	[124]
Polycarbonate	2.86	
Gypsum	2.518	
Brick Wall	3.743	

Table 3.1: Relative Permittivity of various common building materials [123, 124].

Wollongong and Sydney, Australia, in both LOS and NLOS scenarios. Cases where the approximator shows good agreement with the experiment have been identified, as well as areas where its accuracy can be improved.

3.6.1 Experimental Work

Physical experiments were carried out in order to obtain measurement of K-factor and PDR in live traffic situations. The approximator was verified by comparison of its K-factor estimates with the results of these experiments. The methodology applied here will also be adopted for the simulation.

The experiments were conducted in local urban and industrial areas of Wollongong and Sydney, NSW. These areas were selected based on their accessibility, the type and number of buildings in the area, and whether there were times at which vehicle density was very low. Two sites were chosen: Kingsway Road (Figure 3.3) in Miranda, a southern suburb of Sydney; and Burelli Street (Figure 3.4) in Wollongong.

Burelli Street is a long, straight stretch of four-lane road punctuated by multiple intersections. It is frequented by heavy traffic during daylight hours and is virtually deserted at night, providing a site where the channel can be analysed in the presence and absence of vehicles in LOS conditions. The site in Miranda is a built-up urban block, with one street being a part of a highway and the others bordering residential apartments. The presence of Non-Line-Of-Sight (NLOS) areas allows K-factor comparisons with the LOS areas of Burelli street.

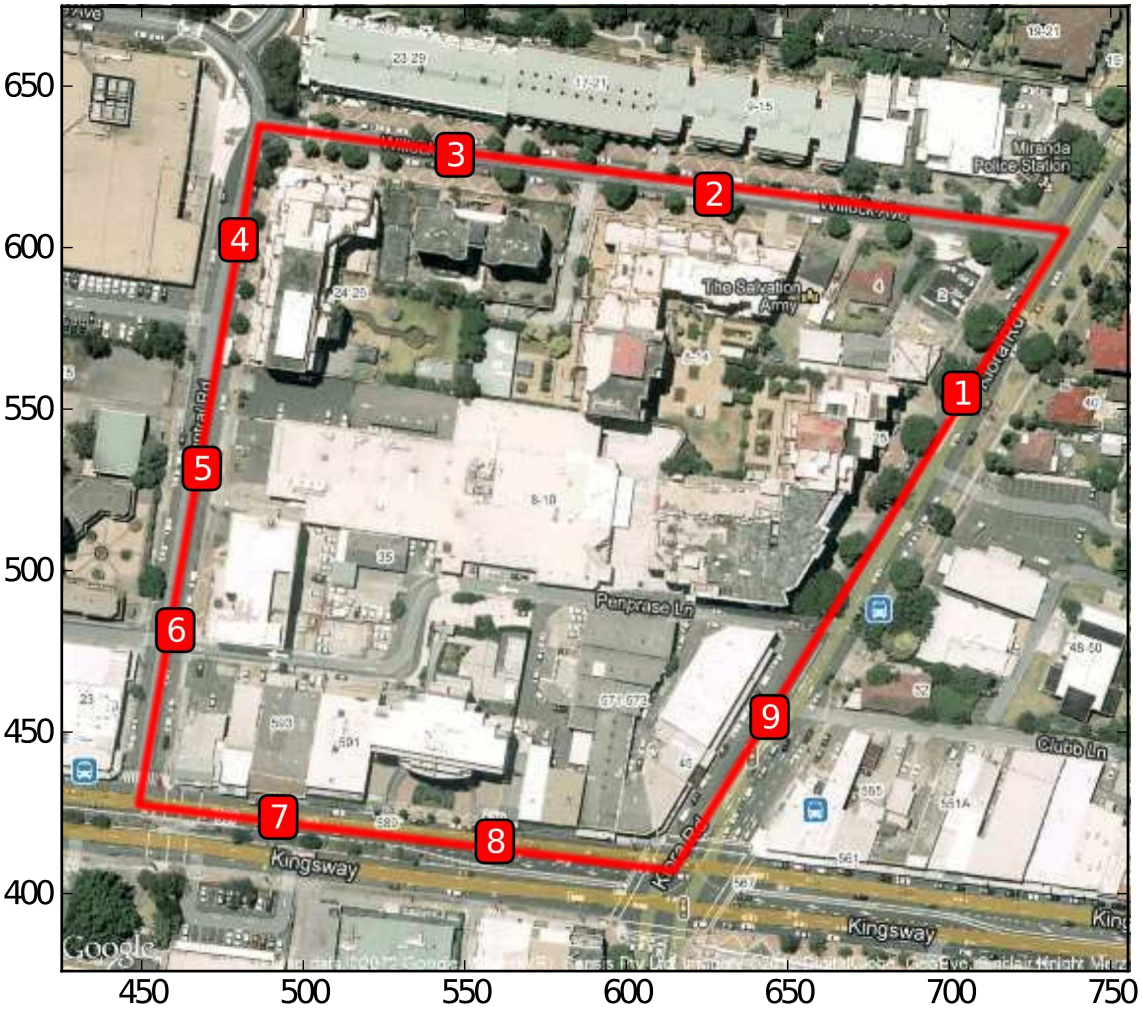


Figure 3.3: Satellite map of the Miranda experiment site. The red line indicates the path travelled by the mobile transmitter; the numbers on the path indicate the position of the receiver at each stage of the experiments.

Figure 3.4: Satellite map of Burelli Street, NSW. The location of the receiver, path of the transmitter (cyan), intersections (green) and point of maximum range (red) are indicated.

The Burelli Street experiments were conducted in the presence (10:00 AM – 12:00 PM) and absence of vehicles (11:00 PM – 1:00 AM) during May 2012. The maximum speed limit on Burelli Street is 50 km/h. Miranda experiments were conducted only in the absence of vehicles, between the hours of 11:00 PM and 5:30 AM, also during May 2012.

Experimental Apparatus

Two wireless testing platforms were used, each consisting of the following:

- PCEngines Alix 3d3 systems running Voyage Linux 0.8
- Ubiquiti SR-2 400mW 802.11b/g cards (ath5k drivers)
- Generic 5dBi gain omni-directional antennas
- Laptop for data logging and system coordination

One platform served as a stationary receiver; the other was a mobile transmitter, which travelled along the route indicated by Figures 3.3 and 3.4, transmitting CBR bursts of 100 500-byte packets three times per second. The average data rate was 1.2 Mb/s with a carrier frequency of 2.412 GHz and a physical data rate of 11 Mb/s. This channel was selected for minimal interference from existing wireless networks following a brief survey. The receiver was additionally equipped with a Rohde and Schwarz FSH8 spectrum analyser which sampled signal spectra during as the packet burst transmissions. Packets additionally contained the GPS location of the transmitter, which were recorded together with signal power and Packet Delivery Ratio (PDR) measurements for each burst. Data was collected as the transmitter made ten circuits, at a constant speed of 20 km/h.

Transmitter positions were quantised to ten metre increments over the total length of the test path. Signal measurements were then collated at each of these transmitter positions, from which K-factor values were extracted. PDR measurements were collated for MAC layer performance comparison.

Estimation from Experimental Data

While the most accurate method of K-factor estimation would be a best-fit curve matching of the Signal Power CDF, this method was difficult to reliably automate.

There exist a number of alternative moment-based methods in the literature. Of these methods, the estimator proposed by Greenstein [125] was most simple to implement. Greenstein's estimator is given by

$$\hat{K}(n) = \frac{V(n)}{G_a(n) - V(n)} \tag{3.16}$$

where

$$\begin{aligned} G_a(n) &= \frac{1}{N_W}(|c(n)|) * w(n) \\ G_v^2(n) &= \frac{1}{N_W} \sum_{i=-\infty}^{\infty} \left[w(i) \left(|c(n-i)| - G_a(n) \right)^2 \right] \\ V(n) &= \sqrt{G_a^2(n) - G_v^2(n)} \end{aligned}$$

and where $c(n)$ is the vector of channel samples and $w(n)$ is a rectangular window of width N_W.

The accuracy of the estimation depends on the value of N_W. In the experiment, it was difficult to ensure that each transmitter position received the same number of channel samples for consistent accuracy. Thus, the analysis required a minimum number of channel samples to ensure the computed K-factor would accurately represent the dynamics of the data collected.

A simple procedure was used to determine an appropriate minimum number of samples for satisfactory statistical significance. A fading waveform of known K-factor, K_n was generated according to

$$c\,(n) = G_f\,(n, K) = \frac{\left(X_1\,(n) + \sqrt{2K} \right)^2 + X_2^2\,(n)}{2K + 1} \tag{3.17}$$

where X_1 and X_2 are independent Gaussian random variables with zero mean. This is based on Jakes' model for Rayleigh fading, which has been used previously in [126]. The error between the estimate and the true value of the K-factor, $E_k = \left| \hat{K} - K_n \right|$ was then computed for increasing values of N_W. Figure 3.5 shows that the error between $\hat{K}$ and K_n decreases exponentially with increased N_W. Based on this, a minimum sample size of 30 is selected for analysis, as the residual error is small, but N_W is not impractically large (from an experimental and simulation perspective).

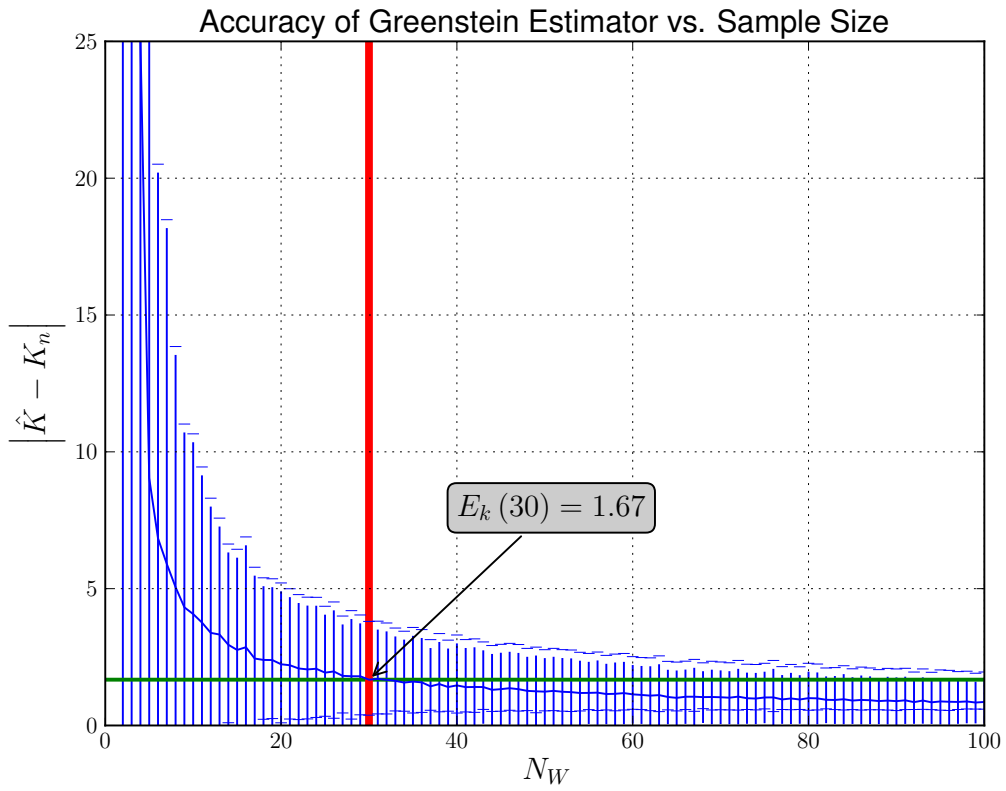


Figure 3.5: The error of the K-factor estimator decreases exponentially with increasing sample size. Error bars represent one standard deviation. At approximately $N_W = 30$, the error is considered acceptably small for the purpose of this work.

Any transmitter positions with fewer than 30 samples were therefore excluded from analysis.

3.6.2 Simulation Description

The K-factor approximator and necessary support classes to load the building edges from the OSM maps were integrated into the OMNeT++ Network Simulator [89]. This is a widely used simulator which can be coupled to the SUMO traffic simulator [98] through the VEINS API [90].

Both simulated nodes use the MiXiM implementation of the 802.11 MAC and PHY layers, without a network layer (as it is not relevant for this study). All packets were broadcast, removing the necessity for routing and association. Packets were sent at a static 11 Mb/s at 2.412 GHz, with the same parameters as used in the experiment (three bursts of 100 500-byte packets transmitted per second). For con-

sistency with the experimental results, power measurements logged by the receiver were used to calculate the sample K-factor using the same analysis techniques as in Section 3.6.1.

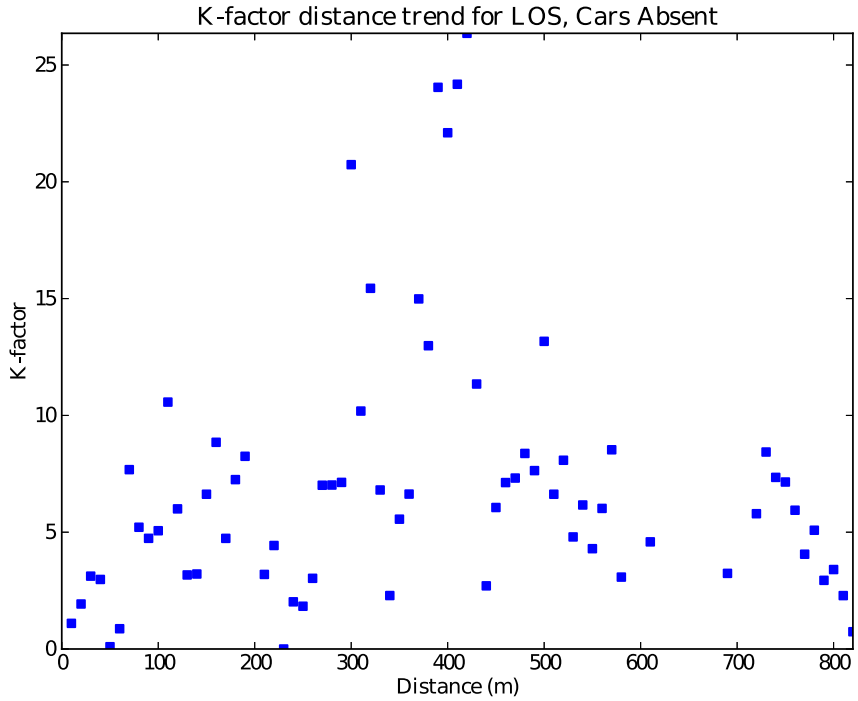
3.7 Results

This section first presents the behaviour in LOS, describing both K-factor dynamics and PDR trends against distance. Note that in all results, K-factors are expressed as a power ratio (i.e. not in decibels). The results are promising, but discrepancies have been noted with respect to vegetation and diffraction phenomena.

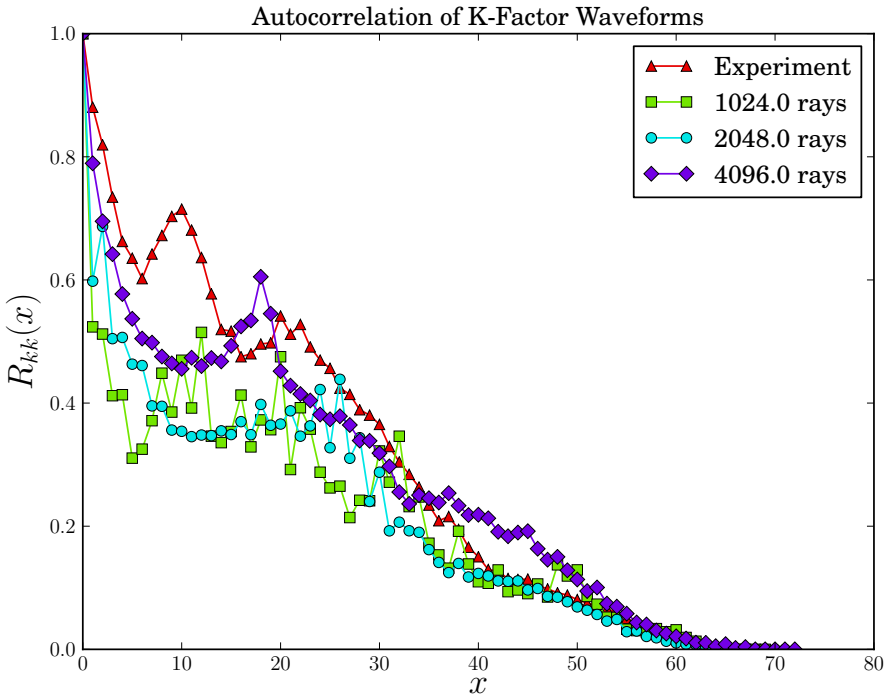
3.7.1 Behaviour in LOS

In Figure 3.6a, the K-factor follows an upward trend with increasing distance, until approximately half-way down the street, at which point it begins to decrease. When the transmitter and receiver are close to each other, with reflectors nearby, the resultant diffuse paths have similar lengths and there are fewer reflections; thus they will interfere with the specular path, reducing the K-factor. As the transmitter moves further from the receiver, the diffuse paths tend to reflect more and travel further, and thus diminish to a greater degree, allowing the specular path to remain dominant. This is analogous to viewing a point light source in a dark room. When near the source, a person can easily see objects illuminated by the reflected light. However, when viewing the source from a greater distance, reflected light becomes progressively dimmer while the point source itself remains strongly visible. At greater distances, path-loss effects dominate the specular path, and the K-factor falls.

The normalised autocorrelation of the K-factor against distance was computed for the simulation with 1024, 2048, and 4096 rays as well as the experimental data. These curves are shown in Figure 3.6b. At a low ray count, the simulation K-factor autocorrelation is lower than the experimental results, and more erratic. However, an increase and subsequent drop in the autocorrelation in the range $x = [5, 13]$ is consistent with the shape of the experimental data. Increasing the ray count to 2048 rays, the simulation results are smoother, but still lower than the experiment. The



(a) K-factor versus distance on Burelli Street.



(b) Autocorrelation of K-factor for experimental results and simulations with 1024, 2048 and 4096 rays.

Figure 3.6: K-factor analysis from the physical experiment.

shape of the curve is also less consistent with the experimental results than 1024 rays. When the rays are further doubled to 4096, the autocorrelation plot is closer to the experiment, with a "bump" in the results over the range $x = [15, 20]$ that corresponds to a similar "bump" over the range $x = [16, 24]$ in the experimental results. The results indicate that the behaviour of the approximated K-factor converges to the experiment as the rays increase.

Figure 3.7 shows a plot of PDR versus distance. The plots show a close agreement between the experimentally measured PDR in the simulation results over the first 100 metres (beyond 100 metres, connectivity is intermittent and so an accurate PDR estimate is difficult to obtain). Increasing the ray count beyond 1024 rays appears to provide little improvement in this case. Furthermore, the mean experimental PDR is higher than the simulation. The discrepancy between experiment and simulation may be attributed to an observed pseudo-waveguide effect caused by building walls [12], which may boost received signal power and thus PDR. Since the simulation does not attempt to account for this effect in the pathloss calculations, the mean PDR is slightly lower than observed in the experiment.

Despite these discrepancies, experiment and simulation exhibit similar behaviour over short range. As mentioned in [9], a transmission range of 100 metres is sufficient for most VANET applications. This is an important result in the application of this K-factor model to the validation of certain VANET scenarios, such as clustering, RSU association, and vehicular collision warning systems.

3.7.2 Approximator Issues

The proposed system provides a good representation of the experimental K-factor for LOS. However, the experiment data reveals certain issues with the approximator in terms of vegetation and NLOS scenarios.

Behaviour in NLOS

Initial analysis of K-factor distribution plots for Miranda, shown in Figure 3.8, revealed that experimental K-factors tend to be higher in NLOS than in LOS at shorter ranges. In the NLOS case, even though there is no direct path, some signal power is diffracted around building corners; it seems that these refracted signals

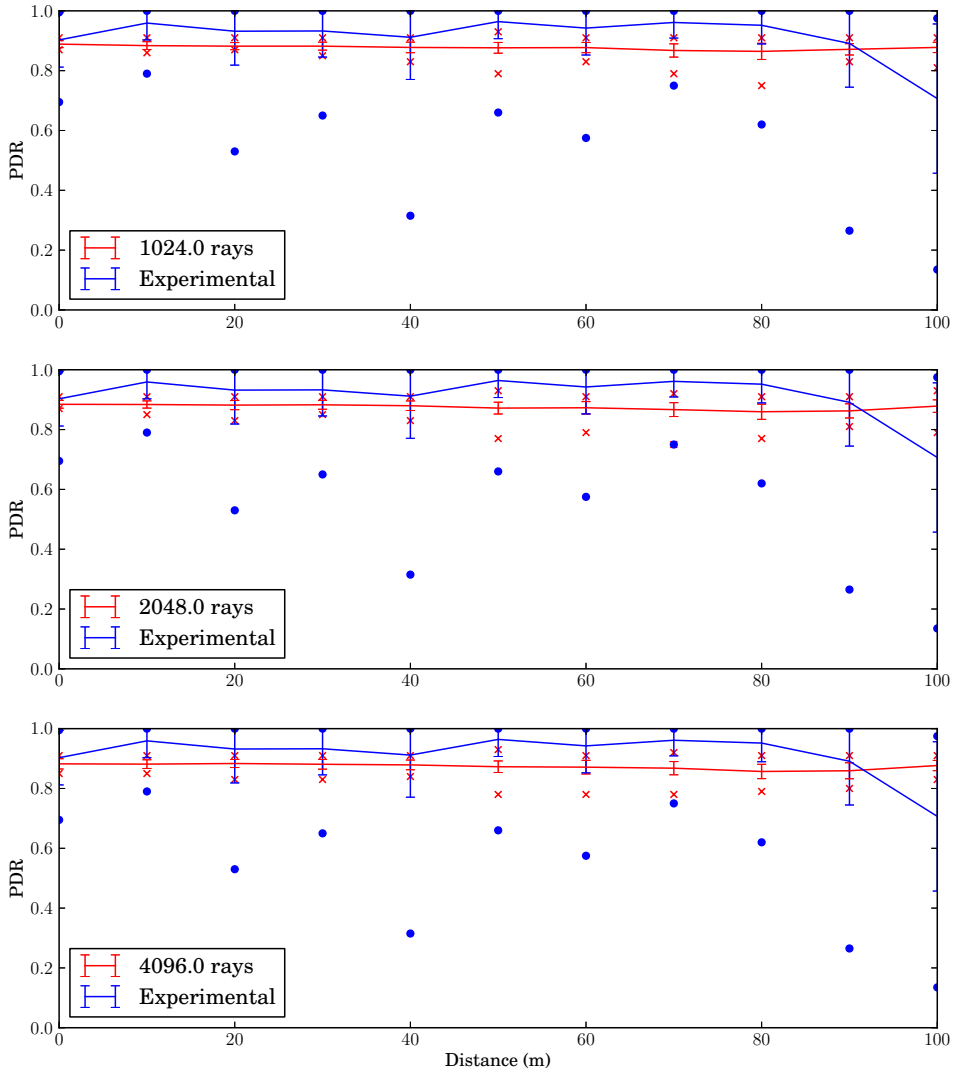


Figure 3.7: PDR behaviour over a range of 100 metres for three 1024, 2048 and 4096-ray simulations in the absence of vehicles. The error bars represent two standard deviations, and the coloured markers indicate maxima and minima.

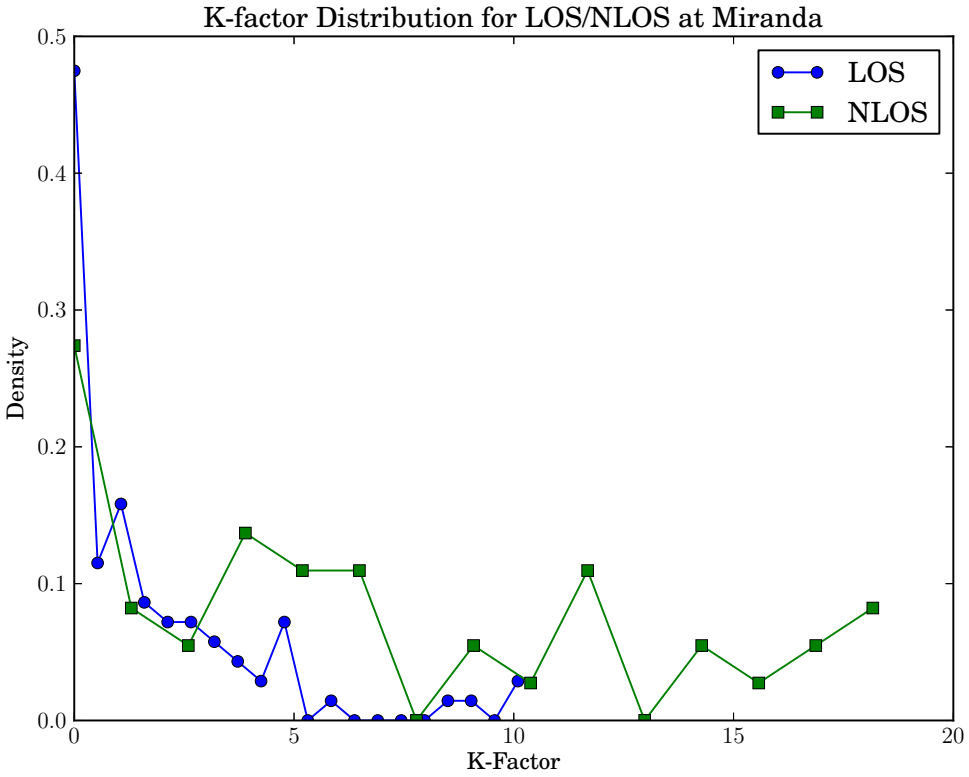


Figure 3.8: K-factor distribution in LOS and NLOS cases.

remain dominant over reflected paths, resulting in a higher K-factor in the NLOS case.

Figure 3.9 shows that in the NLOS case, the K-factor can be up to four times greater than the corresponding values in the LOS case. Figure 3.9d shows a sudden increase in K-Factor values beyond approximately 120 metres. This matches with the four readings after the turn into Penprase Lane in the corresponding satellite image, suggesting that significant power is lost through diffuse rays which escape via intersecting side-streets. Additionally, these areas have significant numbers of road-side trees and other foliage, which can act as an absorbent or scattering medium.

These results indicate a current drawback of the approximator presented in Section 3.5.2. Specular rays having undergone no reflections are required for a non-zero K-factor, meaning that NLOS scenarios should result in Rayleigh fading. However, the experiment results show the possibility for the K-factor to be non-zero in NLOS. In future work, the approximator may need to be modified to incorporate diffraction effects that lead to non-zero K-factors when the source and destination

are not in direct line of sight.

Vegetation

The results of the experiment and corresponding simulations are included in Figure 3.9. The results shown are for the simulation performed with 1024 rays. There is very limited agreement between simulation and experiment for locations 2 and 3. The local environment at these positions includes both significant amounts of vegetation as well as other vehicles, which the model does not yet take into account. The graphs for locations 7 and 8, which included much less vegetation, show a closer agreement between experiment and simulation.

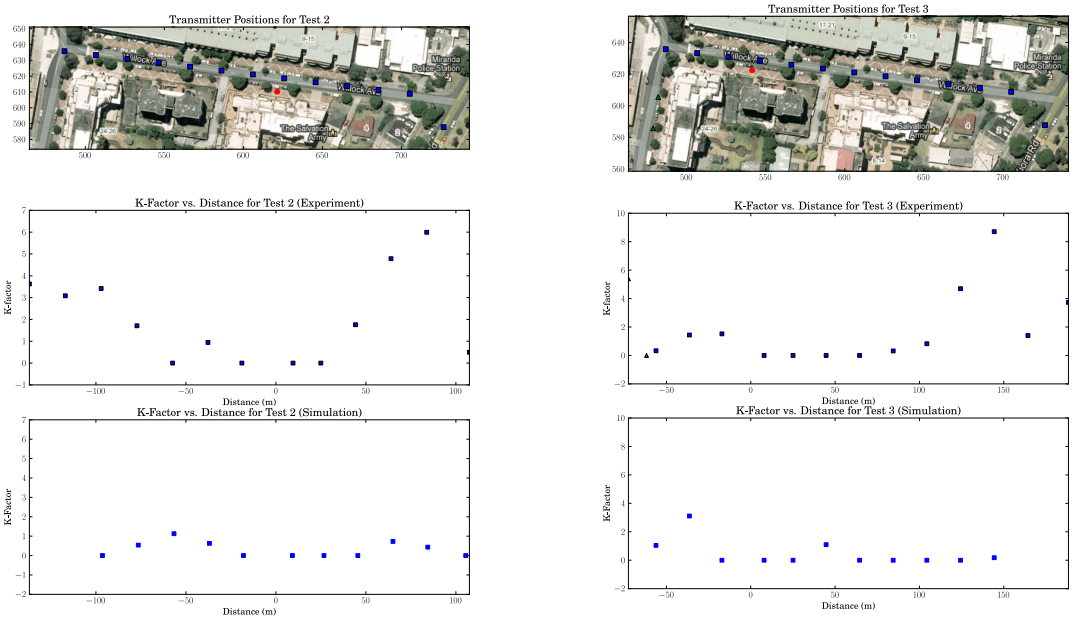
The experiment results indicate that vegetation is absorbing signal power, which prevents it from reflecting off buildings and other objects. As a result, less diffuse power is reaching the destination and the K-factor is thus higher. Future work will entail further experiment to determine the extent of absorption to adequately account for the presence of vegetation in K-factor approximation.

3.7.3 Computational Complexity

The simulation recorded the time taken for each pass of the path resolver at different ray counts, and the results compared with the predicted time complexity given in Section 3.5.3, shown in Figure 3.10.

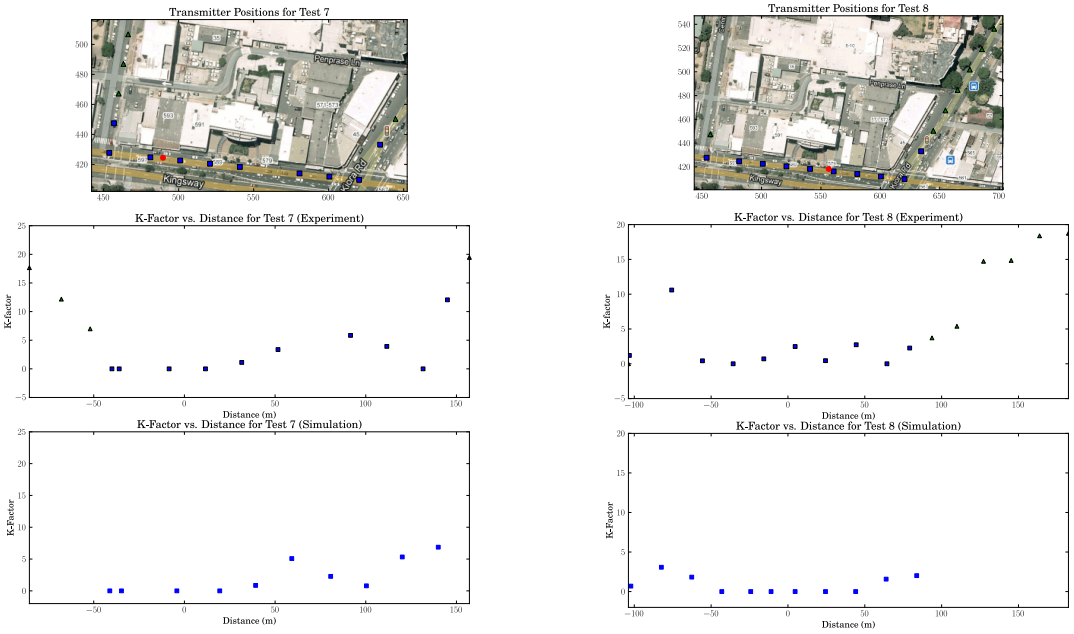
The simulations broadly followed the predicted execution time. It should be noted that in the simulation, only one car was transmitting with no acknowledgement frames being exchanged. In a full VANET simulation, in which hundreds of vehicles are present and transmitting regularly, execution times make simulation computationally infeasible.

Due to the static nature of buildings, the ray-launcher may be moved into a pre-simulation step. During the preparation of the road network from OSM data, a separate program may calculate and store K-factors for pairs of transmitter and receiver positions to be indexed in simulation. In so doing, the ray-launcher need only be run once, and the file retained along with the rest of the road and traffic load data.



(a) Location 2

(b) Location 3



(c) Location 7

(d) Location 8

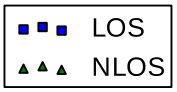


Figure 3.9: K-Factor versus distance for four separate scenarios. The red dots indicate the position of the receiver. The gaps in the lower scatter plots correspond to points where the simulation K-factor reached values higher than observed during the experiment.

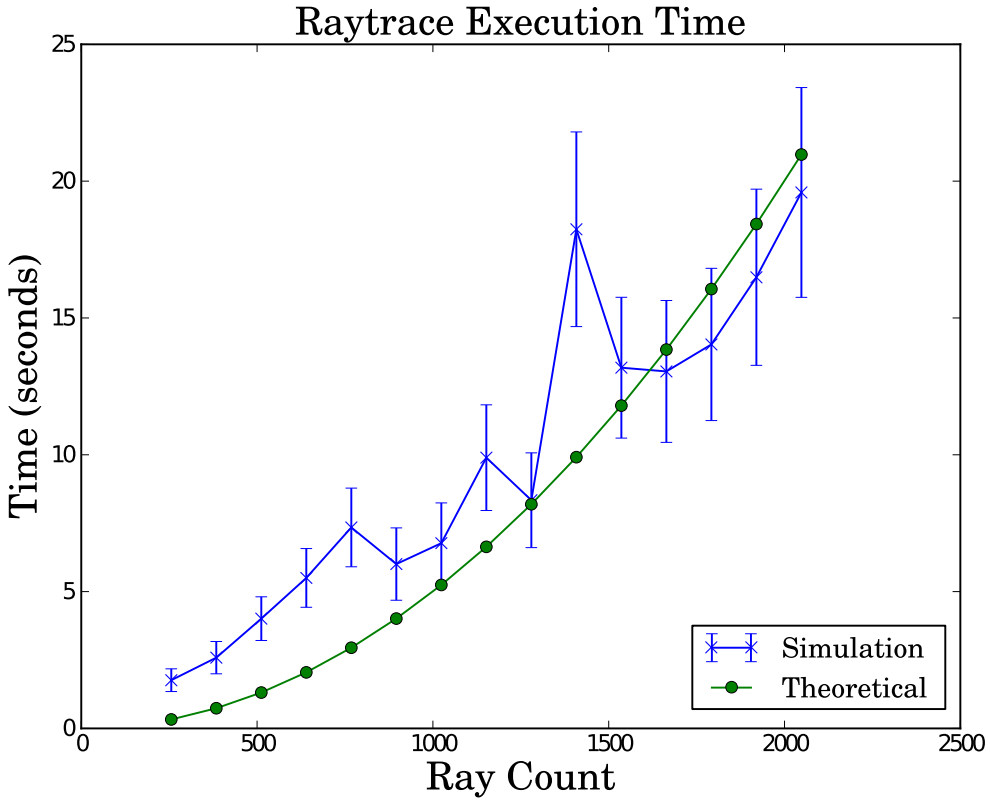


Figure 3.10: Comparison between actual simulation time and analytic estimate of simulation time. One significant outlier is observed for 1408 rays.

3.7.4 Assessment

This section presented a method for generating an approximation of the Rician K-factor through a simple ray-casting model of the propagation environment. Experiments have been performed in a vehicular environment, both in the presence and absence of vehicles, revealing certain properties of the K-factor with respect to the environment. Vegetation appears to exert a shadowing and absorption effect rather than scattering. A particularly interesting result is that intersections result in an increase in the magnitude of the K-factor, which appears to be a result of diffuse signal power escaping and becoming unavailable to the receiver. These experiments, in conjunction with simulation work, support the application of the proposed the K-factor approximator.

