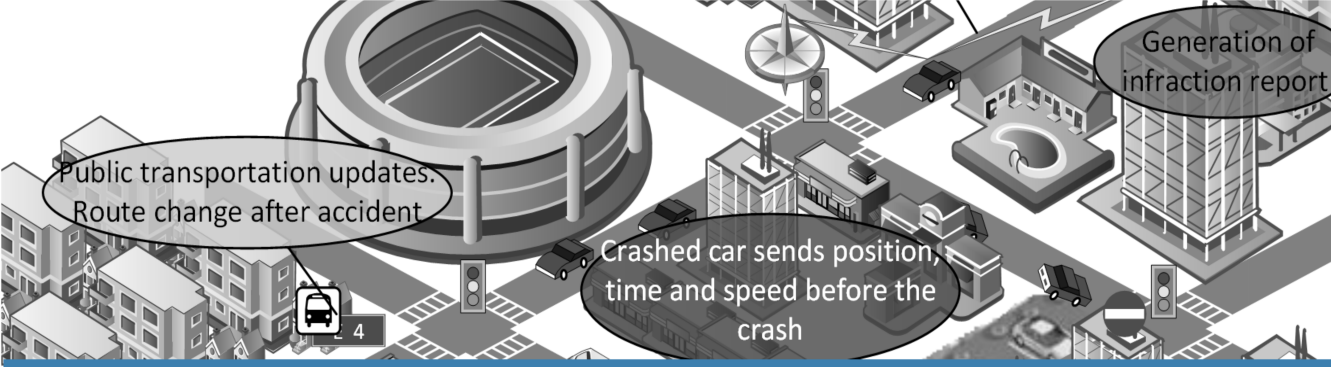


UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Enginyeria Telemàtica

Contribution to the design of VANET routing protocols for realistic urban environments

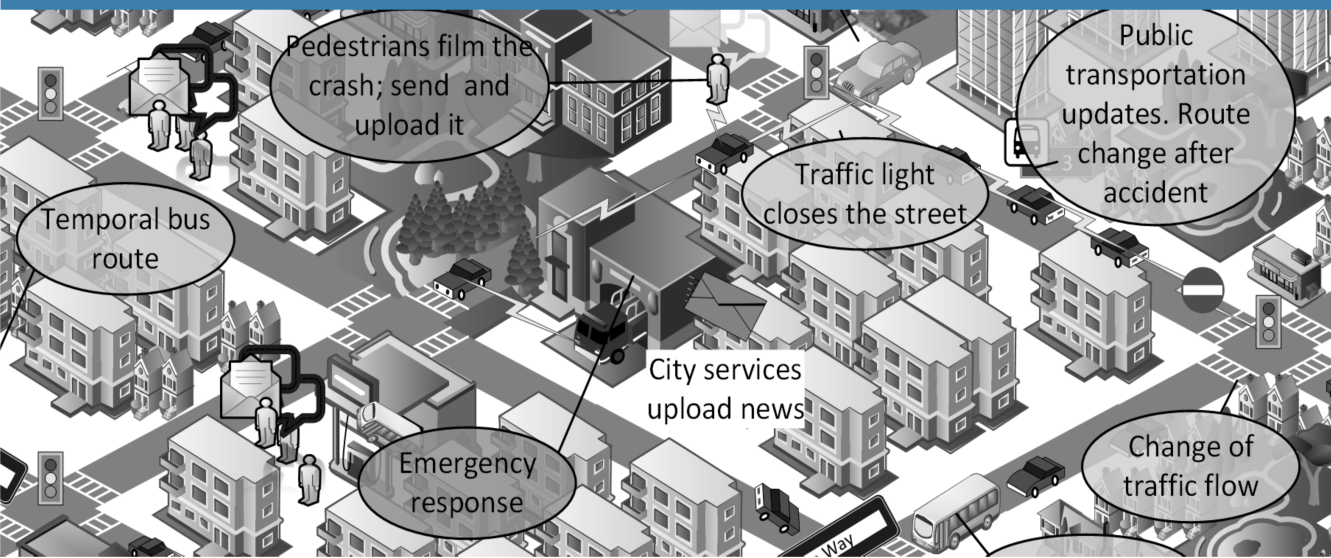
by

Luis Felipe Urquiza Aguiar

Ph.D. Advisor
Dra. Mónica Aguilar Igartua

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Network Engineering
in the
Department of Network Engineering

Barcelona, February 2016



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Enginyeria Telemàtica

Contribution to the design of VANET routing protocols for realistic urban environments

by
Luis Felipe Urquiza Aguiar

Ph.D. Advisor:
Dra. Mónica Aguilar Igartua

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Network Engineering
in the
Department of Network Engineering

Barcelona, February 2016

A mis padres, amor infinito
A mis hermanos, mis amigos
y por supuesto a ti, esposa mia, un ser maravilloso

“El problema con los protocolos de comunicaciones, es que cada uno va por su cuenta, cuando deberían colaborar más en entre ellos. Lo mismo pasa con la personas”

∼ Anónimo

Agradecimientos

Todo día llega si trabajas para que suceda. Me es difícil resumir tanta gente en tan pocas líneas. El apoyo de muchos ha sido crucial para que esta página pueda ser escrita. Esta es la última página que escribo de esta tesis y la más importante porque dar las gracias es lo mínimo que puedo hacer cuando la vida siempre me da más de lo que yo doy.

Quiero agradecer a mi tutora, la Dra. Mónica Aguilar por su inigualable comprensión y apoyo. Por la paciencia en sus correcciones, por motivarme y por siempre creer en las ideas que surgieron a lo largo de este tiempo. Gracias por todo.

A todos los demás autores de las publicaciones en las que participé, gracias por su ayuda. En especial a Carolina Tripp que siempre me ha ayudo, sobre todo en las simulaciones de nuestras propuestas, hacemos y estoy seguro de que seguiremos haciendo un gran equipo junto con Mónica. A Andrés Vázquez, gracias por las conversaciones sobre nuestros trabajos y por tus aportes en los trabajos en conjunto.

Al profesor Jordi Forné, quién me admitió al programa de máster y doctorado de Ingeniería Telemática, gracias por su ayuda. Quiero mencionar también al Dr. David Rebollo, siempre grato escuchar y aprender de él, gracias porque sin saberlo me motivó a aprender más sobre optimización. Debo mencionar también al Dr. Iván Bernal, sin sus estupendas clases de programación en C++ me hubiera sido imposible entender y manipular un simulador de red.

Debo agradecer a Dios tan presente en cada detalle de este mundo y en mi vida a través de mi familia y seres queridos. Mi familia, siempre preocupándose por mi más de lo necesario y haciéndome sentir tan especial aún estando tan lejos. A mi abuelita Luz, no pudiste esperarme a volver pero en la nube donde estés siempre me cuidarás y guiarás mis pasos. A mis amigos los de toda la vida, los de sangre y los que he hecho en el camino gracias por estar ahí, a mis nuevos amigos, los de Barcelona, son otra constancia de las bendiciones que Dios tiene para mí; sin ellos, Barcelona sólo sería esa ciudad maravillosa que es y no ese lugar donde también me siento en casa. Sus nombres tienen que estar aquí: Paúl, Nathaly, Lisseth, Xavier, Daniel, Carlos, Andrés, Francisco, Flor, Olga y Daniel, gracias por todo.

Esta aventura la inicié de cierto modo solo pero no puedo terminarla mejor acompañado. Gracias a mi Mónica, mi "ghost writer", mi amiga, mi compañera, mi motivación, mi amor, mi esposa, gracias por creer en mí, cuando yo no podía. Tú siempre me motivas para ser mejor.

Agradezco a mi país Ecuador y a su gobierno por la oportunidad de estudiar en el extranjero a través de las dos becas recibidas para completar mis estudios de postgrado. Sin esa ayuda yo no sólo no hubiera podido estudiar, también un nuevo mundo permanecería cerrado para mis ojos. Quiero dar las gracias a toda la gente de mi país que sin saberlo hicieron posible esto. Espero poder devolver esa ayuda inconsciente hacia mí cuando regrese.

Abstract

Urban areas account an increasing percentage of the world population and it is expected that this trend will continue growing in the future. The number and size of cities will increase, as well as the inherent problems related with its growth such as longer transfer times, higher generation of waste and pollution, among other issues. City councils in collaboration with committed citizens can try to improve the quality of life in the cities by incorporating information and communications technologies (ICTs) in the city services. A city that can proactively inform and warn citizens and include smart services in its daily operation becomes a smart city.

One of the main concerns of the cities' administration is mobility management. In Intelligent Transportation Systems (ITS), pedestrians, vehicles and public transportation systems could share information and react to any situation in the city. The information sensed by vehicles could be useful for other vehicles and for the mobility authorities. Vehicular Ad hoc Networks (VANETs) make possible the communication between vehicles (V2I) and also between vehicles and fixed infrastructure (V2I) managed by the city's authorities. In addition, VANET routing protocols minimize the use of fixed infrastructure since they employ multi-hop V2V communication to reach reporting access points of the city.

This thesis aims to contribute in the design of VANET routing protocols to enable reporting services (e.g., vehicular traffic notifications) in urban environments. The first step to achieve this global objective has been the study of components and tools to mimic a realistic VANET scenario. Moreover, we have analyzed the impact of the realism of each one of those components in the simulation results.

Then, we have improved the Address Resolution procedure in VANETs by including it in the routing signaling messages. Our approach simplifies the VANET operation and increases the packet delivery ratio as consequence. Afterwards, we have tackled the issue of having duplicate packets in unicast communications and we have proposed routing filters to lower their presence. This way we have been able to increase the available bandwidth and reduce the average packet delay with a slight increase of the packet losses.

Besides, we have proposed a Multi-Metric Map aware routing protocol (MMMR) that incorporates four routing metrics (distance, trajectory, vehicle density and available bandwidth) to take the forwarding decisions. With the aim of increasing the number of delivered packets in MMMR, we have developed a Geographical Heuristic Routing (GHR) algorithm. GHR integrates Tabu and Simulated Annealing heuristic optimization techniques to adapt its behavior to the specific scenario characteristics. GHR is generic because it could use any geographical routing protocol to take the forwarding decisions. Additionally, we have designed an easy to implement forwarding strategy based on an extended topology information area of two hops, called 2-hops Geographical Anycast Routing (2hGAR) protocol. Results show that controlled randomness introduced by GHR improves the default operation of MMMR. On the other hand, 2hGAR presents lower delays than GHR and higher packet delivery ratio, especially in high density scenarios.

Finally, we have proposed two mixed (integer and linear) optimization models to detect the best positions in the city to locate the Road Side Units (RSUs) which are in charge of gathering all the reporting information generated by vehicles.

Resumen

Las áreas urbanas del planeta representan un mayor y creciente porcentaje de la población mundial y no se espera que esta tendencia vaya a cambiar en el futuro. Las ciudades aumentan en número y tamaño y con ello crecen también los inherentes problemas asociados a su desarrollo, como tiempos de desplazamientos más largos, mayor generación de desechos y polución, entre otros. Los ayuntamientos en colaboración con ciudadanos comprometidos pueden lograr la incorporación de las tecnologías de la información y comunicación (TICs) en los servicios de la ciudad para así mejorar la calidad de vida de sus habitantes. Una ciudad que puede anticiparse a las necesidades de sus ciudadanos y adoptar servicios inteligentes en la gestión de su funcionamiento diario, se convertirá en una ciudad inteligente.

Una de las principales preocupaciones en la administración de las ciudades es la gestión de la movilidad de sus vehículos, debido a los problemas de tráfico como atascos y accidentes. En los sistemas inteligentes de transporte (SIT), peatones, vehículos y transporte público podrán compartir información y adaptarse a cualquier situación que suceda en la ciudad. La información obtenida por los sensores de los vehículos puede ser útil para otros vehículos y para las autoridades de movilidad. Las redes ad hoc vehiculares (VANETs) hacen posible la comunicación entre los propios vehículos (V2V) y entre vehículos y la infraestructura fija de la red de la ciudad (V2I). Asimismo, los protocolos de encaminamiento para redes vehiculares minimizan el uso de infraestructura fija de red, ya que los protocolos de encaminamiento VANET emplean comunicaciones multisalto entre vehículos para encaminar los mensajes hasta los puntos de acceso de la red en la ciudad.

El objetivo de esta tesis doctoral es contribuir en el diseño de protocolos de encaminamiento en redes ad hoc vehiculares para servicios de notificaciones (p.ej. reportes del estado del tráfico) en entornos urbanos. El primer paso para alcanzar este objetivo general ha sido el estudio de componentes y herramientas para simular un escenario realista de red ad hoc vehicular. Además, se ha analizado el impacto del nivel de realismo de cada uno de los componentes de simulación en los resultados obtenidos.

Así también, se ha propuesto un mecanismo de resolución de direcciones automático y coherente para redes VANET a través del uso de los propios mensajes de señalización de los protocolos de encaminamiento. Esta mejora simplifica la operación de una red ad hoc vehicular y como consecuencia aumenta la tasa de recepción de paquetes. A continuación, se ha abordado el problema de la aparición inesperada de paquetes de datos duplicados en una comunicación punto a punto. Para ello, se ha propuesto el filtrado de paquetes duplicados a nivel del protocolo de encaminamiento. Esto ha producido un incremento del ancho disponible en el canal y una reducción del retardo medio en la trasmisión de un paquete, a costa de un mínimo aumento de la pérdida de paquetes.

Por otra parte, hemos propuesto un protocolo de encaminamiento multi-métrica MMMR (Multi-Metric Map-aware Routing protocol), el cual incorpora cuatro métricas (distancia al destino, trayectoria, densidad de vehículos y ancho de banda) en las decisiones de encaminamiento. Con el objetivo de aumentar la tasa de entrega de paquetes en MMMR, hemos desarrollado un algoritmo heurístico de encaminamiento geográfico denominado GHR (Geographical Heuristic Routing). Esta propuesta integra las técnicas de optimización Tabu y Simulated Annealing, que permiten a GHR adaptarse a las características específicas

del escenario. Adicionalmente, hemos propuesto 2hGAR (2-hops Geographical Anycast Routing), un protocolo de encaminamiento anycast que emplea información de la topología de red a dos saltos de distancia para tomar la decisión de encaminamiento de los mensajes. Los resultados muestran que la aleatoriedad controlada de GHR en su operación mejora el rendimiento de MMMR. Asimismo, 2hGAR presenta retardos de paquete menores a los obtenidos por GHR y una mayor tasa de paquetes entregados, especialmente en escenarios con alta densidad de vehículos.

Finalmente, se han propuesto dos modelos de optimización mixtos (enteros y lineales) para detectar los mejores lugares de la ciudad d ubicar los puntos de acceso de la red, los cuales se encargan de recolectar los reportes generados por los vehículos.