3.8 Combined Model Performance

This chapter identified three independent components of the channel, and showed that each can be modelled via different processes. These components and their corresponding models are:

1. Path Loss - Accounted for using the CORNER model proposed by Giordano et al [10]. The implementation by Mukunthan et al [12] is preferred.

2. Vehicular obstructions - A deterministic model based on Knife-Edge shadowing was developed and verified experimentally by Wang et al [11].

3. Fading - An environmentally dependant fading model was presented, which uses the building layout to estimate the multi-path components and estimate the parameters of a Ricean fading model.

Now that a means of modelling each of the three attenuation sources has been identified, they are combined into a complete model called Urban Radio Channel (URC) [b]. The implementation in OMNeT++ was compared with the experiment results gathered from Burelli Street.

[b]The source code for URC has been released online under GLGPL. It can be downloaded from `https://github.com/cscooper/URC`. A description of its usage and the data preparation tools has been included in Appendix B.

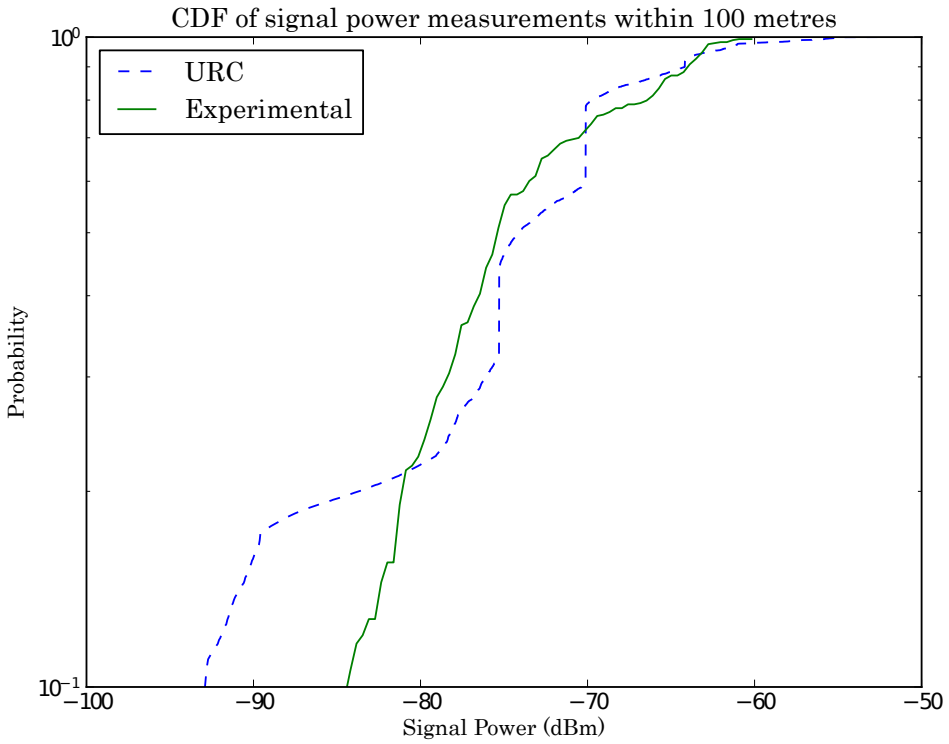


Figure 3.11: CDF of signal power within the first 100 metres.

Figure 3.11 shows the CDF of received power within the first 100 metres. The plot shows a close agreement between experiment and simulation for the physical layer at close range, although the simulation tends toward lower signal power values. Figure 3.12 shows the trends in PDR against distance at close range. The presence of vehicles causes irregularity in the distribution of signal power and the corresponding sharp drops in PDR.

3.9 Discussion

URC will be particularly useful in improving the accuracy of simulations for short-range VANET applications, for example, in simulating clustering and RSU association. It is best suited to modelling built-up urban environments with minimal vegetation, such that the only environmental objects of great influence are buildings and vehicles. The principle bottleneck in such use is the prohibitively time-consuming ray-launching algorithm which would need to be repeated for each transmission. However, it is likely that further opportunities for optimisation still exist, which will

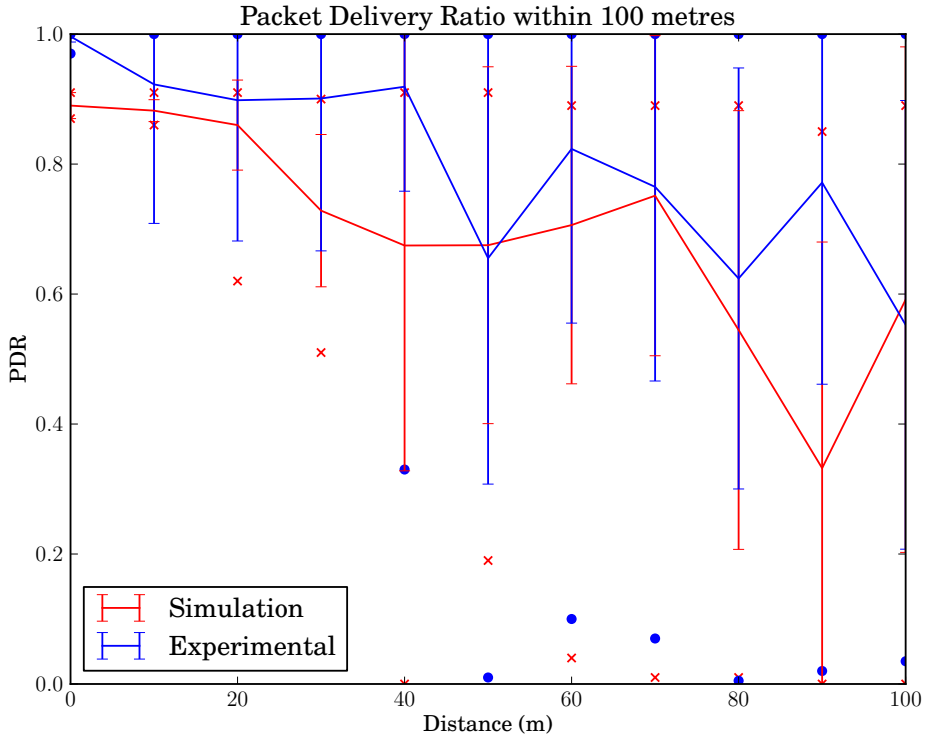


Figure 3.12: The addition of shadowing improves the accuracy of PDR estimates at the MAC layer.

reduce the necessary simulation time.

Other planned future work (not covered in this thesis) includes the incorporation of a wider range of phenomena to be considered in the approximation. This includes smaller static objects, such as lamp-posts which are more likely to scatter signal power than reflect it. Vegetation should also be considered in order to improve the approximator's applicability in a wider range of environments. The antenna's location and radiation pattern also should be considered, as well as V2I properties, since both will influence the behaviour of the VANET channel [14].

As the ray-launching algorithm was chosen purely to prove the concept of K-factor approximation via path resolution, another avenue of improvement is to research different methods of acquiring the necessary multi-path data. Ray-tracing using graphics hardware has been extensively researched for modelling all elements of the channel [104, 110, 127]. Due to the much smaller scale objective of modelling K-factor dynamics, simplified approaches can be used to obtain multi-path component data with faster execution times.

This model will now be applied to the problem of VANET clustering. The

following chapter will contain a detailed simulation survey, identifying clustering techniques that work under the more realistic channel model. In so doing, the model has helped to further clarify the advantages of adequately accounting for the eccentricities of the VANET channel.

Chapter 4

Influence of Channel Model

4.1 Introduction

The Urban Radio Channel (URC) was presented in Chapter 3. The model consists of three parts: a deterministic path loss model to account for diffraction and reflection effects around building corners; a large-scale fading model to account for the presence of vehicles; and a stochastic small-scale fading model that uses a deterministic path-resolution algorithm to approximate its parameters. It is expected that this will give a much more accurate depiction of protocol performance over simpler models.

Additionally, Chapter 3 delineated the consequences of ideal modelling. Protocol performance can be over- or under-estimated, the resultant comparative analysis may be insufficient or misleading, and, as a result, effective cluster strategies cannot be adequately verified. With the URC model, the extent of these issues will be examined.

The study presented in this chapter will compare the performance of three clustering algorithms under FSPL and URC. From the survey presented in Chapter 2, three algorithms were selected as representatives of the various approaches to the clustering problem. MDMAC [31] was selected as the weighted metric approach, using three different weights: Lowest ID (LID), Highest Degree (HD), and LSUF [66]. This approach is compared with the precedence-based hierarchical mechanisms of RMAC [48]. A cluster head hand-off scheme and driver intention analysis is represented by AMACAD [60, 61]. This section will elaborate on the workings of these algorithms, which were taken from the survey in Chapter 2. It will then describe

the simulated highway scenario and the performance metrics to be measured. The results will be presented in terms of channel influence, and used to verify the two consequences of insufficient modelling described in Section 3.1.

4.2 Surveyed Algorithms

4.2.1 MDMAC

Among the algorithms surveyed in Chapter 2, Modified Distributed Mobility-Aware Clustering (MDMAC) [31] was classified as a weighted metric algorithm using periodic beacons for neighbourhood discovery, cluster formation, and maintenance. A more detailed description of its workings is presented.

All nodes begin in an unclustered state, and begin broadcasting their data to other vehicles in the vicinity while listening to beacons from neighbours. Then, based on the data obtained, it computes a CH suitability level according to the function $W_i(\tilde{p})$, where $\tilde{p}$ is the vector of suitability parameters. The higher the value of $W_i(\tilde{p})$, the greater node i's suitability to be CH. The metric is appended to all following broadcasts and recomputed upon the reception of new data.

Upon receiving a data broadcast, a node will insert the received data into a neighbour table, from which the highest-scoring neighbour in the CH state is selected. A JOIN frame is sent to this neighbour, and the node immediately enters the clustered state. If no neighbour exists with a suitability level higher than that of the node, the node will enter the CH state. When a CH receives a JOIN frame, it inserts the ID of the joining neighbour into its cluster table. Periodic broadcasts are then used to update the cluster table, advertise the existence of CHs, and maintain existing CMs. When a CM does not receive a broadcast from its CH within the period, it returns to the unclustered state; similarly, a CH removes a CM that has remained silent within the period. A CH that has lost all its CMs returns to the unclustered state.

In addition to the value of $W_i(\tilde{p})$ and the CH status, joining nodes use a freshness function to assess the suitability of a prospective CH. This function is essentially a Link Expiration Time (LET) computed from the relative velocity of the two vehicles. A CH's freshness value must be greater than a set threshold for it to be considered

a CH candidate.

Functions for $W_i(\tilde{p})$

One of the useful features of MDMAC is that different functions and data can be used for the computation of $W_i(\tilde{p})$. This allows the investigation of different CH suitability criteria on the performance of the algorithm. In this chapter, three separate suitability functions will be used to demonstrate the influence of channel models on comparative analysis.

1. **Lowest ID:** This is among the most popular metrics for MANET algorithms, having been applied since the work of Gerla et al [23]. Each node is assigned a unique ID, which, in a real life scenario, would likely be the device MAC address. This ID is then used as the CH suitability score.

   The principle advantage of LID is the simplicity of implementation. Vehicles need not collect information about their speed or heading, and channel bandwidth consumption is at a minimum, as vehicles need only send their IDs, which are already contained in the source field of MAC frame headers. Clusters implementing this metric in Random Way-Point mobility scenarios exhibit good performance in terms of lifetime and CM stability.

   In vehicular scenarios, however, this approach may not serve the objectives of the network. The assignment of ID is entirely arbitrary and does not reflect more pertinent suitability metrics, such as relative placement of neighbouring vehicles. Furthermore, the mobility of VANETs is constrained to mostly linear movement, and lacks the randomness of RWP scenarios. In this light, Lowest ID is more suited as a final means of distinguishing CHs that have similar scores according to other parameters.

2. **Highest Degree:** In the literature, this is also referred to as Highest Connectivity, and is the number of neighbours with which a node has contact. In the context of MDMAC, this would be the size of a node's neighbour table (provided data is removed if it is not regularly updated).

   HD attempts to ensure a CH with as many neighbours within its effective range as possible. As a result, larger clusters will be promoted, which means

more service coverage for applications such as cellular integration, traffic data collection, and collision warning. The converse is that, while a CH will collect as many nodes as possible into its cluster, there is little attention to the quality of the communications link to those nodes. Therefore, a high-degree CH will have some CMs with a poor communications link, which it will be unable to serve adequately. Additionally, mobility of nodes will become a problem as link quality will fluctuate. So, while cluster size may improve under HD, cluster stability may be lower.

3. **Lane-Sense Utility Function:** LSUF [66] is more complex in its calculation than LID or HD, and uses multiple metrics. The most unique aspect of this metric sum is its analysis of driver intention through the turning direction at the next intersection.

   LSUF defines a traffic flow as the turning direction at the next intersection. If a lane goes straight, it belongs to the straight flow; likewise for those going left or right. A lane which directs traffic in more than one direction belongs to its own separate flow. The weight of a lane, L_k, is computed as the ratio of lanes belonging to the flow f_k – of which lane k is part – to the total number of lanes on the road. The three components – degree, average internode distance, and average relative velocity – are multiplied by this lane weight; the result being that greater importance is assigned to nodes that have more closely-knit neighbours with similar velocities in the most prominent traffic flows.

   This metric has advantages over LID and HD. It uses three different metrics, all of which have relevance to improving cluster stability and longevity. The connectivity component offers the advantages of HD, while the distance component promotes CHs whose neighbours are more closely grouped, meaning the links between them are likely to be stronger. Finally the velocity component ensures a cluster whose members wil remain close to the CH for longer. This combination beats LID in terms of applicability to VANETs, and attempts to overcome the disadvantages of HD.

   The disadvantage of this approach, as was alluded to in Section 2.3.1, is that the metric gives a global suitability value without respect for individual unsuit-

ability. Moreover, a node may compute a high value for one of the components, and very low values for the others, and yet the sum will still put its score above its neighbours. Depending on which component dominates, the cluster could become unstable, very small, or last only a short period of time.

4.2.2 RMAC

Robust Mobility-Adaptive Clustering (RMAC) [48] was classified in Chapter 2 as a precedence-based hierarchical algorithm. The affiliation mechanism uses unicast JOIN requests with handshaking, and periodic unicast polling for maintenance.

Like MDMAC, nodes begin in an unclustered state. After obtaining neighbour information through Inquiry messages, the nodes sort neighbours in order of CH suitability according to distance, relative velocity, CH status, and cluster size. The best scoring node is sent a JOIN request, to which the prospective CH responds with either an accept or deny frame. A denial will be sent if the current size is equal to the connection limits set by the algorithm. Once a node has joined a cluster, the CH will periodically poll its members to ensure their presence and update neighbour data. The maintenance phase is then very similar to MDMAC, in that a CH will drop non-responsive CMs from the cluster and end its CH role if it has lost its members.

RMAC is distinguished from other algorithms by use of hierarchical structures, in that a CM of one cluster can be the CH of another. This allows the algorithm to work actively toward merging instead of trying to prevent it. Also, merged clusters maintain their own structure, instead of all CMs becoming members of another cluster. Such an approach makes it suitable for routing in VANET environments.

This algorithm is more complex than MDMAC, which relies solely on broadcast frames. Maintenance of RMAC clusters requires a high number of transmissions, which have the potential to interfere with each other and impair the performance of the network. Also, stochastic channel behaviour and obstructing vehicles can impact operation.

4.2.3 AMACAD

Adaptive Mobility-Aware Clustering Algorithm based on Destination (AMACAD) [60] is a weighted metric algorithm that shares some traits with both MDMAC and RMAC. Like MDMAC, it's neighbour discovery and maintenance schemes use Hello frames; like RMAC, it requires a handshake for affiliation. However, it has two unique features:

1. It's election criteria is a function of distance, relative velocity, and distance between its destination and those of its neighbours. This is along the same thread as LSUF, in that it attempts to consider driver intention in its analysis.

2. AMACAD adds an additional phase to the affiliation scheme, in which a CH assesses the suitability of a joining node to usurp the CH role. If the new node is a better candidate, the CH will instruct all the nodes in the cluster to join the new CH.

4.2.4 Discussion

The advantages and disadvantages of these clustering strategies are assessed through simulation. This analysis will show the importance of the channel model in adequately differentiating the most suitable approaches, while indicating how an ideal model will inaccurately estimate the performance of these methods on their own.

4.3 Description of simulation

The study presented in this chapter use the same simulation platform as described in Section 3.6.2. The K-factor approximator described in Chapter 3 was moved into a pre-simulation step, and the computed fading statistics were loaded into the simulation at runtime (Appendix B.3.3 contains a description of how this data is stored). Since the simulations do not source OSM data, building data was constructed from the road map (See Appendix A for a description of this method). There is no application level traffic, as the objective is to understand the clustering performance. Table 4.1 contains the wireless parameters for the simulation.

Parameter	Symbol	Value
Carrier Frequency	f_C	2.412 GHz
Transmit Power	P_{TX}	7 mW
System Loss Factor	L	4875
Thermal noise	T_N	-110 dBm
Bit Rate	R_B	11 Mbps
Simulation Time	T_{sim}	1500 sec

Table 4.1: Wireless simulation parameters. L is a dimensionless factor accounting for signal losses in processing hardware. It was determined from the specifications of the hardware used in the experiments described in Chapter 3.

A highway scenario is used for simulation, with a speed limit of 80 km/h, four lanes, and node density of five vehicles per transmission range. The length of the road is computed such that each vehicle spends fifteen minutes of simulation time on the road. Exits are distributed equidistantly along the length of the highway. At each exit, the vehicle has a 50% chance of leaving the simulation.

4.3.1 Performance Metrics

This study measures the following metrics:

- **Reaffiliation Rate:** The is the rate at which nodes change their CH.
- **Cluster Lifetime:** Clusters are considered dead if the CH loses all its CMs, either to broken connections or reaffiliations to other clusters. The lifetime is the time between when a CH obtains its first CM and it loses its last CM.
- **Cluster Size:** This is the maximum number of CMs a CH has accrued during its lifetime. Larger clusters mean greater network coverage.
- **Faulty Affiliation Rate:** This is the frequency of affiliation malfunctions that can be caused by a lossy channel (see Section 2.3.4).

These measures were chosen for being the most commonly selected metrics for assessing performance in the proposal of new algorithms. The most prominent metrics are lifetime, indicating the stability of CHs; and reaffiliation rate, which

indicates stability of CM connections. An algorithm therefore is considered to offer better performance if clusters last longer periods of time with low reaffiliation rate.

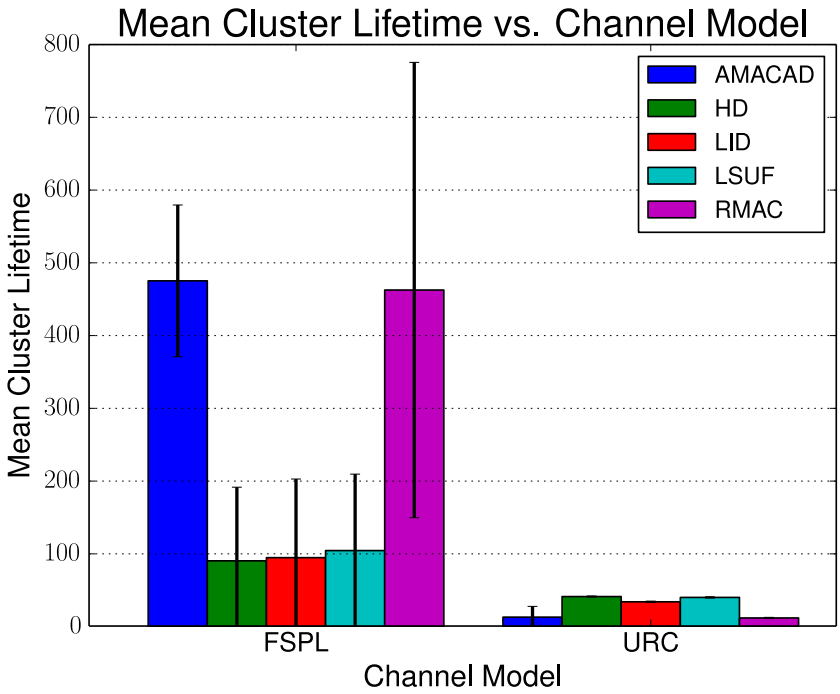
As was alluded in Section 2.3.4, MDMAC's lack of handshaking makes it vulnerable to faulty affiliations. The FSPL model, not accounting for multiple packet loss factors, is expected to present minimal faulty affiliations, if any. Furthermore, algorithms that present fewer faults under URC than competitors will indicate the action of some technique that prevents nodes from attempting to join CHs over a poor channel.

4.4 Results of study

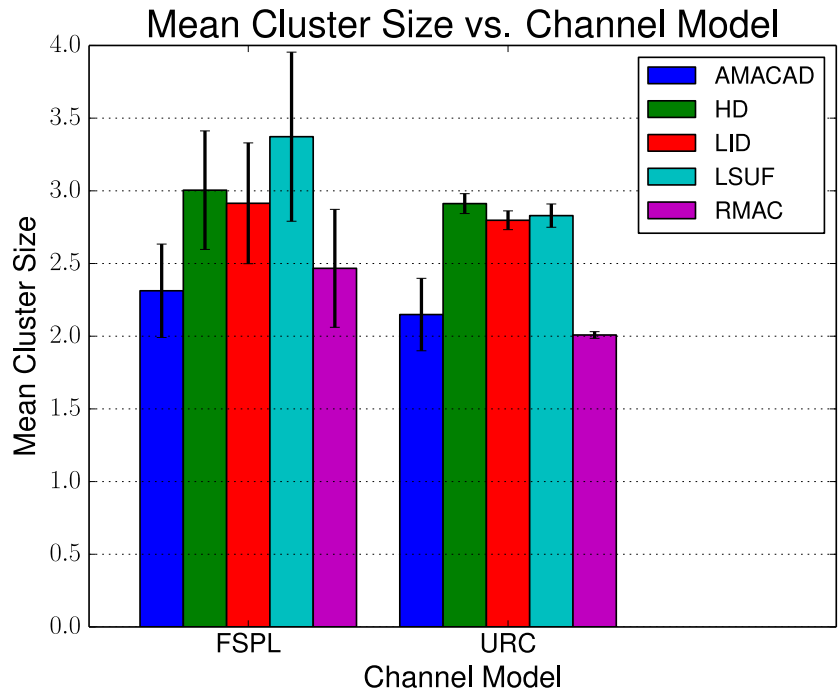
4.4.1 Lifetime and Size of Clusters

The cluster lifetime results are presented in Figure 4.1a. The expected finding is that the algorithms performs significantly better under FSPL. The variance in lifetime is much higher, with clusters lasting as long as three minutes. Since broadcasting is MDMAC and AMACAD's method of cluster maintenance, FSPL makes reception of packets much easier, and data updates are lost infrequently. Additionally, the CHs advertisement frames are able to reach more vehicles with greater success, allowing them to obtain new CMs as old ones are lost. This allows clusters to last longer. With URC, vehicles and multi-path fading interfere with direct transmission, reducing the range and effectiveness of CH coverage, in turn impacting the algorithms' ability to establish and maintain clusters.

However, the effect of the channel model is more than one of scaling. Under FSPL, LID performs better than HD, with LSUF beating both. AMACAD and RMAC are evenly matched in terms of mean lifetime, with RMAC having greater prevalence of longer lived clusters. URC, on the other hand, presents LID performing poorer than both of its competitors. As was discussed in Section 4.2.1, LID is arbitrary and disregards important relevant factors of the network, while HD and LSUF attempt to account for these elements. AMACAD and RMAC, which dominated under FSPL, are now performing worse than MDMAC, possibly due to their more complex affiliation mechanisms. Simulations with URC reflect this inequality properly.



(a) MDMAC clusters last more than twice as long with FSPL, which is more accommodating to MDMAC's maintenance scheme. RMAC and AMACAD also perform very well.



(b) Reduced CH coverage under URC causes smaller clusters, especially with LSUF, RMAC, and AMACAD.

Figure 4.1: Lifetime and size behaviour of the surveyed algorithms between ideal and realistic channel models.

Cluster Size, presented in Figure 4.1b, behaves similarly in response to the realism of the channel. In particular, HD and LID show minimal change in terms of mean cluster size, but have a higher prevalence of large clusters than with URC. The results here show how the effectiveness of CH coverage is affected by channel dynamics. The influence is more pronounced with LSUF, which lends greater importance to CHs belonging to flows that have more lanes associated with them. FSPL allows these CHs to have greater coverage, allowing vehicles in other flows to detect and affiliate with them. URC disrupts communication across flows, and thus clusters will be smaller. RMAC's more comprehensive maintenance mechanism functions better under FSPL, but causes it to suffer a greater drop in size than AMACAD, as its maintenance schemes require more transmissions.

4.4.2 Cluster Stability

Stability of clusters is shown through reaffiliation rates and faulty affiliation, presented in Figures 4.2 and 4.3a. As with lifetime, reaffiliations are much less common with FSPL, in which fewer packets are lost due to channel dynamics. The frequency of reaffiliations are also stable across multiple repeats of the experiment, attributed to the purely deterministic nature of FSPL. URC's combination of deterministic and stochastic elements cause much more reaffiliations, and the incidence was more varied across the experiment runs. An important finding is that URC shows LSUF to function much better than HD or LID, while FSPL shows all three approaches closely matched.

Figure 4.3a shows the incidence of faulty affiliations as a fraction of reaffiliations for MDMAC. For URC, the fractions were very similar between the three algorithms, with LID experiencing slightly more than HD or LSUF. The difference is more pronounced and more interesting under FSPL. LID, surprisingly, causes significantly fewer faults than HD or LSUF, the former experiencing the most. It is unclear why LID is more robust to the malfunction than the others, given that there is no environmental consideration.

Owing to it's handshake-based affiliation strategy, RMAC experienced no faulty affiliations. AMACAD should not have experienced them either but the results presented in Figure 4.3b show a very high incidence of the malfunction under URC.

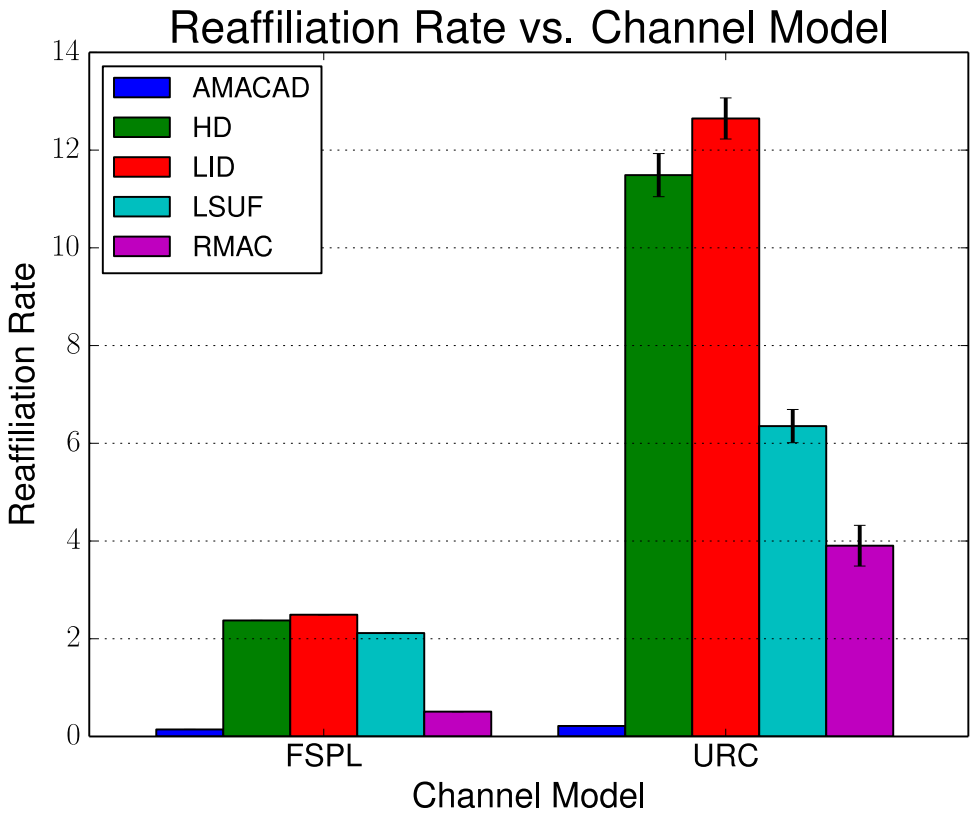
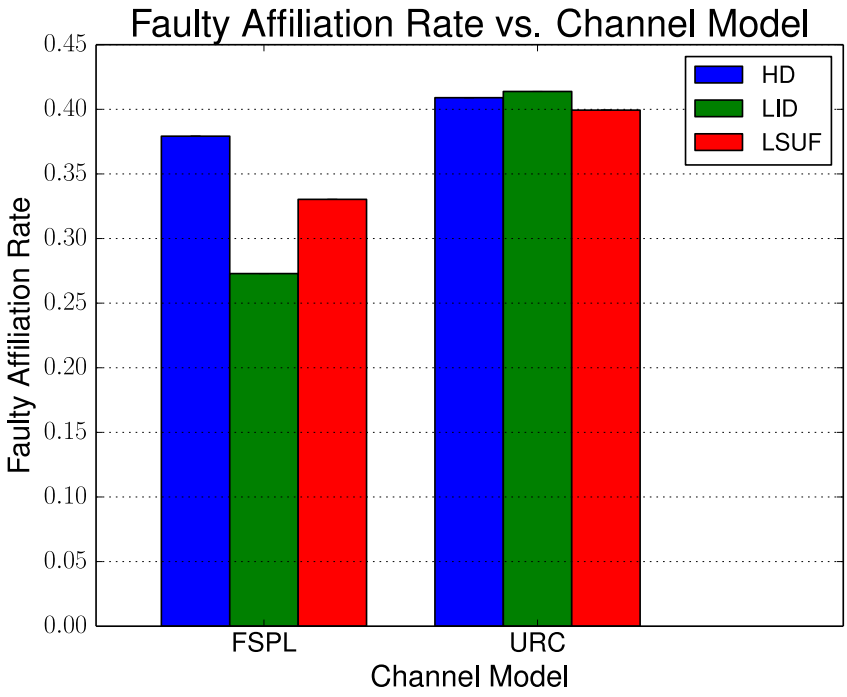


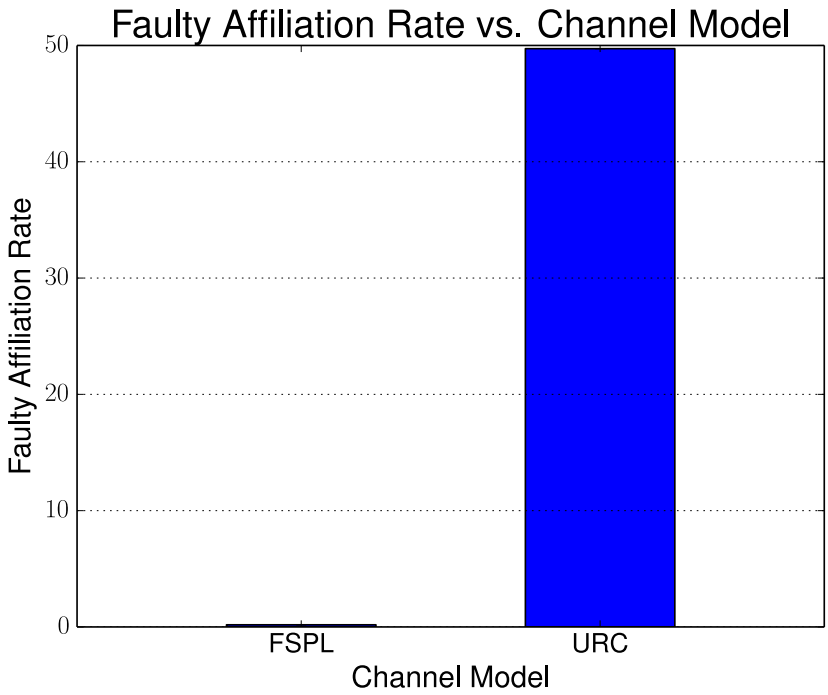
Figure 4.2: Stability is much higher with FSPL, again indicating its accommodating, if unrealistic, environment.

It is notable that under FSPL, the malfunction hardly ever occurs. Inspection of the implementation, as well as the description in the original publications show that AMACAD seeks out the first CH candidate it can find, whereas RMAC sorts according to distance (among other characteristics) before affiliating. As a result, the channel over which AMACAD sends its join requests is more likely to be poor. An affiliation request may reach the CH candidate, but the acknowledgement may not. Thus a CH would believe the requesting node to be part of its cluster, while the node (having not received a response) moves onto the next CH candidate. However, this would not account for the high rate of the malfunction.

AMACAD's unicast-based hand-off scheme does not employ corresponding handshakes. If the initial response, telling the affiliating node to become a CH, is lost due to a poor channel (a high probability due to the aforementioned weakness of AMACAD's handshake scheme), both the old and new CHs will be unaware of it. As the old CH tells its members (via unidirectional frames) to affiliate with the new CH, the cluster collapses. The cluster now erroneously believes itself to be affiliated



(a) Under URC, fraction of faulty affiliations remains mostly stable between the scoring strategies, while the difference is more pronounced under FSPL.



(b) Using FSPL, AMACAD's faulty affiliation rate is non-existent.

Figure 4.3: Stability of the algorithms between simple and realistic channel models.

with an unclustered node that has moved on to its next candidate. Even if the initial response is successfully received, a lost CH transfer message to the CMs can cause faulty affiliations. Since the simulation does not log the CH transfer as a reaffiliation, the incidence of reaffiliation in the above plots is low. In the case of AMACAD, a faulty affiliation is not associated with a failed JOIN message, but a collapse of the clustering structure. The heavy incidence of faulty affiliations indicates AMACAD is highly vulnerable to this unique malfunction, which undermines its applicability in a VANET environment.

4.4.3 Discussion of Channel Model Influence

All the results presented verify the first consequence of insufficient channel modelling: protocol performance can be over/under estimated. In the case of MDMAC, FSPL presented high performance of all three strategies. This is primarily attributed to a channel environment more hospitable to broadcast-based maintenance schemes. The greater CH coverage with FSPL leads to larger and more stable clusters, with a much lower likelihood of faulty affiliation. Similarly, both AMACAD and RMAC functioned very well under FSPL, producing stable long-lived clusters as a result of their establishment procedures. These same procedures, when subjected to fading and shadowing, offered much shorter-lived clusters.

The second consequence of insufficient modelling has also been demonstrated. FSPL gave a much more level playing field, and did not put the algorithm to rigorous tests. The choice of election criteria appears to have much less affect on cluster stability under FSPL, as reaffiliations were very similar compared to the greater disparity under URC. LSUF's use of driver intention through traffic flow analysis enabled it to construct clusters that were more stable under the realistic channel model. Ideal modelling would show only a marginal increase in stability, and be lenient toward the complex formation and maintenance strategies of AMACAD and RMAC. Under URC, they performed less efficiently than MDMAC's simple scheme. URC has revealed that complex, transmission-heavy schemes may be unsuited to the VANET environment compared to more straightforward schemes.

Regarding faulty affiliation, that LID was more robust to the malfunction than HD or LSUF indicates a previously unconsidered facet of this second consequence.

A comparison of LID, HD, and LSUF under FSPL (or a similar ideal model) would demonstrate the small gains achieved by LSUF to be unjustified compared to the straightforwardness of LID, and prompted by the much lower incidence of faulty affiliation, conclude LID to be the better approach. This method would then be built upon, despite its fundamental unsuitability. Under a more comprehensive model like URC, LSUF's advantages would far outweigh the complexity of implementation, and future designs would be constructed around this approach. This scenario is further exemplified by the prevalence of faulty affiliation in AMACAD's results. FSPL showed AMACAD to form long-lived, highly stable clusters, while URC revealed a fundamental flaw in its design that would cripple a network designed around it. A comprehensive channel model detects such flaws, allowing them to be rectified and improved upon.

4.5 Conclusion

Three algorithms were tested under the URC model presented in Chapter 3. The objective was to characterise how the choice of channel model affects the simulated performance of algorithms, thereby verifying the two consequences delineated in Section 3.1. Insufficient modelling has been shown to not only over-estimate protocol performance, but hamper proper comparison and detection of design flaws. Therefore, a clustering approach cannot be fully tested to its limits, and new ideas cannot be honestly investigated. URC gave a better presentation of the protocol performance, and properly identified the efficacious designs of the surveyed algorithms.

In addition to channel modelling, Chapter 2 highlighted that the simulated road scenario and parameterisation of an algorithm may influence its performance. The following chapter will present a parameter study of MDMAC, RMAC, and AMACAD under URC, investigating their behaviour in response to changes in their parameterisations and road structure (e.g. the number of lanes on a road). The study will further elucidate the true performance of these algorithms, and determine the most suitable approach in the given scenarios.

Chapter 5

Influence of Road Structure and Algorithm Performance

5.1 Introduction

Two simulation-based comparisons were presented in Chapter 4, one under FSPL and another under URC. The objective was to determine the insights offered by an ideal channel model, and highlight the unique ones provided by a complex model. The study confirmed the three consequences of inadequate modelling described in Section 3.1, and showed that certain designs function very well under ideal models, but poorly under realistic ones. Moreover, an ideal model may present an even comparative analysis between algorithms that account for environmental factors and those that do not, leading to the conclusion that the perceived small performance gains from environmental analysis are not worth the increased complexity of design and implementation. URC, on the other hand, showed that these approaches are worth the effort, and revealed design flaws that, if rectified, can further improve a protocol's performance.

There are more environmental factors influencing an algorithm's performance, such as lane count, speed limit, and node density. A few of the papers studied in Chapter 2 assessed the influence of these parameters, but it was not a common aspect of verification methodology. Simulation scenarios vary wildly, from simple highway scenarios of different lane counts and speeds to grid scenarios and settings in real-life cities. At present, there exists no standard validation practice for VANET clustering

algorithms that would give reliable comparative analysis or indicate improvement in clustering performance.

This chapter presents a simulation study of clustering in highway and grid scenarios under URC, which has been shown to give a more realistic comparative analysis. The design aspects of each algorithm are assessed under the various scenarios, and the most suitable approaches are highlighted. The influence of road structure parameters (e.g. number of lanes) on protocol performance is quantified, revealing the need for adaptive protocol parameterisation in response to road conditions. The performance of the protocols is compared under URC (see Chapter 3) as well as the commonly-used FSPL model, revealing that realistic models are more suitable for detecting disparity of performance between protocols.

5.2 Simulation

The simulation package used in Chapter 4, VEINS, is adopted for this study. The same transmission parameters are used, and the same performance metrics measured. Please see Sections 3.6.2 and 4.3 for details.

5.2.1 Highway

A highway scenario is used for simulation, with variable lane count N_L, node density ρ_N, and speed limit S_{max}. Vehicles are emitted into the simulation, and accelerate to S_{max}, which is maintained throughout the simulation. The length of the highway is chosen such that a vehicle will take fifteen minutes of simulation time to traverse, and exits are distributed equidistantly along its length. Thus speed influences the mobility of nodes without decreasing the time a node has to cluster with neighbours. At each highway exit, the vehicle has a 50% chance of exiting the simulation.

Vehicle speed is a common independent variable in clustering simulations, as higher speeds can make clusters more unstable. In fact, many algorithms are designed to account for speed deviations of nodes during CH election. The number of lanes on a highway imposes constraints on the permutations of node topology. A low lane count would result in cars forming single-file queues, while a large number of lanes can allow vehicles to move more freely and form more complicated topologies.

Parameter	Value
Dimensions (road segments)	10x10
Road Segment Length	500 m
Lane Count	5
Speed Limit	50 km/h

Table 5.1: Grid simulation parameters

Node density will influence the likelihood of vehicles blocking transmissions. The simulation methodology aims to quantify the effects of these parameters.

Normalising Node Density

Node density is interpreted as the number of vehicles within range of a randomly selected vehicle. The vehicular arrival rate, λ, has then been renormalised in order to prevent speed limit and lane count from unduly influencing node density. Using the vehicular arrival rate, the number of vehicles at time t in a simulation of an N_L-lane highway, is given by

$$N_C(t) = \lambda N_L t$$

The time taken for a car to cross the transmission range is $T_R = 2R_{TX}/S_{max}$, and in this time, ρ_N cars are to be in the simulation. That is,

$$N_C(T_R) = \frac{2R_{TX}\lambda N_L}{S_{max}} = \rho_N$$

Solving for λ gives

$$\lambda = \frac{\rho_N S_{max}}{2R_{TX}N_L} \tag{5.1}$$

This formulation prevents speed or lane count (which are themselves independent) from influencing ρ_N.

5.2.2 Grid

The grid simulations studied the algorithms' behaviour in an urban scenario. This differed from the highway scenario due to the existence of Central Business Districts (CBD) within the urban environment. These act as destinations for incoming traffic

and origins of outgoing traffic. Clustering performance was studied in terms of the number of CBDs, N_{CBD}, and the node density, ρ_N. The parameters of the grid simulation are shown in Table 5.1. The channel and cluster algorithm parameters are the same as in the highway scenario, with the exception of RMAC's Time Threshold parameter, which has been set to 100 seconds.

A set amount of vehicles, ρ_N, are spawned in the simulation when it begins. Each vehicle is assigned a destination, randomly selected from the set of CBDs. The vehicle then makes its way to the CBD, at which point it routes back to its starting position before selecting another CBD as a new destination. This is a variant of Random Way Point that selects destinations from a predetermined set, the goal being to emulate the traffic patterns of city environments (for instance, people leaving their home to travel to work). Increasing the value of N_{CBD} increases the number of potential destinations and thus the diversity of possible routes vehicles will take. The exception is $N_{CBD} = 0$, in which case any street on the road network can be a CBD. This can be considered the worst-case scenario for route similarity.

5.3 Road Structure Influence

MDMAC, RMAC, and AMACAD function on different parameters. The impact of these parameters on performance are discussed in Section 5.5, and for this section they have been kept constant at the values listed in Table 5.2. A description of these parameters can be found in [31, 48, 60] for RMAC, MDMAC, and AMACAD respectively.

5.3.1 Influence of lanes

The highway experiments were performed with 1, 2, 3, and 4 lanes. Results were analysed keeping ρ_N and S_{max} constant at 10, and 40 km/h respectively.

Cluster Lifetime is shown in Figure 5.2a. The large variances, representing one σ, indicate an increasing erratic behaviour in terms of lifetime. However, as lifetime cannot be negative, high variance also indicates a high prevalence of long-lived clusters. This is seen in the results of HD, LID, and LSUF, which show an increase in mean lifetime as well as variance. HD's variance grows until $N_L = 4$,

Algorithm	**Parameter**	**Value**
MDMAC	Initial Freshness	10
	Freshness Threshold	1
	Hop Count	1
	Beacon Period	1s
RMAC	Connection Limits	10
	Distance Threshold	1m
	Time Threshold	1s
	Inquiry Period	1s
	Timeout Period	1s
	Poll Interval	1s
	Poll Timeout	5s
AMACAD	Warning Limit	5
	Speed Threshold	5 km/h
	Time to Live	5s
	Destination Weight	0.33

Table 5.2: Cluster algorithm parameters

where it shrinks. This shows that long-lived clusters are less prevalent, indicating that the topological constraints imposed by lane count have become lax, and thus vehicles have fewer strong connections and clusters are less stable.

For MDMAC, HD tended to have the longer lived clusters, except when $N_L = 4$, where LSUF scores better. A low number of lanes means LSUF is unable to distinguish whether a car will turn right, left, or travel straight at an intersection, as a single lane may carry cars belonging to all three flows. When the number of lanes increases to 4, such that there are two distinct lanes travelling only straight, this assigns higher score to vehicles in these lanes. Thus CHs can hold onto their members for longer. The variance increases with lanes, indicating a higher prevalence of longer lived clusters. This may be as a result of more vehicles having direct line-of-sight with potential CHs due to the greater freedom of movement. However, this may also be the cause of poorer performance of HD at $N_L = 4$, as vehicles may be more dispersed and links become more fragile. LID experiences the poorest

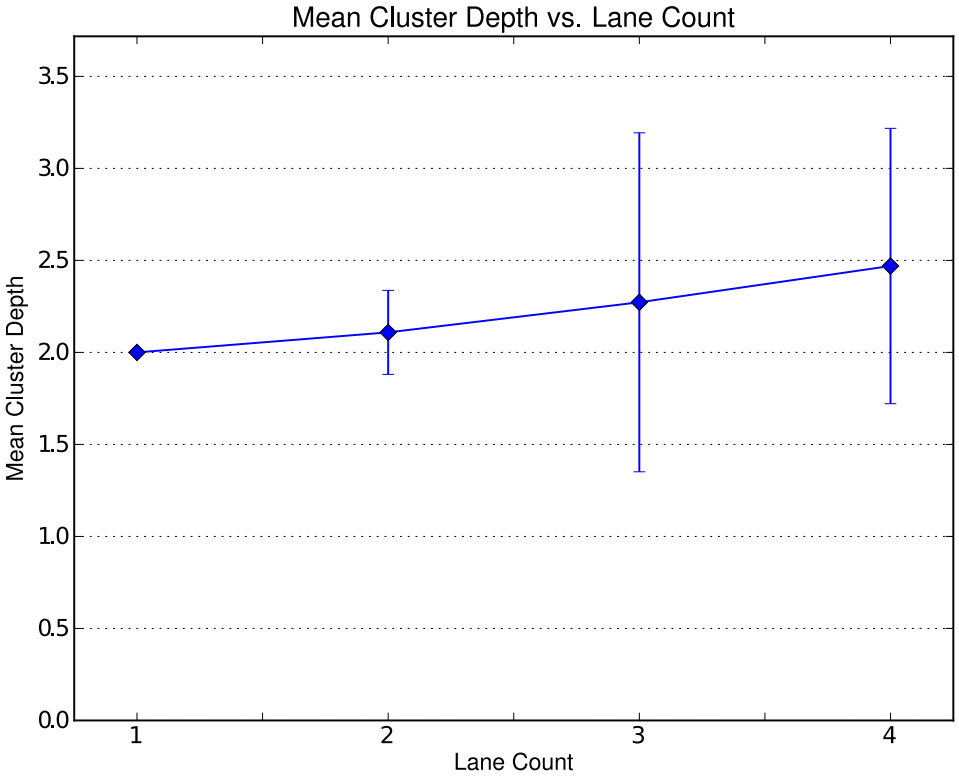


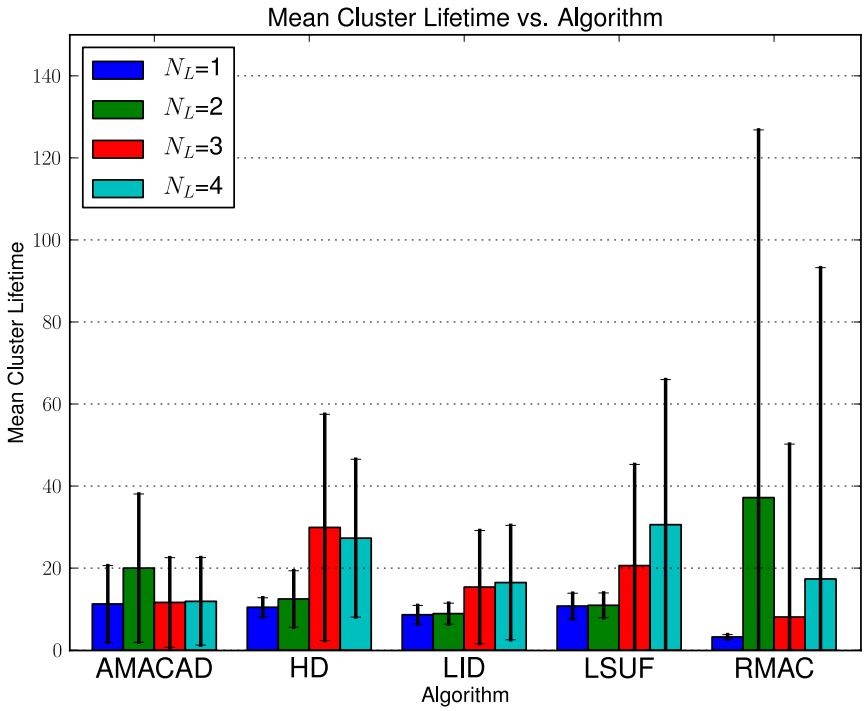
Figure 5.1: RMAC's hierarchical nature produces a backbone with as many as four levels, leading to larger coverage of the network.

performance of all the scoring methods in MDMAC, primarily due to its arbitrary nature.

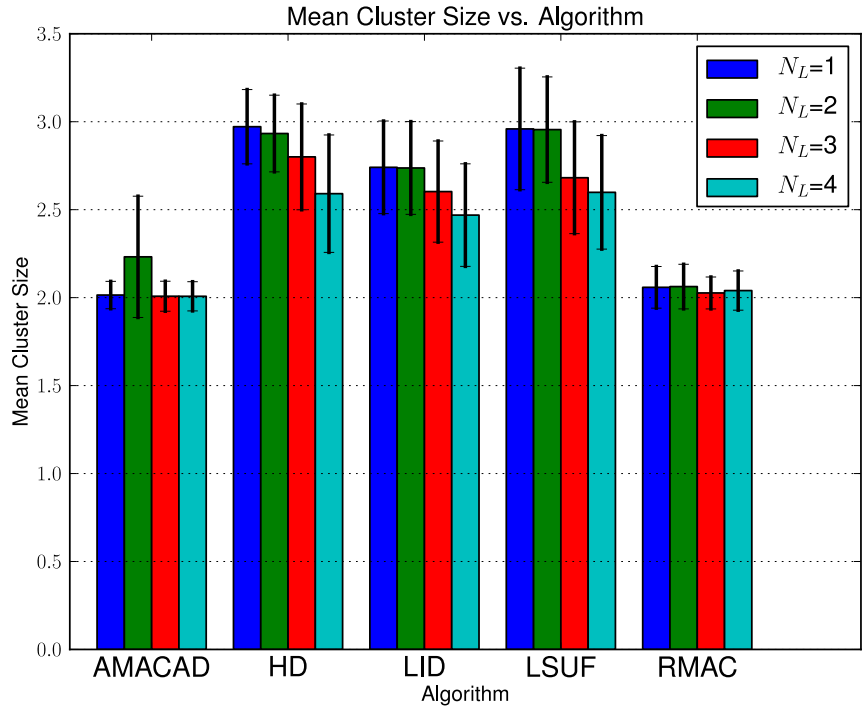
RMAC displays unexpected behaviour with respect to lanes. It appears to perform far poorer than even LID, except when $N_L = 2$. At this point, clusters last significantly longer than with any of MDMAC's variants, with some clusters lasting over two minutes. With more lanes, the trend declines, but the variance in lifetimes remain higher than MDMAC.

AMACAD exhibits similar behaviour to RMAC when $N_L = 2$. With the exception of $N_L = 2$, AMACAD's performance appears largely independent of lane count.

Cluster Size is shown in Figure 5.2b. MDMAC appears to show a downward decline in cluster size with respect to lanes. This also may be caused by the greater freedom of movement, causing vehicles to assume a greater variety of topologies, meaning fewer vehicles will be capable of affiliating with a single CH. Again, the same relative performance scores are seen with LID compared to HD and LSUF,

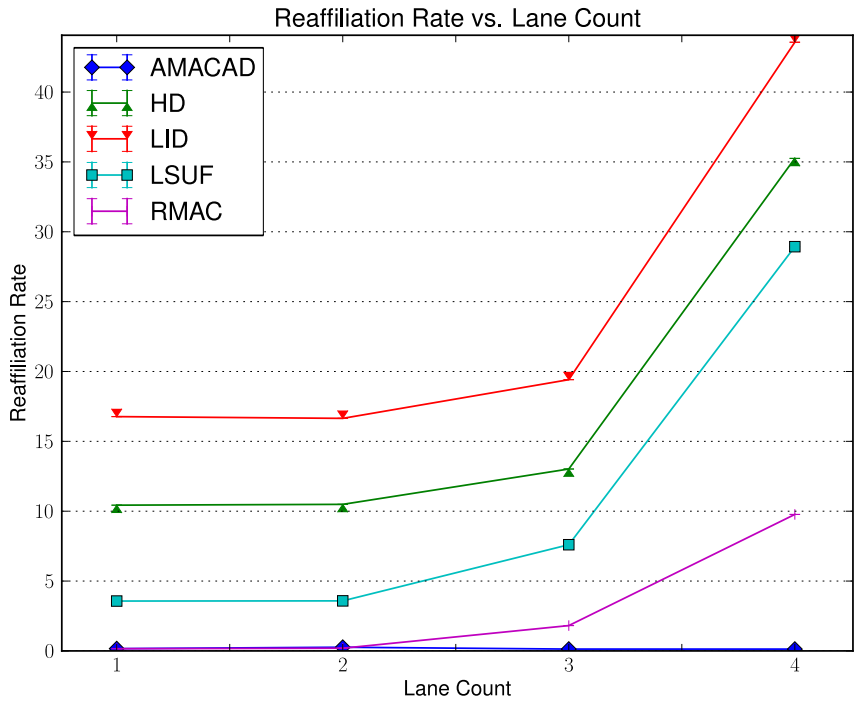


(a) HD appears to provide the longest-lived clusters, except when $N_L = 4$, where LSUF wins. Of the MDMAC scoring functions, LSUF provides the longest-lived clusters due to the use of lane sensing. RMAC however provides erratic behaviour with respect to cluster lifetime. It does tend toward higher lifetimes than MDMAC.

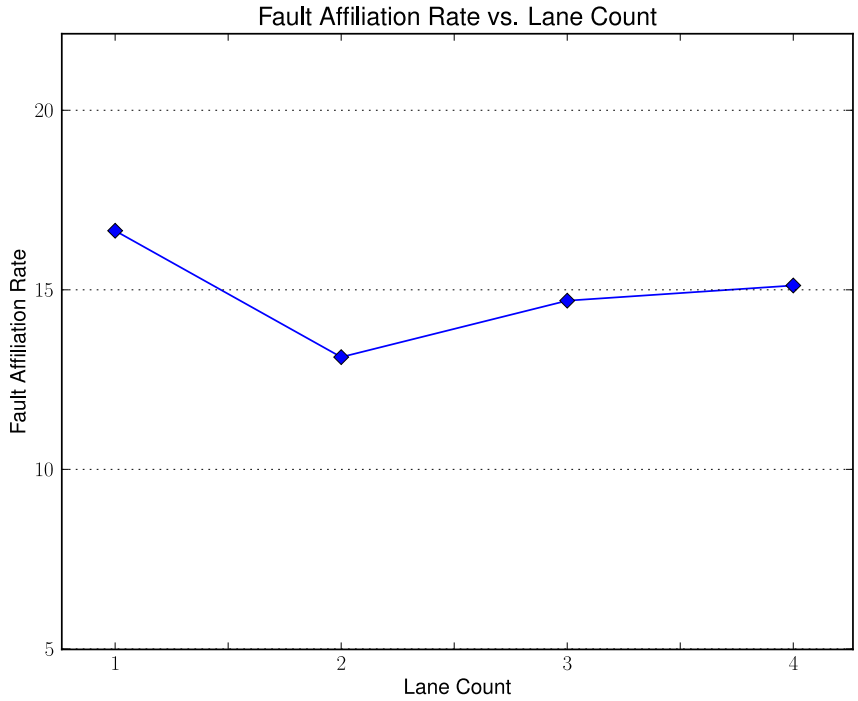


(b) MDMAC is capable of producing clusters of 3 or more nodes, while RMAC data remains tightly around a mean size of 2. However, RMAC's clusters are hierarchical. Although this means the coverage is greater, it also means the structures are snaking significantly.

Figure 5.2: Influence of lane count on cluster lifetime and size.



(a) Member instability increases quadratically with lane count. RMAC is significantly more stable than MDMAC.



(b) AMACAD's handshake mechanism appears not to protect it from faulty affiliation.

Figure 5.3: Influence of lane count on cluster stability.

with LSUF producing larger clusters when $N_L < 3$. Beyond this, HD maintains slightly larger clusters. RMAC produces smaller clusters, and lane count appears to have little impact on the metric. However, RMAC also produces hierarchical cluster structures. The depth of these structures is plotted in Figure 5.1. Cluster depth exceeded four levels, each having on average two members. Thus RMAC allows greater coverage of the network by single clusters. The drawback of this result is that it indicates the clusters are forming snaking structures, as predicted in Section 2.3.1. This would increase the hop-count and delay of any routing algorithm built on top of RMAC.

The Reaffiliation Rate is shown in Figure 5.3a. RMAC and AMACAD provide more stable clusters than MDMAC, due to their more robust and comprehensive maintenance mechanisms. RMAC shows consistently less than ten reaffiliations per second, while MDMAC's CMs were much less stable. In each case, there is an increase in CM instability with increased lanes, again indicating the influence of greater freedom of movement in a vehicular environment. Vehicles are able to change lanes, which can cause moving vehicles to obstruct the direct path to a CH. RMAC's use of hierarchical clustering prevents this, by allowing potential routes around obstructions. AMACAD has the lowest reaffiliation rate of all the nodes, and again appears largely independent of the lane count. However, it should be noted that CH hand-off events do not count toward Reaffiliation Rate in AMACAD, as the nodes have technically not left the cluster, but been asked to transfer to the given CH.

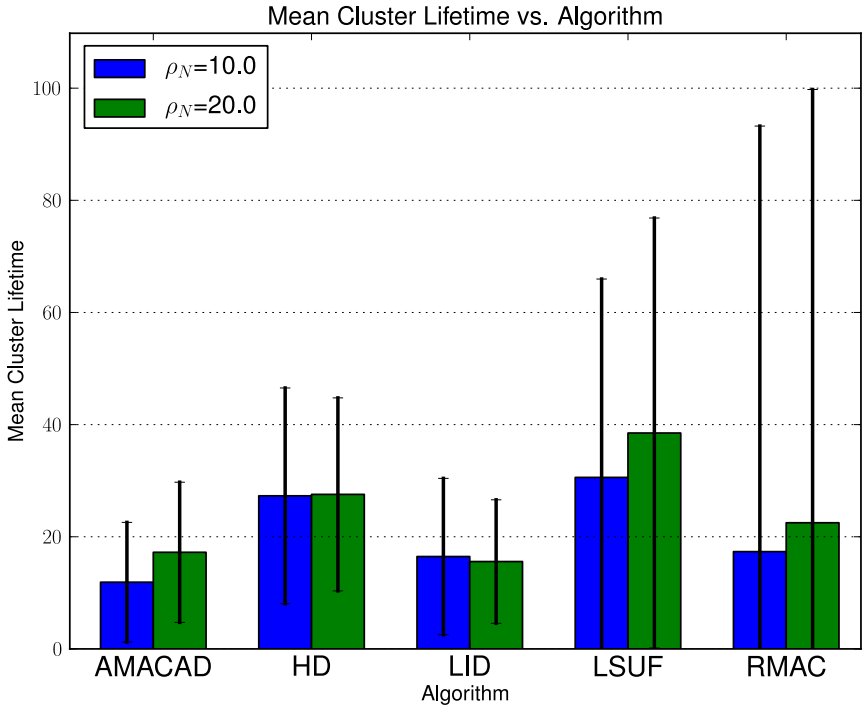
As RMAC has a handshake mechanism, it does not experience faulty affiliations. It was found with MDMAC that as many as 47% of affiliations resulted in a fault. This is a particular problem as the affiliation won't contribute to the cluster size, and the CH thus won't know about it. Furthermore, MDMAC's beacon-based maintenance mechanism causes faulty nodes to be isolated from the network, as the node will continue to hear a CHs signal for up to a minute and will not restart the clustering process. If MDMAC were used as a foundation for data dissemination or routing, a faulty affiliation will isolate potential members from the network and they will be unable to disseminate or receive data. Such a failure would make it inappropriate for collision warning protocols.

As was discovered in Chapter 4, AMACAD's CH hand-off mechanism leads to cluster collapse under realistic channel conditions. The plot in Figure 5.3b shows a very high incidence of the malfunction. The dip in faulty affiliations at $N_L = 2$ can help explain the increase in performance in the other plots. The number of lanes on a road constrains the potential clustering structures. As the lane count increases, there are more degrees of freedom for topology. At $N_L = 1$, the vehicles form a queue, but AMACAD will try to affiliate with the first node it can find, which may not necessarily be the closest. Thus the channel between the two nodes may be poor, increasing the possibility of cluster collapse. However, at $N_L = 2$, the available topologies may be such that nodes have LOS connections to more neighbours, mitigating this effect. As the lane count increases, the inter-node distance increases again, cancelling out the benefit of LOS connections. From a topological perspective, $N_L = 2$ appears to be ideal for AMACAD, as well as RMAC. However, their highly complex maintenance schemes result in lower performance in comparison with the others.

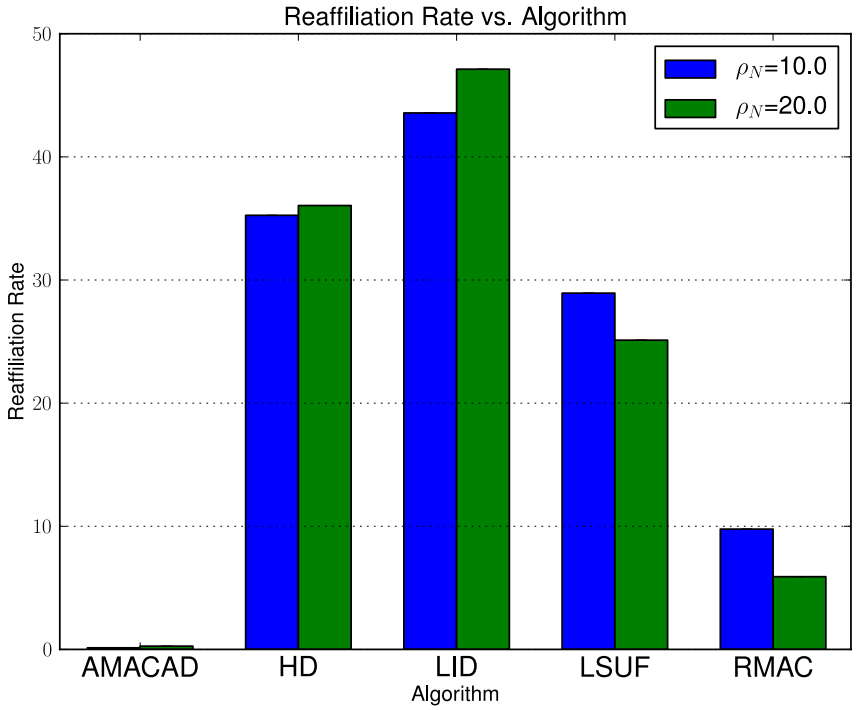
5.3.2 Influence of Node Density

Increasing the density of nodes would, in principle, increase the number of potential CHs as well as the number of vehicles able to associate with them. Conversely, increased node density may also increase the likelihood of vehicular obstructions, causing greater incidence of reaffiliations and possibly faulty affiliations. To study this metric, the number of lanes was kept at 4 and the speed at 40 km/h.

Figure 5.4a shows the influence of node density on cluster lifetime. It appears that HD and LID experience a decrease in the occurrence of long-lived clusters, while LSUF increases in lifetime. RMAC still performs poorer than MDMAC in terms of mean lifetime, but it has a greater occurrence of long-lived clusters, which increases with node density. As LSUF attempts to account for more factors, including relative proximity and similarity of speed of vehicles, a higher node density will lead to more samples from which to compute the eligibility level and thus a greater ability to distinguish between suitable CHs. RMAC's node precedence algorithm also appears to reflect this behaviour. HD drops in performance in part because it only considers the number of vehicles within range of a node, and neither their spatial distribution



(a) The prevalence of long-lived clusters in HD and LID is reduced by increased node density. Meanwhile, LSUF and RMAC improve in cluster lifetime.



(b) HD and LID experience an increase in reaffiliation rate, while LSUF and RMAC's clusters increase in stability, due to their CH selection criteria.

Figure 5.4: Influence of node density.

nor the distribution of their speeds. The same can be said for LID.

This point is illustrated in Figure 5.4b. There is a decrease in CM stability for HD and LID, while LSUF and RMAC improve. This result highlights the advantage of accounting for vehicular dynamics in the CH selection criteria.

There was little change in the sizes of clusters for increased density, indicating that, while more nodes allows for more potential CHs, the shadowing effect of vehicles may prevent vehicles from successfully joining a cluster.

AMACAD's clusters lasted longer with increased node count, in a similar manner to LSUF and RMAC. Higher node density results in a lower probability of cluster collapse, as nodes are closer together and thus affiliation requests are less likely to fail or trigger the malfunction. Moreover, CMs are likely to be closer to their CHs, further reducing the chances of control message loss.

5.3.3 Influence of Vehicle Speed

LSUF and RMAC account for differences in vehicle speed for their CH selection criteria. Thus, simulations were carried out to judge the effectiveness of these schemes to promote stable clusters in different mobility scenarios. Analysis revealed unexpected behaviour from both RMAC and the variants of MDMAC, which is shown in Figure 5.5. The graph indicates an increasing lifetime with speed for MDMAC and AMACAD, which was not expected. RMAC also experiences a performance drop.

The change of Reaffiliation Rate with respect to speed, shown in Figure 5.7a, indicates that CM stability increases with higher speed. While the observation may be counter-intuitive, it can be explained as the functioning of the MDMAC's freshness function and RMAC's node precedence algorithm. A node intending to turn will slow down before the exit, while others travelling straight will overtake and proceed. Both algorithms, using two different mechanisms, attempt to avoid or excise CH candidates that have a relative speed greater than a specific cut-off (in the case of MDMAC, this is the Freshness Threshold; for RMAC, this is the Time Threshold - see Table 5.2). When the maximum speed is 40 km/h, the difference in speed between a vehicle turning and one travelling straight is much smaller than when the speed is 80 km/h. Therefore, faster travelling vehicles will be less likely to associate with cars that are exiting the freeway.

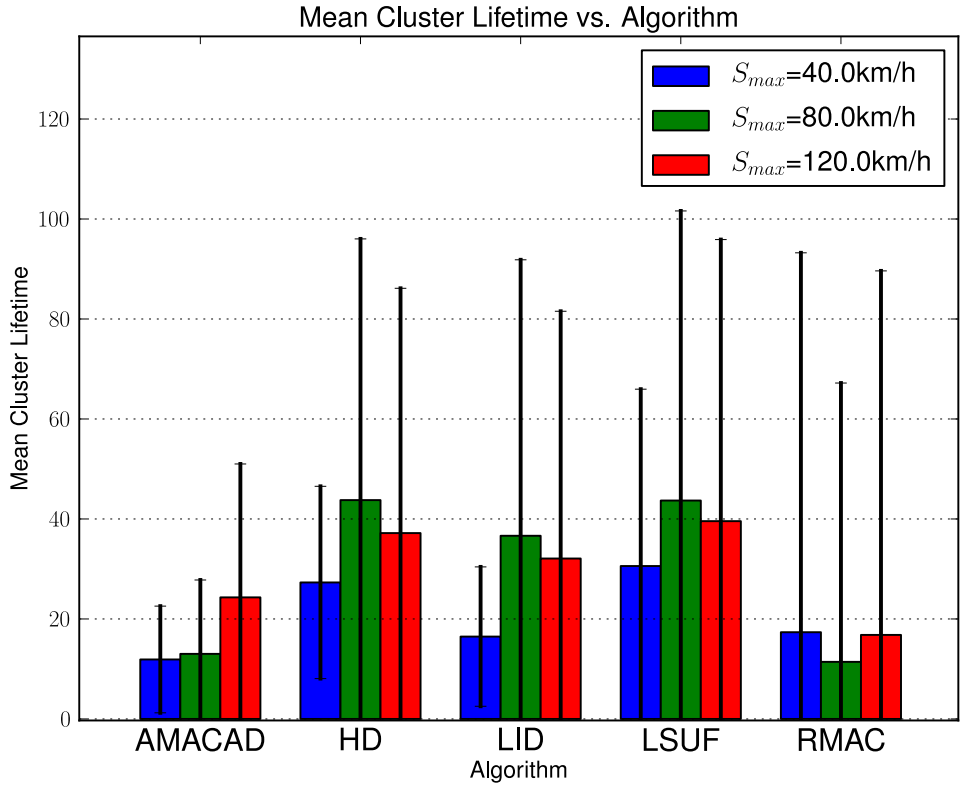


Figure 5.5: Increasing speed appears to increase the longevity of MDMAC and AMACAD clusters, while decreasing the lifetime of RMAC.

AMACAD's Reaffiliation Rate remains small in the simulation. Figure 5.7b shows a considerable increase in Faulty Affiliation Rate, indicating a decrease in stability of AMACAD's handshake scheme.

5.4 Grid Structure Influence

The simulations presented in this section study the performance of the selected algorithms on a grid scenario in order to mimic an urban scenario. The grid is 10x10 intersections, each road link being 500 metres in length with 5 lanes each. As this grid was generated using the *netgen* tool, it lacked building outlines. The *BuildingSolver* tool was used to generate outlines of buildings made of brick (See Appendix A for details).