Contents

List of Figures

List of Tables

Glossary

AAR	Automatic Address Resolution
ABE	Available Bandwidth Estimator
ACK	Acknowledgment frames
AGF	Advance Greedy Forwarding
ANT	Abstract Neighbor Table
AODV	Ad-Hoc On Demand Distance Vector
AR	Address Resolution
AR+	Enhanced Address Resolution
ARP	Address Resolution Protocol
AUs	Application Units
BE	Best Effort
BSP	Binary Space Partitioning
C2C	CAR 2 CAR Communication Consortium
C4R	Citymob For Roadmaps
CAAR	Coherent Automatic Address Resolution
CCCW	CW designed for Congestion Control
CDF	Cumulative Distribution Functions
CI	Confidence Intervals
CRC	Check Redundancy Codes
CW	Contention Window
DCW	Default Contention Window
DP	Dissemination Points
DSRC	Dedicated Short Range Communications
DTN	Delay Tolerant Network
EA	Extra Attenuation
FIFO	First In First Out
FT	Forwarding Technique
GBSR-B	Greedy Buffer Stateless Routing with Buildings detection
GHR	Geographical Heuristic Routing
GPCR	Greedy Perimeter Coordinator Routing
GPCR-D	Greedy Perimeter Coordinator Routing-Density
GPS	Global Position System

GPSR	Greedy Perimeter Stateless Routing
GPSR-MA	Enhanced GPSR with Movement Awareness
GRANT	Greedy Routing with Abstract Neighbor Table
HyBR	A Hybrid Bio-Inspired Bee swarm Routing protocol
iAODV	Irresponsible Ad Hoc On Demand Distance Vector
ICMP	Internet Control Message Protocol
IEEE	Institute Of Electrical And Electronics Engineers
I-GPSR	Improved Greedy Perimeter Stateless Routing
IID	Interface ID
IP	Internet Protocol
IPM	Intersection Point Method
ITS	Intelligent Transportation Systems
LOS	Line-Of-Sight
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MANOVA	Multivariate Analysis Of Variance
MCP	Maximum Coverage Problem
MCTTP	Maximum Coverage with Time Threshold Problem
MFR	Most Forward within the transmission radius R
MIB	Management Information Base
MIB	Management Information Base
MMMR	Multi-Metric Map Aware Routing protocol
MOPR	Movement Prediction-Based Routing
MTU	Maximum Transmission Unit
NCTUns	National Chiao Tung University Network Simulator
ND	Neighbor Discovery
nLOS	Near Line-Of-Sight
NLOS	Non-Line Of-Sight
Non-DTN	Non- Delay Tolerant Network
NS-2	Network Simulator 2
OBU	On-Board Unit
ODE	Ordinary Differential Equation
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing
PDF	Probability Density Function
PDR	Packet Delivery Ratio
PEM	Packet Error Model
PEP	Packet Error Probability
PER	Packet Error Rate
PF	Packet Filters
PLCP	Physical Layer Convergence Protocol
QAM	Quadrature Amplitude Modulation
RD	Router Discovery Process
RERR	Route Error Messages
RFC	Request For Comment
RREP	Route Replay Message
RREQ	Route Request
RSUs	Road Side Units
RT	Recovery Techniques
SA	Simulated Annealing

SINR	Signal to Interference And Noise Ratio
SNR	Signal to Noise Ratio
SOM	Stochastich Optimization Model
STS	Standardized Test Statistic.
SUMO	Simulation Of Urban MObility
SWIM	Shared Wireless Infostation Model
TA	Traffic-Aware
TAR	Traditional Address Resolution
TCP	Transmission Control Protocol
TR	Transmission Range
TSRP	Tabu Search base Routing Protocol
TTL	Time to Live
V2V	Vehicle-To-Vehicle Communications
V2I	Vehicle-To-Infraestructure
VANET	Vehicular Ad-hoc Network
VD	Vehicle Density
VDTN	Vehicular Delay Tolerant Network
VEINS	Vehicles In Network Simulation
VIN	Vehicle Identification Number
VLOCI	VANET LOCation Improve
WAVE	Wireless Access In Vehicular Environments
WMN	Wireless Mesh Networks
WSNs	Wireless Sensor Networks
2hGAR	2-hops Geographical Anycast Routing

1. Introduction

This chapter is dedicated to the definition of the objectives of this thesis. In addition, the chapter presents an overview of the contents dealed along the thesis.

1.1 Motivation

Nowadays, more than half of population live in cities and the proportion is expected to continue growing. It is foreseen that urban population increase up to 6.3 billion by 2050 (around 66% of the world population). Moreover, one out of two people lives in cities with a population with more than 1 million of inhabitants [31]. The cities increase in number, size and importance around the world. City councils and citizens have the responsibility to maintain or even improve the quality of life in their cities. In order to face this challenge, the society has been incorporating information and communications technologies (ICT) [36] in the hands of the cities' administration.

A "smart city" should be based on a seamless, efficient inter-operation of its resources and services to help to achieve the city's goals and daily tasks [57]. In this continuous process, ICTs are in charge of obtaining and analyzing huge data produced by the city. With timely information, cities could adapt proactively to the ongoing and future needs of their dwellers. Thus, "smart cities" are expected to support the wellness and growth of its population. Mobility has been one of the main concerns of cities' administration. How to optimize the trip times, decrease traffic jams, detect public transportation needs or avoid traffic accidents are examples where there is room to further improvement in cities' mobility.

Intelligent Transportation Systems (ITS) are in charge of making safer and more efficient the use of transport networks [108]. ITS use ICTs to increase the knowledge in traffic and mobility management. The recent approval of the European Union for the mandatory e-call system for all new cars in 2018 [35] to automatically inform about an accident in the road, is an example of the countries' commitment with a better and safer transportation. On the other hand, manufacturers are also aware that connectivity becomes more important and they expect that vehicles connected to the Internet reach 20% by 2020 [82]. Therefore, vehicles connected dynamically among them will be the rule and not the exception in near future. Vehicular Ad-hoc Networks (VANETs) [47] [105] is a field in which researches and manufacturers work to make connected vehicles possible.

Nowadays, some manufactures [79] [110] [13] are already working in vehicle-to-vehicle (V2V) communication to share information about the road sensed by vehicles mostly with safety purposes. However, the data that a vehicle senses about the road and its environment could also be of great utility for mobility management authorities to pursue their goal of having more efficient and safer transportation systems. We believe that management centers could take advantage of VANET routing protocols to gather the reports generated by vehicles almost online. More important, VANET routing protocols use multi-hop V2V communication to reach the reporting access point. Notice that this minimizes the installation of fixed infrastructure, not only the related to telecommunications but also those related with sensing. This is particularly important if we consider that the fastest growing urban agglomerations are medium-sized cities [31], which would require a huge investment to have this kind of infrastructure.

Considering the previous context, this thesis aims to contribute in the design of VANET routing protocols to enable reporting services in urban environments.

1.2 Objective of this thesis

The main motivation of this thesis is to contribute in the development of multi-hop vehicular communications. A vehicular communication system would increase the roads' safety and improve the quality of life in cities through a better management of the mobility issues.

In this thesis we aim at designing a delay tolerant routing protocol for vehicular ad hoc networks, that can be used in urban environments for reporting traffic management services. However, to achieve this goal it is needed a realistic simulation environment with emphasis on the aspects of channel and physical layers that guarantee trustworthy results. Also, in the design of proposals it has a tremendous importance all the aspects that contribute to increase the performance of the routing protocols and the vehicular network. Moreover, a vehicular multi-hop communication requires simple and efficient forwarding mechanisms that can work with limited topology information and that have to adapt rapidly to the fast changes of the network.

To achieve this general objective, we have worked in a down-up procedure step-by-step. First, we have studied the different components and tools that in conjunction build a realistic simulation scenario. We have tested systematically which channel and physical aspects produce dramatic changes in the simulation results.

Then, we have studied straightforward modifications of the different communication layers that can improve the overall performance of a multi-hop VANET communication. In particular, we have improved the Address Resolution procedure of a VANET suppressing the signaling messages of this task. We were also interested in the issue of having duplicate packets in unicast communication and we have proposed two simple filters to mitigate them.

Next, we have concentrated our effort in improving the performance of our geographical, traffic aware and delay tolerant Multi Metric Map aware (MMMR) routing protocol. Trying to maximize the number of packets delivered to their destination, we have researched heuristic techniques that could be applied to geographical routing protocols as MMMR. We aimed to merge these heuristic mechanisms in a single generic routing algorithm that can adapt its behavior to the scenario characteristics. Additionally, we have designed an easy to implement forwarding strategy based on an extended topology information area of two hops. The objective is to test the advantages and costs of a wider network topology knowledge.

Vehicles require fixed network points, typically called road side units (RSUs), to access to public city services like information about vehicle traffic state, among others. Furthermore, VANETs must be able to work without or with minimal fixed infrastructure. Hence, our research proposes a model to efficiently deploy RSUs in a city to allow vehicles to access smart services, although keeping the number of RSUs as low as possible.

1.3 Organization of this thesis

This thesis is organized in nine chapters. In this section we point out the contents of each of the chapters.

Chapter 2 describes the main components of a VANET and provides a survey of the routing approaches with emphasis in geographical routing. This chapter also summarizes some routing protocols including one designed by our research group called Multi-Metric Map aware Routing protocol (MMMR) [113], in which we test all the improvements proposed in this thesis.

Chapter 3 reports the VANET simulation software, the simulation scenario and the statistical tools employed to analyze the results obtained from our proposals.

In Chapter 4 we concentrate on surveying the channel and physical layer modeling in VANETs and on evaluating their impact in VANET simulations. We carried out performance comparisons among different models to mimic propagation, packet error probability and building attenuation. Our performance comparisons among different models intend to establish a trade-off between complexity and accuracy of the models.

In Chapter 5, the use of the Address Resolution (AR) procedures is studied for vehicular ad hoc networks VANETs. We analyze the poor performance of AR transactions in such networks and we present a new proposal called Coherent, Automatic Address Resolution (CAAR). Our approach inhibits the use of AR transactions and instead increases the usefulness of routing signaling to automatically match IP and MAC addresses.

In Chapter 6, we explain how the interaction between reliability mechanisms at MAC and routing layers lead to an unexpected generation of duplicate packets. Vehicles in VANETs use reliability mechanisms at different layers to mitigate packet losses. We propose two filters to detect duplicate packets at routing layer to prevent the propagation of unnecessary packet copies, so that the available bandwidth will increase as a consequence. We have incorporated this feature in our following contributions.

In Chapter 7, we propose different local-search heuristics to improve the performance of geographical routing protocols in VANETs. The presented algorithms are based on modifications of the well-known meta-heuristics Simulated Annealing and Tabu [135]. We divide our heuristics according to their operation in forwarding and recovery algorithms. Moreover, we combine them in a generic Geographical Heuristic Routing (GHR) protocol to improve the overall performance of the system. Besides, we seek the best configuration for GHR according to the vehicle density. We also propose 2-hops Geographical Anycast Routing (2hGAR) protocol that use information of nodes located two hops away to forward packets. Finally, we compare the performance of our both proposals GHR and 2hGAR.

Chapter 8 present two stochastic, mixed-integer linear optimization models to select the positions of RSU in the deployment of VANETs' fixed infrastructure. Both models take advantage of the inherent stochasticity provided by the vehicles' movements by using mobility traces to determine which are the best positions to place RSUs to maximize connectivity with the lowest number of RSUs. The first model mimics the routing behavior of such network and takes into account the maximum bandwidth capacity of the nodes and gateways. The second model simplifies the first one by using pre-computed multihop connectivity information. This makes that the second model can be solved for large data instances (e.g., big city areas).

Finally, conclusions, publications generated from this thesis and some future work guidelines are exposed in Chapter 9.

2. VANET Routing protocols

With the rapid development of wireless communication technologies, a new decentralized architecture based on vehicle-to-vehicle communications (V2V) has raised interest among car manufacturers, the research community and telecom operators. Vehicular Ad hoc NETwork (VANET) has emerged as a promised concept to improve road safety and traffic efficiency. VANETs are a specific type of traditional mobile ad hoc networks (MANETs), whose particular features make the routing process in this kind of network a challenge and an important research topic. This chapter describes the main components of a VANET, provide a survey of the routing approaches with emphasis in geographical routing. Finally, we summarize some routing protocols including the one designed by our research group called Multi-Metric Map aware Routing protocol (MMMR) [113], in which we test all the improvements proposed in this thesis.

2.1 Vehicular Ad Hoc Network Architecture

The VANET architecture can be divided into three domains: in-vehicle, ad-hoc and infrastructure domain [45] as it is depicted in Fig. 2.1. The "in-vehicle" domain refers to a local network inside each vehicle. It is composed of two types of units: an on-board unit (OBU) and one or more application units (AUs). An OBU is a device placed in the vehicle with communication capabilities. On the other hand, an AU is a device executing a single or a set of applications while making use of the OBU's communication capabilities. The AU and OBU are usually connected with a wired connection, while wireless connection is also possible (using e.g., Bluetooth, wireless universal serial bus or Ultra wide band). The ad-hoc domain is a network composed of vehicles equipped with OBUs and roadside units (RSUs) that are placed stationary along the road. OBUs of different vehicles form a mobile ad hoc network (MANET). OBUs and RSUs can be seen as nodes of an ad-hoc network, being mobile and static nodes, respectively,. An RSU can be attached to an infrastructure network, which in turn can be connected to the Internet. RSUs can also communicate to each other directly or via multi-hop, and their primary role is the improvement of road safety, by executing special applications and by sending, receiving, or forwarding data in the ad hoc domain. Infrastructure, whether roadside or embedded in the highway, is an important part of vehicular network systems because they can be used to help in the provision of security and privacy for VANET applications.

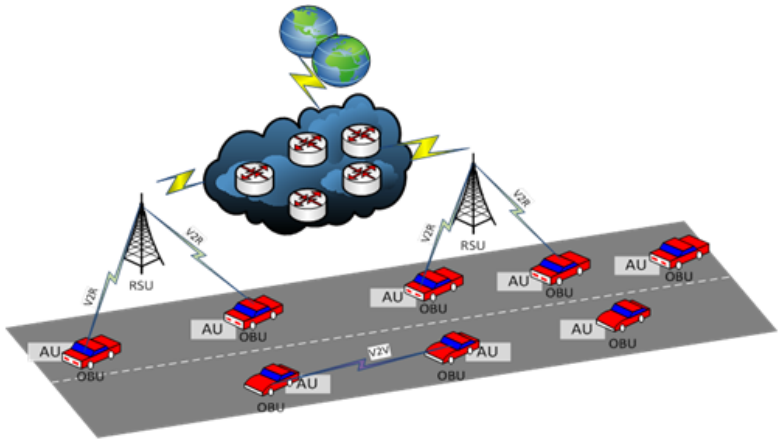


Figure 2.1: VANET architecture. Image adapted from [45]

2.2 VANET Characteristics

According to [69], VANETs can be distinguished from other kinds of ad hoc networks as follows:

Highly dynamic topology. Due to the high speed of movement of vehicles, the topology of VANETs is always changing.

Frequently disconnected network. Due to the same reason, the connectivity of a VANET may also change frequently. Especially when the vehicle density is low. There is a high probability that the network is disconnected.