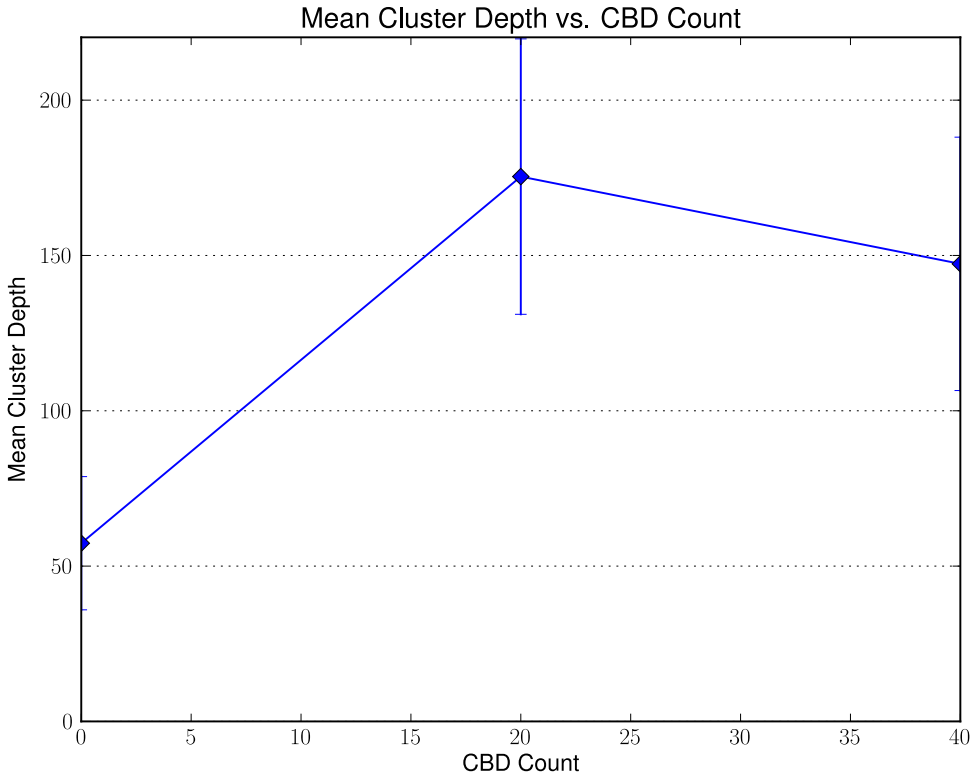
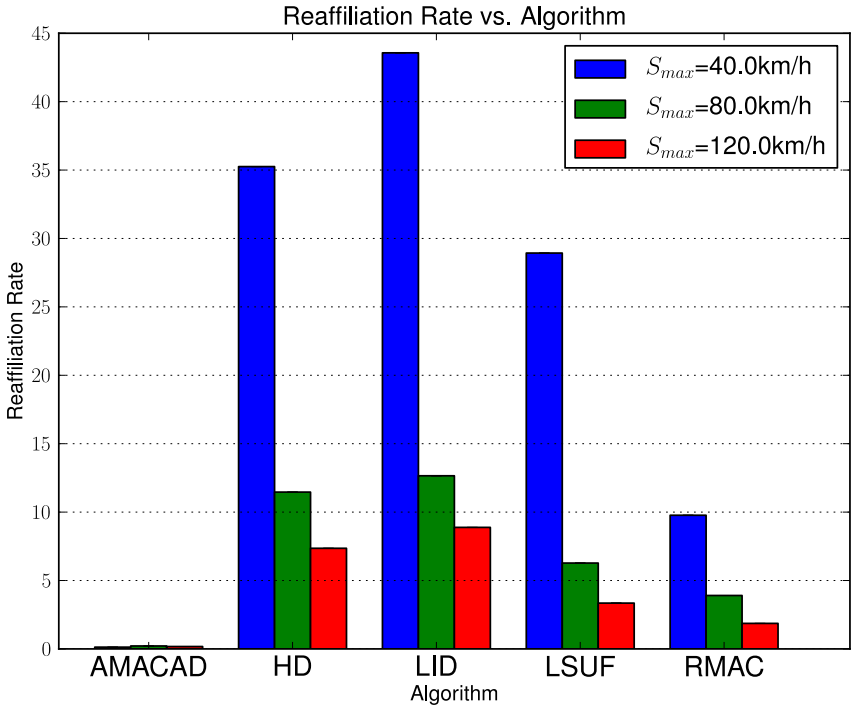
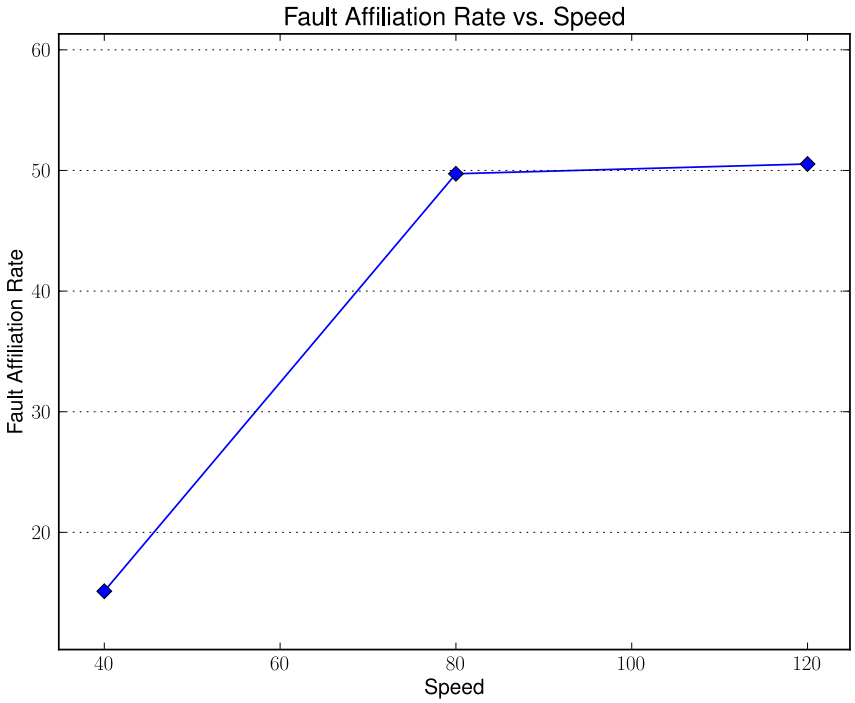


Figure 5.6: High variance of routes inhibits the formation of deep cluster structures.

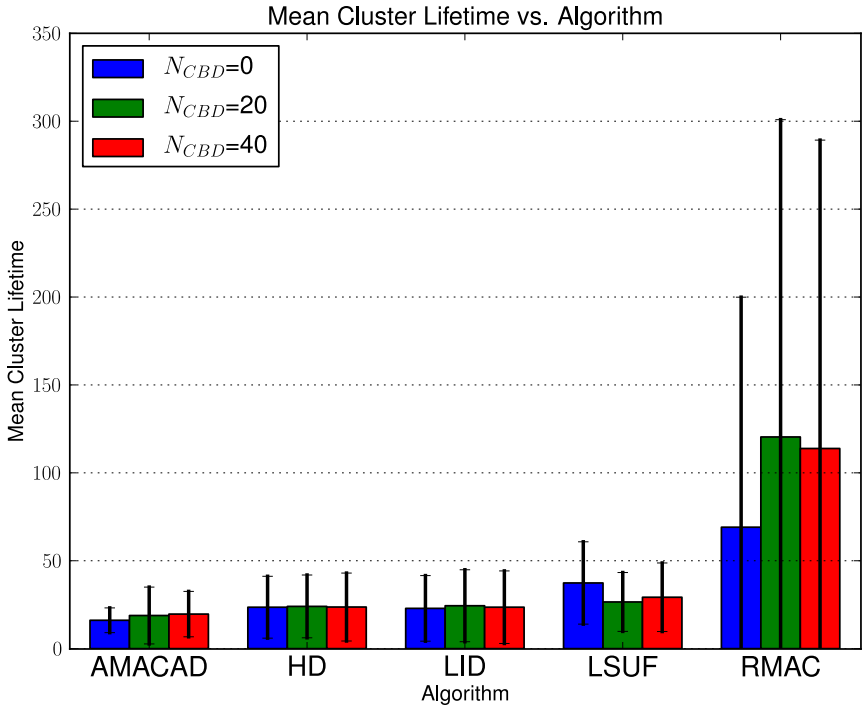


(a) Reaffiliation rate appears to drop with increasing speed. This is likely due to a greater difference in speed between vehicles slowing to turn at exits and those travelling straight.

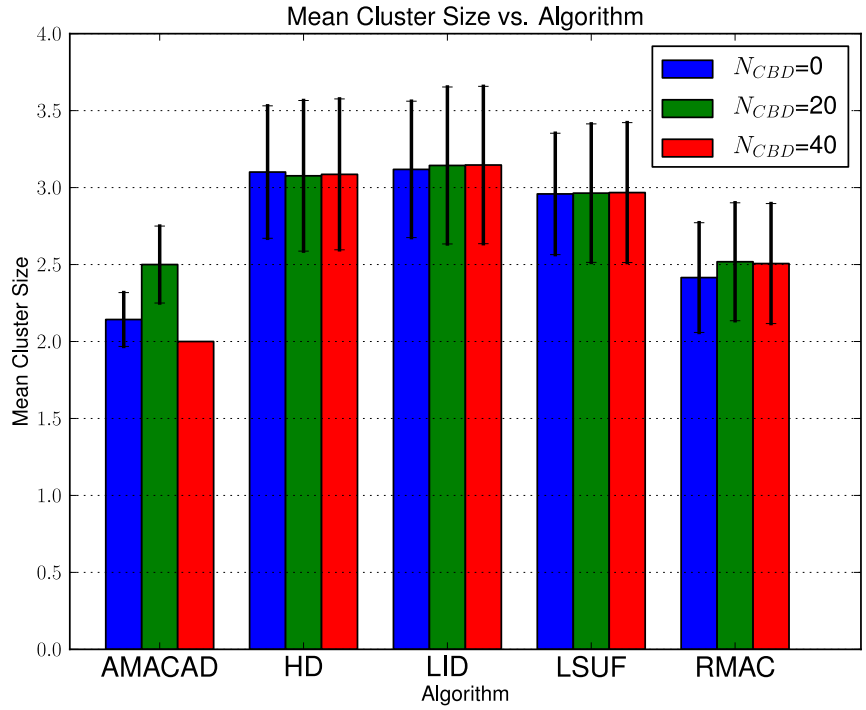


(b) Increasing speed increases the likelihood of cluster collapse in AMACAD.

Figure 5.7: Influence of vehicle speed on cluster stability.



(a) RMAC has the longest lived clusters on the grid.



(b) AMACAD is the only algorithm exhibiting significant changes with respect to N_{CBD}

Figure 5.8: Influence of CBD count on lifetime and size.

5.4.1 Influence of CBD Count

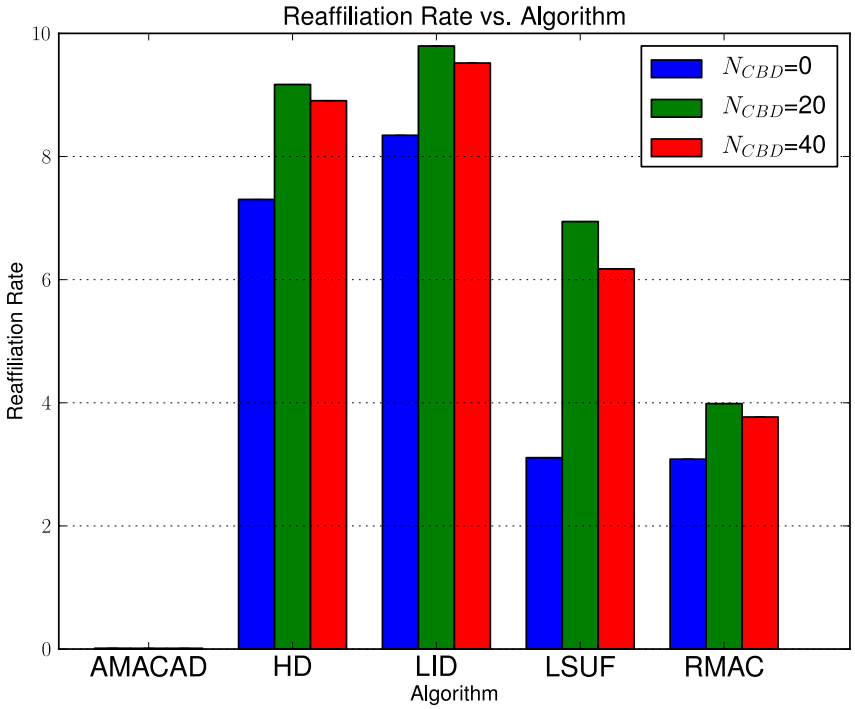
The simulations were conducted with 20, 40, and no CBDs. Node density was kept constant at 500 nodes.

Figure 5.8a shows the influence of N_{CBD} on cluster lifetime. RMAC exhibits much higher longevity than any other algorithm. It also exhibits much greater dependence on CBD count. The decrease in RMAC's lifetime from $N_{CBD} = 20$ to $N_{CBD} = 40$ can be explained as routes growing more dissimilar and thus CHs are not holding onto their CMs for as long. As $N_{CBD} = 0$ is the extreme case, it has the lowest lifetime. Despite this, RMAC's performance is the best of all the algorithms, exceeding MDMAC and AMACAD by at least a factor of 2. As seen in Figure 5.6, the depth of the hierarchical structure decreases with CBD count. AMACAD, HD, and LID show minimal dependence, while LSUF's clusters live slightly longer when $N_{CBD} = 0$.

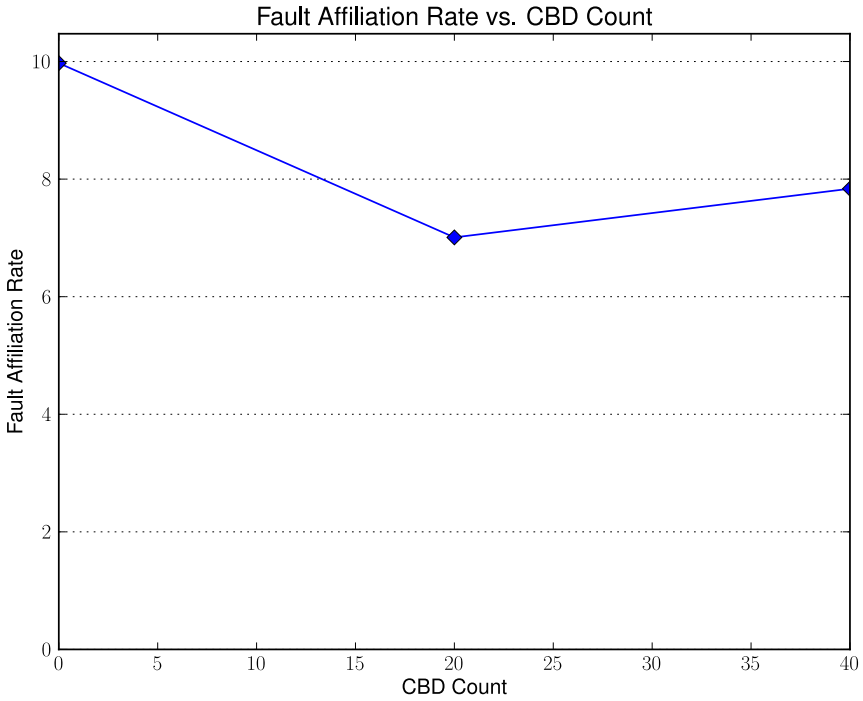
MDMAC's cluster size exhibits minimal dependence on N_{CBD}, though its clusters are the largest of the algorithms tested. There is also minimal change with RMAC, though the trend resembles that of lifetime. AMACAD experiences increased size at $N_{CBD} = 20$, but drop to a constant size of 2 at $N_{CBD} = 40$. Again this can be explained by the decreased route similarity as a result of increased number of destinations.

Reaffiliation Rate undergoes unexpected changes with respect to CBD count. With increased number of destinations (including $N_{CBD} = 0$), the stability of CM connections increase. AMACAD is the exception, as it experiences minimal reaffiliation events (see Section 5.3).

Regarding faulty affiliations, 40% of MDMAC's reaffiliations result in a fault. This is stable and independent of CBD count. However, AMACAD's incidence of cluster collapse increases, due to nodes trying to join CHs coming from converging traffic flows. After two traffic flows merge at an intersection, it is highly unlikely that a CH's suitability would remain constant as newly discovered nodes try to affiliate. Furthermore, the large number of lanes and AMACAD's flawed CH selection scheme make the algorithm more susceptible to the malfunction, particularly if nodes are affiliating with CHs that are unlikely to remain close for long or are doing so in close proximity to an oncoming intersection.



(a) CM stability increases with more unique destinations.



(b) Incidence of cluster collapse increases.

Figure 5.9: Influence of CBD count on cluster stability.

These results show that diversity of potential vehicular routes has an impact on cluster performance. An algorithm that attempts to account for the route and intention of drivers would mitigate this impact. This is seen in the Reaffiliation Rates between HD/LID and LSUF in Figure 5.9a. The former two make no effort to predict driver intention, while LSUF at least accounts for the turning direction. Thus its CM connections are more stable. AMACAD accounts for similarity of the vehicles' destinations by incorporating Euclidean distance between two nodes' destinations in its election criteria. But this potential advantage is undone by the flaws in its affiliation and maintenance schemes (See Section 5.3). RMAC's LET analysis appears to improve stability as well. It also has a maintenance mechanism whereby clusters can merge with newly-detected CHs. Unclustered nodes also affiliate with the nearest established CH, allowing clusters to replenish lost members after crossing an intersection. As a result, clusters can last longer and are more stable, making RMAC more promising for routing in vehicular environments.

5.4.2 Influence of Node Density

The influence of increasing node density will be discussed here. The simulations were conducted with 500, 1000, and 2000 nodes, with $N_{CBD} = 0$.

AMACAD and LID experience minimal effect from node density in terms of lifetime, as seen in Figure 5.11a. HD and LSUF's lifetime decreases with node density, owing potentially to the larger node density resulting in more potential CHs, which – due to MDMAC's broadcast-based neighbour discovery and affiliation strategies – cause CMs to jump between CHs. This, coupled with nodes separating and merging in different traffic flows, causes the drop in lifetime. LSUF's performance is slightly higher as the algorithm attempts to group vehicles in similar traffic flows, which mitigates these effects. RMAC, by contrast, performs far better than the others, by virtue of its more robust CH selection mechanism and its capability to hold onto and replenish CMs from converging traffic flows. In Figure 5.10, depth of clusters increases with node density, but appears to taper off, suggesting an intrinsic upper limit on depth of the hierarchy.

The size of clusters, shown in Figure 5.11b, increases for MDMAC and RMAC. It is interesting to note the LID exhibits the highest incidence of clusters larger

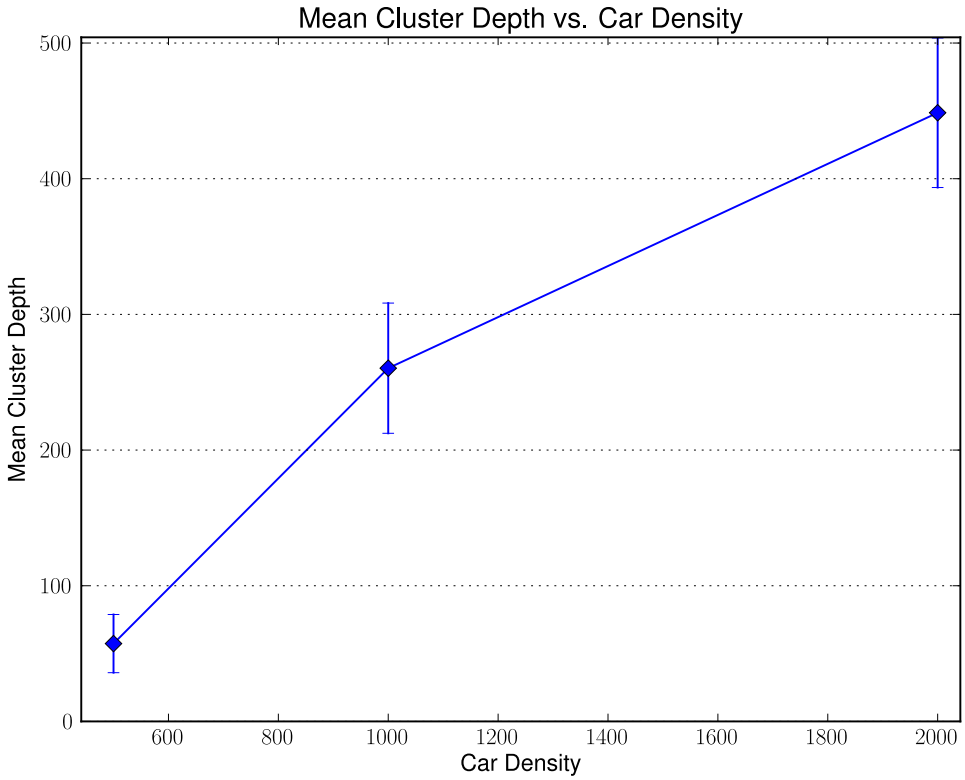
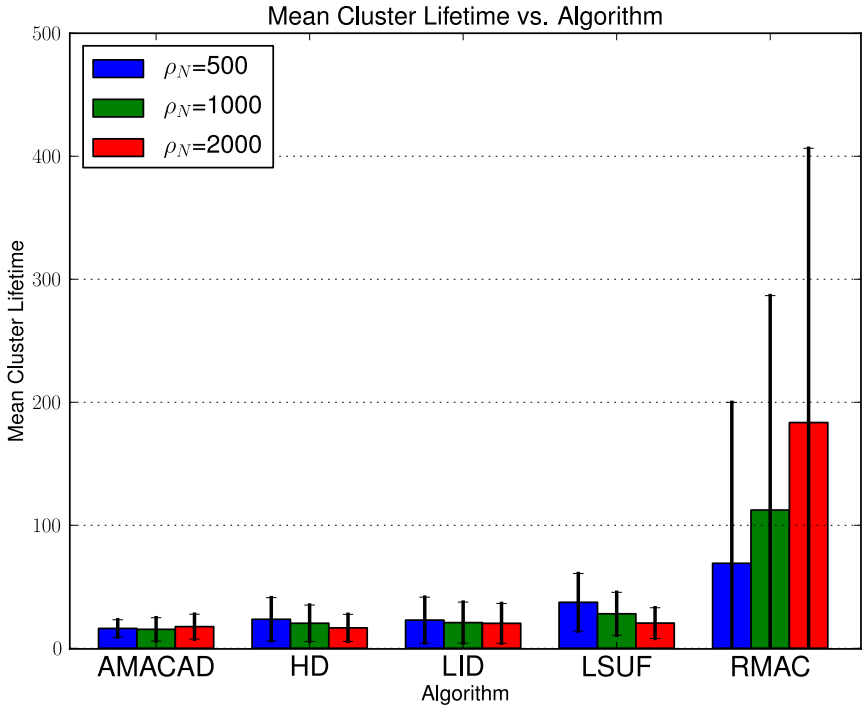
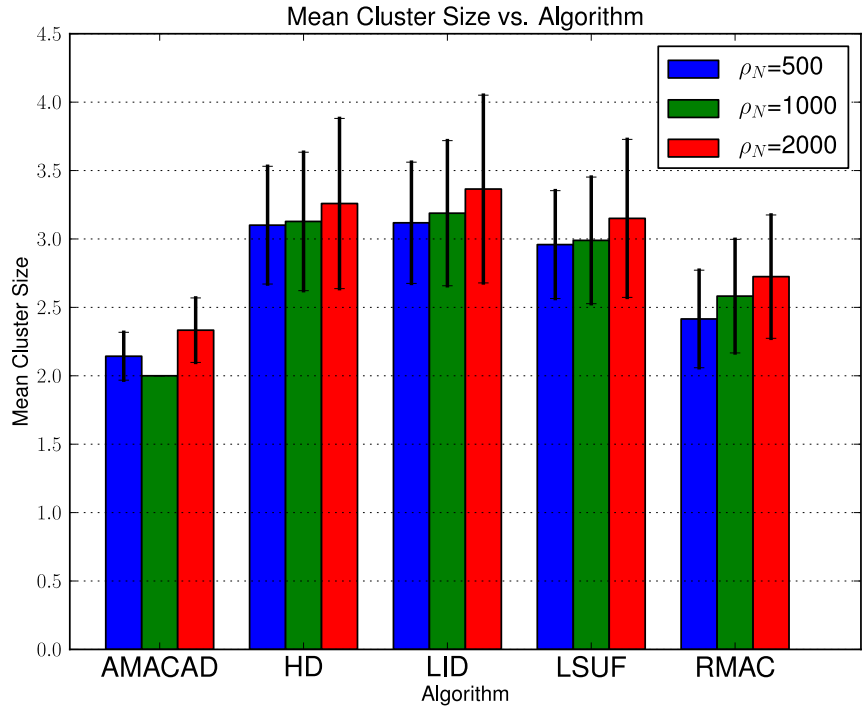


Figure 5.10: RMAC's structure depth increases due to the larger number of nodes.

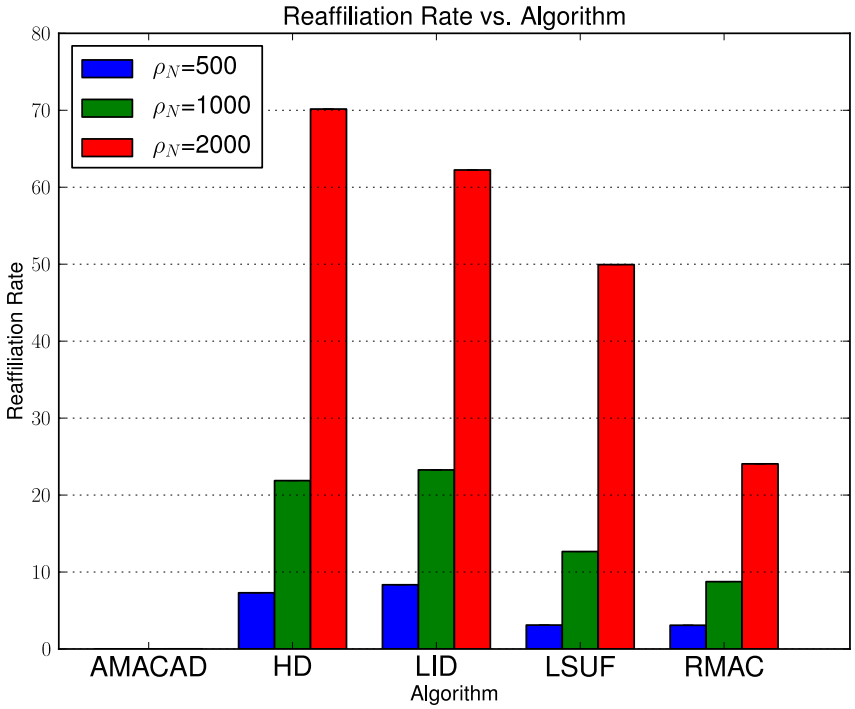


(a) RMAC is the longer-lived of the algorithms studied, and its performance increases with node density.

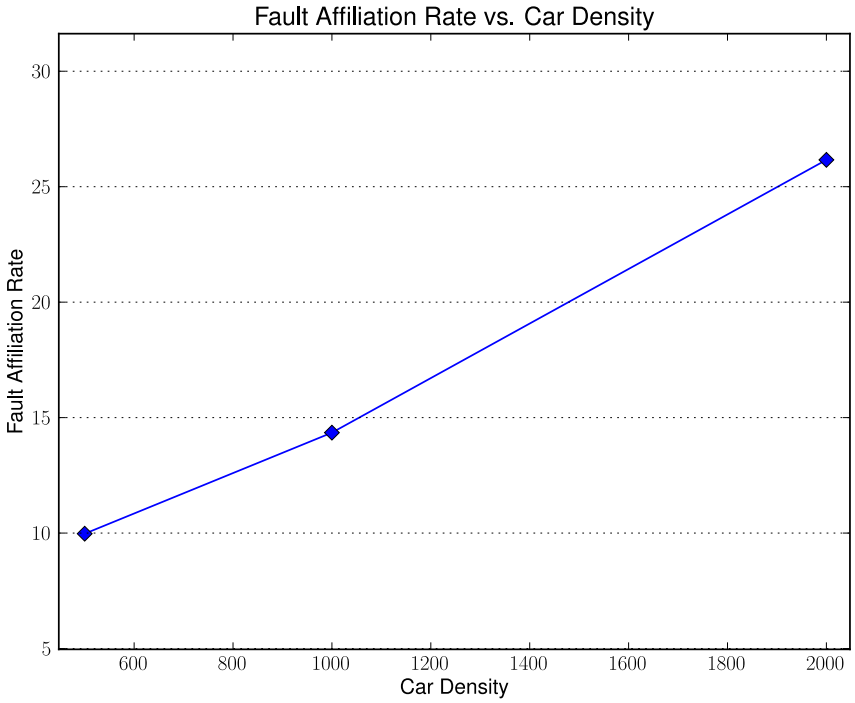


(b) Expectedly, the cluster size increases with density, except for AMACAD, which drops at $\rho_N = 1000$.

Figure 5.11: Influence of node density lifetime and size.



(a) Higher node density reduces CM stability



(b) AMACAD's incidence of cluster collapse appears to increase linearly with density.

Figure 5.12: Influence of node density cluster stability.

than 4. This is likely due to the static nature of LID, which allows a CH to gather more CMs when nodes are stationary at a red light. HD and LSUF are susceptible to fluctuations in their scores even when stationary, as even a slight change in the relative positions of vehicles can have an impact on the channel. A temporarily good channel will allow the reception of more beacons from further away, increasing the degree of a node, and thus its suitability. As LID is resistant to these changes, it can take greater advantage of standstill traffic. This is, however, eclipsed by its high reaffiliation rate (see Figure 5.12a) and low lifetime (see Figure 5.11a). AMACAD experiences a drop to a static value of 2 nodes per cluster at $\rho_N = 1000$. This may be due to the higher density of nodes causing vehicular obstructions on LOS transmission paths, preventing nodes from announcing their presence.

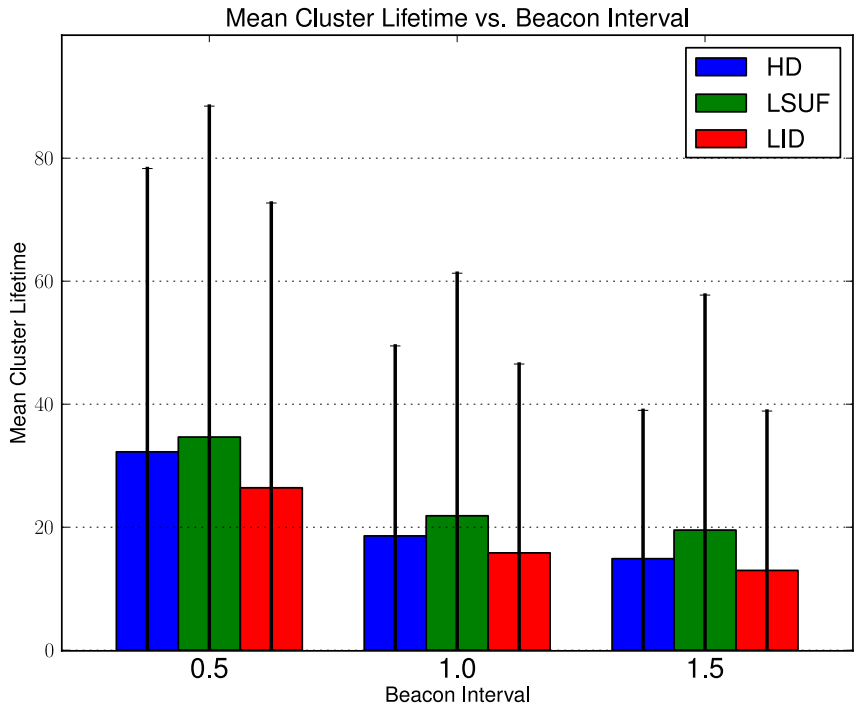
Reaffiliation Rate increases quadratically as seen in Figure 5.12a. LID experiences lowest CM stability until $\rho_N = 2000$, at which point HD is the least stable. Again this is likely due to nodes changing CH as degree of nodes fluctuates. RMAC's clusters are the more stable. AMACAD's reaffiliations are comparatively negligible, while its incidence of faulty affiliation increases linearly with density (see Figure 5.12b). With more nodes causing more vehicular obstructions while simultaneously seeking CHs to join, AMACAD's clusters collapse more frequently.

5.5 Algorithm Parameter Study

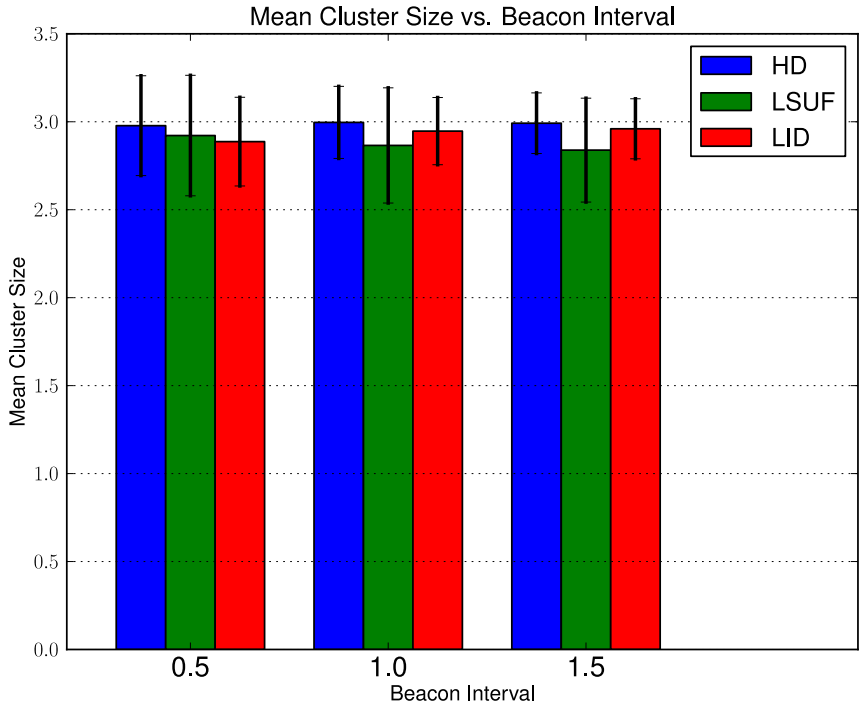
Each of the algorithms surveyed in this study depend on a set of parameters, listed in Table 5.2. The influence of these parameters on clustering performance is presented in this section. As the parameterisation of each algorithm is different, they will be discussed in separate sections. The simulation scenario is identical to the highway, with $N_L = 4$, $S_{MAX} = 80km/h$, and $\rho_N = 10$.

5.5.1 MDMAC

The weighted metric sums applied in MDMAC behaved differently in response to the parameterisation. The trends in Figure 5.13a show that lifetime decreases with increased beaconing period. Combined with the reduced percentage of faulty affiliations, shown in Figure 5.14b, the results suggest a higher beacon period reduces

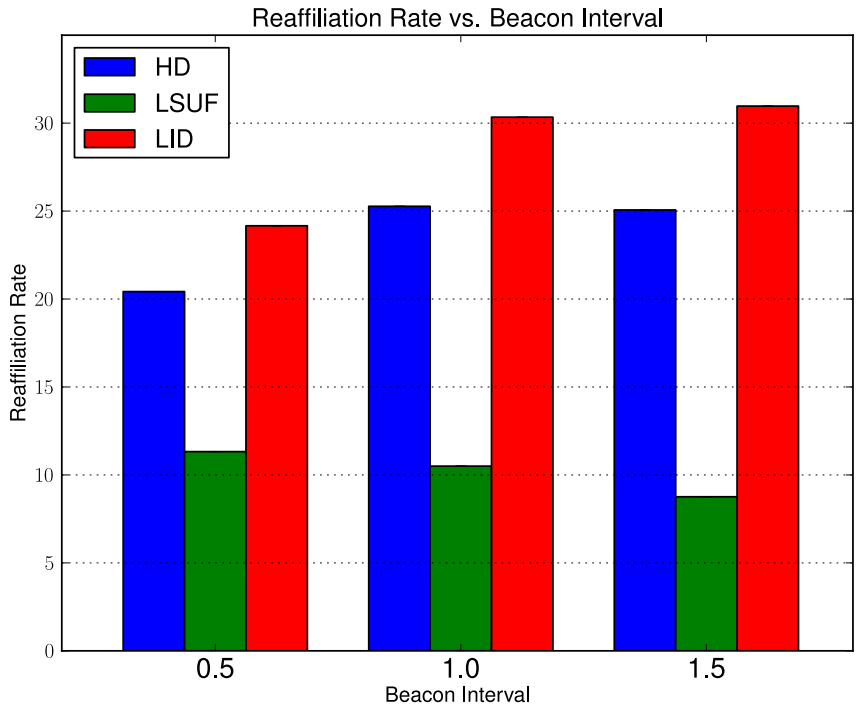


(a) Higher beaconing period reduces lifetime, as there is lower likelihood of *Hello* collisions and thus nodes can disseminate their data more easily, resulting in more potential CHs.

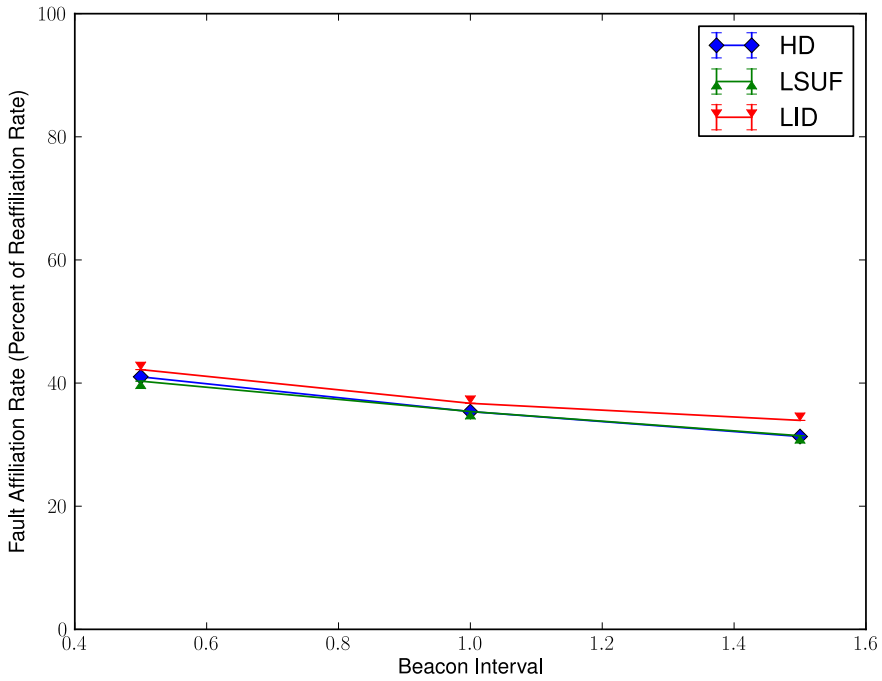


(b) There appears to be minimal influence on cluster size.

Figure 5.13: Influence of MDMAC's beaconing period on lifetime and size. The beaconing period is given in seconds.



(a) Higher beaconing period leads to greater CM instability, except for LSUF, which decreases.



(b) A slight decrease in faulty affiliation percentage indicates increased success of JOIN messages.

Figure 5.14: Influence of MDMAC's beaconing period cluster stability. The beaconing period is given in seconds.

the probability of *Hello* collision, allowing more nodes to transfer their data to their neighbours, and increasing the number of potential CHs. The lifetime is then reduced as CMs jump between CHs of fluctuating score. This is supported by the Reaffiliation Rates for LID and HD in Figure 5.14a, which increase for higher beaconing periods. However, the beaconing interval appears to have the opposite effect on LSUF's CM stability. LSUF prevents reaffiliations by incorporating lane sensing and mobility. Thus it does not fluctuate as much as LID and HD, allowing LSUF to benefit from reduced *Hello* collision probability. The mean lifetime drops for LSUF, but its variance seems to remain the same, while LID and HD's variance in lifetime decreases – LSUF has higher prevalence of longer-lived clusters even though the mean drops. Beaconing period has little influence on cluster size.

Initial Freshness is a constant used as part of the computation for CH suitability. When two nodes have equal speed, the initial freshness is taken as a default value, thus specifying an upper limit on the LET calculation. Figure 5.16a shows that cluster lifetime increases with the initial freshness. One could argue that the apparent linear increase is a result of CHs taking longer to recognise when they have lost their CMs (the higher initial freshness causing a higher countdown time). CMs likewise take longer to recognise the loss, meaning they will log fewer reaffiliations because they are changing CH less frequently. Thus CM stability cannot be said to increase as a result of the increased initial freshness value. There also appears to be minimal increase in the size of clusters, and no change in the percentage of faulty affiliations.

These results highlight a potential drawback of timer-based methods of detecting CM disconnection. A short-lived timer (i.e. 1 or 2 beaconing intervals) makes CM connections more susceptible to temporarily poor channels or one-time broadcast collisions. A long-lived timer prevents nodes from recognising lost neighbours as the connection fails, which can effectively isolate nodes from the functionality for which the cluster may serve as the foundational structure. Determining a suitable timeout would be context-dependent, which could be difficult to design. A disconnection detection scheme based on channel sensing and driver intention would be far more suitable.

Hop Count influences two elements of MDMAC's performance:

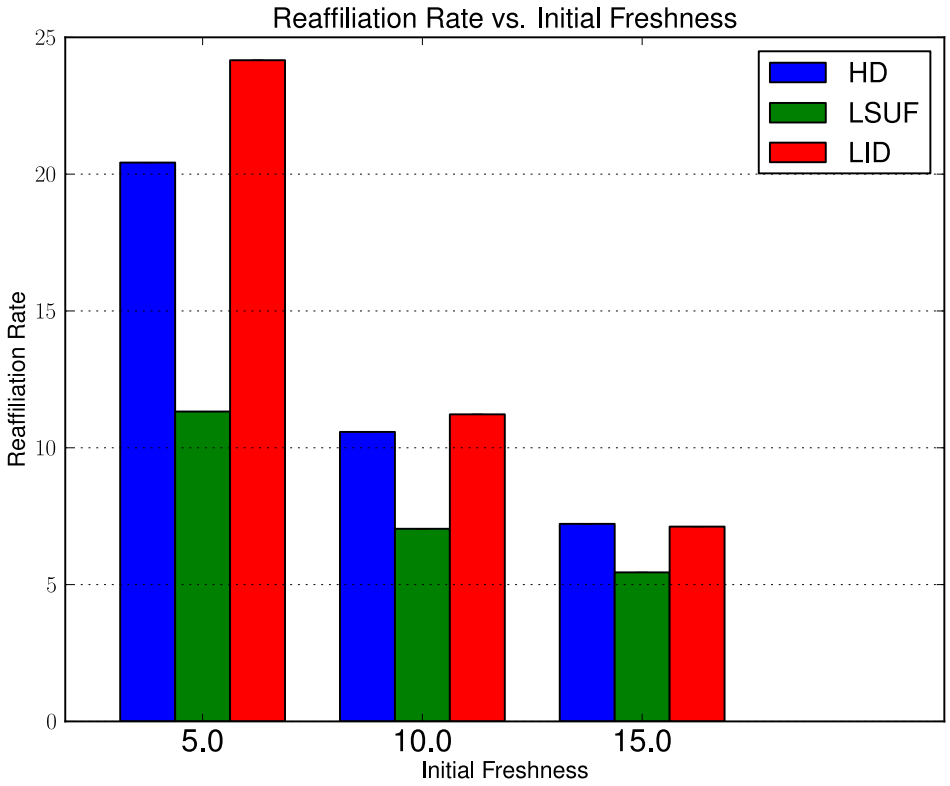
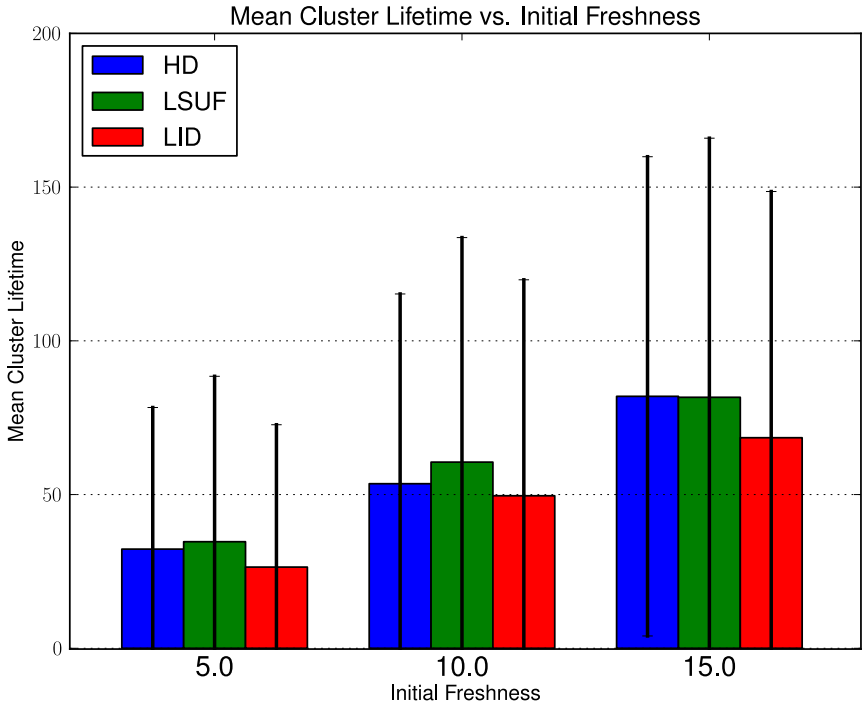
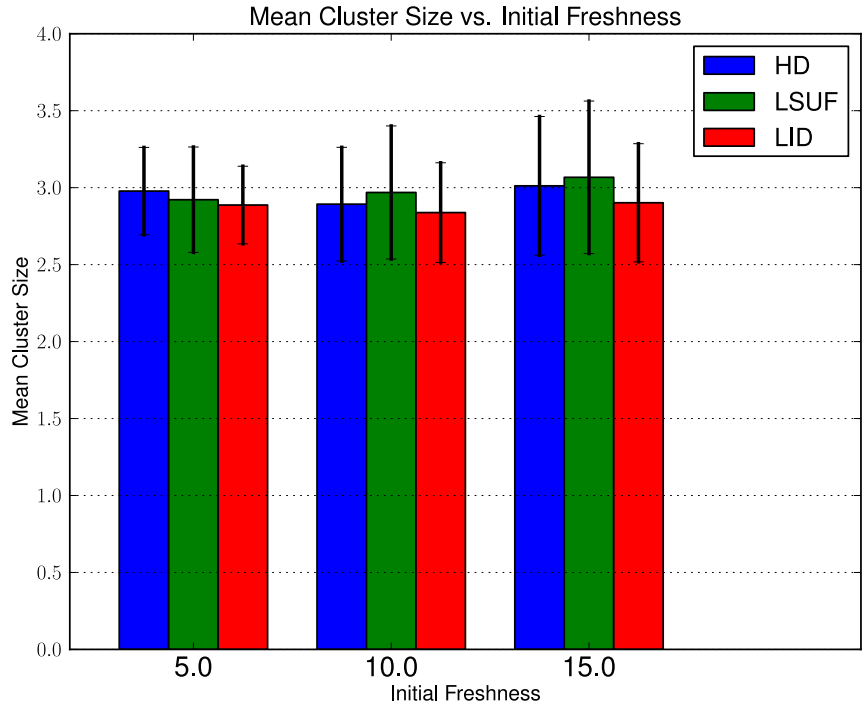


Figure 5.15: As with cluster lifetime, CMs take longer to realise they have lost their CHs, meaning reaffiliations are less frequent.

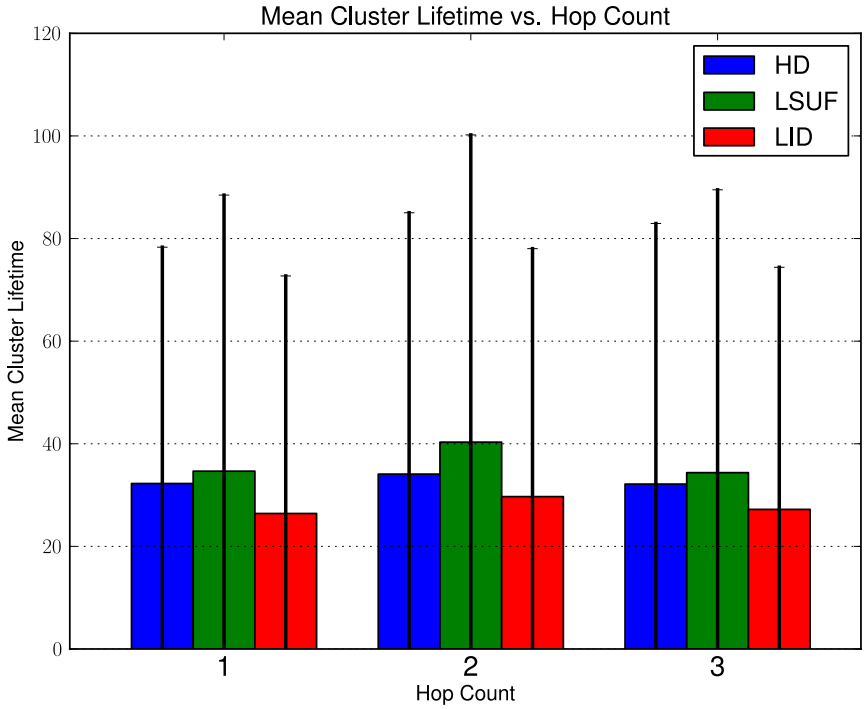


(a) Linear increase may mean CHs are taking longer to recognise lost CMs.

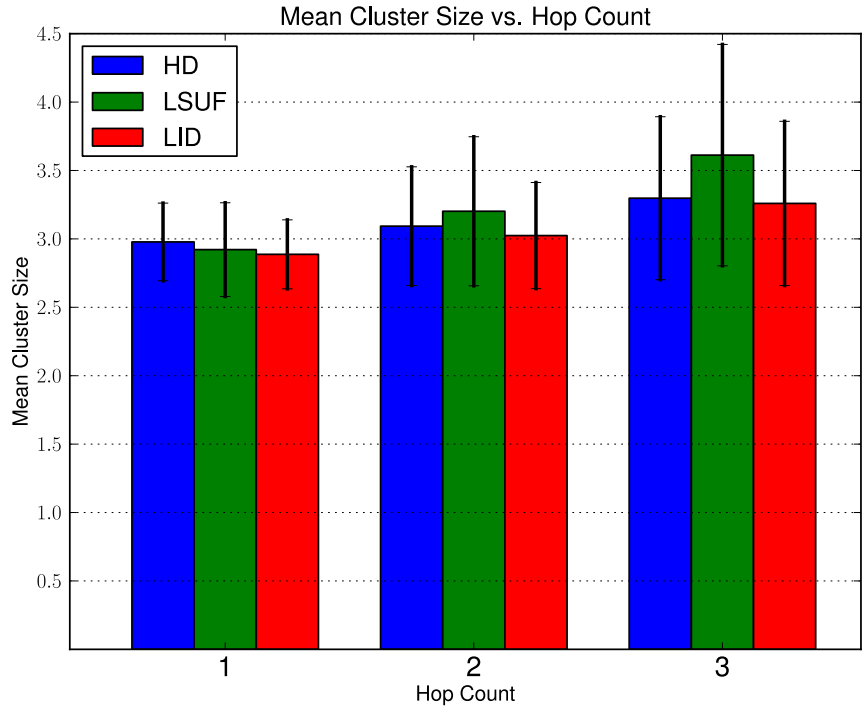


(b) There appears to be minimal change in cluster size.

Figure 5.16: Influence of MDMAC's initial freshness value.

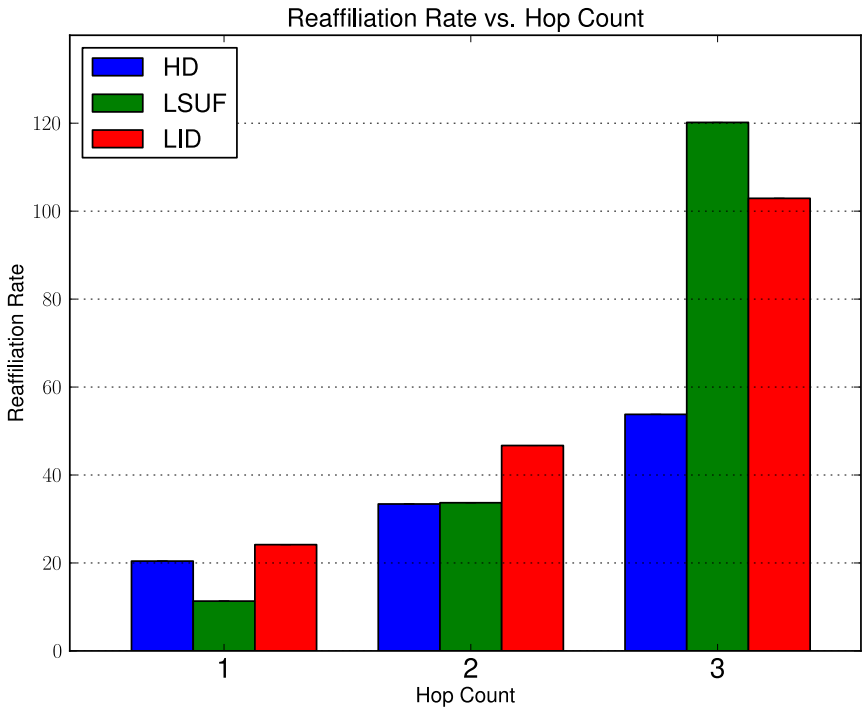


(a) The lifetime peaks at 2 hops, before dropping.

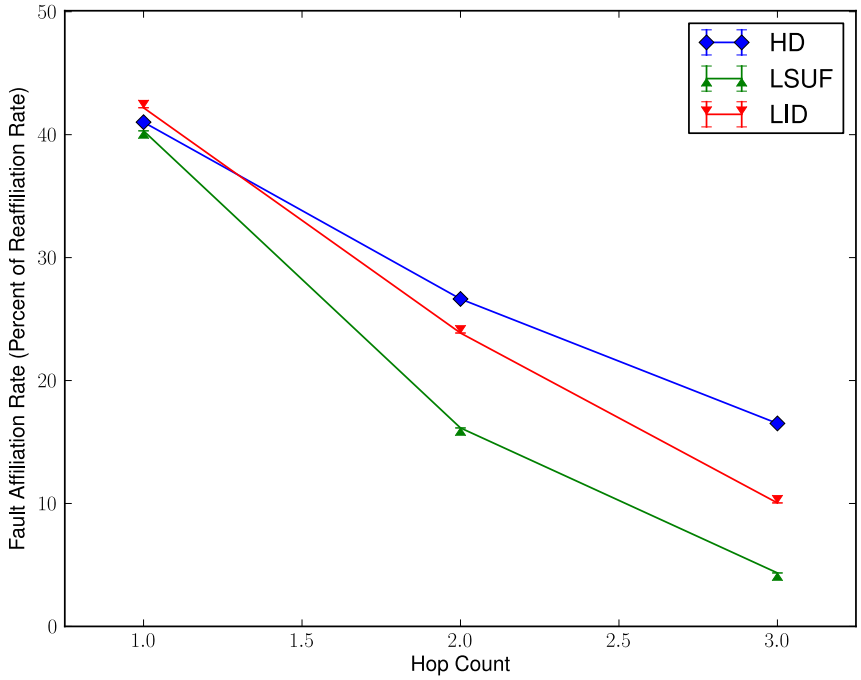


(b) Higher hop count allows CHs to acquire more members. LSUF benefits most from this.

Figure 5.17: Influence of MDMAC's hop count on lifetime and size.



(a) Reaffiliation rate grows exponentially with hop count, as CMs jump between CHs of fluctuating score.



(b) Multiple hops allows a greater degree of redundancy, which would be improved if a more elegant cooperative retransmission scheme were used.

Figure 5.18: Influence of MDMAC's hop count on cluster stability.

1. A node's zone of influence is increased, allowing data to be disseminated to neighbours outside the transmission range;

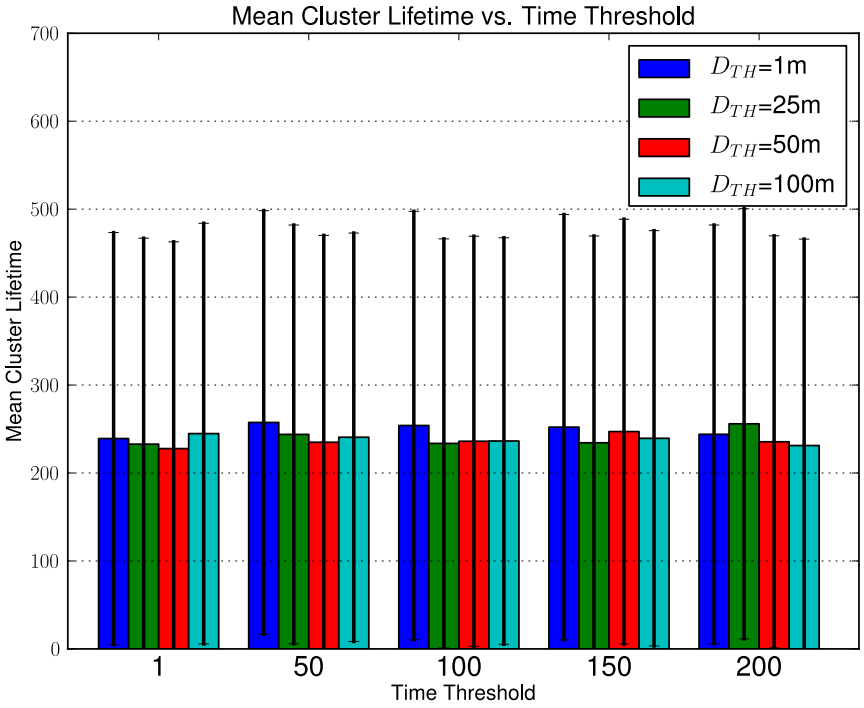
2. A level of transmission redundancy is introduced, in that the retransmission of received broadcasts acts as a form of brute-force diversity.

The increased zone of influence is shown in the lifetime, reaffiliation rate, and size measurements (Figures 5.17a, 5.18a, and 5.17b respectively). There is a marked increase in cluster size as the hop count increases. However, lifetime does not increase considerably. At 3 hops, the lifetime begins to drop. Reaffiliation rate increases exponentially. Even though hop count allows a greater range of neighbour data dissemination, it also allows nodes to obtain information on other CHs that may, as of the last beaconing period, have a better score than the current CH. LSUF is most adversely affected in terms of CM stability by increased hop count due to its eligibility function accounting for differences in speed and turning direction, such that the benefits of driver behaviour prediction are overshadowed by multiple CHs jockeying for position.

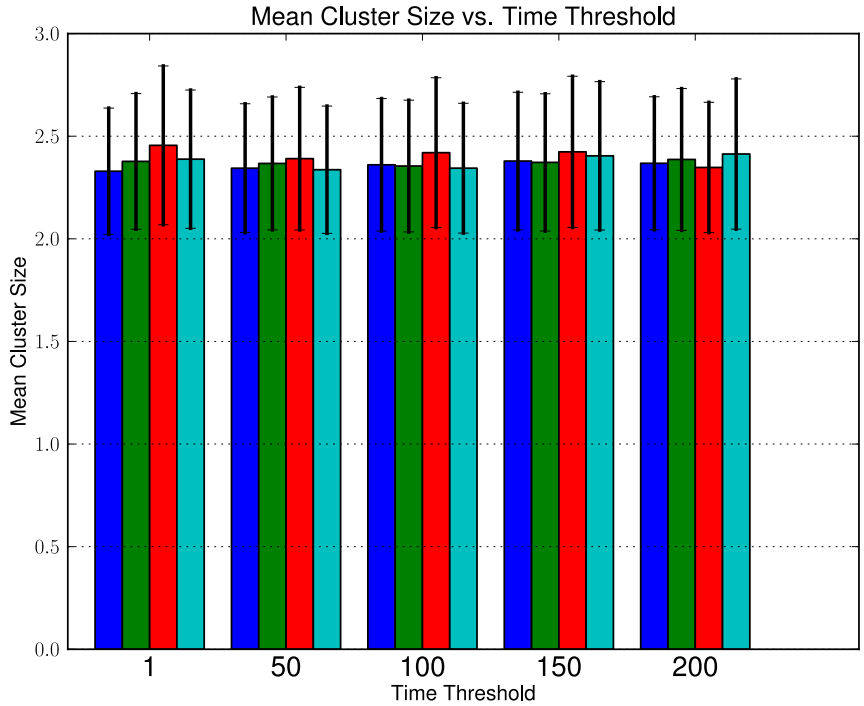
However, the potential benefits of transmission redundancy is highlighted in the percentage of faulty affiliations. The incidence of the malfunction is significantly reduced by retransmissions. However, it does not eliminate the problem, as the retransmissions occur indiscriminately of neighbour activities. The retransmission scheme could be supplemented by any number of cooperative diversity schemes (e.g. [128, 129]) in order to eliminate the malfunction altogether.

5.5.2 RMAC

The parameters of RMAC selected for this study are Time Threshold, T_{TH}, and Distance Threshold, D_{TH}. These two parameters configure the node precedence algorithm, specifying the size of tiers of internode distance and LET, into which prospective CHs are grouped. Larger values of D_{TH} reduce the number of distinct distance groupings, within which larger T_{TH} reduces the number of distinct LET (and thus speed) groupings. As a result, internode distance has a higher precedence in CH selection over LET. Increasing D_{TH} ultimately reduces the importance of distance ordering, as neighbours will not be sorted based on their distance from

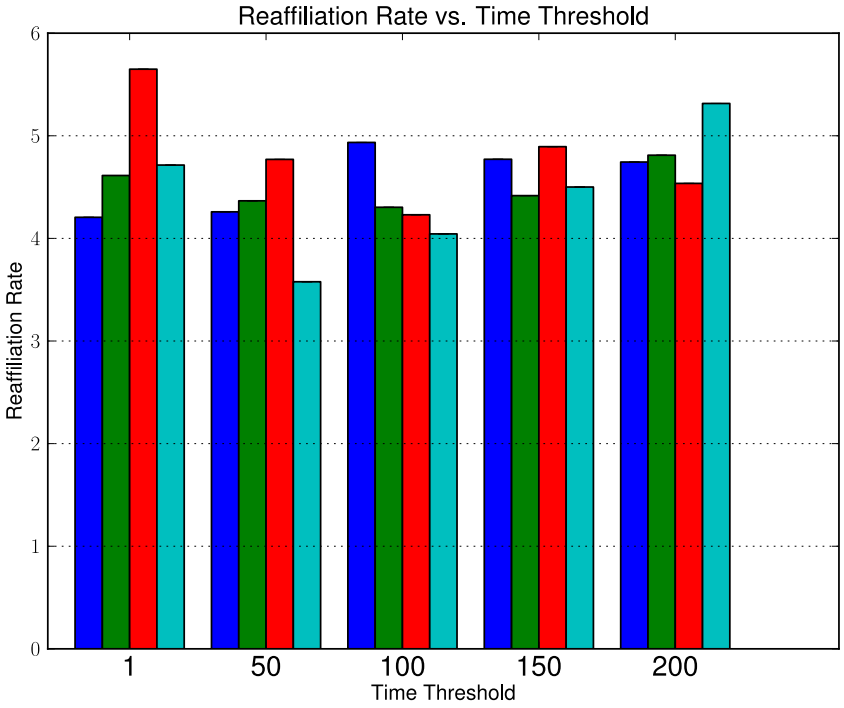


(a) There is minimal impact of T_{TH} and D_{TH} on lifetime.

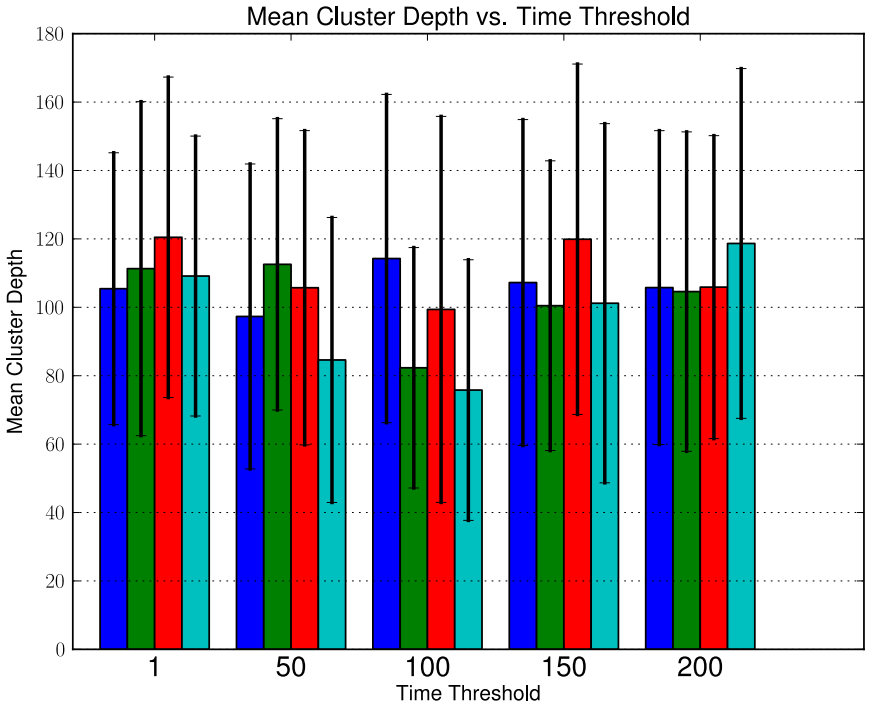


(b) There is minimal impact of T_{TH} and D_{TH} on size. See Figure 5.19a for the legend.

Figure 5.19: Influence of RMAC's time and distance thresholds on lifetime and size.



(a) As the number of LET and distance groupings decrease, the results become increasingly erratic. See Figure 5.19a for the legend.



(b) As the number of LET and distance groupings decrease, the results become increasingly erratic. See Figure 5.19a for the legend.

Figure 5.20: Influence of RMAC's time and distance thresholds on reaffiliation rate and cluster depth.

a node if the difference between those distances is less than D_{TH}. Conversely, $D_{TH} = 1m$ ensures that LET will never be considered in the sort.

Cluster lifetime and size (Figures 5.19a and 5.19b) appear unaffected by the changes to the thresholds, while Reaffiliation Rate and Hierarchy Depth (Figures 5.20a and 5.20b) show more variation. At $T_{TH} = 1$, the importance of LET is at maximum. As D_{TH} increases, the incidence of reaffiliation increases, as nodes may be attempting to join more distant CHs over a poorer channel. The reaffiliation rate drops at $D_{TH} = 100m = R_{TX}$, where a single distance grouping is in effect and LET dominates. This could be explained by greater precedence being assigned to nodes of highest LET, which would otherwise have been sequestered by multiple distance groupings. The same trend is seen at $T_{TH} = 50s$. At higher T_{TH}, the pattern appears more erratic, as neighbours are being sorted neither by distance nor LET, and thus nodes' ability to select a suitable CH is impaired. The same trends are visible in cluster depth. Increased importance on LET allows nodes with longer LET among their neighbours to accrue CMs rather than losing out to closer CHs, causing more CMs to become CHMs.

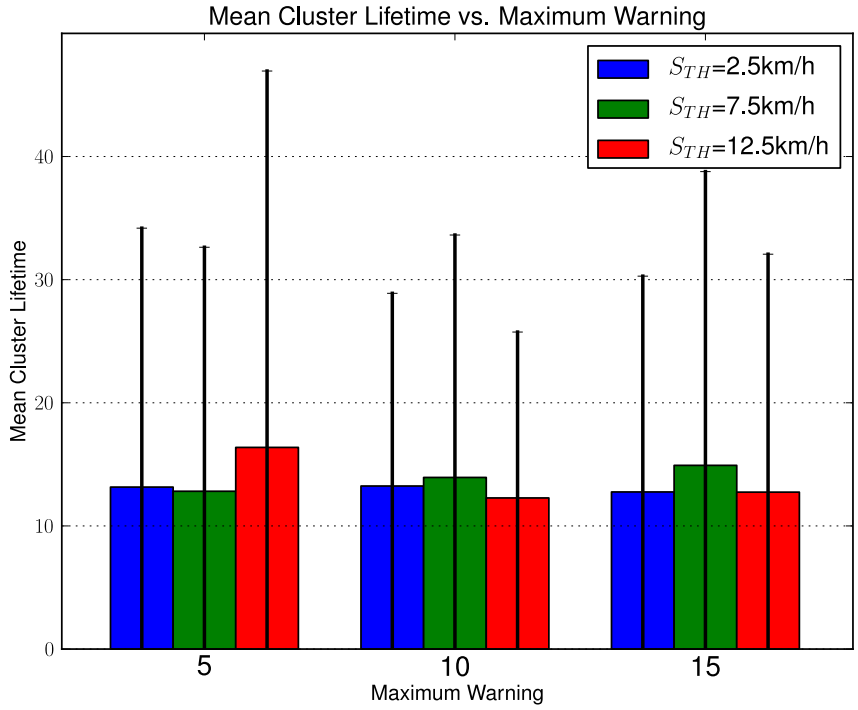
Sorting neighbours stringently by distance allows vehicles to find nearer CHs with which they will have stronger connections. However, such a high focus on distance ignores LET, as LET is sorted after distance. A possible improvement upon the design is to sort neighbours by LET with a small number of groupings (i.e. $T_{TH} = 50s$), and then sort by distance with a large number of groupings. Nearer vehicles with higher mutual LET are then more likely to affiliate and cluster stability may improve. As seen in Section 5.4.1, RMAC's clusters are more robust and long-lived, which would be even more stable if CHs and CMs attempted to predict driver intention, as seen with LSUF's performance measurements.

5.5.3 AMACAD

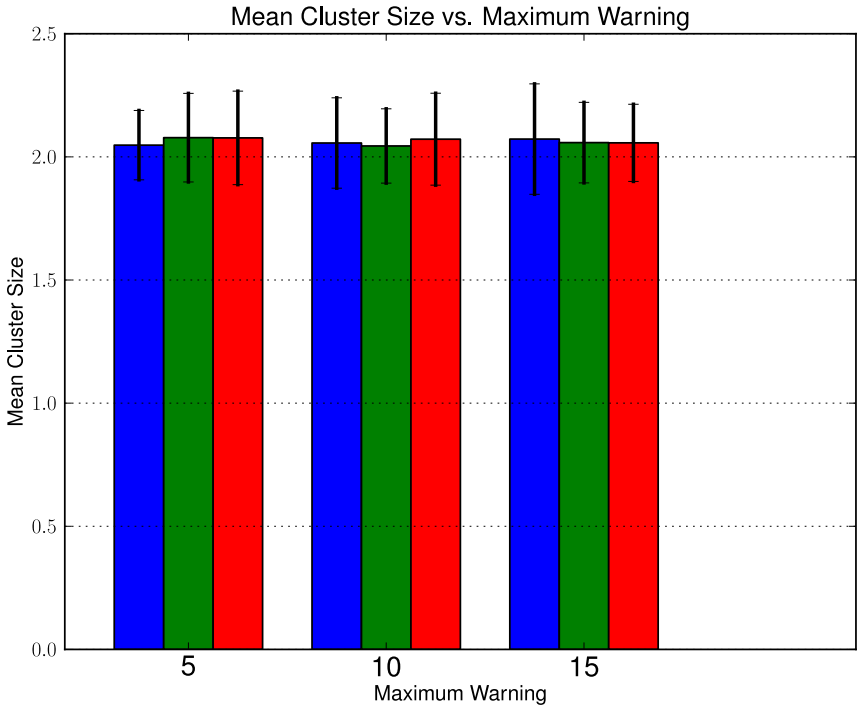
AMACAD was simulated with four parameters:

1. Speed Threshold: CMs monitor the relative speed of their CH, and transmit a unicast warning when the speed exceeds this threshold.

2. Maximum Warning Count: When a CH receives this number of warning mes-

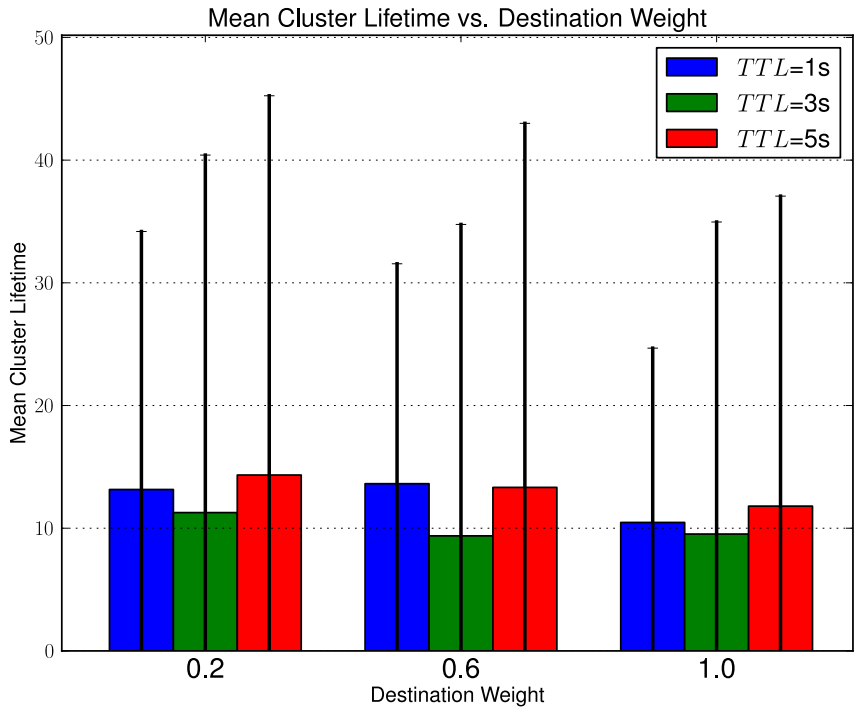


(a) A low warning limit and high speed threshold yields longer-lived clusters.