Sufficient energy and storage. A common characteristic of nodes in VANETs is that nodes (i.e., vehicles) have sufficient energy and computing power (including both storage and processing).

Geographical type of communication. Compared to other networks that use unicast or multicast where the communication end points are defined by ID or group ID, VANETs often have a new type of communication where data are addressed taking into account the geographical positions of the intermediary nodes and destination node.

Mobility modelling and prediction. Due to the high mobility of nodes movement and the dynamic topology, mobility models and prediction play an important role in the design of network protocols for VANETs. Vehicular nodes are usually constrained by prebuilt highways, roads and streets. VANETs are usually operated in two typical communication environments. In highway traffic scenarios the mobility model usually is a one-dimensional movement; while in city scenarios the mobility model becomes much more complex. The streets in a city are often separated by buildings, trees and other obstacles.

Hard delay constraints In some VANET applications, the network does not require high data rates but has hard delay constraints. A representative example are safety applications.

Interaction with on-board sensors It is assumed that nodes are equipped with on-board sensors to provide information which can be used to form communication links and for routing purposes.

2.3 Routing Protocols classification

There are mainly two categories of routing protocols for VANETs [68] [100]: topology-based and geographic routing. Topology-based routing uses the information about links that exist in the network to perform packet forwarding. Geographic routing uses neighboring location information to perform packet forwarding.

2.3.1 Topology-based Routing

These routing protocols use the information of the links in the network to perform packet forwarding. They can be divided into proactive (table-driven) and reactive (on-demand) routing. Nevertheless, in fact the two options construct a routing table.

Proactive (table-driven). In Proactive routing, the routing information such as the next forwarding hop is maintained in the background regardless of communication requests. Control packets are constantly broadcast and flooded among nodes to maintain the paths or the link states between any pair of nodes even though some of the paths might be never used.

Reactive (On Demand). Reactive routing establishes a route only when it is necessary for a node to communicate with another node. It maintains only the routes that are currently in use, thereby reducing the burden on the network. Reactive routings typically has a route discovery phase where query packets are flooded into the network in search of a path. The phase completes when a route to destination is found.

2.3.2 Geographic (Position-based) Routing

In geographic routing, also known as position-based routing, the forwarding decision of a node is primarily made based on the position of a packet's destination and the positions of the node's one-hop neighbors. The position of the node's one-hop neighbors is obtained by the beacons sent periodically with random jitter. Nodes that are within a node's radio range will become neighbors of that node. Geographic routing assumes that each node knows its location, and the sending node knows the receiving node's location by the Global Position System (GPS) device.

Geographical routing protocols were designed as an alternative to the classical topological routing approach. Geographical routing attracted the attention of the VANET research community because their routing procedures could deal with the inherent fast topology changes in VANETs. Two procedures can be recognized in the operation of these protocols:

1. *Forwarding mechanism,* which determines the rules that a node must follow to choose the next forwarding hop until a packet reaches its destination.
2. *Recovery strategy,* which defines the actions that a node must perform when it does not have any neighbor that meets the forwarding criteria.

Recovery strategy classification

One of the first classifications of VANET routing protocols presented in [68] identifies two main types of position-based protocols depending on the recovery methods:

- *DTN* (*Delay Tolerant Network*). This kind of protocols implement the so-called "carry and forwarding" procedure that consists on storing a packet until the node finds a suitable next forwarding node. Carry and forwarding is a mechanism adequate only

for delay tolerant applications because this strategy introduces considerable delay in the data transmission.

- *Non-DTN*. Non-DTN protocols change the forwarding criteria to find a suitable node to forward the packet and avoid carrying it. Two are the most common alternatives: (1) the use of the right hand rule and (2) the construction of a recovery path through request/reply signaling messages. The destination of the recovery path could be either the final destination or a node that fulfills the forwarding requirements of the protocol.

Classification based on the forwarding mechanism

In the same classification [68], authors also differentiate position-based protocols according to the forwarding mechanism.

- *Non-overlay.* The forwarding mechanism takes routing decisions at each hop. Normally, these protocols only employ in their decision local information like GPS positions or additional data obtained through the exchange of messages between nodes.
- *Overlay.* In these routing protocols the routing decisions are taken by a set of representative (e.g. those located at junctions) nodes and the other nodes only relay the packet according to the taken decision. To select the next forwarding node, some of these protocols require additional information that has to be obtained from external sources.

  Typical examples are the topology of area (maps) and vehicle traffic information.

In [29] a classification of geographical traffic-aware routing protocols was presented. The routing decisions in traffic-aware (TA) routing protocols are influenced by traffic and network status through the integration of new routing metrics in them. Authors considered the “routing strategy" to differentiate the protocols. Despite the proposed categorization was done for TA protocols, it can also be used for non-TA protocols.

- *Full Path.* Vehicles construct a complete path between source and destination based on geographical information like distance between nodes and vehicles' density, among others.
- *Junctions or Anchor.* Routing decisions are taken only in the junctions based on routing metrics that include the vehicle traffic conditions.
- *Node.* In this class all the nodes take the forwarding decision at each hop seeking to forward the packet closer to destination at each hop.

The aforementioned classification can be seen as a subdivision into the assortment proposed by Lee in [68]. In this sense, *Junctions or Anchor* and *Full path* strategies can be included into the Overlay category. On the other hand, *Node* routing strategy is equivalent to the non-overlay forwarding mechanism. Moreover, traffic awareness by itself can be considered a factor to differentiate protocols that utilize more parameters in addition to the distance to select the next forwarding node.

As carry and forwarding mechanisms have increased their use as recovery mechanisms in VANET routing protocol, a recent classification of VDTN (Vehicular Delay Tolerant Network) protocols subdivides unicast position-based routing protocols based on the geographical knowledge needed to perform the routing decision. It divides the protocols in:

- *Geographical Location.* Here are classified protocols that only need to know the geographical position of the neighbors and their moving direction to estimate positions and compute any particular metric related to the distance.

- *Road maps.* It includes all the protocols that need in any way a map to take the forwarding decision.
- *Online.* Additionally to the use of maps, these protocols require real-time information of the vehicle's traffic.

This last classification can be applied to any of the previous two classifications. On the other side, online information is used in traffic-aware routing protocols that create full paths or take decisions at junctions due to the inherent additional cost that involves the use of external sources of information. *Road maps*, which is a halfway level of geographical knowledge can be used in both overlay and non-overlay networks.

Fig. 2.2 shows a taxonomy of geographical routing protocols, which combines the categorization factors previously reviewed: type of network, routing strategy, type of metric, knowledge that the protocols require to operate and recovery mechanism.

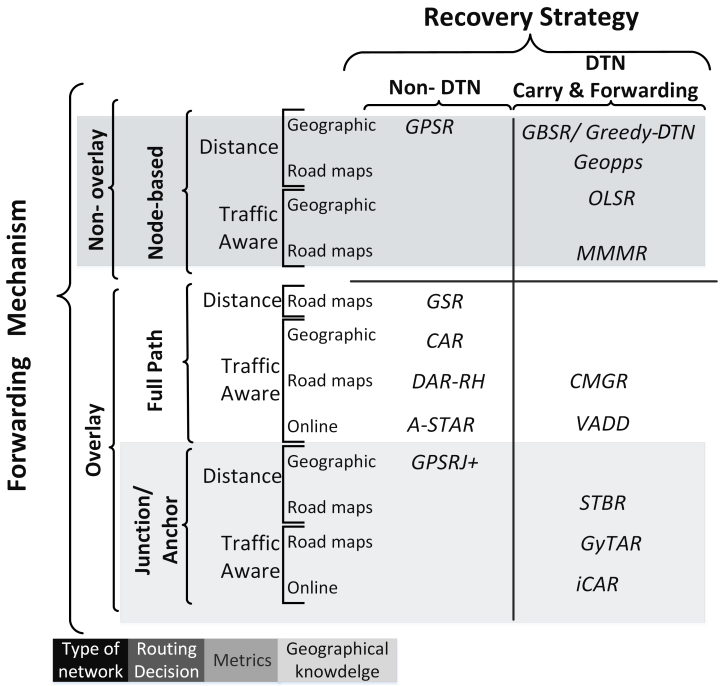


Figure 2.2: Taxonomy of geographical routing protocols for VANETs. It includes the name of a protocol located in each category. The details of each one can be found in [29], [100] and [22].

A recent survey of routing protocols for VANETs [100] maintains the simple taxonomy presented by Lee in [68] with an updated list of proposals in each category. Also, a careful explanation of the first geographical proposals, on which some current improvements are based, can be found in [22]. More detailed comparison between some geographical proposals have been done in [29] and [72].

2.4 Representative VANET routing protocols

2.4.1 Topology-based Protocols

Ad-hoc On demand Distance Vector (AODV)

Ad-hoc On demand Distance Vector AODV [90] is a topological, reactive protocol that uses the Bellman-Ford distance vector algorithm adapted to work in a mobile environment. It determines an end-to-end route to a destination only when a source node wants to send a packet. Routes are maintained as long as they are needed by the source while there is connectivity between nodes in the path.

To keep updated the routing table the AODV protocol has two tasks well defined: *Discovery of routes* and *route maintenance*. The *Route Discovery* takes place when there is no route to a certain destination. This task utilizes the Route Request (RREQ) broadcast message sent by the source in the upstream path till reach destination. Then, a Route Replay message (RREP) is unicasted downstream from destination to source. This way, source nodes set up forwarding nodes to their destinations. The second task is *Route Maintenance*, which is in charge of keeping a route while there is a communication between a source and a destination. Timers associated to each route are triggered to delete obsolete routes from the routing tables. The *Route maintenance* uses hello messages to send a *keep alive* signal to the other nodes. Route Error messages (RERR) inform that a node of a route is not available anymore and that the route has to be repaired. AODV uses two FIFO buffers as protection mechanism. A buffer to store packets while the route is being established and another buffer to prevent packet losses when a route breakage occurs with a short lifetime. AODV has been evaluated in a VANET scenario under different circumstances showing that AODV can still be suitable for VANET scenarios, especially for high density areas [37] [46] and with a low number of active connections [106].

Irresponsible Ad hoc On demand Distance Vector (iAODV)

Irresponsible AODV (iAODV) [42] is a modification of AOD Vthat improves the route discovery process carried out through broadcast dissemination of Route Request (RREQ) messages. In AODV, all nodes that receive a new RREQ can reply to the source through a Route Reply message or rebroadcast the RREQ packet. A node only sends a RREP message if it has a fresh-enough route or it is the destination. With Irresponsible AODV, a node only rebroadcasts a message with a probability given by:

$$p_{\text{forwarding}} = exp- \left(\frac{\sqrt{\rho}(z-d)}{c} \right) \tag{2.1}$$

where z is the transmission range of the current forwarding node n, d is the distance from the sender of the RREQ message to node n, ρ is the vehicle density in the neighborhood of n and c is a tunning parameter. The idea behind iAODV is to reduce the number of RREQ broadcasted each time that a new route is needed. Nodes close to the RREQ sender are more likely to inhibit the propagating of the RREQ message. Author of iAODV tested it in pedestrian and vehicular scenarios where it improves the overall performance of AODV in terms of packet losses and end-to-end delay.

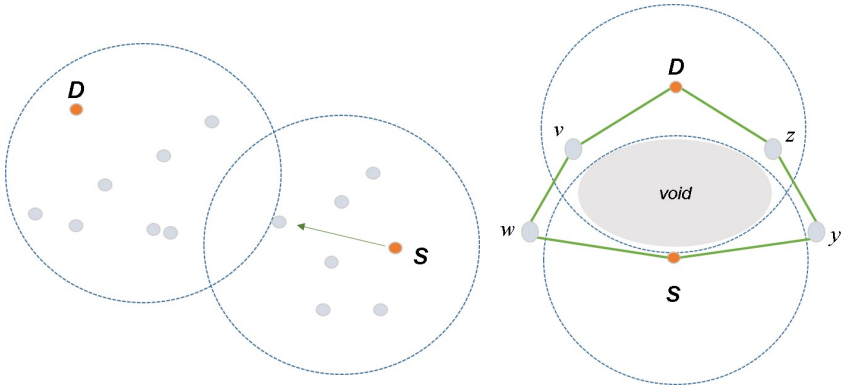
2.4.2 Geographical Protocols

In this section we briefly summarize the main geographical routing protocols proposed for VANETs in the last years.

Greedy Perimeter Stateless Routing for Wireless Networks (GPSR)

Greedy Perimeter Stateless Routing (GPSR) for Wireless Networks [62], is a well-known geographical routing protocol specially designed for VANETs. It is the base for many novel protocols that intend to improve the GPSR performance results, changing the requirements to forward packets or the logic to find a neighbor node, but always keeping the principle of forwarding the packet to the closest node to destination following a hop-by-hop fashion.

GPSR operates in two modes: *greedy forwarding*, which is used by default; and *perimeter forwarding*, which is used in those regions where greedy forwarding cannot be used. With greedy forwarding, the neighbor geographically closest to the packet's destination (greedy choice) is chosen as the packet's next hop, see Fig. 2.3a. When there is not a closest neighbor to destination, GPSR seeks to exploit cycle-traversing properties of the well known right-hand rule to route around voids. Fig. 2.3b, shows the path $(S \rightarrow w \rightarrow v \rightarrow D \rightarrow z \rightarrow y \rightarrow S)$ by using the right-hand rule to navigate around the pictured void. The sequence of edges traversed by the right-hand rule is called *perimeter*, see Fig. 2.3b.



(a) Greedy forwarding mode in GPSR.

(b) Perimeter forwarding mode in GPSR.

Figure 2.3: Forwarding operation modes in Greedy Perimeter Stateless Routing *(GPSR)*. Images adapted from [62]

GPSR poses two important drawbacks: the first one reported in [86] is the use of outdated information; the second one is the inefficient perimeter forwarding scheme [40]. There are many variants for GPSR that try to improve the GPSR performance by adding new restrictions to do an accurate forwarding decision or to use a more reliable recovery mechanism than perimeter.

Greedy Perimeter Coordinator Routing (GPCR)

GPCR [74] uses an improved version of greedy routing with a better recovery strategy than perimeter. Beacons sent periodically information about whether the sender is currently

located on a junction or on a street. To detect if a node is on a junction, they propose two strategies that do not require any map knowledge. (1.) A node is located on a junction if two neighbors are in transmission range but do not list each other in their neighbor tables. This approach needs that all nodes add the positions of their neighbors to the continuously sent beacon packets. Hence, the transmission of all neighbors needs much bandwidth space in the beaconing process. (2.) The node is on the same street with a one-hop neighbor if the driving direction of the neighbor is close to the driving direction of the node (i.e., the vehicle). Each node calculates a correlation coefficient for all its neighbors. A value close to 0 indicates that they are not in the same street, then the considered node is located in a junction.

When a node wants to start GPCR, it uses a restricted greedy routing, in which packets are sent to the one-hop neighbor who is closest to the position of destination. The difference is that neighbors in a junction are preferred even if their distance to destination is larger. Junction nodes have the benefit that they can reach neighbors located on all the connected streets. In the recovery mode a junction node decides, based on the right hand rule, to which further junction the packet should be forwarded.

Enhanced GPSR Routing in Multi-hop Vehicular Communications through Movement Awareness (GPSR-MA)

GPSR-MA [44] uses a metric calculated as a combination of speed (s) and direction (θ) between the straight line from destination to source and the line from destination to candidate forwarding node and the distance (d) between the Next Hop and destination, with a tolerance range computed as a percentage of the coverage range.

$$m(s,d,\theta) = \alpha_{\text{speed}} \cdot f(s) + \alpha_{\text{distance}} \cdot f(d) + \alpha_{\text{movement}} \cdot f(\theta) \tag{2.2}$$

With $\alpha_{\text{speed}}, \alpha_{\text{distance}}, \alpha_{\text{movement}}$ are weights associated with each function. The route with the highest total score $m(s,d,\theta)$ is selected as Next Hop.

Movement Prediction-based Routing (MOPR)

MOPR [78] implements an estimation of connectivity called link stability (LS) to select the next forwarding node. The link stability is defined as:

$$LS[i,j] = Lifetime[i,j]/\sigma \tag{2.3}$$

The link's lifetime between nodes i and j ($Lifetime[i,j]$) is defined as a period in which a neighbor is into the coverage range of the node and σ is a constant value that depends on the used routing protocol. The neighbor with the highest Link Stability is selected.

In MOPR over GPSR when a vehicle wants to send or forward data, it first estimates the future geographic location for each neighbor after a duration time T in seconds. Then, it selects as Next Hop the closest neighbor to destination which has not a future location out of its communication rage after T seconds.

Advance Greedy Forwarding (AGF)

In Advanced Greedy Forwarding over GPSR presented in [86] nodes periodically send Hello Messages with speed and direction information. Also, the travel time of a data packet is added in the header.

A node receiving a packet checks if the destination is listed in its neighbor table and the entry is still valid, taking into account the packet travel time and the node's and the destination's

velocity vectors. If the destination is in the neighbors table, but the new position estimation tells that the destination is most likely out of the range, then the node closest to the new position of the destination is chosen as Next Hop. If the destination node is not in the neighbors' table then the node consults the packet travel time and estimates whether it may reach the position of the destination recorded in the packet header within one hop. If it is possible, a non-propagating broadcast (RREQ) is sent around, with the search for the destination. If no answer is received then the next node closest to the destination node is chosen, and the process repeats until the packet reaches the destination node.

Improved GPSR Routing Protocol

The proposal of Improved GPSR [137] consist on nodes using Hello Messages to inform of their position, speed $v(t)$ and number of neighboring nodes $\rho(t)$. With this information each node can predict the direction of a node comparing two consecutive positions received by that neighbor. The next forwarding node is chosen from the neighbors that are traveling towards the destination node, and it is the node that has the minimum value of a mixed metric $D(t)$ shown in equation 2.4.

$$D(t) = \alpha \frac{d}{v(t)} + (1-\alpha)\frac{\pi R^2}{\rho(t)} \qquad \alpha \in \{0,\ 1\} \tag{2.4}$$

where d is the distance from the forwarding node to destination, R is the transmission range and α is a weigh factor to balance vehicle density and distance to destination. If there were two neighbors with the minimum D(t), the slower node would be selected because the communication link could be maintained relatively longer.

Greedy Routing with Abstract Neighbor Table (GRANT)

In GRANT [97], each node maintains the forwarding information of the neighbors located at $x-$hops. This gives to every node a far sighted vision of the best route to be chosen to avoid a local maximum. To select the next hop neighbor s, each node n computes a routing metric as the product of the distance between n and s, the distance between s and destination d, and a penalty factor per hop for multi-hop neighbors. The node s offering minimum metric is chosen as the next hop. GRANT sends its Abstract Neighbor Table (ANT) in the beacon message. The ANT separates the map into areas and includes only one representative neighbor per area.

Simulations performed with x=2 (i.e., 2-hop neighbors), prove that the path lengths of most of the routes in GRANT are shorter than the path lengths obtained by traditional greedy routing. Also, in traditional greedy routing the number of times the packet is recovered per route is higher than in GRANT. In the simulations the x hop neighbors were assumed to be available, which is not always possible. There are large beacon overheads and inaccuracies in calculations which were not taken into consideration.

GPCR-D: A Topology and Position Based Routing Protocol in VANET

Greedy Perimeter Coordinator Routing based on Density (GPCR)-D [138] is a topology and position based routing protocol. It can detect dynamically the network density and establish the local areas with high node density, where the vehicles' speed is limited and topology changes slowly. The authors use a shortest path algorithm aimed at delivering

packets speedily. Outside the local areas topology changes rapidly, so greedy forwarding is used to avoid restoring and maintaining links frequently. Thus, GPCR-D makes full use of their respective advantages. The recovery mode of GPCR-D selects the neighbor whose direction is the closest to destination in the neighbor table as the next forwarding hop. The simulation shows that GPCR-D works more effectively than GPCR in terms of average delivery success rate and end-to-end time delay.

A Hybrid Bio-inspired Bee swarm Routing protocol (HyBR)

HyBR [12] is a hybrid protocol which applies a topology-based routing approach when the network density is high, like a city, and applies a geography-based routing approach when the network is not dense. Network density is used to determine the type of routing method to be used in the VANET environment. Using its positions table, the source node checks the network between source and destination after dividing the network in a set of subnetworks where each one has a perimeter equal to the transmission range. If the number of nodes in a sub-network is higher than a threshold α called density coefficient, calculated using Eq.(2.5), the topology-based routing is applied; otherwise, the geography-based routing is used.

$$\alpha = \frac{TR}{\beta} \tag{2.5}$$

where α represents the number of vehicles in the checked sub-area, and TR is the transmission range. Topological routing is based on AODV operation, whereas geographical routing establishes a route, based on the distance between nodes. The route with the minimum summation of distance between the nodes in the route is selected. Results show that this protocol improves GPSR but not AODV in high density scenarios. However, it uses an unrealistic assumption that a node can be aware of all the other nodes' positions in the network.

2.5 Our routing proposals

In this section, two protocols developed by our research group are described. Both will be the basis for the contributions presented in this thesis that aim to improve the performance of VANET routing protocol in urban environments.

These two protocols follow a geographical approach. Greedy Buffer Stateless Routing-Building detection (GBSR-B) [111] is an improvement of GPSR. On the other hand, the Multi-Metric Map aware Routing (MMMR) protocol [113] employs all the enhanced of GBSR-B and also adds three other new routing metrics to take the best forwarding decision.

2.5.1 Greedy Buffer Stateless Routing with Buildings detection (GBSR-B)

GBSR-B [111] tackles the out-of-date information problem of a neighbor to improve the next forwarding decision. Also, GBSR-B uses carry and forwarding instead of the Perimeter mode when a local maximum arises, i.e. when a next forwarding node cannot be found using the greedy mode.

Improving the position estimation of a node

To determinate if a neighbor is actually reachable and consequently a good next forwarding hop, GBSR-B adds to the 1-hop signaling hello messages three new fields: (1.) speed in X coordinate, (2.) speed in Y coordinate and (3.) the sensibility of the antenna. The speed fields

are used to estimate and update the position of a neighbor. The sensibility of the antenna is used to decide whether a node is still reachable or not.

GBSR-B estimates the current neighbor's position and the received power of a packet at that neighbor's position. A neighbor is reachable if the estimation of the packet's power at reception is higher than the sensibility of its antenna plus 1db to guarantee the stability of the link. In addition, GBSR-B ignores vehicles which are behind buildings.

Forwarding procedure

To take a forwarding decision, GBSR-B uses the same criterium as the original GPSR but also includes two additional restrictions. The next forwarding node is a node in the table of neighbors that satisfies three conditions:

1. It is reachable. This is verified through the calculation of the current position of the node and the estimation of the reception power of a packet.
2. It is located in a non obstructed position. This is checked through a validation of the line of sight between the two nodes.
3. Among the nodes that fulfill the two previous conditions, the selected node is the nearest neighbor to destination according to Euclidean distance.

In case that there is not a neighbor closer to destination than the current forwarding node, then packets are stored in a buffer in order to be sent latter.

To avoid the problems by using perimeter forwarding, GBSR-B stores the packets in a buffer when there is no neighbor that satisfies all the requirements needed to be a *next forwarding node*. After a timeout packets still stored in the buffer would be dropped.

GBSR-B tries to initialize the new forwarding process of the packet stored in the local buffer of a node in a mixed way. This is, the node attempts to send the packet stored in that buffer every t seconds (proactive way) and each time a hello message is received (reactive way). The reception of a hello message is a signal that a change could have been produced in the neighbor list and therefore it is a good moment to try to find an optimal next forwarding hop. The value of t in proactive mode could be set as a function of the mobility of nodes, the stability of the neighbors or any other criteria. If a packet is not sent after a threshold time, it will be discarded.

2.5.2 Multi-Metric Map aware Routing Protocol (MMMR)

Multi-Metric Map aware Routing (MMMR) [113] Protocol is a traffic aware, delay-tolerant protocol that can be seen as an improvement of GBSR-B. MMMR determines if a neighbor in the list can be reachable and unobstructed. Also, MMMR considers three additional routing metrics in addition to the distance to select the best next forwarding node for each packet in each step towards its destination.

Multi-Metric to select the best next forwarding node

MMMR seeks to improve the next forwarding node decision based on four metrics, which are the distance to destination, the vehicle density, the vehicle trajectory and the available bandwidth. MMMR adds two fields in the hello message defined in GBSR-B. These fields are the percentage of idle time sensed by each candidate node, and the number of neighbors of each candidate node. With these two fields, the multi-metric score used by the routing protocol can be calculated. The $u_{n,i}$ value of each metric i is computed for each node n with eq. (2.6).

$$u_{i,n} = e^{-f_i(x_{n,i})} \qquad i = 1..4, \quad n \in \text{Nodes} \tag{2.6}$$

where $u_{n,i}$ is between 0 and 1. The function f_i evaluates the components employed by the corresponding metric and it returns values on $[0, +\infty)$. Function f_i will depend on the metric that is evaluated. Notice that low $f_i(x_{n,i})$ values provide higher values in metric the $u_{n,i}$ due to the use of the negative exponential. This way, MMMR penalizes drastically those neighbors with bad values in this metric. Conversely, the best forwarding node will get a score notably higher than the others.

Function metrics

In the following, there is a description of the four functions f_i used by the MMMR in the multi-metric score computation.

Distance: It is one of the most used metrics in VANETs. MMMR includes the Euclidean distance[1] between a forwarding node and the destination node. The function for distance in MMMR is obtained with Eq. (2.7).

$$f_{1,n} = \left(\frac{d_n}{d_{ref}}\right)^{\alpha} \tag{2.7}$$

where d_n is the distance of node n to destination, d_{ref} is a distance below which the probability that a packet reaches destination is very low and α is an attenuation factor that equals 0.77, obtained after a mathematical regression. With this function, the metric u_1 rewards higher more those candidates located closer to destination.

Trajectory: An important issue in VANETs is the accurate knowledge of the trajectory of vehicles. We consider the trajectory as a comparison of the future distance to destination with the current distance of a node. We obtain the trajectory function $f_{2,n}$ of node n using the future distance $d(t)$ to destination of that node, where $d(0) = d$. We compute the $f_{2,n}$ value using Equation (2.8) where $\Delta_d(t)^{\alpha} = d^{\alpha}(t) - d^{\alpha}$ is the variation of distance function ($f_{1,n}$) in a time t. The distance $d(t)$ is computed by estimating the future position of that vehicle using its speed, according to Eq. (2.9). The speed of the node, $\vec{v_n}$, helps us to give a higher score to nodes that sooner will be closer to destination (i.e., the Access Point AP). The idea is that with a higher speed, nodes may arrive sooner to destination given that the distance to destination decreases.

$$f_{2,n} = \left(\frac{\Delta_d(t)}{d_{ref}}\right)^{\alpha} \tag{2.8}$$

$$d(t) = \| \vec{x} + \vec{v}_n \cdot t - \vec{x}_D \| \tag{2.9}$$

Density: It is computed as the number of vehicles NV_n in the neighbors list of each node n at the moment of sending the current hello message, divided by the transmission range TR_n. Each node computes the density of nodes ρ_n and includes it in the next hello message. The algorithm gives a higher score when the node has a higher value of the node density

[1] In Appendix A, we analyze the use of other distance functions

ρ_n, by using Eq. (2.11) in the metric equation 2.6 for u_3. Nodes with a denser area in the transmission range will have more possibilities to forward the packet to a better next node.

$$\rho_n = \frac{NV_n}{TR_n} \tag{2.10}$$

$$f_{3,n} = \frac{1}{\rho_n} \tag{2.11}$$

Available bandwidth: To provide QoS, MMMR includes an estimator of the available bandwidth between two nodes in VANET based on a previous approach developed for IEEE 802.11 networks called available bandwidth estimator (ABE) developed in [96]. ABE is an estimation of the available bandwidth based on the idle times of the sender T_s and receiver T_r, the fraction of time needed for the backoff process K, the probability of m-sized packet loss $p_{(m,N,s)}$, which is based on the average speed of the vehicles s and the number of nodes in the area N, and the capacity of the channel C, as it is shown in Eq. (2.12).

$$\text{ABE}_i = (1-K)\cdot(1-p_{(m,N,s)})\cdot T_s \cdot T_r \cdot C \tag{2.12}$$

Following the same procedure proposed in [96], we provided in [112] an equation to obtain the probability of packet losses specifically designed for VANETs. This way, we obtained $p_{(m,N,s)}$ from the probability of hello messages $p_{hello}(m,N,s)$ and its relation with the length of the packet m, the density of vehicles N and their speed s, through a function $f(m,N,s)$.

$$p_{(m,N,s)} = f(m,N,s)\cdot p_{hello}(m,N,s) \tag{2.13}$$

$$f(m,N,s) = -7.475\cdot 10^{-5}\cdot m - 8.983\cdot 10^{-3}\cdot N - 1.428\cdot 10^{-3}\cdot s + 1,984 \tag{2.14}$$

The inverse of ABE as a metric is used by our forwarding decision algorithm to choice the best next node according to Eq. (2.15). In this way, nodes high bandwidth estimation will obtain high metric values u_4 through the negative exponential of Eq. (2.6).

$$f_{4,n} = \frac{1}{\text{ABE}_i} \tag{2.15}$$

Computing the multi-metric score

A node evaluates and assigns a total multi-metric qualification to each neighbor when it has to route a packet, applying a geometric average of the four metrics evaluated. A geometric score is used, because it is less sensitive than the arithmetic metric in the extreme values of the metric components. As a starting point, MMMR assigns equal weights (w_1, w_2, w_3, w_4) to each metric (u_1, u_2, u_3, u_4), in the qualification of each neighbor. The neighbor with the highest score is chosen as next forwarding node. If there is not any suitable next hop, MMMR stores the packet in a buffer (i.e., the node carries the packet). Packets have associated a timeout to limit their transmission time in the network.

$$\bar{u_n} = \prod_{i=1}^{4} u_{i,n}^{w_i} = u_{1,n}^{w_1}\cdot u_{2,n}^{w_2}\cdot u_{3,n}^{w_3}\cdot u_{4,n}^{w_4} \tag{2.16}$$

2.5.3 Performance Evaluation

GBSR-B and MMMR were evaluated under two different scenarios: sparse and medium density. We compared them to AODV [90], GPSR [62], I-GPSR [137]. The well known routing protocols AODV and GPSR we evaluated as references. We used I-GPSR for comparison purposes because it has some similarities with our proposal.