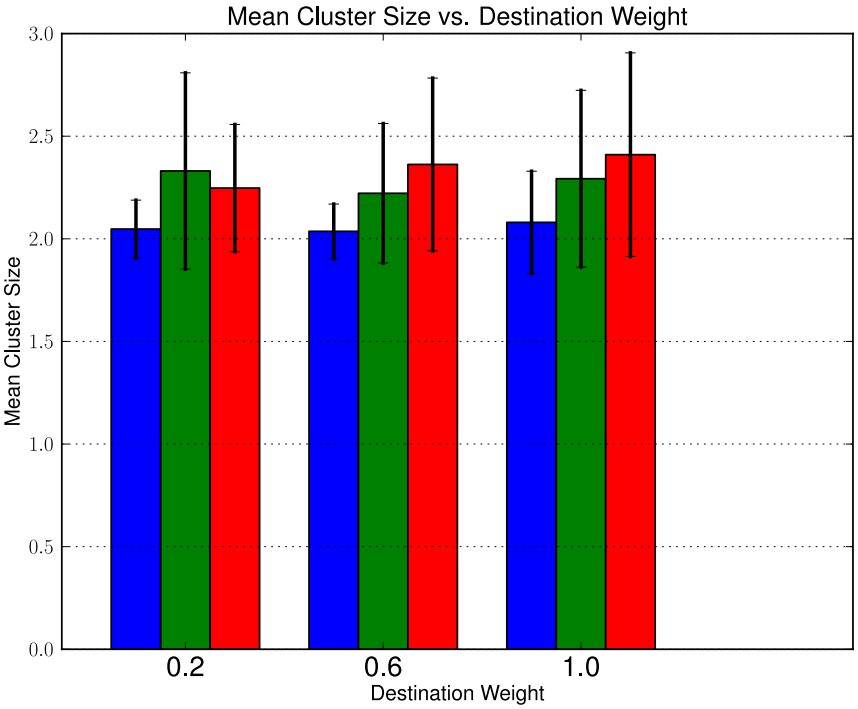


(b) There is minimal effect on cluster size. See Figure 5.21a for the legend.

Figure 5.21: Influence of AMACAD's Warning Speed on lifetime and size.



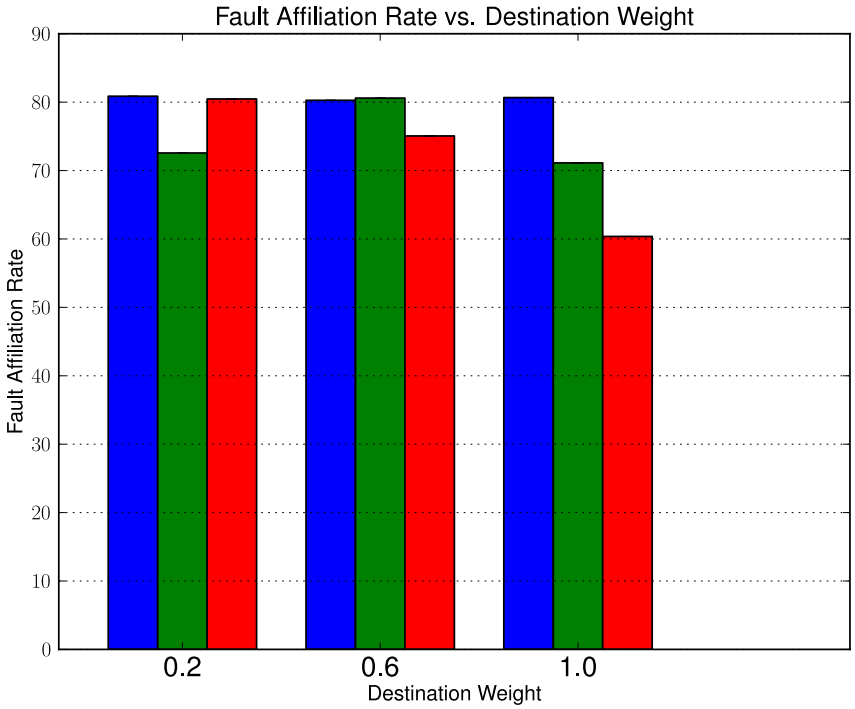
(a) There is a greater variance in lifetime, though a decrease in mean lifetime.



(b) Mean cluster size increases with TTL, and its effects are more apparent when greater importance is applied to destination in the metric calculation. See Figure 5.21a for the legend.

Figure 5.22: Influence of AMACAD's destination weighting on lifetime and size.

(a) Less stringent demands on cluster stability leads to fewer reclustering events, reducing faulty affiliation rate. See Figure 5.21a for the legend.



(b) Faulty affiliation exhibits a similar effect as with cluster size. See Figure 5.21a for the legend.

Figure 5.23: Influence of AMACAD's parameterisation on cluster stability.

sages from its CMs, it initiates a reclustering phase to identify a better CH.

3. Time To Live: This is the upper limit on the age of neighbour data in a node's table. If a node does not receive an update from its CH (in the form of a *Hello* message) within this time, it returns to an unclustered state.

4. Destination Weight: AMACAD's clustering metric is a weighted sum of internode distance, relative velocity, and Euclidean distance between destinations in the road network. The rationale is that nodes with similar destinations will follow similar paths. This is the weight applied to destination distance, between zero and one, with the remainder divided between internode distance and relative velocity.

Figures 5.21a, 5.21b, and 5.23a show the joint influence of warning limit and speed threshold. The results indicate that when the warning limit is set to a low value – meaning more stringent requirements for cluster stability - and the speed threshold is high, the clusters last longer than more lax requirements. As with MDMAC's initial freshness configuration, there exists an ideal configuration between warning limit and speed threshold that allows long-lived clusters. However, cluster size shows minimal dependence on these parameters. Incidences of faulty affiliation are lesser with more lax configurations of these parameters, which can be explained as clusters disbanding less often due to the higher instability requirements.

Figures 5.22a, 5.22b, and 5.23b show the influence of TTL and Destination Weight. Increasing the TTL results in a higher variance in lifetime. At $TTL = 1s$, this is the most strict maximum age, as CHs broadcast their updates with a period of 1 second. Increasing this value postpones reclustering, allowing clusters more time to stabilise. This is supported by the cluster size and faulty affiliation results, which show an increase in mean cluster size and stability for increasing TTL. A greater focus on destination difference causes a decrease in cluster lifetime, possibly due to nodes having similar destinations but are too distant from each other or their speeds are too different. Thus there is a need for balance. However, cluster size does improve slightly for higher destination weight, and at a weight of 1, the TTL changes have a much greater effect on stability.

AMACAD follows in the direction of LSUF, attempting to improve clustering stability by accounting for destination of vehicles. This makes intuitive sense, and is supported by LSUF's positive results. However, AMACAD is crippled by it's hand-off flaw. The CH hand-off scheme should be thoroughly redesigned in order to improve the system's robustness to cluster collapse. An adaptive assignment of weightings to the three elements of the clustering metric will allow the system to determine importance of speed, distance, and destination based on the context. It must be noted that there is a potential flaw in AMACAD's assessment of destination similarity. In a grid environment, similar destinations are likely to have similar routes, as in highways. Real life roads, by contrast, are not so simply designed. Two locations may be geographically close to each other, but the distance through the road topology may be significantly longer. Assessing driver intention through common consecutive streets in two vehicles' routes could be a much better indicator of suitability.

5.6 Discussion

A simulation survey of VANET clustering paradigms has been presented. The performance of weighted metric multi-hop and precedence-based hierarchical algorithms have been assessed under highway and grid scenarios, and their parameterisations have been studied. Three observations have been made, supplementing the contribution of Chapter 4.

1. algorithm development should tend toward design decisions that absorb and maintain existing clusters rather than permitting them to disband;

2. better approaches are required for detecting CM disconnection;

3. finally, driver intention and readily-available road map data should be applied in both CH election and cluster maintenance schemes.

A particularly useful observation is the instability of architectures that require CMs to jump to more suitable CHs upon detection. As highlighted by Ghosh and Basagni [30], VANET mobility causes frequent fluctuations in CH suitability metrics, and this results in frequent reaffiliations. While this problem has been addressed in

recent publications, it is still a drawback of weighted metric mechanisms, especially since CH eligibility scores typically involve averages, which ignore the individual node's requirements. By contrast, RMAC's precedence-based scheme allows nodes to select the CH that is most suitable for them. This, plus a hierarchical algorithm that absorbs disjoint clusters into existing structure rather than simply disbanding a cluster, makes for a combination of design choices that allow RMAC to function well in an urban environment. The main issue with RMAC's hierarchical structure is its tendency to form long chains, which should be rectified with further processing of the clustering structure – perhaps by having CMs redirect affiliating nodes to the nearest CH, provided some given criteria (assessed by the CM in question) is met.

This study has also demonstrated difficulties with time-outs as a viable disconnection detection method. Time-outs set too long prevent vehicles from detecting lost connections to their neighbours, while short time-outs make an architecture susceptible to a temporarily poor channel causing unnecessary reclustering. Adaptable time-out calculation based on environmental conditions would help, but the question then becomes how the time-out is calculated and what factors should influence the computation. Alternatively, nodes can measure received signal power statistics to assess the quality of the channel. Steadily diminishing channel quality between two nodes alerts the nodes to the failing link. There is a wealth of readily-available information in VANETs, such as road map data and the route of the vehicle through the road map, that can be exploited by a node to determine exactly when two vehicles' paths will deviate, indicating one of the more common disconnection events. Unfortunately, to the knowledge of this author, this information has gone untapped.

Road map information has not yet been fully utilised to its potential in the field of CH election and maintenance. LSUF and AMACAD attempt to move in this direction by predicting driver intention. Unfortunately LSUF is hampered by vulnerabilities to low lane count and the inability to distinguish a car's turning direction in lanes belonging to traffic flows that turn in multiple directions. AMACAD is similarly undone by the assumption that closeness of destination equates to similarity of route, which isn't necessarily the case. Alternatively, utilisation of a vehicle's route as a sequence of street identifiers, and counting the consecutive common streets between two nodes may offer a far more robust measure of driver intention. Assess-

ment of neighbour topology and minute intentional deviations (such as lane changes – which can be detected by the on-board unit via the turn indicator switch) can allow CHs to dynamically reform segments of the cluster hierarchy without the need for complete reclustering.

The following chapter introduces an extension to RMAC, which incorporates two of the aforementioned proposals. The node precedence algorithm is extended with a route similarity metric intended to group vehicles that will stay together for the longest period of time. The polling mechanism is upgraded with a check of the channel conditions, preventing reaffiliations due to temporarily poor channel conditions.

Chapter 6

Extensions to RMAC

6.1 Introduction

The experiment survey presented in Chapter 5 found that CH election mechanisms accounting for vehicular dynamics allows more stable clusters, which are further improved by incorporation of driver intention in the decision-making process. Additionally, the study found RMAC's precedence-based scheme is more stable than the weighted metric approaches. This was primarily attributed to the fact that weighted metric schemes often involve an average of CH suitability parameters (such as mean relative velocity), in which individual disadvantages can be lost. As a result, nodes affiliate with high-scoring CHs that are unsuited to their needs, causing frequent reaffiliations as CHs jockey for position. By allowing nodes to choose their own CHs via independent assessment, the connections can be much stronger, and when combined with a hierarchical structure provides a sturdier foundation for VANET routing architectures.

While RMAC's predicate accounts for vehicular dynamics such as position and speed, there are vulnerabilities that serve as sources of instability. In particular, diversity of routes among vehicles cause CMs to break away from their CH upon turning down different streets at intersections. LSUF [66] uses the turning direction of the vehicle's lane to assess the driver's probable intention, while AMACAD [60] uses the Euclidean coordinates of a vehicle's destination. However, as was shown in Chapter 5, LSUF can fail if the number of lanes is low as the algorithm can no longer distinguish whether a lane will turn left, right, or travel straight; AMACAD can also

fail if two vehicles have close destinations, but the structure of the road network results in different routes to the destinations. By contrast, it is expected that GPS navigation – already widespread in modern vehicles – will be integrated into VANET systems in the future, providing the vehicle's route as a list of consecutive street identifiers. This potentially provides a much greater method of assessing driver intention throughout a vehicle's journey.

Another weakness of RMAC is found in its poll-based maintenance mechanism, as discovered in Chapter 4. The CH waits a set time for an acknowledgement to its polls from each CM. Those that do not respond during the time-out are dropped from the cluster. Among the results in Section 5.3 was the prevalence of the *faulty affiliation* malfunction. A CM's affiliation request may be lost due to a temporarily poor channel, but if no handshake is required for affiliation, the node will erroneously believe it has joined a CH that does not know it exists. While RMAC does not suffer this malfunction thanks to its requirement for *JOIN* acknowledgement, a poll message or corresponding acknowledgement could be lost as a result of a temporarily poor channel, which causes unnecessary reaffiliations. Routine assessment of the VANET channel to determine transmission conditions, indirectly explored with MOBIC in [24], could allow CHs and CMs to more reliably recognise when they have lost connections to each other.

This chapter presents Channel and Route Aware Clustering (CRAC), which extends RMAC with a Route Similarity metric and a mechanism for classifying missed poll events. The proposed algorithm sorts nodes according to highest Route Similarity before sorting according to speed and inter-node distance, in order to account for driver intention in the CH election process. The algorithm also analyses the VANET channel to detect when the polling mechanism is disrupted due to a temporarily poor channel. The results show that both these modifications improve cluster lifetime and CM stability.

6.2 Details of Modification

There are two modifications presented in this chapter. The node precedence algorithm is supplemented with a Route Similarity check, encouraging nodes to affiliate

with vehicles that will remain close to them for longer periods of time. The poll-based maintenance mechanism is augmented with an additional channel assessment to give CMs more leeway.

6.2.1 Route Similarity

The nodes transmit their next S_{TH} route segments, I, to their neighbours. Then, for each car v in a node c's neighbour table V, a Route Similarity factor S_{cv} is calculated.

$$S_{cv} = \sum_{i=0}^{A} f_w\left(I_i^c, I_i^v\right) \tag{6.1}$$

where

$$A = \min\left(S_{TH}, |I^c|, |I^v|\right)$$

and

$$f_w\left(a, b\right) = \delta_{ab} \tag{6.2}$$

where δ_{ab} is the Kronecker Delta. Note that the summation does not always go to completion. The summation stops when $f_w\left(I_i^c, I_i^v\right)$ yields a zero.

The node precedence algorithm is supplemented by comparing the Route Similarity before inter-node distance and LET. As $S_{cv} \in \mathcal{Z}$, the Route Similarity will place all neighbours with the highest similarity at the top of the list before further sorting them according to the other relevant metrics. Thus, vehicles will seek out CHs that will remain with them for the longest time possible.

To the knowledge of this author, the use of a vehicle's route as a list of consecutive road IDs has not been explored for improving cluster stability. This is because most simulation programs do not offer the functionality to perform such analysis. Most clustering algorithms are assessed using a network simulator that imports a pre-computed trace from a traffic mobility model (e.g. ns-2 [85] loading an exported SUMO [98] trace). By contrast, bi-directional VANET simulators like VEINS [90], which run the traffic and network simulations in tandem, offer the capability to extract far more information about the traffic scenario than just location and velocity. This includes the vehicle's route. The Route Similarity modification is the first known attempt to leverage the features of bi-directional simulators that have

Algorithm 1: Response to missed polls

Data:
CH node, i
CM node, j
Loss Probability of j w.r.t i, P_{ij}^{loss}
Route Similarity of j w.r.t i, S_{ij}
Number of i's polls j has missed, M_j
Result:
Should j be dropped from i's cluster?
begin
  $M_j \leftarrow 1 + M_j$
  $a, b, c \leftarrow 0$
  if *i crossed intersection since last poll* **then**
    $a \leftarrow S_{ij} \leq 1$
  $b \leftarrow M_j P_{ij}^{loss} \geq P_{crit}$
  $c \leftarrow M_j > M_{max}$
  if a ***or*** b ***or*** c **then**
    Drop j from i's cluster.
  else
    Retain j and wait for next poll.

otherwise gone overlooked in clustering research.

6.2.2 Poll-based Maintenance Modification

The CH uses channel measurements between consecutive packet receptions to assess whether a missed poll response was caused by a temporarily poor channel. It also uses awareness of the road and the Route Similarity metric to determine if a missed poll response was caused by a CM turning down a different street at the last intersection. This assessment then determines whether the CM is dropped from the cluster.

When a CH, i, successfully receives a frame from a neighbour, j, it logs the signal power measurements in the neighbour table. From these signal power measurements, it computes two values: the K-factor; and Ω, the sum of the specular and diffuse signal strengths. K is derived using the estimator applied in Chapter 3, while Ω is taken as the second raw moment of the channel sample. These are then used to calculate the statistical parameters of the channel distribution, A and σ, using formulae in [130].

$$A^2 = \frac{K\Omega}{1+K} \tag{6.3}$$

$$\sigma^2 = \frac{\Omega}{2(1+K)} \tag{6.4}$$

The CH uses A and σ to determine P_{ij}^{loss}, the probability that the last frame i received from j could have been lost. That is,

$$P_{ij}^{loss} = P(P_{RX} \leq S) \tag{6.5}$$

where S is the receiver sensitivity. The possible presence of obstructing vehicles is not explicitly considered in the calculation, as the data available to i covers only a small subset of the network and may not include fast-moving vehicles. Furthermore, the value of Ω will be influenced by any vehicles present anyway, and as was noted in Chapter 3, the statistics of Small- and Large-Scale fading are considered to be independent. Thus, the calculation of Ω indirectly brings vehicular presence into consideration. In light of this, the channel is assumed to be Rice-faded and undergoing Free-Space Path Loss. The received signal power is given by

$$P_{RX} = \frac{P_{TX} G_i G_j \lambda^2 G_f}{16\pi^2 d_{ij}^2} = \frac{\tau G_f}{d_{ij}^2} \tag{6.6}$$

where P_{TX} is the transmission power; G_i and G_j are the gain of i and j's antennas, respectively; λ is the carrier wavelength; d_{ij} is the distance between i and j; and G_f is the fading gain. G_f is a Rice-distributed random variable having the parameters A and σ. Substituting 6.6 into 6.5 and solving for G_f gives:

$$P_{ij}^{loss} = P\left(G_f \leq \frac{d_{ij}^2 S}{\tau}\right) \tag{6.7}$$

This is essentially the CDF of the Rice distribution. Therefore, the loss probability can be expressed as

$$P_{ij}^{loss} = 1 - Q_1\left(\frac{A}{\sigma}, \frac{d_{ij}^2 S}{\sigma\tau}\right) \tag{6.8}$$

where Q_1 is the Marcum Q function. Figure 6.1 shows a plot of Equation (6.8). As expected the loss probability increases with distance, but a lower K-factor reduces loss probability at greater distances, possibly as a result of greater incidence of constructive interference.

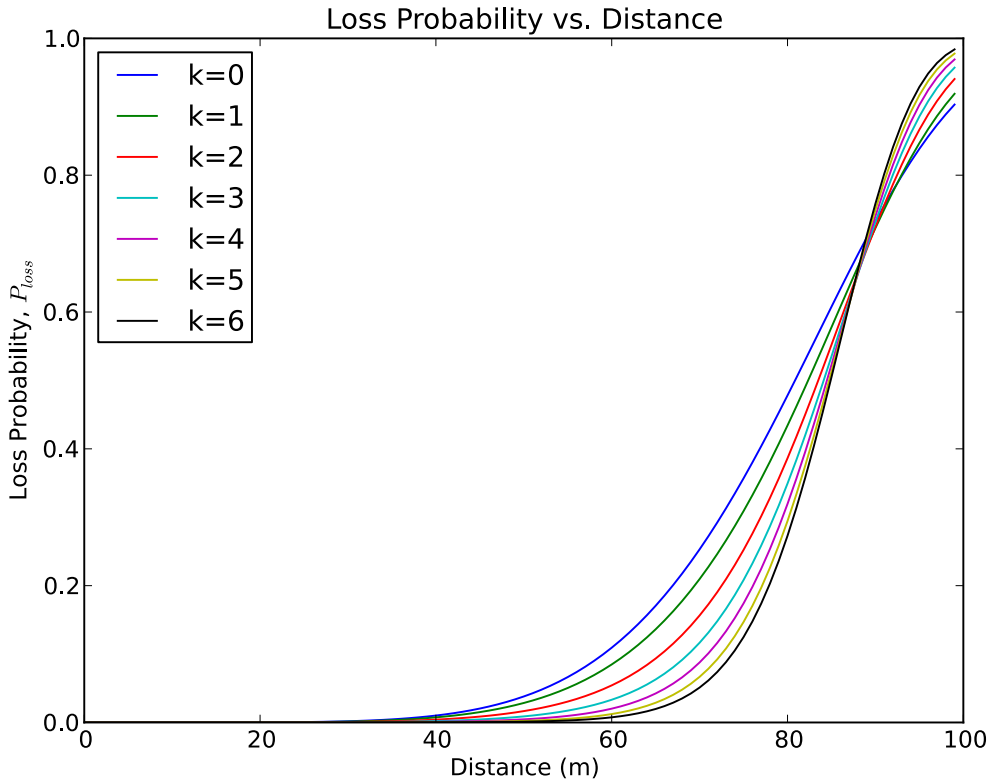


Figure 6.1: Low K-factor reduces maximum loss probability at greater distances. This is due to greater incidence of constructive interference from multipath components.

Algorithm 1 shows how P_{ij}^{loss} is applied when a CM fails to respond to a poll. The algorithm specifies P_{crit} as a critical loss probability, above which the channel is deemed too lossy. It also specifies M_{max} as an absolute margin for delinquent CMs.

When a CH, i, has not received a poll ACK message from a CM, j, it performs two checks. First, i checks if it has crossed an intersection and entered a new road segment since the last poll cycle. If it has, it ensures $S_{ij} > 1$, which signifies that j is still on the same road. If $S_{ij} = 1$, then j likely turned in a different direction at the last intersection, and is no longer on the same road as i. In this case, the CM is determined to have left the cluster and it is removed from the CH's cluster record. If $S_{ij} > 1$, the CH proceeds to the next check.

If $P_{loss} > P_{crit}$, the CH can conclude that the link with the CM has been severed and remove it from the cluster. However, if $P_{loss} < P_{crit}$, the CH will then retain the CM in the cluster record, having concluded that the channel was temporarily disrupted.

Every time j fails to respond to a poll, M_j is incremented. Consecutive channel assessments as a result of lost polls then require $M_j P_{loss} < P_{crit}$, such that eventually the CM will be dropped from the cluster if it continuously fails to respond. If M_j exceeds M_{max}, a maximum missed poll count, the member is dropped regardless of the value of P_{loss}.

6.3 Results

The simulation will study the impact of variable Route Similarity Threshold S_{TH} and Critical Loss Probability P_{crit}. Higher values of S_{TH} allows the algorithm more foresight to detect route dissimilarities. Lower P_{crit} means more stringent channel quality requirements on delinquent CMs.

The simulation package used in Chapters 4 and 5, VEINS, is adopted for this study. The same transmission parameters are used, and the same performance metrics measured. Please see Sections 3.6.2 and 4.3 for details.

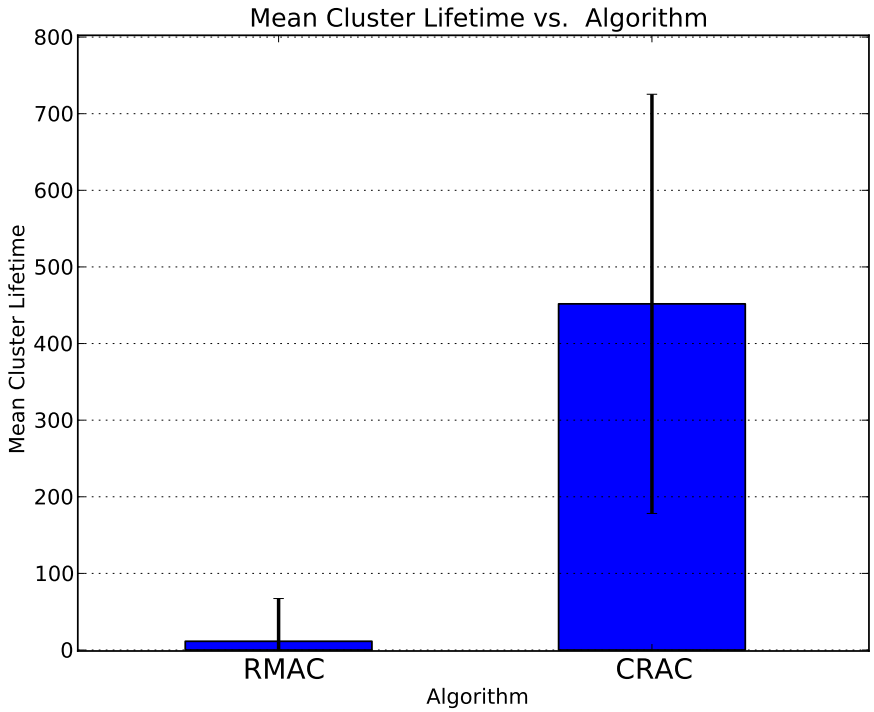
The modifications were tested under both highway and grid scenarios. The highway was run with speed limit $S_{max} = 80km/h$, lane count $N_L = 4$, and node density $\rho_N = 10$. The grid scenario was run with $\rho_N = 1000$ and $N_{CBD} = \{0, 5\}$.

6.3.1 Comparison with RMAC

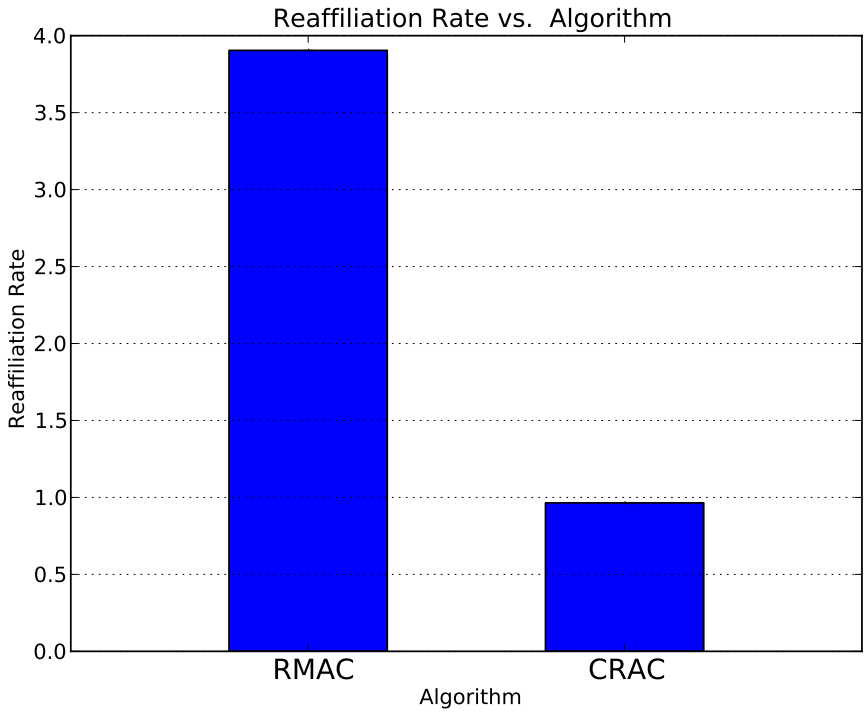
CRAC was compared with RMAC to gauge performance improvements. The algorithm was run with $S_{TH} = 2$ and $P_{crit} = 0.01$, and the performance compared to results from Chapter 5. The lifetime of clusters, shown in Figure 6.2a, show a significant improvement over RMAC, with clusters lasting as long as 700 seconds. This can be attributed to CHs taking slightly longer to detect lost CMs, allowing time to replace them. CRAC also provides better protection against changes in channel dynamics, as indicated in Figure 6.2b, showing an approximate 75% reduction in reaffiliation rates.

6.3.2 Influence of P_{crit}

The lifetime of clusters with respect to P_{crit} and S_{TH} on the highway are shown in Figure 6.3a. First observation is that just by increasing P_{crit} slightly above zero



(a) Lifetime



(b) Reaffiliation Rate

Figure 6.2: CRAC clusters last significantly longer than RMAC, as CHs are able to obtain new CMs before detecting lost ones.

almost doubles lifetime. After $P_{crit} = 0.25$, the lifetime begins to drop, but remains higher than with $P_{crit} = 0$. Logically, an increase in P_{crit} should cause CHs to wait for at most M_{max} failed polls before discarding a CM, which would in theory make the clusters last longer. Conversely, this may isolate nodes for longer periods of time, meaning that they do not go back to the unclustered state as often. As a result, more suitable CHs cannot replenish lost CMs as readily and thus die quicker. The data indicates that $(0, 0.15]$ is a suitable range for P_{crit}.

Reaffiliation Rates on the highway also experience a dramatic drop with P_{crit}, showing that CM stability is significantly improved by the Loss Probability modification (see Figure 6.3b). The rate continues to drop after $P_{crit} = 0.25$, which is consistent with the explanation presented for the behaviour of lifetime.

The grid lifetime results are shown in Figure 6.4. The effects of P_{crit} are less significant with $N_{CBD} = 0$. As the possible routes for vehicles are far more varied, clusters will die more frequently from route divergence than from missed polls. When $N_{CBD} = 5$, the effect is much more pronounced. Here, there is a smaller number of available routes, along which temporarily poor channels will be more predominant in causing reaffiliation. The results shown in Figure 6.5 indicate a similar trend. Any positive value of P_{crit} gives a mostly constant performance gain, particularly when $N_{CBD} = 5$.

6.3.3 Influence of S_{TH}

Figures 6.3a and 6.3b also show the influence of S_{TH}. For low values of P_{crit}, lifetime improves with S_{TH}, with an increase of approximately 35-40 seconds. After $S_{TH} = 6$, lifetime begins to drop. When $P_{crit} \in [0.1, 0.25]$, changes in S_{TH} have minimal impact on lifetime.

At $P_{crit} = 0$, higher S_{TH} causes higher CM instability, as indicated by the reaffiliation rates. The exception is at $S_{TH} = 4$, which shows a significant drop. Certain higher P_{crit} values allow S_{TH} to diminish reaffiliation rates, though performance varies.

This may be due to nodes sorting first according to Route Similarity, which may place precedence on CHs with more similar routes, but may also segregate nodes with lower similarity that are in fact closer and have lower relative velocity. Thus

sorting according to S_{ij} first may not be the most optimal order.

On the other hand, the grid results show a more pronounced, though still small, improvement in lifetime and stability with higher S_{TH}. The reaffiliation results show that N_{CBD} influences the significance of the impact. Fewer diverse routes means less route divergence, and thus the Route Similarity check offers smaller advantage.

6.3.4 Cluster Size and Depth of Hierarchy

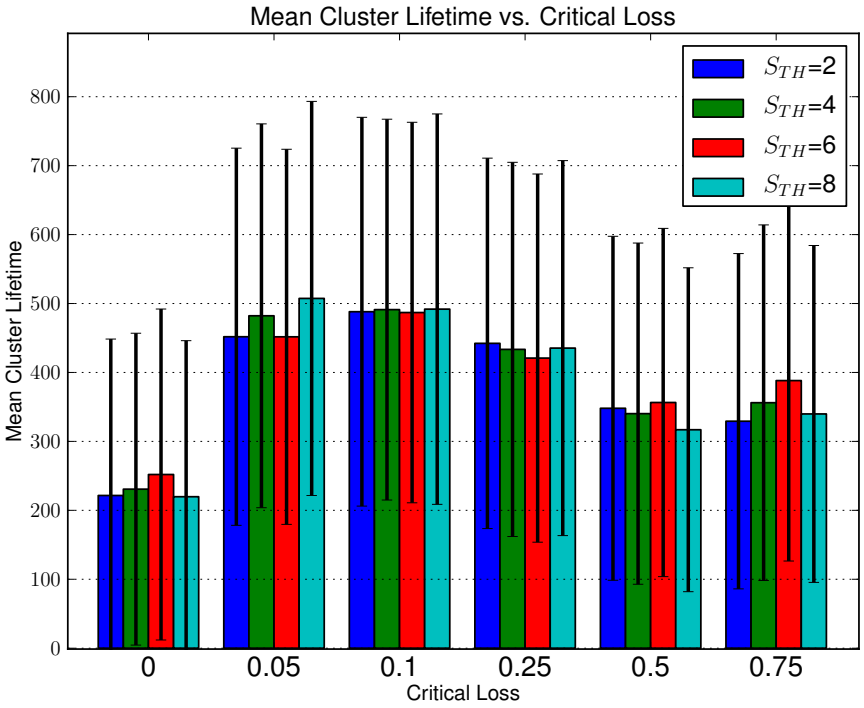
Cluster Size, shown in Figure 6.6, shows minimal change with respect to both P_{crit} and S_{TH}. The size of clusters is constrained to two or three members, with two being more prevalent. The depth of the cluster hierarchy is shown in Figure 6.7. Results are highly variable, but with respect to P_{crit} the trends show similarity to those of lifetime. Depth is improved by higher S_{TH} as well.

6.4 Conclusion

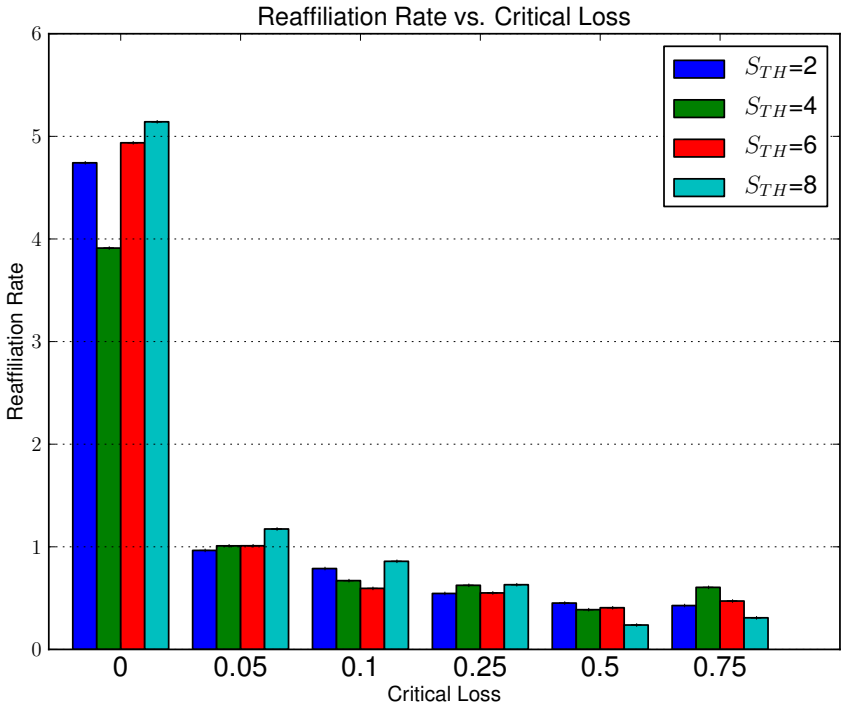
This chapter presented two modifications to the RMAC algorithm, accounting for Route Similarity as a metric of driver intention and channel sensing to prevent unnecessary reaffiliations. The results show that the Loss Probability modification has the most profound positive influence on performance, drastically decreasing reaffiliations while improving cluster lifetime. Route Similarity allows some improvement in certain cases, but results are varied.