To do this we carried out several simulations using the NCTUns 6.0 [133] simulator. NCTUns makes possible to use an own mobility model, include walls to attenuate the signal, among other features. We used a typical Manhattan grid scenario, depicted in Fig. 2.4 to model a common urban scenario formed by streets and crossroads. We used this simple simulation scenario in [113] as a straiting point to carry out our performance evaluations. Nevertheless, in this doctoral thesis we have used a more realistic simulation scenario taken from OpenStreetMap [89], including the presence of real buildings and realistic propagation models. We describe such realistic scenario in the next chapter and use it along the next chapters, which present the main contributions of this thesis.

We used Citymob [75] to generate the movements of vehicles that follow streets and respect the presence of other vehicles and traffic lights. The simulation area was 1000 m x 1000 m. Each street was 100 m long with intersections of 40 m according to the area of the Eixample in Barcelona, Spain. We uses in the scenarios blocks (orange lines) to simulate buildings. These walls block the signal in the simulation process. We considered two densities of vehicles (60 and 120 vehicles) which were randomly positioned. The simulations were carried out using IEEE 802.11p standard on physical and MAC layers with only the Best Effort (BE) access category. All the figures are presented with confidence intervals (CI) of 95% obtained from five simulations per point. Table 2.1 summarizes the main simulation settings.

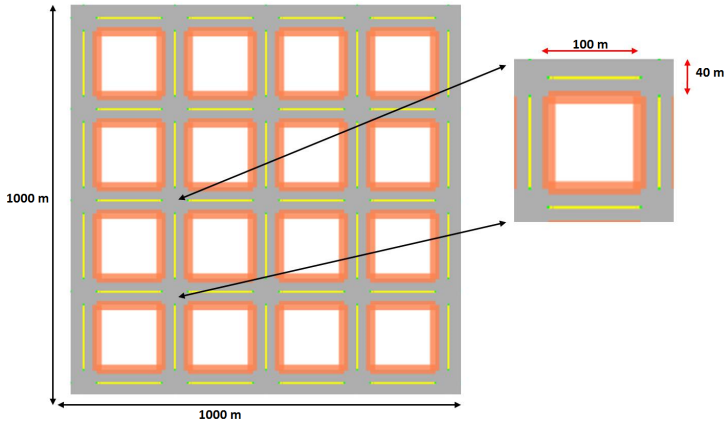


Figure 2.4: Manhattan simulation scenario.

Results show that MMMR (5th column in Fig. 2.5) improves significantly, in terms of packet losses the previous proposal GBSR-B (4th column) and the other protocols (see Fig. 2.5a) in a sparse scenario (i.e., 60 nodes scenario). For the delay (Fig. 2.5b) GBSR and MMMR performs worse than AODV, I-GPSR and GPSR, which is the price to pay for the reduction of packet losses shown in Fig.. 2.5a. Nonetheless our last proposal MMMR outperforms the

Parameter	Value
Simulation area	1000 m x 1000 m
Number of nodes	60 and 120 vehicles
Max. nodes speed	50 km/h
Transmission range	250 m
Sensing range	300 m
Mobility model	Manhattan
Mobility generator	Citymob
MAC specification	IEEE 802.11p (BE access category)
Bandwidth	12 Mbps
Simulation time	1000 sec.
Maximum packet size	1000 bytes
Traffic profile	CBR 4kbps
Routing protocol	AODV, GPSR, GBSR-B, I-GPSR, MMMR

Table 2.1: Simulation settings.

former GBSR in half a second (Fig. 2.5b). This fact confirms the benefits of considering more metrics than only using the distance as GBSR. On the contrary, in medium vehicle density scenario, the benefit of the multi-metric approach of MMMR in the performance metrics is not so high compared to GBSR-B. Recall that, GBSR-B is in fact equal to MMMR with the weight associated to the distance metric $w_1 = 1$ and the other weight with a value zero.

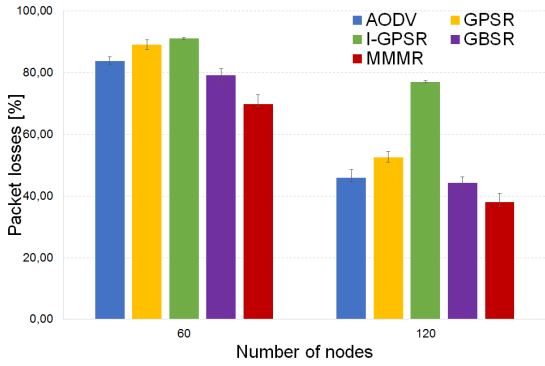
Regarding AODV, it is not in general an efficient routing protocol for VANETs because it establishes a full end-to-end paths. Nevertheless, AODV can successfully send a high number of packets in the medium dense scenario (See Fig. 2.5a). In that case, packets that used the recovery process due to the often path breakage, had a higher probability to achieve destination due to the presence of a high number of neighbors that offer more options to forward the packets. However, the recovery process introduces a high delay, as it can be seen in Fig. 2.5b for 120 nodes.

GPSR uses the perimeter mode (as recovery path process) that is not very efficient and produces a considerable number of packet losses, specially in low density scenarios (60 nodes in Fig. 2.5a). GPSR achieves the lowest delay (Fig. 2.5b) because most of the packets that arrive at destination come from greedy routing.

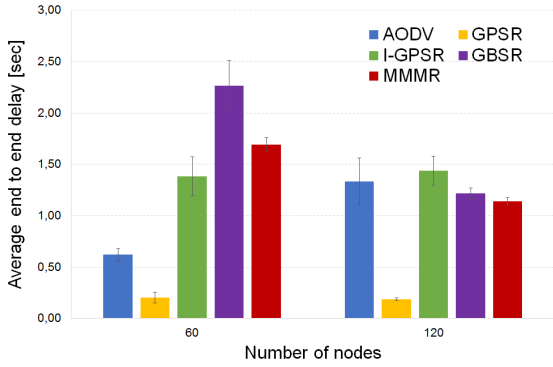
I-GPSR has high packet losses and average delay in both low and medium density scenarios. This is due to how I-GPSR selects the next hop. I-GPSR prefers those vehicles that approach faster to destination, which is good in VANETs. Nevertheless, it may happen that sometimes vehicles remain static (e.g., in a crossroads due to a traffic light), which affects its forwarding score. I-GPSR thinks that such a stopped node is not a good forwarder node.

Regarding the average number of hops, depicted in Fig. 2.5c, the results are very related to the scheme to choose the next forwarding node used by each routing protocol.

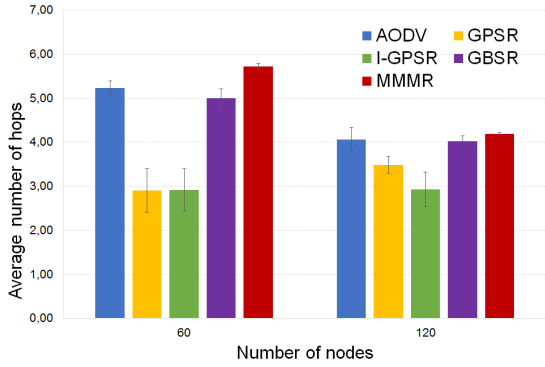
Let us focus our attention in the GBSR-B and MMMR routing protocols We can see that GBSR-B reduces losses (see Fig. 2.5a) although more hops are used. This is mainly due to the building aware scheme of GBSR-B that avoids choosing vehicles behind buildings. Notice that our MMMR presents the highest number of hops in both density scenarios (Fig. 2.5c). The reason is that MMMR not only uses distance as the metric to choose next



(a) Percentage of packet losses.



(b) Average packet delay.



(c) Average number of hops.

Figure 2.5: Comparison of the performance evaluation for GBSR-B [111] and MMMR [113] forwarding nodes, MMMR also uses trajectory, density and available bandwidth, so that it might happen that longer number of hops are used.

We can conclude that considering other metrics in addition to the distance is more important when the density of nodes in the area is low. Also, we can conclude that the most important metric in the forwarding decision is the distance to destination, although including other metrics improves the forwarding decision.

3. Methods and Materials

The results of this thesis have been obtained by using the Estinet network simulator [34]. A network simulator is typically used to study VANET services and applications in large scale scenarios due to the infeasibility of deploying real testbeds. This chapter describes the VANET simulation software, the simulation scenario and the statistical tools employed to analyze the results obtained from our proposals.

3.1 Introduction

Deploying VANET testbeds is quite expensive, and it is not a feasible solution in large-scale scenarios, which would require the deployment of hundreds of VANET vehicles. Due to this fact, network simulation is widely used for investigation in this field. Network simulators are useful and powerful tools to test a broad spectrum of proposals before their actual implementation. Hence, network simulation is becoming the first step in the process of developing new VANET protocols and services. Complex VANET scenarios with several nodes can be easily managed by simulators, and using realistic simulation scenarios is critical to obtain reliable results.

In this chapter, we describe the VANET scenario in which we are focused to study the contributions of the thesis. Then, we overview the simulation tools including the network simulator, a movement generator and a buildings importation tool. After that, the most relevant aspects of the configuration of our simulation scenario are explained. This section ends up with a description of the statistical analysis of the metrics computed from the results that we obtained from simulations.

3.2 Characterization of the VANET Scenario

Authors of [19] provide a classification of vehicular applications and their communication requirements. These categories are: safety, vehicular traffic efficiency and infotainment. Traffic flow control or environmental conditions monitoring are some aims of such applications. They might include delivering information to traffic authority centers about road conditions, traffic accidents, air pollution, noise level, infraction reporting, reckless drivers, etc.

All the efficiency-oriented applications require a continuous monitoring phase of the streets and city conditions. Vehicles are the most suitable collectors of this information so they can be in charge of feeding the monitoring centers. Our application scenario incorporates data traffic generated by an application during the phase of gathering data. The characteristics of the data traffic are:

a) Vehicles obtain data from their sensors, they process such data and generate constant-length packets.

b) The packets are sent to the closest RSU. This is a unicast and unidirectional traffic, since the information is valid only to the authority. With the purpose of facilitating the delivery process of data, RSUs can be configured with an anycast address, which does not semantically differ from unicast IPv6 addresses.

c) This kind of application does not have restrictive end-to-end constraints as safety related ones, so the transport of its traffic is suitable for end-to-end-tolerant protocols which deal with variable and high end-to-end, since the data is still useful.

It is worth mentioning that, when the traffic authority has to send data related to traffic conditions to vehicles or drivers, this information will be interesting for the whole group of vehicles located in a specific area. Consequently, this information could be transmitted using geocast communication to a multicast group associated to a specific geographical area.

3.3 Simulation tools

In this section we describe the Estinet network simulator [34], which is the main tool used in this thesis to test our proposals. Also, we summarize the mobility tools employed to provide realism to our simulation scenario.

3.3.1 Estinet Network Simulator

Estinet is the commercial version of NCTUns (National Chiao Tung University Network Simulator) [132] [133]. Estinet is a discrete-time network simulator that implements two-way communication with a single events queue. To deal with the simulations of vehicular communications, Estinet includes some vehicular traffic simulation capabilities, such as designing maps or importing a real layout and realistic vehicles mobility.

The Estinet simulator directly uses the real-life Linux TCP/IP protocol stack to generate high-fidelity simulation results, it provides a highly-integrated GUI environment. Estinet supports the IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4 standards to simulate V2V and V2I networking and communication. These standards are the basis of Wireless Access in Vehicular Environments (WAVE) to allow data transmission in the Intelligent Transportation System (ITS) applications.

On the other hand, the movement of each vehicle is controlled by a software agent that mimics a human driving behavior. To obtain this goal, the software agent considers the following parameters in its operation: initial speed, maximum speed, initial acceleration, maximum acceleration and maximum deceleration. To enhance the realism, the agent is able to perform car following, lane changing, overtaking, and compliance with traffic light signals.

3.3.2 Mobility generation tools

A key factor in the simulation of a VANET is the mobility of the vehicles in the scenario. Estinet can be used to evaluate VANET protocols without interacting or needing any

additional software because it uses the protocols stack in combination with its own vehicular traffic pattern. Nevertheless, since the vehicular traffic model is one of the most important factors that contribute to achieve realistic simulations, we decided to use a widely accepted tool to generate the mobility pattern of vehicles. Moreover, the choice of the traffic pattern in VANETs has a notable impact in the simulation results. We employ CityMob for Roadmaps (C4R) [38] that relies on Simulation of Urban MObility (SUMO) [65] to model a realistic vehicle behavior.

CityMob for Roadmaps (C4R)

CityMob for Roadmaps (C4R) [38] is a mobility pattern generator for vehicular networks that uses layouts of real cities from OpenStreetMap [89]. OpenStreetMap is an open project with updated map information of the world, which is increasing every day thanks to the fact that anyone is free to collaborate and edit or add map information.

Wtih C4R, researchers can generate several mobility traces in one step and all of them will share the same characteristics. Moreover, in C4R it is possible to define downtown areas with different attraction factors. C4R is also capable to consider different types of vehicles as it happens in real cities. C4R generates Network Simulator 2 (NS-2) compatible traces. We exported the NS-2 traces to Estinet format, using our own translating software, available at www.lfurquiza.com/research/estinet [119].

The route to be followed by the vehicles can be set by the user or randomly by C4R. To establish the routes C4R considers random origins and destinations for each vehicle. These points are located with higher probability in areas specified by the user (downtown/attraction points). The path for a specific start and end point is computed through the Dijkstra's algorithm in the directed graph formed by the streets and crossroads of the real map (as a GPS-based navigation system computes a route).

C4R uses the Simulation of Urban MObility (SUMO) engine [65] to model the vehicular behavior. SUMO is summarized in the following section.

Simulation of Urban MObility (SUMO)

SUMO (Simulation of Urban MObility) [65] is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. Its main features include collision free vehicle movement, different types of vehicles, multi-lane streets with lane changing, junction-based right-of-way rules, hierarchy of junction types. All these features makes SUMO able to manage different types of city topologies and together with large environments.

Once C4R sets up the vehicular scenario (e.g., type of vehicles, routes, simulation time), it uses SUMO to offer a realistic interaction among vehicles in the scenario. To do this, SUMO implements several mobility models.

Among them, in this thesis we use the modified Krauss mobility model [64] that includes multi-lane behavior as well as the basic Krauss model [66] that implements collision avoidance among vehicles. The mobility model uses four predefined parameters to adjust the speed of a vehicle to its predecessor. These parameters are maximum acceleration and deceleration of vehicles, reaction time of the driver and an imperfection factor in the driving behavior.

3.3.3 Buildings importation tool

In addition to a realistic vehicular movement, a realistic VANET simulation should also consider the presence of obstacles. Buildings are the main kind of obstacles that a VANET has to face in urban areas.

To incorporate building information in our scenario we extracted the buildings' location from the OpenStreetMap [89] with the polyconvert tool, which is incorporated in the SUMO package. Polyconvert provides an output file with the vertices' positions of a building. We use this file to create an empty Estinet project with obstacle objects in the corresponding building positions. The creation process of a VANET simulator project with building information is carried out with a command line program coded by us to this purpose, and also available at [119].

The presence of buildings in the simulation scenario allows any propagation model to use this building information in its operation. Futhermore, our proposed routing protocols will take building information into account in the forwarding decision algorithm to find out if neighbors are in the line of sight (LOS) or behind buildings.

3.4 Description of our Simulation Scenarios

The simulation scenario in which we analyzed the performance of the different contributions along this thesis consists of a multi-hop VANET in a urban environment. The main simulation settings are shown in table 3.1. The information of this table is divided in three blocks, which are settings about the city topology and vehicle movement generation, data traffic generated in the scenario and parameters about the protocol stack including the wireless channel.

Simulation area.

We used a real city area of 1.5 km^2, obtained from the Eixample district of Barcelona, which is depicted in Fig. 3.1). This urban area is formed by a high density of streets and crossroads with similar features like maximum vehicle speed, number of lanes well connected, which make it possible the generation of many realistic vehicle routes. This area of Barcelona has also main avenues (i.e., green, red and orange streets in Fig. 3.1) that allows generating sectors with high vehicle density. In the following we describe the simulation settings summarized in table 3.1.

Vehicle density.

Notice in Fig. 3.1 that the simulated area has a high density of buildings. This fact makes more challenging the communication among nodes due to the presence of obstacles. We considered three vehicle densities of 67, 100 and 167 vehicles/km^2. Each of these densities could represent different situations of a day, e.g. early morning, day/night and rush hour, respectively. The objective of using three different densities is to test if the difference among the results obtained from our contributions depends on the vehicle density in the scenario. In general, a high vehicle density helps to avoid discarding packets, since a suitable next forwarding hop would surely be always available; however, for very high densities, data transmissions would be more prone to interfere and collide.

Mobility traces and Confidence intervals.

Seeking to simulate a realistic scenario, the mobility traces were obtained with CityMob for Roadmaps (C4R) [38] with the SUMO engine [65]. The movements of the vehicles are

Parameter	Value
Map Zone	**Example District of Barcelona**
Area	1.5 km $\times$ 1 km
Number of junctions	712 (475 $\times$ km^2)
Number of streets (edges)	920 (613 $\times$ km^2)
Number of buildings (polygons)	2216 (1477 $\times$ km^2)
Number of nodes	100, 150 and 250 vehicles
Vehicles' density	67, 100 and 167 veh/km^2
Mobility generator	SUMO [65] / C4R [38]
Mobility model	Krauss [64] [66]
Max speed	60 km/h
GPS precision	10 m
Inter-packet generation time	t$\sim$U(2,6) s E(t) = 4 s
Maximum packet size	1024 bytes
Simulation time	300 s
Path loss model	Empirical model of IEEE 802.11p radio shadowing [104]
Building models	$\beta = 9db/wall \quad \gamma = 0.4db/m$ [104]
Fading model	Ricean (LOS) and Rayleigh (not in LOS)
Power transmission	23 dbm
Receiving sensing	-82 dbm ($\sim$ 400 m in LOS)
PHY & MAC specification	IEEE 802.11p
Bandwidth	6 Mbps
Routing protocol	MMMR [113]
Hello packets interval	1 s

Table 3.1: Simulation settings of our urban scenario.

randomly defined in each repetition by the movement generator C4R to take into account the intrinsic randomness that movements of vehicles introduce in the performance of VANET protocols and applications. We use at least ten repetitions per value to asses our contributions, plot the results with confidence intervals (CI) of 95% and perform their corresponding statistical analysis, which will be explained in the next section.

GPS.

In the simulations we have included a 10 m precision value for Global Positioning System (GPS) equipment of the vehicles. The aim of this parameter is to use a not very exact position of a node provided by the simulator in order to mimic the operation of a real GPS equipment. The reason is that we are interested in including realism in the simulated GPS device. We use a precision range of 10 meters, which is a typical value for these equipments depending on the supported technologies and vendors [77]. This means that we perturb the exact position of the node by modifying its coordinates with a radius uniformly distributed $r \in [0, 10]$ using Alg. 3.1.

The Alg. 3.1 uses the exact position x and y obtained from the simulator to provide the distorted $(\tilde{x}, \tilde{y})$. First it computed the distortion radius as a uniform random number between 0 and 10 (line 1). Next, the distortion distance on the x coordinate r_x is calculated as a uniform number between 0 and r (line 2). The distortion on y coordinate is computed on line 3 of the algorithm. After that, the signs of the distortions distance r_x and r_y are randomly and

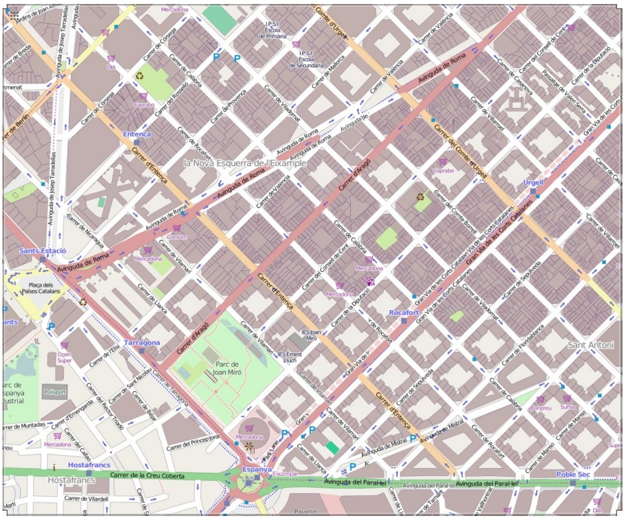


Figure 3.1: Map of the simulated area located at the Eixample district of Barcelona, Spain obtained from the OpenStreetMap OSM [89].

GPS-POSITION(x, y)

Require: exact coordinates $(x,\ y)$
Ensure: not exact GPS coordinates $(\tilde{x}, \tilde{y})$
1: $r \leftarrow$ Random (0, 10)
2: $r_x \leftarrow$ Random (0, r)
3: $r_y \leftarrow \sqrt{r^2 - r_x^2}$
4: $\text{sign}_x \leftarrow$ Random (0, 1)
5: **if** $\text{sign}_x \leq 0.5$ **then**
6: $\quad r_x \leftarrow (-1) \cdot r_x$
7: $\text{sign}_y \leftarrow$ Random (0, 1)
8: **if** $\text{sign}_y \leq 0.5$ **then**
9: $\quad r_y \leftarrow (-1) \cdot r_y$
10: $\tilde{x} \leftarrow x + r_x$
11: $\tilde{y} \leftarrow x + r_y$
12: **return** $(\tilde{x}, \tilde{y})$

Algorithm 3.1: GPS coordinates computation.

independently chosen from step 5 to 9 in the Alg. 3.1. Finally, the distortion distances r_x and r_y are added to the exact position x and y to obtain the GPS coordinates $(\tilde{x}, \tilde{y})$.

Access Point (AP).

The scenario has a common destination in a fixed node, which are henceforth called access point or AP. We focus our interest in reporting messages (e.g., traffic accidents) from vehicles to the single AP. The AP enables the connection, directly or by using multiple hops, to the city services in the network. We used a single AP in the scenario because in this way we obtained a long range of route lengths, which depends on the position of the source

vehicles into the scenario. Each vehicle during the simulations sends 1000-byte packets to the destination AP, during 300 s. The inter-packet time follows a uniform distribution between 2 and 6 s that has a average value of 4 s. We refer this information as "t∼U(2,6) s E(t) = 4 s" in table 3.1.

Protocol stack.

Regarding to the protocol stack of the vehicles, simulations were carried out using the IEEE 802.11p standard on physical and MAC layers. We used an empirical model of radio shadowing [104] in IEEE 802.11p networks as path loss model. This propagation model uses two factors to consider attenuation due to buildings presence: β applied to the number of walls that block the line of sight (LOS) between the nodes and γ that considers the obstructed distance between de vehicles. Also, as fading model we used Rician when vehicles are in LOS and Rayleigh when vehicles are not in LOS [95]. We set a receiving sensing of -82 dbm according to the receiver performance requirement for 6 Mbps specified in the IEEE 802.11 standard [55], which corresponds to a sensing range around 400 m in LOS.

As we anticipated in Sec. 2.5 of the previous chapter, in which we describe our routing proposals, we used our Multi-Metric Map aware (MMMR) routing protocol [113]. MMMR is based on Greedy Perimeter Stateless Routing (GPSR) [62]. MMMR is a end-to-end-tolerant protocol that improves the next forwarding node decision by employing four metrics: the distance to the destination, the vehicle density, the vehicle trajectory and the available bandwidth. A multi-metric score is obtained by each node for every neighbor in the transmission range, and it is used to choose the neighboring node that is the best candidate to be the next forwarding node.

Figure 3.2 shows our simulation scenario in the graphical interface of Estinet simulator. The figure includes the building information (orange lines) and the trace of vehicle movement that we imported from C4R. The high level of realism in our scenario can be appreciated by comparing it to the map area of Fig. 3.1. The buildings attenuate the signal of vehicles during the simulation process and there are more vehicles in the main streets of the area (red and green streets in Fig. 3.1)

3.5 Analysis of Simulation Results

This section introduces a description of the performance metrics and statistical tests that we employed to evaluate our contributions along this thesis.

3.5.1 Performance metrics

We use the simulation logs files to extract information about three widely used metrics to compare the performance of VANET routing protocols. These metrics are:

- *Packet Delivery Ratio* (*PDR*). It is the total number of packets that successfully reached the AP divided by the total number of packets sent from the vehicles. PDR does not increase due to the reception of copies of packets. Hence, this metric measures the effectiveness of the routing protocol in terms of different delivered packets.
- *Average end-to-end packet end-to-end.* It is the average time elapsed from the transmission of a packet until it arrives to destination (computed either for the original packet or for a copy).

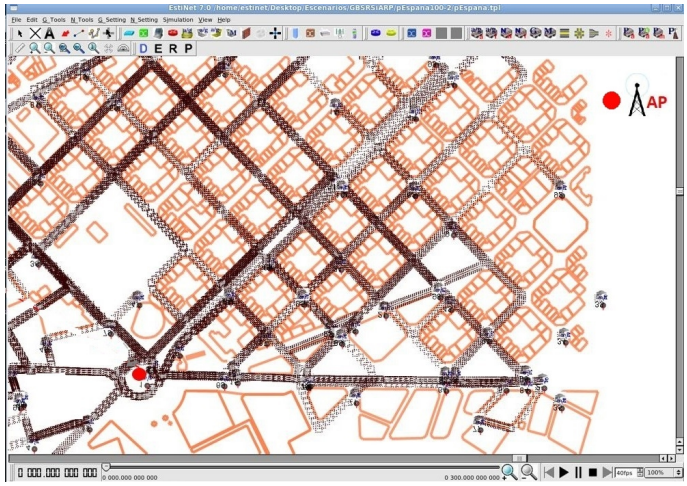


Figure 3.2: Barcelona simulated scenario with an access point (AP). The scenario includes buildings imported from the OpenStreetMap [89] and vehicles' mobility pattern generated with C4R [38].

- *Average number of hops.* It is the average number of hops that a packet needs to reach the AP. This average includes the hops performed either by the original packet or any possible packet copy.

3.5.2 Statistical Analysis

Since the data of these three metrics (i.e., PDR, average end-to-end and average number of hops) are obtained from the same simulations (although using different mobility patterns in each repetition), it is expect that they show some correlation among them. For instance, a high end-to-end could be related with a high number of hops. Hence, we use MANOVA [60] (Multivariate ANalysis Of VAriance) tests to consider the inherent correlation among the performance metrics in case they are not independent from each other.

MANOVA asses if the vector of averages of different groups are statistically equal among them or not. In our performance evaluations the groups (typically called levels) correspond to a set of simulations obtained from our contributions and other similar proposals. The MANOVA test is based on the decomposition of each observation of each group, as Eq. (3.1) shows.

$$x_{lj} - \bar{x} = (\tau_l) + (x_{lj} - \bar{x_l}) \quad \forall l = 1..g, \forall j = 1..n_l \tag{3.1}$$

where:

- x_{lj} is j^{th} observation of the l group. A *group* consist of the results obtained either from our contribution or from another approach of the state of the art. x_{lj} is a vector of metrics (i.e., PDR, average end-to-end and average number of hops).
- $\bar{x}$ is the average of the overall samples, which considers for its computation all the observations regardless the level they belong to.
- $\bar{x_l}$ is the average of the samples of the l group x_l.
- $x_{lj} - \bar{x}$ is the value of observation that is not represented by the overall sample mean.

- $\tau_l = \bar{x_l} - \bar{x}$ represents the difference between the sample mean of group and the overall sample mean. It represents the value of the observation that can be explained through the sample mean of the group and cannot by the overall sample mean. τ_l is named as *treatment effect*.
- $x_{lj} - \bar{x_l}$ is the portion of the observation value that can not be represented by the sample mean of the group.

If the sample means x_l are the same among groups then they should be equal to the overall sample mean $\bar{x}$ and consequently all τ_l would be equal to 0. To do this MANOVA uses the sum of squared Eq. (3.1) over groups l and samples of groups j which leads to the expression of Eq. (3.2).

$$\begin{aligned} \sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj}-\bar{x})(x_{lj}-\bar{x})' &= \sum_{l=1}^{g}(\tau_l)(\tau_l)' \quad + \quad \sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj}-\bar{x_l})(x_{lj}-\bar{x_l})' \\ &= \quad T \quad + \quad W \end{aligned} \tag{3.2}$$

T and W are known as the sum of squares and cross products of the treatment and residuals, respectively. If there is at least one $\tau_l \neq 0$ then the ratio $\frac{|W|}{|B+W|}$, known as Wilk's λ, is very small. Therefore, at lest one level (either our contribution or another approach) behaves significantly different from the others. For the MANOVA tests, we report the value of the statistics Wilk's λ and its corresponding F- value, which allow us to obtain a probability called the p-value. The p-value is the probability that the different levels of a variable produce the same result in a metric. The p-value is compared to a threshold named the *significance level* to determine if the simulation findings are statistically relevant (i.e., the p-value is lower than the *significance level*) or not. We use for our test a typical value of 0.05 for the *significance level*.