Further research is needed to better incorporate the Route Similarity metric into CH selection methods. A small set of high similarity nodes may not necessarily include nodes close to or having low velocity relative to an unclustered vehicle, and an algorithm will need to consider the short term suitability metrics (such as internode distance and relative velocity) separate from long term metrics like Route Similarity. An alternative application of Route Similarity would be in prediction of future behaviour of CMs and CHs. CHs can use the metric to determine when it will leave the current road and which CMs it will lose. In the case that it loses all or most of its members, it can hand-off the CH role to one of the CMs, or order a suitable CM to become a CHM prior to losing contact.

The Loss Probability function in Algorithm 1 allows much greater stability of

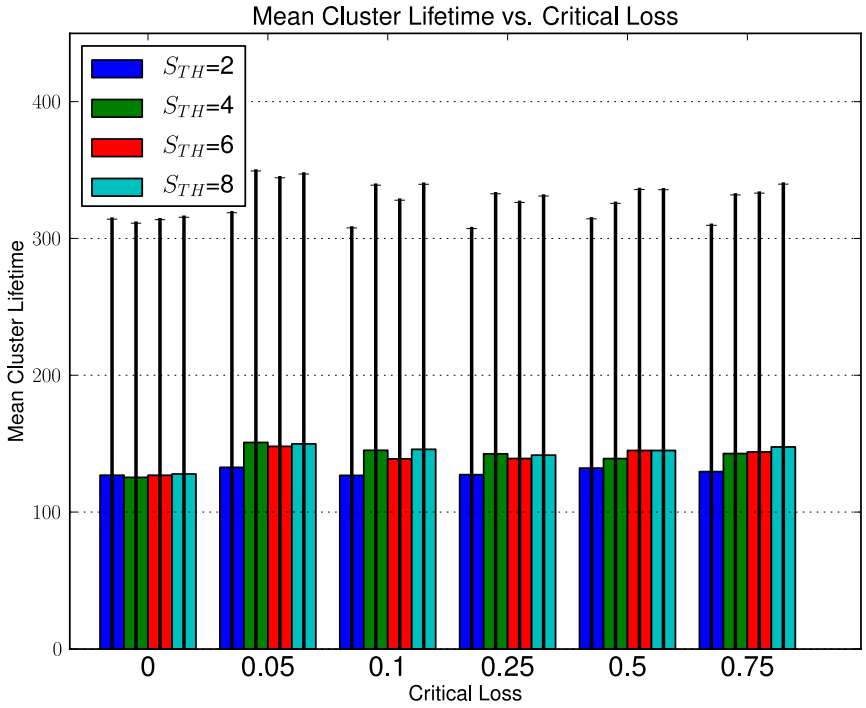


(a) Even a low P_{crit} has a high impact on performance.

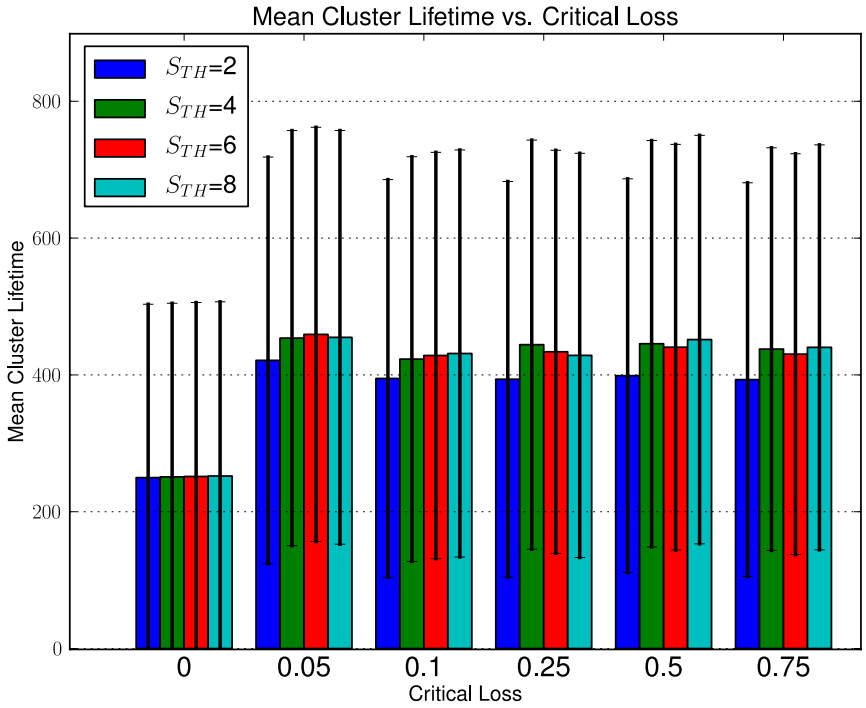


(b) CM stability is significantly improved by the Loss Probability modification.

Figure 6.3: Cluster performance on the highway.

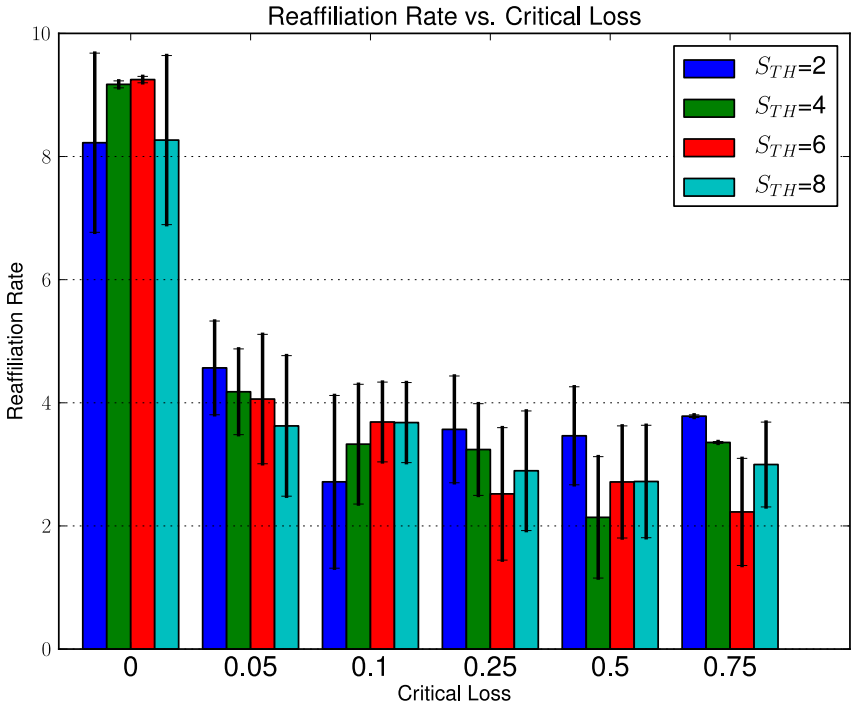


(a) No CBDs.

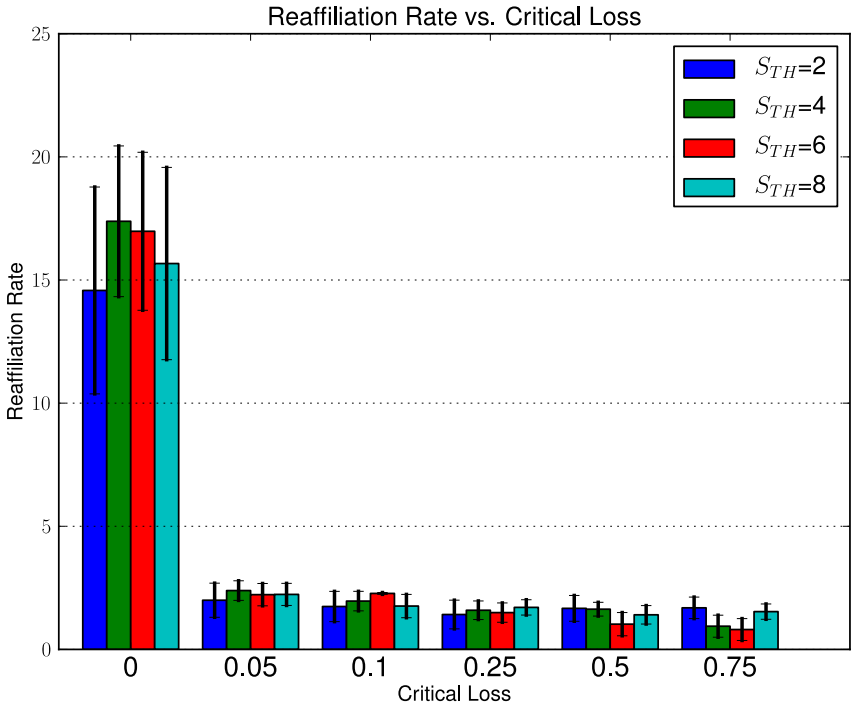


(b) Five CBDs.

Figure 6.4: Cluster lifetime on the grid with different CBD counts.



(a) No CBDs.



(b) Five CBDs.

Figure 6.5: Reaffiliation rate on the grid with different CBD counts.

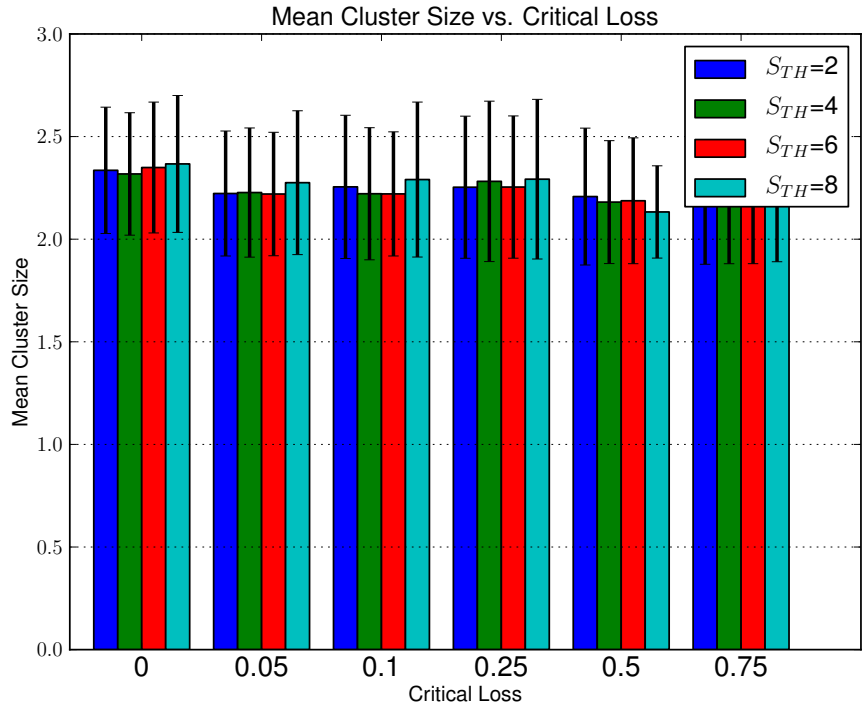


Figure 6.6: Size appears to be unaffected by either P_{crit} or S_{TH}.

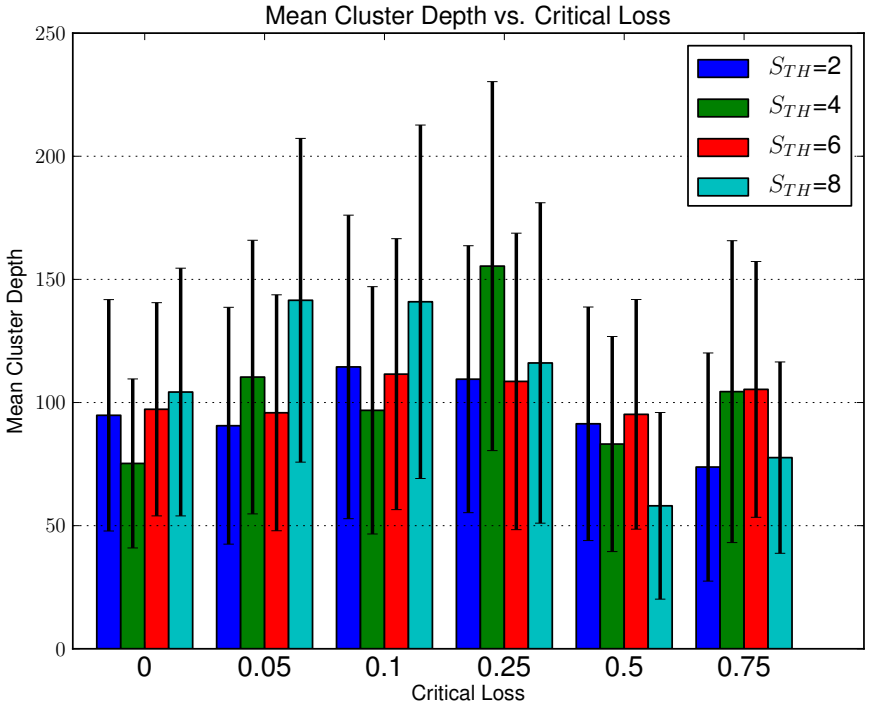


Figure 6.7: Erratic results, but there appears some improvement with respect to both P_{crit} or S_{TH}.

clusters in the face of variable channel dynamics. This modification could be applied to other clustering algorithms in literature to improve performance, particularly when routing is built on top of the clustering structure. However, there are avenues for advancement. Firstly, the Loss Probability function explicitly assumes a Rice-distributed channel, which is not always the case. This problem can be solved with a channel analysis phase to determine its distribution (using curve-fitting or Kolmogorov-Smirnov analysis), after which an appropriate Loss Probability function can be selected. Secondly, the calculation considers only variances in received signal strength, which is just one cause of packet loss. Bit Errors are another source, which can be influenced by channel dynamics, and should be considered in this method. As seen in Chapter 5, accounting for more elements of the channel allows a more accurate analysis of performance. In this same thread, accounting for more factors influencing packet reception probability will enable CHs to more accurately assess CM connections.

Chapter 7

Conclusion

7.1 Summary

This thesis primarily addressed two observations made in VANET clustering research. *Firstly*, protocol designers tend to employ unrealistic channel modelling. The extent of this problem in clustering research was presented through a literature survey, and the consequences of inaccurate modelling delineated. *Secondly*, new algorithms in clustering are often compared to older mobile network algorithms not suited for the challenges of VANET communication. Because of these trends, it is hard to determine which algorithm designs are beneficial. In this thesis, a reductionist environment-dependent channel model was developed through experimentation, and then used in a comparative simulation campaign of clustering algorithms. The design of an improved polling mechanism for RMAC resulted from the findings of the experiment survey.

7.2 Findings

The conclusions of the work are presented here.

7.2.1 Reductionist Channel Modelling

The gold standard of channel modelling, Raytracing, was deemed too complex and resource-hungry for most researchers to use. This thesis suggested that sufficient accuracy in channel modelling could be achieved via a reductionist approach, split-

ting the channel into separate and independent components. Separate models were selected from literature to address each of the individual components in order to achieve greater accuracy without a significant increase in simulation runtime.

It was identified that experiment campaigns to gather statistics of Small-Scale fading would also incur too great a resource expenditure, especially if a researcher wishes to control the structure of the road in simulation. An alternative approach to approximating Small-Scale fading parameters was presented, in which signal paths are resolved through a 2D representation of the static environment. Physical experiment on streets in Australia showed the viability of this approach and the circumstances under which it was useful. The three components were combined into the Urban Radio Channel (URC).

7.2.2 Extent of Channel Modelling Consequences

The finding was that ideal modelling gives an optimistic assessment of protocol performance. Packet loss, being unaffected by the changing environment, is significantly lower under the ideal models, and thus clusters were more stable and lasted longer. Additionally, under these ideal models, LID, HD, and LSUF appeared to function at the same level in terms of lifetime and reaffiliation rate. In terms of faulty affiliations, LID was shown to perform better than the others – this was unexpected since LID is such an arbitrary method. A study of these results would then conclude that the performance gains from multi-variable election criteria and management schemes are not justified, and a researcher would then opt for the simpler broadcast-based approaches that work in MANET scenarios.

By contrast, under URC, the algorithms performed much worse in terms of lifetime and reaffiliation rate, owing to vehicular obstructions and the presence of buildings. Moreover, the use of URC provides a better comparison, in that CH election criteria that account for vehicular dynamics are rewarded with improved performance over those that do not. Unsuitable designs are better distinguished under URC as well. AMACAD was particularly vulnerable to the realistic model, which revealed a fundamental flaw in its CH hand-off mechanism leading to cluster collapse. Mobility of nodes may cause a vehicle to come between a CH and a CM, disrupting the channel between them. With this in mind, the algorithm could be

redesigned to anticipate such a change, and turn the obstructing vehicle into a relay node between a CH and the CMs it blocks. The more pertinent finding was that ideal modelling prevents researchers from recognising performance disparity between competing algorithms. The key conclusion is that an accurate channel model will properly distinguish good design choices and efficacious clustering techniques from unsuitable ones, give a better understanding of protocol behaviour, and open new areas for investigation.

7.2.3 Road structure influences

Road structure, in particular the number of lanes, has a noticeable impact on cluster performance. This is because it constrains network topology in a way very different from MANET clustering. Certain algorithms (such as LSUF) perform poorly with low lane counts, and their advantages become apparent on highways with four or more lanes.

In light of this, research should tend toward designs that account for local road structure and vehicle density. Context-sensitivity with respect to lane count would allow clusters to modulate their behaviour to maximise performance in a variety of scenarios, both on the highway and in urban areas.

7.2.4 Using driver intention in CH election

RMAC was modified with a Route Similarity metric for CH election. The sum was a count of common consecutive route segments between travelling vehicles. To the knowledge of the author, this is the first time such methodology has been investigated, as bi-directional simulators, which exposes route and other data to the network simulation, are not yet widely used. However, the improvements were not as significant as anticipated.

The finding here is that a distinction should be made between short-term and long-term suitability metrics in CH election. Relative position and LET are short-term metrics, as they change with mobility within short periods of time following cluster formation. In this respect, they are better for initial CH election. Route Similarity is a long term metric, which can be utilised in the selection of better CH candidates once the cluster has been established. Furthermore, it will have a much

greater application in maintenance, such as selecting back-up CHs and predicting when clusters will disband.

7.2.5 Using channel analysis in cluster maintenance

Supplementing the Route Similarity modification, RMAC's polling scheme was augmented with channel analysis steps. A CH would drop a non-responsive CM if the channel between the two nodes following the last successful poll was significantly poor – that is, the probability of packet loss (assuming a Rice-faded channel) was high. Otherwise, the CH would keep the CM's data in its table and wait for the next polling round. Route Similarity found use as a long-term metric here, in that the CH would use the measurement to determine if the path of a delinquent CM had diverged. This allowed the algorithm to bypass the channel check altogether. This modification allowed a significant improvement in lifetime and stability.

It is then concluded that channel sensing and intention awareness are key metrics to be considered in cluster design. This approach was reactive in nature, responding to changes after-the-fact. However, there is no reason to think proactive analysis is impossible.

7.3 Future Work

Further work is required to improve the accuracy of the URC model and ensure its accuracy over a wider range of areas. The simulations conducted in Chapter 5 considered only buildings and other cars. Vegetation has an impact on channel behaviour as well – the results in Chapter 3 suggesting shadowing and absorption phenomena were in effect. Also, the Miranda experiments showed the K-factor can be non-zero in NLOS conditions, which is explained by a dominant component diffracting around building edges. The K-factor approximator should be modified to account for these effects. Additional work will incorporate a wider range of phenomena for consideration in the K-factor approximation. This includes smaller static objects, such as lamp posts, which are more likely to scatter signal power than reflect it; and vegetation that had been shown to absorb signal energy.

A new avenue for clustering research is the use of *Long-term* suitability metrics

in cluster maintenance. The criteria discussed in Chapter 2 with respect to mobility and spatial distribution are enough to elect a CH and form a cluster, while Route Similarity and driver intention/interest can be used for transferring the CH role and proactively anticipating cluster collapse. Small changes in driver intention, such as lane changes, can be used to predict changes in topology and act accordingly. All this data can be obtained through the use of bi-directional simulation while a comprehensive channel model will reveal the true performance of designs.

The experimental methodology applied in this thesis may be useful in the verification of VANET applications. Assuming an application is built on top of a clustering architecture, the findings of this thesis can be used to strengthen the architecture and improve application performance – especially in content streaming and other comfort applications. Traffic monitoring and infotainment, when investigated with respect to road structure and realistic channel conditions, may exhibit the same context-sensitivity as clustering. The cellular integration and P2P file sharing schemes of [79] and [68] respectively could be extended or modified with functionality employed based the environment detected by the vehicle. The methodologies put forward in this thesis – realistic channel modelling, road structure influence, and comparison between designs directed toward the same purpose – will identify the most efficient application solutions.

It should be noted that, in the surveyed literature, there seems to be an implicit assumption that the agent driving the vehicle will be a human – but there is little reason to think this will not change. Google [131], the UK Autodrive Consortium [132], and Phoenix Wingsin [133] are among the many companies testing autonomous transportation systems. These self-driving cars use combinations of LIDAR, GPS, and digitised maps to detect, process, and navigate their surroundings. Though this technology is making great strides, it focuses only on the individual car and its perceptions and decision-making processes. A VANET system will allow not only the sharing of network resources for infotainment applications, but the distribution of sensor data across a cluster. A cluster could then be turned into a processing farm, in which computing power is pooled to analyse the surrounding environment reconcile group decisions with an individual vehicle's agenda. One vehicle's blind-spots could be covered by a neighbouring vehicle watching its back – this

concept is an avenue for cooperative vehicular safety. Trailing vehicles can alert its cluster when a fast-moving car, such as an ambulance, is approaching from behind and trigger a coordinated lane change to allow the speeding vehicle past; similarly, a leading vehicle will detect and direct the cluster away from traffic congestion up ahead. A ubiquity of Internet access will allow municipalities to do away with road signing and lane marking (and corresponding maintenance costs), as vehicles could gather information about their environment from an online database in the form of "virtual signposts." This data may contain information about speed limit, changes in road topology, traffic density, and status of road work; even serve as traffic light systems that require no maintenance. Speeding fines and highway patrols could become a thing of the past with this technology, which can govern itself and free law enforcement officers to focus on other matters.

The key to deploying this technology is in adequate preparation of the necessary infrastructure and consistently high public interest. To do both of these, the benefits of the technology need to be clearly visible and easily demonstrated. As this thesis has shown, simulation is the best way to demonstrate the benefits, but the simulation must accurately depict reality – as shown in Chapter 2, in the field of clustering at least, the required level of reliability has not been reached. When it is achieved, researchers can show consumers how the technology will positively affect them and their interests, and approach investors and governments with deployment strategies. Simulations will reveal the best designs to solve the challenges of communication and the adoption ratio necessary for the advantages to be realised, thereby putting VANET technology on the road to roll-out.

Bibliography

[1] Y. L. Morgan. Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics. *IEEE Commun. Surv. Tutorials*, 12(4):504–518, 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.033010.00024.

[2] 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)*, pages 1–51, July 2010.

[3] IEEE Communications Society Emerging Technologies Committee. Vehicular networks and telematics applications sub-committee.

[4] International Standards Organization. ISO/TC 204 Intelligent transport systems.

[5] European Telecommunications Standards Institute. Intelligent Transport Systems.

[6] IEEE Vehicular Technology Society. Intelligent transportation systems committee (vt/its).

[7] L. Wischhof, A. Ebner, and H. Rohling. Information dissemination in self-organizing intervehicle networks. *Intelligent Transportation Systems, IEEE Transactions on*, 6(1):90–101, March 2005.

[8] J. Jakubiak and Y. Koucheryavy. State of the art and research challenges for vanets. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 912–916, Jan 2008.

[9] J.J. Blum, A. Eskandarian, and L.J. Hoffman. Challenges of intervehicle ad hoc networks. *IEEE Trans. Intell. Transport. Syst.*, 5(4):347–351, December 2004. ISSN 1524-9050.

[10] Eugenio Giordano, Raphael Frank, Giovanni Pau, and Mario Gerla. CORNER: a radio propagation model for VANETs in urban scenarios. *Proceedings of the IEEE*, 99(7):1280–1294, July 2011.

[11] Pao-Jen Wang, Chi-Min Li, and Hsueh-Jyh Li. Influence of the shadowing on the information transmission distance in inter-vehicle communications. In *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 3015–3019. IEEE, 2009. ISBN 978-1-4244-5122-7. doi: 10.1109/PIMRC.2009.5450288.

[12] A Mukunthan, C Cooper, D Franklin, M Abolhasan, F Safaei, and M Ros. Experimental validation of the corner propagation model based on signal power measurements in a vehicular environment. In *IEEE WCNC 2013*, Shanghai, 2013. IEEE WCNC.

[13] C. Cooper, A. Mukunthan, M. Ros, D. Franklin, and M. Abolhasan. Dynamic environmental fading in urban vanets. In *Communications (ICC), 2014 IEEE International Conference on*, pages 5641–5646, June 2014.

[14] C.F. Mecklenbrauker, A.F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink. Vehicular channel characterization and its implications for wireless system design and performance. *Proceedings of the IEEE*, 99(7):1189–1212, July 2011.

[15] C. Cooper, M. Ros, F. Safaei, D. Franklin, and M. Abolhasan. Simulation of contrasting clustering paradigms under an experimentally-derived channel model. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–6, Sept 2014.

[16] S. Vodopivec, J. Bester, and A. Kos. A survey on clustering algorithms for vehicular ad-hoc networks. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pages 52–56, July 2012.

[17] Rasmeet S Bali, Neeraj Kumar, and Joel J.P.C. Rodrigues. Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. *Vehicular Communications*, 1(3):134 – 152, 2014. ISSN 2214-2096.

[18] Craig Cooper, Montserrat Ros, Farzad Safaei, Daniel Franklin, and Mehran Abolhasan. A comparative survey of VANET clustering techniques. *IEEE Surveys and Tutorials*, 2015.

[19] Craig Cooper, Abhinay Mukunthan, Farzad Safaei, Montserrat Ros, Daniel Franklin, and Mehran Abolhasan. Including general environmental effects in k-factor approximation for rice-distributed VANET channels. *Physical Communication*, 14:32 – 44, 2015. ISSN 1874-4907.

[20] Craig Cooper, Montserrat Ros, Daniel Franklin, and Mehran Abolhasan. A simulation study of VANET clustering techniques. *IEEE Transactions on Intelligent Transportation Systems*, 2015.

[21] Craig Cooper, Montserrat Ros, and Farzad Safaei. Employing route similarity and channel sensing to improve cluster stability in vanets. In *IEEE Transactions on Intelligent Transportation Systems*, 2015.

[22] A. Ephremides, J.E. Wieselthier, and D.J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75:5673, 1987.

[23] Mario Gerla and Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 1(3):255–265, 1995. ISSN 1022-0038.

[24] P. Basu, N. Khan, and T. D C Little. A mobility based metric for clustering in mobile ad hoc networks. In *Distributed Computing Systems Workshop, 2001 International Conference on*, pages 413–418, 2001.

[25] Yan Zhang and Jim Mee Ng. A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 3161–3165, May 2008.

[26] Charalampos Konstantopoulos, Damianos Gavalas, and Grammati Pantziou. Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction. *Computer Networks*, 52(9):1797 – 1824, 2008. ISSN 1389-1286.

[27] W. Liu, C. Chiang, H. Wu, and C. Gerla. Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel. In *Proc. IEEE SICON'97*, pages 197–211, April 1997.

[28] Peng Fan. Improving broadcasting performance by clustering with stability for inter-vehicle communication. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 2491–2495, April 2007.

[29] Stefano Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks*, ISPAN '99, pages 310–, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0231-8.

[30] Rituparna Ghosh and Stefano Basagni. Mitigating the impact of node mobility on ad hoc clustering. *Wirel. Commun. Mob. Comput.*, 8(3):295–308, March 2008. ISSN 1530-8669.

[31] G. Wolny. Modified dmac clustering algorithm for vanets. In *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*, pages 268–273, 2008.

[32] Slawomir Kuklinski and Grzegorz Wolny. Density based clustering algorithm for VANETs. In *Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pages 1–6. IEEE, 2009.

[33] Slawomir Kuklinski and Grzegorz Wolny. Density based clustering algorithm

for vehicular ad-hoc networks. *International Journal of Internet Protocol Technology*, 4:149–157, 2009.

[34] Weiwei Li, A. Tizghadam, and A. Leon-Garcia. Robust clustering for connected vehicles using local network criticality. In *Communications (ICC), 2012 IEEE International Conference on*, pages 7157–7161, June 2012.

[35] Zhenxia Zhang, Azzedine Boukerche, and Richard Pazzi. A novel multi-hop clustering scheme for vehicular ad-hoc networks. In *Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '11, pages 19–26, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0901-1.

[36] E. Souza, I. Nikolaidis, and P. Gburzynski. A new aggregate local mobility (ALM) clustering algorithm for vanets. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5, May 2010.

[37] L.A. Maglaras and D. Katsaros. Distributed clustering in vehicular networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, pages 593–599, Oct 2012.

[38] L.A. Maglaras and D. Katsaros. Enhanced spring clustering in vanets with obstruction considerations. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–6, June 2013.

[39] Zhigang Wang, Lichuan Liu, MengChu Zhou, and N. Ansari. A position-based clustering technique for ad hoc intervehicle communication. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):201–208, March 2008.

[40] Peng Fan, JamesG. Haran, John Dillenburg, and PeterC. Nelson. Cluster-based framework in vehicular ad-hoc networks. In *Ad-Hoc, Mobile, and Wireless Networks*, volume 3738 of *Lecture Notes in Computer Science*, pages 32–42. Springer Berlin Heidelberg, 2005.

[41] Z.Y. Rawshdeh and S.M. Mahmud. Toward strongley connected clustering structure in vehicular ad hoc networks. In *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pages 1–5, Sept 2009.

[42] Zaydoun Y. Rawashdeh and Syed Mahmud. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. *EURASIP J. Wireless Comm. and Networking*, page 15, 2012.

[43] K.A. Hafeez, Lian Zhao, Zaiyi Liao, and B.N.-W. Ma. A fuzzy-logic-based cluster head selection algorithm in vanets. In *Communications (ICC), 2012 IEEE International Conference on*, pages 203–207, June 2012.

[44] I. Tal and G.-M. Muntean. User-oriented fuzzy logic-based clustering scheme for vehicular ad-hoc networks. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5, June 2013.

[45] Francesco Chiti, Romano Fantacci, and Giovani Rigazzi. A mobility driven joint clustering and relay selection for ieee 802.11p/wave vehicular networks. In *IEEE International Conference on Communications (ICC) 2014*, 2014.

[46] Yvonne Gunter, Bernhard Wiegel, and Hans Peter Grossmann. Medium access concept for VANETs based on clustering. In *2007 IEEE 66th Vehicular Technology Conference*, pages 2189–2193, Baltimore, MD, USA, September 2007. doi: 10.1109/VETECF.2007.459.

[47] Yvonne Gunter, Bernhard Wiegel, and Hans Peter Grossmann. Cluster-based medium access scheme for VANETs. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 343–348, Bellevue, WA, USA, September 2007.

[48] R.T. Goonewardene, F.H. Ali, and E. Stipidis. Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks. *IET Intell. Transp. Syst.*, 3(2):148, 2009. ISSN 1751956X.

[49] Jie Luo, Xinxing Gu, Tong Zhao, and Wei Yan. Mi-vanet: A new mobile infrastructure based vanet architecture for urban environment. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1–5, Sept 2010.

[50] T. Kwon, M. Gerla, V.K. Varma, M. Barton, and T.R. Hsing. Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks. *Proceedings of the IEEE*, 91(8):1210–1220, Aug 2003.

[51] Sheng-Shih Wang and Yi-Shiun Lin. Performance evaluation of passive clustering based techniques for inter-vehicle communications. In *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*, pages 1–5, May 2010.

[52] Ö. Kayiş and T. Acarman. Clustering formation for inter-vehicle communication. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 636–641, Sept 2007.

[53] R.R. Sahoo, R. Panda, D.K. Behera, and M.K. Naskar. A trust based clustering with ant colony routing in vanet. In *Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference on*, pages 1–8, July 2012.

[54] Luciano Bononi and Marco di Felice. A cross layered MAC and clustering scheme for efficient broadcast in VANETs. In *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pages 1–8, Pisa, Italy, October 2007. doi: 10.1109/MOBHOC.2007.4428735.

[55] Sheng-Tzong Cheng, Gwo-Jiun Horng, and Chih-Lun Chou. Using cellular automata to form car society in vehicular ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1374–1384, Dec 2011.

[56] Samo Vodopivec, Janez Beter, and Andrej Kos. A multihoming clustering algorithm for vehicular ad hoc networks. *International Journal of Distributed Sensor Networks*, 2014:8, 2014.

[57] N. Maslekar, M. Boussedjra, J. Mouzna, and L. Houda. Direction based clustering algorithm for data dissemination in vehicular networks. In *Vehicular Networking Conference (VNC), 2009 IEEE*, pages 1–6, Oct 2009.

[58] N Maslekar, M Boussedjra, J Mouzna, and H Labiod. C-DRIVE: clustering based on direction in vehicular environment. In *2011 4th IFIP International*

Conference on New Technologies, Mobility and Security, pages 1–5, Paris, February 2011.

[59] N. Maslekar, M. Boussedjra, J. Mouzna, and H. Labiod. A stable clustering algorithm for efficiency applications in vanets. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 1188–1193, July 2011.

[60] M.M.C. Morales, Choong seon Hong, and Young Cheol Bang. An adaptable mobility-aware clustering algorithm in vehicular networks. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pages 1–6, Sept 2011.

[61] M.M.C. Morales, Eung Jun Cho, Choong seon Hong, and Sungwon Lee. An adaptable mobility-aware clustering algorithm in vehicular networks. *Journal of Computing Science and Engineering*, 6:227–242, Sept 2012.

[62] C. Shea, Behnam Hassanabadi, and S. Valaee. Mobility-based clustering in vanets using affinity propagation. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, Nov 2009.

[63] A. Ahizoune, A. Hafid, and R. Ben Ali. A contention-free broadcast protocol for periodic safety messages in vehicular ad-hoc networks. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 48–55, Oct 2010.

[64] E. Dror, C. Avin, and Z. Lotker. Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 1–8, June 2011.

[65] Efi Dror, Chen Avin, and Zvi Lotker. Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks. *Ad Hoc Networks*, 11(7):2002 – 2015, 2013. ISSN 1570-8705.

[66] M.S. Almalag and M.C. Weigle. Using traffic flow for cluster formation in vehicular ad-hoc networks. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 631–636, 2010.

[67] M.S. Almalag, S. Olariu, and M.C. Weigle. Tdma cluster-based mac for vanets (tc-mac). In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–6, June 2012.

[68] M.S. Almalag, S. Olariu, M.C. Weigle, and S. El-Tawab. Peer-to-peer file sharing in vanets using tc-mac. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 84–89, March 2013.

[69] Ameneh Daeinabi, Akbar Ghaffar Pour Rahbar, and Ahmad Khademzadeh. Vwca: An efficient clustering algorithm in vehicular ad hoc networks. *Journal of Network and Computer Applications*, 34(1):207 – 222, 2011. ISSN 1084-8045.

[70] T. Gazdar, A. Benslimane, and A. Belghith. Secure clustering scheme based keys management in vanets. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5, May 2011.

[71] Neeraj Kumar, Naveen Chilamkurti, and JongHyuk Park. Alca: agent learning-based clustering algorithm in vehicular ad hoc networks. *Personal and Ubiquitous Computing*, 17(8):1683–1692, 2013. ISSN 1617-4909.