In the evaluation of our contributions, we are interested in knowing if the differences achieved by our proposals against other similar protocols are statistically significant or not. To carried out the statistical analysis, we use the three steps procedure depicted in Fig. 3.3.

Step 1 *Tests to determine interactions* among the factor formed by our proposals and similar approaches (*PR*) against the factor of vehicle density (*VD*) . If an interaction is detected (i.e., p-value is lower than 0.05) then the performance differences in the first factor will depend on the vehicle density of the scenario. In our analysis this means that if the $PR \times VD$ interaction is present then the comparison of the metrics should be performed for each vehicle density separately (right hand of Step 1 in Fig. 3.3). Otherwise, the rest of the tests do no need to differentiate the vehicle density in the analysis of the performance results.

If, in any particular study, we identify more than two factors that could be correlated, then further interaction tests will be performed to determine if the behavior of a factor may change according to the levels of another factor. This step allows us to do not miss any possible difference in the performance metrics thanks to a proper grouping of simulation data.

If the number of levels in the factor composed by our proposal and other similar approaches (PR) of the literature is higher than 2, then it is necessary to determinate if there is a difference among the set of levels of the analyzed variable (Step 2). Otherwise, the analyzed factor consist only of two levels: our proposal and other state of the art approach. Therefore, a pairwise test to compare the two levels for each metric can be performed directly (Step 3).

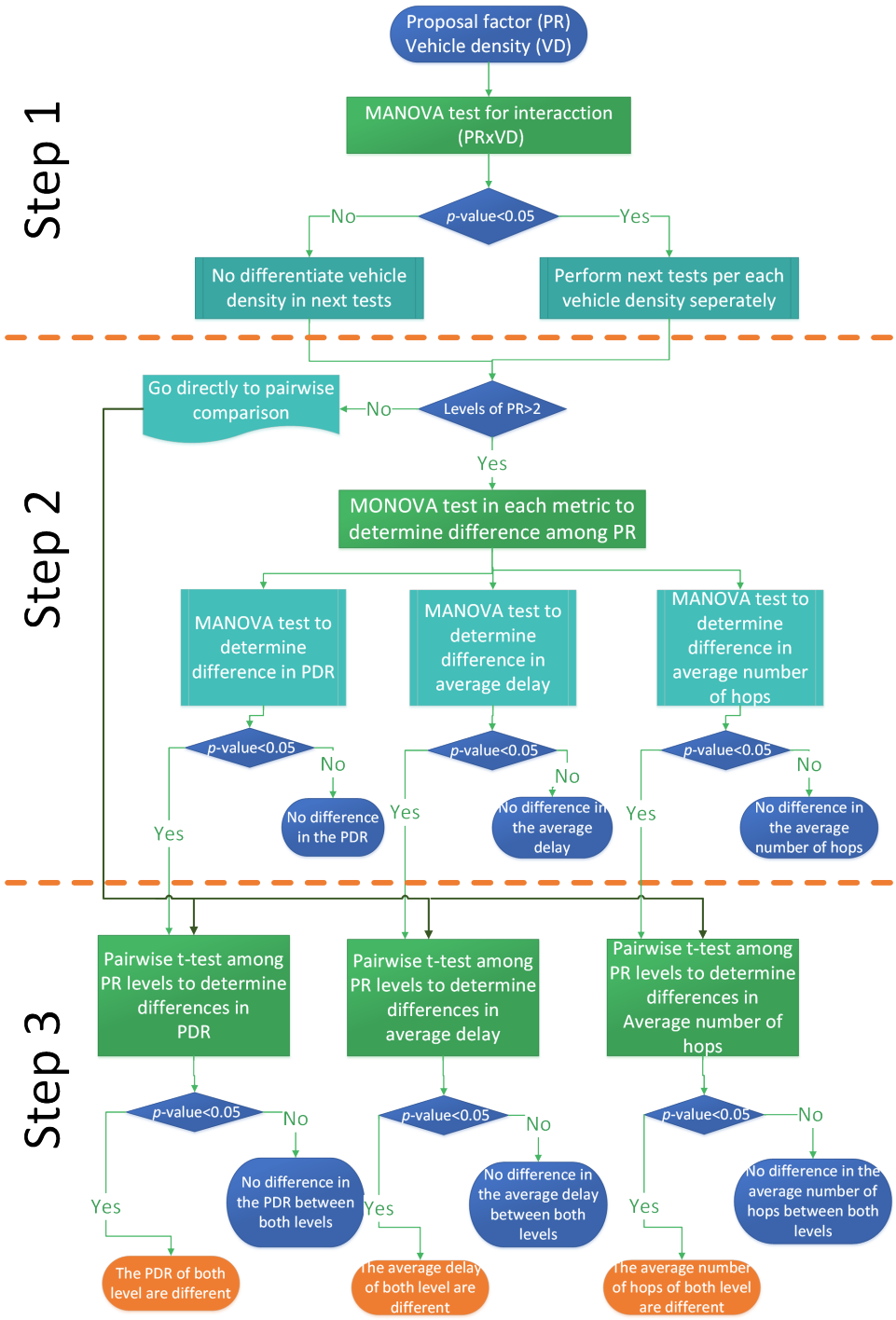


Figure 3.3: Statistical procedure to analyze our contributions.

Step 2 *Tests to determine if there is statistical difference in each metric* for each one of the groups in which the data set was divided because of the presence of interactions. If there is not a statistical difference in a metric, (i.e., the p-value is higher than 0.05) then it means that this metric behaves similarly under the different levels of the studied factor and no further analysis is required. Otherwise, the test tells us that there is difference but it does not indicate between which levels of the studied factor this difference is present. Thus, pairwise test among the levels are performed in Step 3.

Step 3 *Pairwise comparisons for a metric.* If the previous test determines a significant difference in a metric, then we run paired t-test pairwise comparison for the different combinations of levels of the factor under analysis. The objective of this step is to determine the performance relation between our proposal with other approaches of the state of the art.

By following the above test order, we are able to provide a detailed and accurate analysis of advantages and costs of our proposals. Since we used the same set of different vehicle movements to control better the variability in our studies we carried out the MANOVA version known as repeated measure MANOVA. All the previous tests in our simulation data were performed with the statistical software SPSS [52].

4. Propagation and Packet Error models

Simulation is a very valuable mechanism to carry out the performance evaluation of vehicular ad-hoc networks (VANETs), due to the high cost and effort that would be involved in the deployment of vehicles in urban scenarios. Accordingly, as in real VANETs, simulations should consider several factors related to realistic propagation, error and obstacles modeling, to real city topologies, vehicle movements, etc. In this chapter we concentrate on the channel and physical layer modeling in VANETs evaluate their impact through VANET simulations. On one hand, our numerical results indicate that different propagation models specifically designed for VANETs obtain very close results in channels with intermediate capacity. On the other hand, simulation results for our evaluation scenario of a traffic-efficiency application indicate that a basic packet error model can generate very similar results to the ones obtained from a realistic packet error model, when the configuration of the former is properly set at a low or medium channel capacity. Finally, we show that modeling the influence of buildings in urban areas as the total absence of communication between vehicles approximates pretty well to modeling such influence in a more realistic but complex fashion and could be considered a conservative bound in the performance metrics.

4.1 Introduction

Applications in road safety have encouraged study and research in wireless vehicular communications, both in the industry and in the research community. Deploying VANET testbeds is quite expensive, and it is not a feasible solution in large-scale scenarios, which may require the deployment of hundreds of vehicles. Due to this fact, network simulation is widely used for investigation in this field. Network simulators are useful and powerful tools to test a broad spectrum of proposals before their implementation. Hence, network simulation is becoming the first step in the process of developing new VANET protocols or services. Complex VANET scenarios with several nodes can be easily managed by simulators, and realistic simulation scenarios are critical to obtain reliable results. A realistic simulation environment requires a node mobility model that guarantees an appropriate distribution of vehicles and a channel model that mainly reproduces the effects of interference and attenuation in different scenarios. These channel models must capture the effects of

interference and attenuation depending on the scenario. Furthermore, a realistic simulation environment of an urban area should include the effects of shadowing caused by buildings, as part of the channel model.

In this chapter, we describe how a network simulator works in channel and physical layer levels. Also, we survey some of the most representative proposals to model propagation, packet error and obstacle attenuation in a VANET simulation. We make fair comparisons in each of these three simulation elements to evaluate their behavior. In brief, this chapter offers a thorough study of the different channel modeling techniques applied in simulation to reproduce the effects of propagation, packet error and attenuation caused by buildings in VANETs over an urban scenario. This analysis was very useful to find the impact of such obstacles in the overall performance of the simulated network.

The chapter is organized as follows: Sec. 4.2 describes how a network simulator works at channel and physical layer levels. Then, Sec. 4.3 surveys the models that composes the packet transmission process over a wireless channel and the generation of erroneous packets at physical layer for a VANET. Afterwards, we compared the different propagation model under a realistic packet error generator at physical layer in Sec.4.4. Next, we concentrate our attention to analyze the impact of different building attenuation techniques in Sec.4.5. After that, Sec. 4.6 is devoted to the evaluation of two packet error models. Finally, conclusions are drawn in Sec.4.7.

4.2 Operation of Channel and Physical Modules in a Network Simulator

A network simulator is a software written generally in C or C++. It follows the logic of the protocol stack. This is, a set of modules defines the behavior of a node during the simulation. All the nodes interact among each other according to a global clock that guarantees a proper order in the simulation events (e.g., transmission or reception of a packet). For a wireless node, the first two modules represent the wireless communication channel and the physical layer, which are the most relevant for the objective of our study.

Channel module

The propagation process in a VANET simulation can be summarized in three main tasks:

Path Loss. Usually called as propagation models. In this block are the models in charge of computing the signal attenuation caused by the distance. The average reception power is computed as a function of the transmission power, frequency, antenna gain, etc. Some models include dispersion effects through the addition of a random value from a given probabilistic distribution.

Obstacles. The network simulator includes an attenuation factor or modifies the path loss model due to the obstacles' presence. This attenuation factor could depend on the numbers of obstacles, the obstructed distance or a combination of both.

Fading. After the average reception power has been computed by the two previous blocks, the channel module introduces a variability effect in the computed power, which is called small-scale fading or just fading. Fading simulates effects like reflection and scattering.

When a packet p is sent from node i, the simulator creates packet copies of p in the $n-1$ nodes. Then the channel module of each node computes the reception power in the aforementioned order. If the power computed in the reception of a packet is lower than a minimum threshold, then the packet is discarded and it is not processed by the physical simulator's module. This threshold is typically obtained as a small fraction of the Nyquist noise associated with the channel.

The computation of the obstacles' attenuation is the most costly. Thus, this factor is computed if the reception power from the path loss model is higher than the minimum threshold of power. In addition, another optimization in the simulation process is to compute the reception power for all nodes in the sender node i. This pre-computation allows the network simulator to generate packet copies only for those nodes whose pre-computed reception power is higher than the minimum threshold. This provides a better memory management during the simulation execution.

Physical layer module

This module of the simulator has to control:

Channel State. The module sets the channel to busy state if the node is transmitting a packet or is receiving a signal which power is higher than the antenna sensitivity. The received power can come from an incoming packet or from the summation of interfering signals whose result is higher than the antenna sensitivity, as it can be seen in Fig. 4.1. A node will remain busy while last the transmission or reception of the packets that activate that state.

Interference management. Every packet whose reception power is higher than the minimum power threshold but lower than the antenna sensitivity is considered as interference for the reception of an incoming packet. Fig 4.1 shows how the total interference suffered by a node changes according to the arrival time, power and duration of the packet that contribute to it.

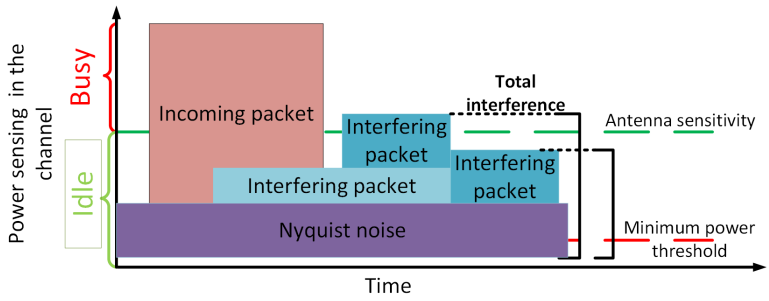


Figure 4.1: State of the wireless channel with interference management of a wireless channel for a node in a network simulator

Packet error computation. This task computes an error probability for every packet whose reception power is higher than the antenna sensitivity. This probability can depend on the Signal to Interference and Noise Ratio (SINR), distance, channel modulation etc.

Collisions. If a packet arrives with a reception power higher than the antenna sensitivity while a packet is being received by a node, then there is a collision between them. Typically, if the power of one of these packets is at least ten times higher than the

other packet then the stronger packet is processed and the other packet is added to the interference. If any interferent packet fulfills the previous condition, both packets are discarded due to the collision.

4.3 Description of Propagation and Packet Error models

This section provides a survey of the models employed in the propagation process of channel module and the packet error model of the physical module, explained in the previous section.

4.3.1 Path loss models

Free Space Model

In the Free Space Model [39] the received power $P_{r_{dBm}}(d)$ calculated through Eq. (4.1) depends on the transmission power $P_{t_{dBm}}$, the antenna gains G_t, G_r, the wavelength λ, the distance d between sender and receiver and α path loss exponent. In addition, the equation can include an attenuation factor L due to the system losses like the attenuation in the antenna and cable coupling. As a radio wave travels away from an (omnidirectional) antenna, the power decreases with the square of the distance (i.e., $\alpha = 2$).

$$P_{r_{dBm}}(d) = P_{t_{dBm}} + 10\log_{10}\left(\frac{\lambda^2 G_t G_r}{16\pi^2 L}\right) - 10\alpha\log_{10} d \tag{4.1}$$

Two Ray Ground Model

This model [95] estimate the received power $P_{r_{dBm}}(d)$ considering the contribution of two waves: a wave that travels in line of sight (LOS) between transmitter and receiver and another wave reflected in the ground. Simulators use Eq. (4.2) for this model.

$$P_{r_{dBm}}(d) = P_{t_{dBm}} + 10\log_{10}\left(\frac{h_t^2 h_r^2 G_t G_r}{L}\right) - 40\log_{10} d \tag{4.2}$$

where h_t and h_r are the height of the transmitter and receive antennas, respectively. This equation is only valid if:

1. The distance d is much longer than $h_t + h_r$. (i.e.,$d \gg h_t + h_r$)
2. The wave reflects in the ground perfectly.
3. The phase difference between the arrival of the LOS wave and the reflected wave is very small. This only happens when $d \gg \frac{h_t h_r}{\lambda}$.

To guarantee the last condition, network simulators only use Eq. (4.2) when $d \geq d_c$, where $d_c = \frac{4\pi h_t h_r}{\lambda}$. For shorter distances than d_c, network simulators use Eq. (4.1) for free space.