[72] Hang Su and Xi Zhang. Clustering-based multichannel mac protocols for qos provisionings over vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 56(6):3309–3323, Nov 2007.

[73] Xi Zhang, Hang Su, and Hsiao-Hwa Chen. Cluster-based multi-channel communications protocols in vehicle ad hoc networks. *Wireless Communications, IEEE*, 13(5):44–51, October 2006.

[74] A. Ahizoune and A. Hafid. A new stability based clustering algorithm (sbca) for vanets. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 843–847, Oct 2012.

[75] Fan Yang, Yuliang Tang, and Lianfen Huang. A multi-channel cooperative clustering-based mac protocol for vanets. In *Wireless Telecommunications Symposium (WTS), 2014*, pages 1–5, April 2014.

[76] T. Taleb, A. Benslimane, and K. Ben Letaief. Toward an effective risk-conscious and collaborative vehicular collision avoidance system. *IEEE Trans. Veh. Technol.*, 59(3):1474–1486, March 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2040639.

[77] Liren Zhang and Hesham El-Sayed. A novel cluster-based protocol for topology discovery in vehicular ad hoc network. *Procedia Computer Science*, 10(0):525 – 534, 2012. ISSN 1877-0509.

[78] T. Taleb and A. Benslimane. Design guidelines for a network architecture integrating vanet with 3g & beyond networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5, Dec 2010.

[79] Abderrahim Benslimane, Tarik Taleb, and Rajarajan Sivaraj. Dynamic clustering-based adaptive mobile gateway management in integrated VANET & 3G heterogeneous wireless networks. *IEEE J. Select. Areas Commun.*, 29 (3):559–570, March 2011. ISSN 0733-8716.

[80] R. A. Santos, R.M. Edwards, and N. L. Seed. Cluster-based location routing algorithm for inter-vehicle communication. In *PostGraduate Networking Conference (2003)*, April 2003.

[81] R. A. Santos, R.M. Edwards, and A. Edwards. Cluster-based location routing algorithm for inter-vehicle communication. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 2, pages 914–918 Vol. 2, Sept 2004.

[82] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, USA, 1999.

[83] L. Bernado, T. Zemen, F. Tufvesson, A.F. Molisch, and C.F. Mecklenbrauker. The (in-) validity of the wssus assumption in vehicular radio channels. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 1757–1762, Sept 2012.

[84] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee. Clustering in vehicular ad hoc networks using affinity propagation. *Ad Hoc Networks*, 13, Part B(0):535 – 548, 2014.

[85] ns. The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`, .

[86] R. Barr, Z. J. Hass, and R. van Renesse. JiST/SWANS: Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator. http://jist.ece.cornell.edu/. URL `http://jist.ece.cornell.edu/`.

[87] ns. The Network Simulator NS-3. `http://www.isi.edu/nsnam/ns/`, .

[88] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[89] Andras Varga. OMNeT++, 2009. URL `http://www.omnetpp.org/`.

[90] Christoph Sommer. Veins (Vehicles in network simulation), 2010. URL `http://veins.car2x.org/`.

[91] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. *SIGSIM Simul. Dig.*, 28 (1):154–161, July 1998. ISSN 0163-6103.

[92] M. Treiber. Traffic Simulator 3.0. `http://vwisb7.vkw.tu-dresden.de/~treiber`, 2005.

[93] J. Härri, F. Filali, C. Bonnet, and Marco Fiore. Vanetmobisim: Generating realistic mobility patterns for vanets. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, VANET '06, pages 96–97, New York, NY, USA, 2006. ACM. ISBN 1-59593-540-1. doi: 10.1145/1161064.1161084.

[94] *NCTUns 4.0: An Integrated Simulation Platform for Vehicular Traffic, Communication, and Network Researches*, 2007.

[95] P. Gburzynski and I. Nikolaidis. Wireless network simulation extensions in side/smurph. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 2225–2233, Dec 2006.

[96] OPNET Network Performance Management. `http://www.riverbed.com/products/performance-management-control/opnet.html?redirect=opnet`, 2014.

[97] Eugenio Giordano, Raphael Frank, Giovanni Pau, and Mario Gerla. CORNER: a realistic urban propagation model for VANET. In *Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 57–60. IEEE, February 2010.

[98] Daniel Krajzewicz, Eric Nicolay, and Michael Behrisch. SUMO - simulation of urban mobility, 2011. URL `http://sumo.sourceforge.net/`.

[99] Axel Wegener, Michal Piórkowski, Maxim Raya, Horst Hellbruck, Stefan Fischer, and Jean-Pierre Hubaux. TraCI: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium on - CNS '08*, page 155, Ottawa, Canada, 2008.

[100] Andreas Viklund. Mixim project, 2011. URL `http://mixim.sourceforge.net/`.

[101] Shie-Yuan Wang and Yi-Bing Lin. Nctuns network simulation and emulation for wireless resource management. *Wireless Communications and Mobile Computing*, 5(8):899–916, 2005. ISSN 1530-8677.

[102] Y. Hormann, H.P. Grossmann, W. H. Khalifa, M. Salah, and O.H. Karam. Simulator for inter-vehicle communication based on traffic modeling. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 99–104, June 2004.

[103] Martin I Krzywinski, Jacqueline E Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 2009. doi: 10.1101/gr.092759.109. URL `http://genome.cshlp.org/content/early/2009/06/15/gr.092759.109.abstract`.

[104] S. Bai and D.M. Nicol. Acceleration of wireless channel simulation using gpus. In *Wireless Conference (EW), 2010 European*, pages 841 –848, april 2010.

[105] OSM - open street maps. URL `http://www.openstreetmap.org/`.

[106] J. Nuckelt, D.M. Rose, T. Jansen, and T. Kurner. On the use of openstreetmap data for v2x channel modeling in urban scenarios. In *Antennas and Propagation (EuCAP), 2013 7th European Conference on*, pages 3984–3988, April 2013.

[107] 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*, pages 1–5, June 2013.

[108] Laura Bernad, Thomas Zemen, Er Paier, Gerald Matz, Johan Karedal, Nicolai Czink, Fredrik Tufvesson, Martin Hagenauer, Andreas F. Molisch, and Christoph F. Mecklenbruker. Non-wssus vehicular channel characterization in highway and urban scenarios at 5.2 ghz using the local scattering function. In *in Workshop on Smart Antennas (WSA*, pages 9–15, 2008.

[109] Ruisi He, Andreas F. Molisch, Fredrik Tufvesson, Zhangdui Zhong, Bo Ai, and Tingting Zhang. Vehicle-to-vehicle channel models with large vehicle obstructions. In *Communications, 2014. ICC '14. IEEE International Conference on*, 2014.

[110] D. Olivadese, F. Berizzi, A. Cacciamano, and A. Capria. A radar oriented ionospheric channel model based on ray-tracing theory. In *Radar Conference (EuRAD), 2010 European*, pages 105 –108, 30 2010-oct. 1 2010.

[111] Q. Sun, S.Y. Tan, and K.C. Teh. Analytical formulae for path loss prediction in urban street grid microcellular environments. *Vehicular Technology, IEEE Transactions on*, 54(4):1251–1258, July 2005.

[112] Mate Boban, Tiago T. V. Vinhoza, Michel Ferreira, Joao Barros, and Ozan K. Tonguz. Impact of vehicles as obstacles in vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(1):15–28, January 2011. ISSN 0733-8716. doi: 10.1109/JSAC.2011.110103.

[113] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2 edition, January 2002. ISBN 0130422320.

[114] A.F. Molisch, F. Tufvesson, J. Karedal, and C.F. Mecklenbrauker. A survey on vehicle-to-vehicle propagation channels. *Wireless Communications, IEEE*, 16(6):12–22, December 2009.

[115] C. Tepedelenlioglu, A. Abdi, and G.B. Giannakis. The ricean k factor: Estimation and performance analysis. *IEEE Transactions on Wireless Communications*, 24(5):799–810, May 2003. ISSN 1536-1276. doi: 10.1109/TWC.2003.814338.

[116] G. Azemi, B. Senadji, and B. Boashash. Ricean K-Factor estimation in mobile communication systems. *IEEE Communications Letters*, 8(10):617–619, October 2004. ISSN 1089-7798. doi: 10.1109/LCOMM.2004.835344.

[117] L. Bernado, T. Zemen, J. Karedal, A. Paier, A. Thiel, O. Klemp, N. Czink, F. Tufvesson, A.F. Molisch, and C.F. Mecklenbräuker. Multi-dimensional k-factor analysis for v2v radio channels in open sub-urban street crossings. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 58–63, Sept 2010.

[118] Laura Bernadó, Thomas Zemen, Fredrik Tufvesson, Andreas F. Molisch, and Christoph F. Mecklenbräuker. Time-, frequency-, and space-varying k-factor of non-stationary vehicular channels for safety relevant scenarios. *CoRR*, abs/1306.3914, 2013.

[119] G.G. Messier and J.A. Hartwell. An empirical model for nonstationary ricean fading. *IEEE Transactions on Vehicular Technology*, 58(1):14–20, January 2009. ISSN 0018-9545, 1939-9359. doi: 10.1109/TVT.2008.924988.

[120] L. Bernado, T. Zemen, A. Paier, J. Karedal, and B.H. Fleury. Parametrization of the local scattering function estimator for vehicular-to-vehicular channels. In *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pages 1–5, Sept 2009.

[121] J. Karedal, F. Tufvesson, N. Czink, A. Paier, C. Dumard, T. Zemen, C.F. Mecklenbrauker, and A.F. Molisch. A geometry-based stochastic mimo model for vehicle-to-vehicle communications. *Wireless Communications, IEEE Transactions on*, 8(7):3646–3657, July 2009.

[122] O. Renaudin, V.-M. Kolmonen, P. Vainikainen, and C. Oestges. Wideband measurement-based modeling of inter-vehicle channels in the 5-ghz band. *Vehicular Technology, IEEE Transactions on*, 62(8):3531–3540, Oct 2013.

[123] J. Jemai, T. Kumer, A. Varone, and J.-F. Wagen. Determination of the permittivity of building materials through WLAN measurements at 2.4 GHz. In *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 1, pages 589–593. IEEE, 2005. ISBN 978-3-8007-29. doi: 10.1109/PIMRC.2005.1651504.

[124] C.A. Grosvenor, R.T. Johnk, J. Baker-Jarvis, M.D. Janezic, and B. Riddle. Time-Domain Free-Field measurements of the relative permittivity of building materials. *IEEE Transactions on Instrumentation and Measurement*, 58(7): 2275–2282, July 2009. ISSN 0018-9456, 1557-9662. doi: 10.1109/TIM.2009.2013916.

[125] L.J. Greenstein, D.G. Michelson, and V. Erceg. Moment-method estimation of the ricean k-factor. *IEEE Communications Letters*, 3, June 1999.

[126] A Mukunthan, C Cooper, D Franklin, M Abolhasan, and F Safaei. Studying the impact of the CORNER propagation model on VANET routing in urban environments. In *Vehicular Technology Conference, 2012. VTC Fall 2012. IEEE 76th*, pages 1–5, Quebec City, 2012. IEEE VTS.

[127] J. Becerra, J.A. Nazabal, V. Torres, F. Esparza, M. Navarro, M. Beruete, C. Fernandez, and F. Falcone. Wireless channel modeling for campus sensor networks. In *Antenna Technology and Applied Electromagnetics the American Electromagnetics Conference (ANTEM-AMEREM), 2010 14th International Symposium on*, pages 1 –4, july 2010.

[128] B. Hagelstein, M. Abolhasan, D. Franklin, and F. Safaei. A general performance model for mac layer cooperative retransmission contention protocols. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 1584–1589, Dec 2013.

[129] C.S. Cooper, B. Hagelstein, and D. Franklin. Implementation of opportunistic cooperative diversity in an ad-hoc network using commodity hardware. In

Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International, pages 165–168, Aug 2012.

[130] A Abdi, C. Tepedelenlioglu, Mostafa Kaveh, and G. Giannakis. On the estimation of the k parameter for the rice fading distribution. *Communications Letters, IEEE*, 5(3):92–94, March 2001.

[131] Alexis Madrigal. The trick that makes google's self-driving cars work. 2014.

[132] Alexis Santos. Researchers testing frugal autonomous car system, aim for $150 price tag. February 2013.

[133] Roger Stansfield. First driverless 'car' on sale. January 2014.

[134] Shawn Hargraves. Allegro 5.0, 2014. URL `http://alleg.sourceforge.net/`.

Appendix A

When lacking building data

Chapter 3 detailed an alternative to experiment campaigns for modelling K-factor dynamics. Part of the solution involved extraction of building geometry and material data from the OSM file. Most areas of the world currently contained in OSM contain building data, and where such data is missing, it can be easily added by tracing around the buildings in a satellite image. This process is shown in Figure A.1. In recent years the system has grown more comprehensive in the features it offers, and more robust in the face of changing technology and mapping standards, allowing the experimental verification of the K approximator in Chapter 3.

The simulation surveys in Chapters 5 and 6, by contrast, occurred in simple grid and highway scenarios that do not match any street in reality. As a result, there are no building outlines available for these scenarios, and an alternative had to be found. This appendix describes a geometric algorithm that constructs outlines of buildings using the SUMO road network file. The algorithm has been called *BuildingSolver*.

A.1 The BuildingSolver

A.1.1 Input Data

The *BuildingSolver* algorithm operates on link and node data extracted from a SUMO net file. In this context, *node* means an intersection of two or more roads, and *link* means a single road connecting two nodes. In the SUMO net file, a single pair of nodes may share multiple links, each representing traffic flows in different directions. Furthermore, each link will contain at least one lane. A pre-computation

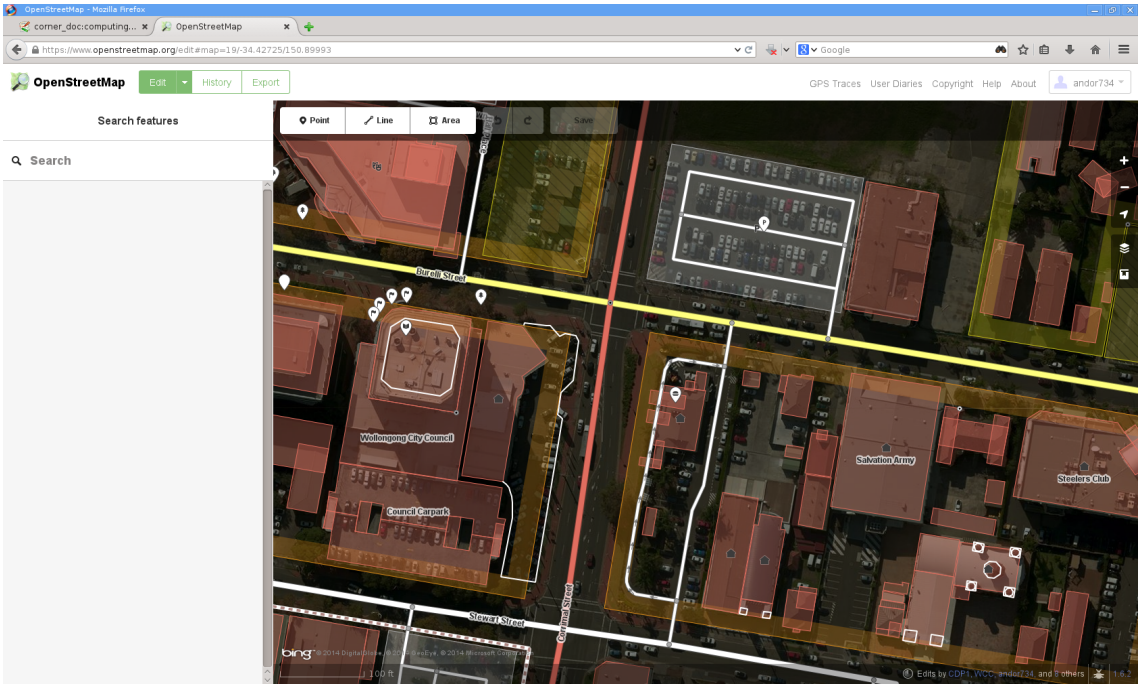


Figure A.1: Screen capture of OpenStreetMap editor on Burelli Street, Australia.

step is required to translate this data into a set of nodes share only one link with their connected neighbours, which is detailed in Algorithm 2.

A.1.2 Computation of building edges

After computing the reduced link set using Algorithm 2, the BuildingSolver iterates through each link and computes the vectors between its nodes, and assigns them to a vector set for each node. The next step is to sort these vectors clockwise from the positive y axis. Consecutive vectors are used to compute the vertices of each building edge adjacent to the node. The link is used as a means of matching each building vertex and connecting them to form a set of edges corresponding to building walls. The complexities of the process are shown in Algorithm 3.

A.1.3 Post-processing

Once the building edges are computed, they are tessellated into polygons representing the building geometry. This is done simply by iterating through all the edges, locating all the edges that have common vertices. Each building can be tagged with a particular material permittivity to be used in Equation 3.8.

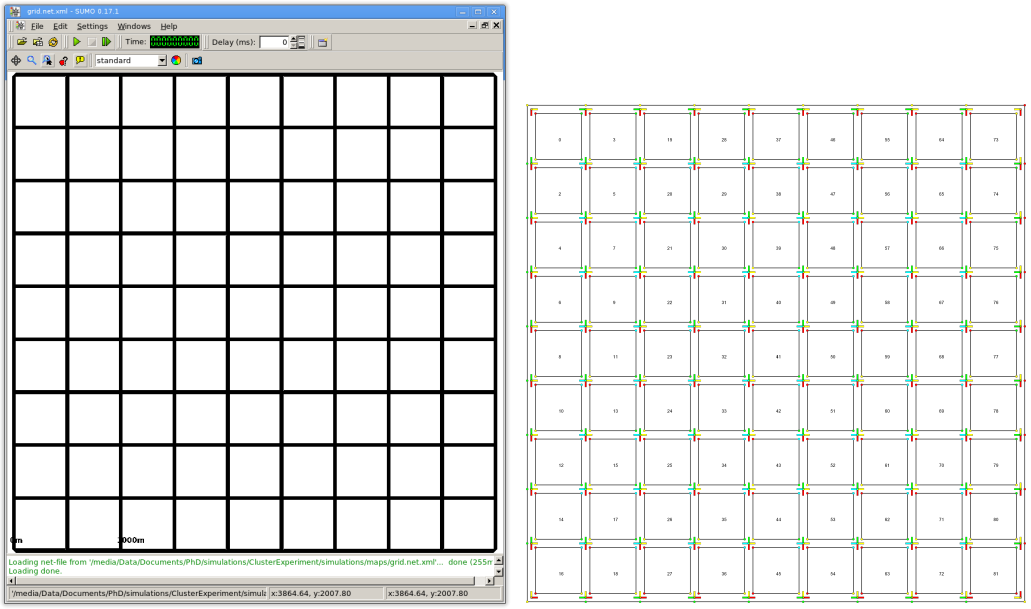
Algorithm 2: Data Preparation Step

Data:
Lane Width in metres, W_L
Set of nodes N
Set of raw links L where $L_i = (a_i, b_i, l_i)$ is a link between nodes $a_i, b_i \in N$ with l_i lanes
Result:
A reduced link set L^{red} where $L_i = (a_i, b_i, w_i)$ is the only link between nodes $a_i, b_i \in N$, having a width of w_i metres
begin
  $P = \emptyset$
  $L^{red} = \emptyset$
  for $L_i \in L$ **do**
    if $(a_i, b_i) \in P$ **then**
      Let L_x be $\{L_x \in L^{red} | \, (a_x = a_i \text{ and } b_x = b_i)\}$
      $w_x = w_x + W_L l_i$
    else
      $L^{red} \Leftarrow (a_i, b_i, W_L l_i)$
      $P \Leftarrow (a_i, b_i)$

A.2 Performance

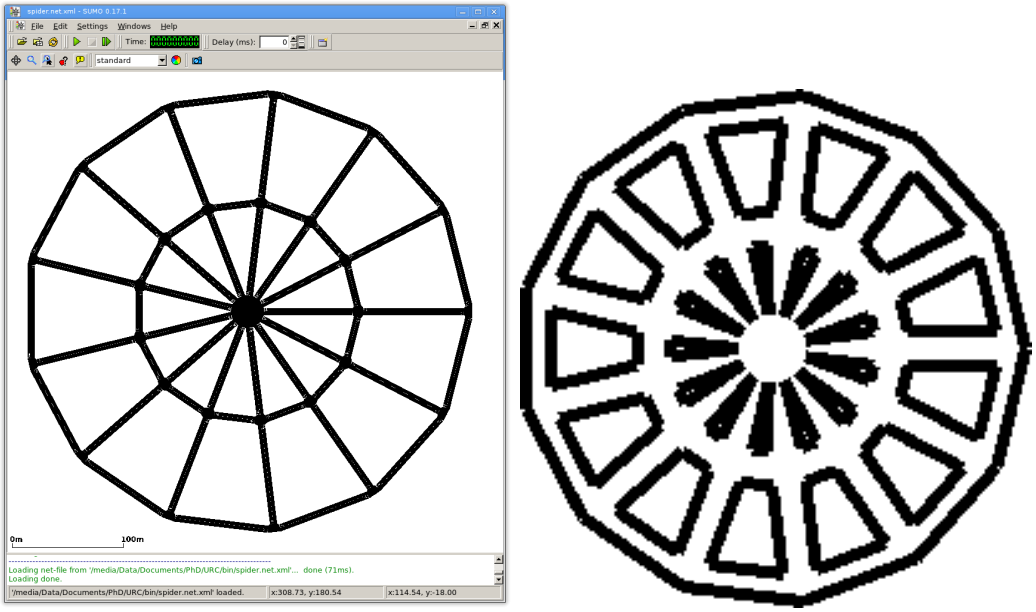
Figure A.2 shows the output of the BuildingSolver, given a SUMO file. The system works well with roads meeting at right angles, as shown in the images, but it also handles different angles as well. Figure A.3 shows the output of the BuildingSolver on a spider network generated by SUMO's *netgen* program.

The program nicely produces building geometry in the absence of OSM source data, which can then be fed into the K-factor approximator algorithm from Chapter 3.



(a) Input **(b)** Output

Figure A.2: The BuildingSolver program takes the SUMO data and constructs a building outline.



(a) Input **(b)** Output

Figure A.3: The BuildingSolver algorithm also works well with irregular angles.

Algorithm 3: Computation of Building Edges

Data:
Lane Width in metres, W_L
Set of nodes N where N_n is located at $\tilde{n}$
Set of reduced links L where $L_i = (a_i, b_i, w_i)$ is a link between nodes $a_i, b_i \in N$ having a width of w_i metres
Result:
A set of lines E representing building walls.
begin
  for $N_n \in N$ **do**
    $V_n = \emptyset$
    $C_n = \emptyset$
  for $L_i\,(a, b) \in L$ **do**
    $u = \tilde{b} - \tilde{a}$
    $V_a \Leftarrow \hat{u}$
    $V_b \Leftarrow -\hat{u}$
  for $N_n \in N$ **do**
    if $|V_n| > 1$ **then**
      `/* This node has multiple links. */`
      Sort V_n in order of increasing angle from $+\hat{y}$
      for $V_{n,v} \in V_n$ **do**
        Let $L_v(a_v, b_v, w_v)$ be $L_v \in L$ corresponding to $V_{n,v}$
        $\theta = \arccos\left(V_{n,v} \cdot V_{n,v+1}\right)$
        if $\theta = \pi$ **then**
          `/* Two links meeting in a straight line. */`
          $C_n \Leftarrow \tilde{n} + \frac{w_v}{\sqrt{2}} V_{n,v} \times \mathrm{Rot}\left(\frac{\pi}{2}\right)$
        else
          `/* Two links meeting at an angle. */`
          $v_{res} = \frac{w_v}{\sqrt{2}}\left(V_{n,v} + V_{n,v+1}\right)$
          $\theta = \arccos\left(V_{n,v} \cdot \hat{y}\right)$
          $\hat{\alpha} = V_{n,v+1} \times \mathrm{Rot}\left(-\theta\right)$
          if $\hat{\alpha} \cdot \hat{x} < 0$ **then**
            `/* Next vector is on left-hand side. */`
            $C_n \Leftarrow \tilde{n} - v_{res}$
          else
            `/* Next vector is on right-hand side. */`
            $C_n \Leftarrow \tilde{n} + v_{res}$
    else
      `/* This is a dead-end. Compute end vertices. */`
      $\tilde{u}_1 = V_{n,1} \times \mathrm{Rot}\left(\frac{\pi}{2}\right)$
      $\tilde{u}_2 = -\tilde{u}_1$
      $C_n \Leftarrow \tilde{n} + \frac{w_v}{\sqrt{2}}\left(\tilde{u}_1 - V_{n,1}\right)$
      $C_n \Leftarrow \tilde{n} + \frac{w_v}{\sqrt{2}}\left(\tilde{u}_2 - V_{n,1}\right)$
  `/* Calculate the building edges. */`
  $E = \{(A, B)\,\forall A \in C_a, B \in C_b | A$ and B correspond to link $L_i\,(a, b) \in L\}$

Appendix B

URC Read-Me

B.1 Introduction

This is the help document for the Urban Radio Channel (URC), developed by Craig Cooper and Abhinay Mukunthan. It has been released under the GLGPL.

This documents contains instructions on how to use the model in VEINS [90], a simulation framework that combines OMNeT++ [89] with SUMO [98] via the TraCI communications protocol [99]. The modules are built upon MiXiM [100]. To build URC, VEINS 2.0 or later is required.

The model combines calculations to account for three separate elements of the channel:

1. Path Loss - Accounted for using the CORNER model proposed by Giordano et al [10]. This uses the layout of an urban environment to determine whether a destination node is within Line-Of-Sight of a transmitting vehicle. Different path loss calculations are applied based on this classification. The model was verified and improved upon by Mukunthan el al [12].

2. Fading - An environmentally dependant fading model was presented in [13], which uses the building layout to estimate the multi-path components and estimate the parameters of a Ricean fading model. A pre-simulation step to compute the fading parameters is required.

3. Vehicular obstructions - A deterministic model based on Knife-Edge shadowing was developed and verified experimentally by Wang et al [11]. This was

selected to account for the transient nature of vehicular obstructions.

B.2 Setup

B.2.1 Building the library

To build URC for OMNeT++, open a terminal and navigate to where you unzipped the URC folder. Type the following command:

```
make install_OMNETPP
 OMNETPP_INSTALL_DIR=<install_dir>
 VEINS_ROOT=<veins_dir>
```

This will compile the URC library, and copy the headers and library files to the given installation directory. The directories provided must be relative directories.

If you want to build in debug mode, use

```
make install_OMNETPP DEBUGMODE=1
 OMNETPP_INSTALL_DIR=<install_dir>
 VEINS_ROOT=<veins_dir>
```

Note, you should make sure you've built VEINS before building URC, otherwise you'll get some annoying errors about cpp and cc files in OMNeT++.

B.2.2 Building the utilities

URC has several utilities for preparing data for the engine. Most of these are written in Python (See Section B.3.1), but two are written in C++ and need to be built:

1. *BuildingSolver* - a tool that constructs an outline of buildings from a SUMO map.

2. *Raytracer* - a tool that performs the K-factor approximation method in [13] to a given SUMO map.

To build these, type:

```
make BuildingSolver
```

or

```
make Raytracer
```

The binary will be stored in the *bin* directory of the URC root folder.

B.3 Utilities

URC has a number of utilities that are used to prepare road data for simulations.

B.3.1 Sumo2Corner

Our CORNER implementation is different from that of UCLA [10]. Their approach used geometric distance calculations to determine the link on which a transmitter and receiver were travelling, and then classified the Line-Of-Sight (LOS) on the fly. This method is not computationally efficient for very large simulations, unfortunately. We have leveraged the unique features of VEINS, which allows us to simply fetch the current road IDs from SUMO. The LOS state can then be indexed in a database of link-pairs and corresponding classifications. By matching the textual Road ID to a numerical ID, the model is made more efficient. This necessitates a presimulation step of analysing the road network and analysing the classifications between all links in the network.

The script *Sumo2Corner.py* contains all the preprocessing functions for preparing a map to work with the CORNER components of URC. It can operate on either a SUMO network file, or an OSM file.

The script accepts a valid filename, analyses the road network, and generates five files:

1. Link Lookup - A file of road links and numerical IDs of their corresponding nodes. Number of lanes is also included here.

2. Node Lookup - A file of road intersections and their geographical locations within the map.

3. Classification database - This database contains a list of source-destination road links, their LOS state (LOS, NLOS1, or NLOS2), and additional street information required for the pathloss calculation.

4. Link-Name Lookup - The classification database stores links and nodes as numerical indices for performance consideration. However, certain functions like the K-factor lookup relies on knowing the ASCII name of the link the current vehicle is occupying, as well as its lane. Thus, this lookup table allows URC to find the ID of a link based on its name.

5. Internal Link Lookup - SUMO includes internal links, which connect lanes across intersections. If a map is created with internal links, an additional lookup is required to match an internal link to the intersection the car is crossing. This will allow the CORNER functionality to determine an LOS state for cars crossing intersections.

Sumo2Corner.py may also work on OSM maps. When given an OSM map, it extracts the road geometry into a SUMO net file, while also extracting the building geometry data for use with the Raytracer (See Section B.3.3). The script may be given the option of stripping out paths such as walkways and bicycle tracks, which the SUMO converter mistakes for vehicular roads.

To use with a SUMO network file, use the command:

```
Sumo2Corner.py -n <netfile>
```

This will use the base filename, sans extension, to name the files it generates.

To use with an OSM map, use the command:

```
Sumo2Corner.py -o <osmfile>
```

To strip walkways from the OSM file before conversion, use the **-s** option.

Note, if you specify your own SUMO map, you won't be able to get building geometry using this script. To generate building geometry, use the *BuildingSolver* tool (See Section B.3.2).

B.3.2 BuildingSolver

Sumo2Corner.py can generate building geometry data from an OSM map, but not from a SUMO network file. The *BuildingSolver* tool traces around the road network and generates building outlines in between roads. This data can then be fed into the Raytracer (See Section B.3.3) for K-factor calculation.

To use program, type the following command:

```
BuildingSolver <basefile>
  <lanewidth> <footPathWidth>
```

This script will load the CORNER files generated in the previous section (specified by **basefile**) and generate a set of buildings. The filename will be **basefile.corner.bld**.

B.3.3 Raytracer

The Raytracer is significantly more complicated than the others. As this program can be computationally intensive, we have attempted to write it in a manner conducive to parallel processing. To that end, the program first generates configurations, dividing the map area into several smaller segments. This allows multiple instances of the program to be run on different areas of the map simultaneously.

Note: This technically isn't a raytracer, but a ray-launcher.

To begin, run *Raytracer* with the **-g** option, followed by these configuration tags:

- **-b** - Specify base filename of CORNER files as generated by *Sumo2Corner*. Manditory.
- **-r** - Specify a ray count. Default: 256
- **-G** - Receiver Gain. Default: 1
- **-i** - This is the number of metres between consecutive positions for K-factor approximation. Default: 10
- **-c** - Number of cores to use in tracing the rays. Default: 2
- **-N** - Number of areas to divide the map into. Default: 1
- **-l** - Road width in metres. Default: 5
- **-F** - Filename to write configurations into. Default: config
- **-V** - Visualise the raytracing in progress. See Section B.3.3

Once the configurations have been generated, run the program like so:

```
Raytracer <config> <run_number>
```

If you have a script or system that allows you to run a program with multiple configurations (such as a simulation cluster program), you can adapt your system to distribute *Raytracer* over multiple computers, to be run in parallel. When the entire run is complete, you will be presented with a file named **basename-run#.urc.k**. The basename will be the one specified in the configuration. These files will need to be combined into one file. At present, such a functionality has yet to be programmed.

Each file begins with the increment value specified in the file, followed by the number of source links contained within. The data is then organised in order:

1. Source Link ID
2. Increment along Source Link
3. Source Lane
4. Destination Link ID
5. Increment along Destination Link
6. Destination Lane
7. K-factor

When combining, ensure that the database is sorted according to Source Link ID.

Visualiser

It is possible to visualise the Raytracer program in progress. The program must be built for this first, using the command:

```
make Raytracer USE_VISUALISER=1
```

This uses Allegro 5.0 [134], a cross-platform multimedia library, to display the raytracer functioning. In order to build the visualiser, you will need a working installation of this library.

B.3.4 vehicleTypes.py

The script *vehicleTypes.py* allows you to assign specified vehicle types to cars in an existing SUMO route definition file. The script is executed with the command:

```
vehicleTypes.py <car_def> <in> <out>
```

The car definition file is a csv file. Each line has the format:

1. **id** - Name of the car type.
2. **accel** - Acceleration coefficient.
3. **decel** - Deceleration coefficient.
4. **sigma** - Driver imperfection.
5. **color** - Colour in SUMO GUI interface.
6. **length** - Length of car.
7. **width** - Width of car.
8. **height** - Height of car.
9. **weight** - Probability of this car being selected at random $(0, 1)$.

The first six parameters are explained in more detail on the SUMO wiki. The dimensions of the car are used in the shadowing model. Note that the total weight of all the cars in the file must sum to 1. An example vehicle definition file is in *data/vTypes.csv*.

B.4 Running Simulations

B.4.1 Incorporating the model

The simplest way to start simulations using URC is to subclass the *UrcScenarioManager* class. This object, when initialised, loads your CORNER and URC files into memory automatically. Then you must create your vehicles in OMNeT++, specifying *UrcPhyLayer*, or a subclass thereof, as your PHY layer module. Then

you can specify which parts of the model in the analogue models XML file. This is an example of the analogue models section of the channel configuration file.

```
<AnalogueModels>
    <AnalogueModel type="CORNER">
        <parameter name="interval"
                   type="double"
                   value="0.1"/>
    </AnalogueModel>
    <AnalogueModel type="CarShadow">
        <parameter name="interval"
                   type="double"
                   value="0.1"/>
        <parameter name="wavelength"
                   type="double"
                   value="0.124378109"/>
    </AnalogueModel>
</AnalogueModels>
```

URC is modular, meaning you can use only the components you wish. If you want only to use CORNER, you can omit the *CarShadow* segment. The environmental K-factors can be ignored in favour of a static value by adding the parameter *"k"* to the *CORNER* segment and setting a desired value. Conversely, if you want only the shadowing model, the *CORNER* segment can be removed. At present, the fading and CORNER functionality are inseparable.

B.4.2 Configuration

The *UrcScenarioManager* requires the following parameters to be specified.

- **linksFile** - Link file as created by *Sumo2Corner*.
- **nodesFile** - Node file as created by *Sumo2Corner*.
- **classFile** - CORNER Classification database.
- **linkMapFile** - Link name-ID lookup file.

- **intLinkMapFile** - Internal Link-node lookup file.
- **riceFile** - Precomputed K-factor database file, generated from *Raytracer*.
- **laneWidth** - Width of the lanes. This must be the same as was specified to the pre-simulation programs.
- **txPower** - Transmission power in milliwatts. This should be the same as was specified to the *Raytracer*.
- **systemLoss** - A loss factor associated with thermal noise. The value used in [12, 13] was 1142.9.
- **sensitivity** - Receiver sensitivity.
- **lossPerReflection** - Loss per reflection. This is not the value used in the *Raytracer*, but is used in the CORNER calculation.
- **componentFile** - Name of a datafile containing uncorrelated fading waveforms. This is used in calculation of the fading gain. The data used in [12, 13] is in *data/default.fading*.

B.5 Open Problems

The following problems need to be addressed.

B.5.1 BuildingSolver

The building material is preprogrammed as brick. The algorithm needs to be fixed so that buildings can be constructed around roads meeting at any angle, and different building materials need to be specified. Ideally, this program should be built into the *Sumo2Corner* script.

B.5.2 Raytracer

This program requires support for scattering from other stationary elements of the environment, such as trees and poles. Additionally, RSU support should also be implemented.

B.6 Conclusion

Bug reports or changes made, email us!