Log-normal Shadowing

It is an empirical approach [95] that estimate the received power $P_{r_{dBm}}(d)$ at a given distance d according to Eq. (4.3)). The shadowing model consists of two parts. The first one predicts the mean received power at distance d. It uses a closer distance d_0 as a reference. $P_r(d_0)$ can be computed from Eq. (4.1). From the reference distance d_0 the average power value in the reception point falls with a slope equal to 10α, where α is a path loss exponent and is usually empirically determined by mean of field measurement.

$$P_{r_{dBm}}(d) = P_{r_{dBm}}(d_0) - 10\alpha\log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma_{dB}} \tag{4.3}$$

The second part of the shadowing model perturbs the received power at certain distance using a log-normal random variable $X_{\sigma_{dB}}$ measured in dB with a standard deviation σ_{dB}. σ_{dB} is usually called the shadowing deviation, and it is also obtained by measurement, with values from 4 to 12 in outdoor environments.

Nowadays, more accurate propagation models are based on 3-dimensional *Ray tracing*, which needs a big amount of time due to the computational complexity involved in this process. This makes these techniques impractical to be used in network simulators. The more successful and widely used, specific path loss models designed for VANETs are those which modify or add parameters to the three previous models based on real measurements.

Empirical Inexpensive

In [104], the authors present an empirical and computationally inexpensive simulation model for IEEE 802.11p radio shadowing in urban environments. This model and its validation are based on real world measurements using IEEE 802.11p/DSRC devices in urban (buildings) and suburban (residential) areas. They modify the value of the attenuation factor α in Eq. (4.1) to ajust the equation to the measurements they obtained when two vehicles are in LOS. They found that the value that best fit their data was $\alpha = 2.2$.

Log-normal Shadowing Dual Slope

Authors of [140], adapt the attenuation slope of Log-normal Shadowing from Eq. (4.3) as a function of the distance between vehicles, which leads to employ two different slopes as it can be seen in Eq. (4.4).

$$P_{r_{dBm}} \begin{cases} P_{r_{dBm}}(d_0) - 10\alpha_1 \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma_1} & d_0 \leq d \leq d_c \\ P_{r_{dBm}}(d_0) - 10\alpha_1 \log_{10}\left(\frac{d_c}{d_0}\right) + X_{\sigma_1} - 10\alpha_2 \log_{10}\left(\frac{d}{d_c}\right) + X_{\sigma_1} & d > d_c \end{cases} \tag{4.4}$$

From distance d_c, the attenuation is more severe, so the two slopes ae needed to capture this effect. Also, the shadowing deviation σ increases with the distance. The values that authors suggest for the model and that will be used in this chapter are: d_0=10 m, d_c=100 m, α_1=2.1, α_2=3.8, σ_1=2.6 and σ_2=4.4.

4.3.2 Fading models

In this section we briefly summarize the main features of Rician and Rayleigh models.

Rician model

Rician [95], [71] is a fading model to provided randomness to deterministic power estimation models like *Free space* and *Two-ray ground*. It captures the effect of anomalies caused by partial cancelation of a radio signal by itself; the signal arrives at the receiver by two different paths, exhibiting multipath interference. Rician fading occurs when one of the paths, typically a LOS signal, is much stronger than the other (NLOS signal). The probability density function (PDF) of the received power under Rician fading is given by Eq.(4.5)

$$f(p) = \frac{(K+1)}{\Omega} \exp\left(-K - \frac{(K+1)p}{\Omega}\right) I_0\left(\sqrt{\frac{4K(K+1)p}{\Omega}}\right) \tag{4.5}$$

where Ω is the mean signal power, K is the ratio between the power in the LOS path and the power in the NLOS path and $I_0(.)$ is the 0th order modified Bessel function of the first kind. An efficient generator of values for Rician fading was developed for NS-2 in [92].

Rayleigh model

The Rayleigh fading model [95], [71] is used in situations when there is NLOS, and there are only multipath components. the PDF for the received power under Rayleigh fading can be obtained from the PDF of received power under Rician fading by setting path ratio $K = 0$ which leads to Eq. (4.6). This is because there is not a LOS path in the Rayleigh fading model.

$$f(p) = \frac{1}{\Omega} \exp\left(-\frac{p}{\Omega}\right) \tag{4.6}$$

Currently, network simulators like NS-3 include fading effect using the Nakagami model [71], from which it is possible to obtain Rician and Rayleigh distributions, as well as other distributions by changing its parameters' values. The Nakagami model is implemented using the Gamma probability distribution from the C++ standard libraries.

4.3.3 Obstacle models

This section summarize the relevant models to simulate obtacles in VANETs.

Discrete attenuation factor

In [98], the authors consider three different states for the mutual positions between each transmitter and receiver devices: line-of-sight (LOS), near line-of-sight (nLOS) and non-line of-sight (NLOS). These states are used to categorize the existing condition between two nodes in a fast and straightforward fashion, by discretizing x,y positions into x*,y*. Each one of these states, which depends on the line of sight from one node to another, is associated with an extra attenuation (EA). The possible discrete positions for a vehicle in a Manhattan grid scenario are shown in Figure 4.2. Equation (4.7) can be used to obtain the corresponding EA factor for two nodes in this scenario proposed in [98].

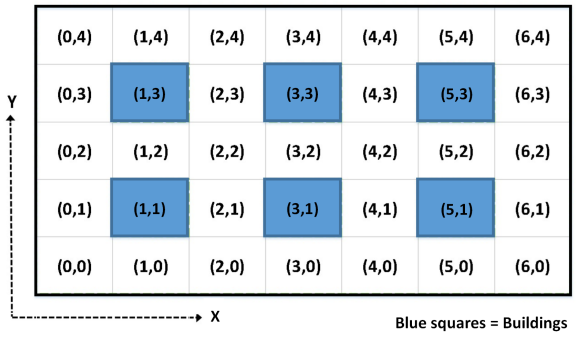


Figure 4.2: Discrete positions of a vehicle in a Manhattan scenario proposed in [98].

$$EA = \begin{cases} 0 & \text{if} \quad ((x_1^* = x_2^*) \wedge ((x_1^*, x_2^* \quad \text{even})) \vee \\ & \quad ((y_1^* = y_2^*) \wedge ((y_1^*, y_2^* \quad \text{even})) \\ -13dB & \text{if} \quad (\mid x_1^* - x_2^* \mid = 1) \wedge (\mid y_1^* - y_2^* \mid = 1) \\ -30dB & \quad \text{otherwise - NLoS} \end{cases} \tag{4.7}$$

Since the attenuation model relies on discrete positions, the attenuation factor for the communication link between two nodes can be calculated offline (not necessarily during the simulation process). This factor can be pre-computed and included in the network simulation process as an additional configuration file.

Attenuation as function of obstructed distance

In [84], the authors take into account the influence of obstacles, such as buildings, as a parameter for the computation of the reception power in simulations scenarios. They propose the usage of environment geometry as an input for a channel model. The influence of obstacles, such as buildings, are modeled by a 2D polygonal baseline that describes the obstacle's boundaries. In order to implement an efficient data retrieval strategy, the baseline boundaries should be stored in a recursive binary space partitioning (BSP) tree, which has a complexity of $O(n) = \log n$ to get the information of any obstacle. Next, during the simulation, the positions of sender A and receiver B form a line-of-sight (LOS) rectangle. This rectangle is used by the BSP algorithm to find buildings that might obstruct the LOS. In the next step, the intersection of all of the obstacle's faces with the LOS path is checked. This process is depicted in Fig. 4.3.

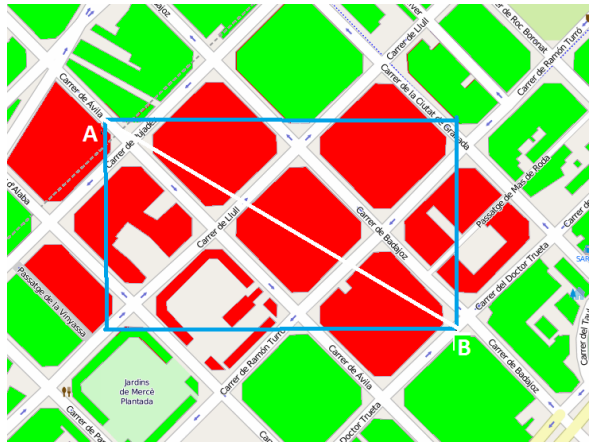


Figure 4.3: Detection of relevant building influence in the transmission process between nodes A and B. Red buildings are considered to have a relevant influence. We have used a zone map of Barcelona, Spain.

The total distance in LOS $d = d_f + d_o$, is equal to the distance traveled in free-space d_f plus the distance traveled through obstacles d_o are used in conjunction with a double-regression path loss model, called dual-slope model, where the distance d_f denotes the breakpoint from

the sender defined as Eqs (4.8) and (4.9).

$$L_0 = -20\log_{10}\frac{\lambda}{4\pi} \tag{4.8}$$

$$L_p = L_0 + 10\cdot\begin{cases} \alpha_f \log_{10} d & d \le d_f \\ \alpha_f \log_{10} d_f + \alpha_o \log_{10}\frac{d}{d_f} & d_f < d \end{cases} \tag{4.9}$$

L_0 denotes the reference path loss for the wavelength λ at a distance of one meter. The path loss exponents α_f and α_o are also wavelength dependent and have been set to α_f = 18 dB/decade and α_0 = 61 dB/decade in [25] after several simulations.

In order to improve VANET simulation results, the authors of [134] design and implement a more realistic radio propagation model, called the U.K. model (New University Kangaku) on NCTUns 6.0. This model was specifically proposed for VANET simulations in Tokyo, which represents a highly-populated urban environment. The New U.K. model considers both the line-of-sight (LOS) and non-line-of-sight (NLOS) conditions in its equations to compute the path loss. For the LOS condition, the distance between two vehicles, d, can be easily determined. In addition to the direct distance d, the computation of the LOS, shown in Equation (4.10), path loss involves several other parameters, such as the transmitter's height (h_t), the receiver's height (h_r), the widths of the streets in the scenario (W_s, W_1, W_2), the brake distance point (d_b), and the frequency f in GHz. For NLOS path loss computation using Equation (4.11), d_1 represents the distance between a vehicle (which can be either a transmitter or a receiver) and the intersection, and d_2 represents the distance between another vehicle and such an intersection. The computation of the NLOS path loss also involves many parameters, such as the transmitter's height, the receiver's height, the brake point, and the frequency. Different from the LOS path loss computation, for the NLOS, the sum of d_1 and d_2 ($d = d_1 + d_2$) is used as the distance between the transmitter and the receiver.

$$\begin{aligned} L_{LOS} \quad &= \left\{7.2 + 7.1\cdot\log\left(\tfrac{h_t\cdot h_r}{\lambda}\right)\right\}\cdot\log(d) + 28.3\cdot\log\left(1+\tfrac{d}{d_b}\right) \\ &\quad -1.2\cdot\log(f) - 19.6\cdot\log(W_s) + 65.9 \end{aligned} \tag{4.10}$$

$$\begin{aligned} L_{NLOS} \quad &= \left\{47.6 + 6.6\cdot\log\left(\tfrac{h_t\cdot h_r}{\lambda}\right)\right\}\cdot\log(d) + \left\{89.1 - 33\cdot\log\left(\tfrac{d_1}{\lambda}\right)\right\} \\ &\quad \cdot\log\left(1+\tfrac{d}{d_b}\right) + 19.9\cdot\log(f) - 11.3\cdot\log(W_1\cdot W_2 + 2.8 \end{aligned} \tag{4.11}$$

The authors in [76] focus on the development of an adaptive algorithm to determine the condition of LOS between two vehicles, depending on the their position in the scenario's streets. Some characteristics of the transmitted signal may determine if the nodes can directly communicate with each other or not. To this aim, three different cases are described:

- Vehicles on the same street: For two vehicles on the same street, there is an LOS between them, since no buildings interfere with the signal's path.
- Vehicles on different streets: If a couple of vehicles are located on different streets, it is necessary to check if there is an open area allowing communication between them (LOS). This involves identifying whether existing buildings completely interfere with the wireless signals. Success in communication, however, also depends on the distance between nodes and on the attenuation scheme used.

- Vehicles near junctions: Although there is no LOS between two vehicles, some electromagnetic phenomena of signals may help to obtain a successful communication. If the vehicles are on different streets, but near the corner where the streets meet, reflection, refraction or diffraction of signals over solid obstacles might sometimes produce such a positive effect. Some empirical results show that only vehicles close enough (< 20 m) to junctions are able to communicate with each other under NLOS conditions.

The following flowchart (Fig. 4.4) shows the conditions used to determine if a packet is successfully received using the proposed model in [76].

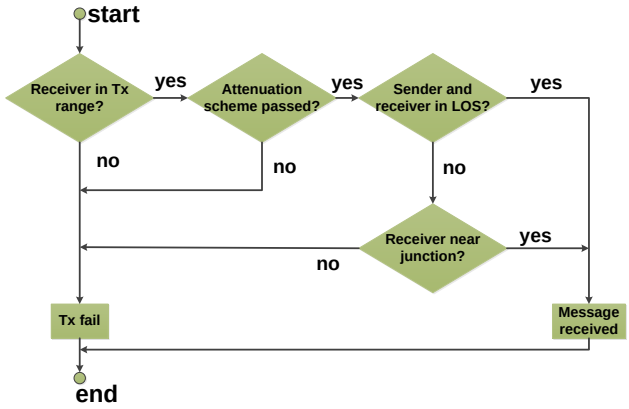


Figure 4.4: Flowchart of the visibility model proposed in [76].

As illustrated, the computation to determine if two vehicles are in the LOS is only done after two discarding steps. These steps are based both on the reception probability of packets and on the transmission range of nodes. This model tries to reduce the number of times that the LOS operations have to be done, since they are computationally expensive.

Attenuation as a function of the number of obstacles and the obstructed distance

In addition to provide the attenuation factor $\alpha = 2.2$ for Eq. (4.1) to estimate the received power in a vehicle communication for LOS situation, authors of an empirical and computationally inexpensive simulation model for IEEE 802.11p radio shadowing in urban environments [104] estimate the effects of building and other obstacle influence on radio communications between vehicles. The proposal considers building geometry and sender/receiver positions, and its model relies on building outlines, which are commonly available in modern geodatabases as OpenStreetMaps [89]. Furthermore, to keep the model computationally inexpensive, it only considers the line of sight between sender and receiver. By using the idea of [84] to detect the blocking effect of a building in the LOS between sender and receiver, the authors propose a generic model extension which is built on well-known propagation models, as shown in eq. (4.12), where $P_{r_{LOS}}(d)$ represents the received power at distance d employing Eq. (4.1) and L_{obs} captures the additional attenuation caused by an obstacle in the transmission process, based on the number of times n that the border of the obstacle is intersected by the LOS, and the total length d_m of this intersection. L_{obs} is

computed through Eq. 4.13.

$$P_{rdBm}(d) = P_{r_{LOS}}(d) - L_{obs_{dB}} \tag{4.12}$$

$$L_{obs}[dB] = \beta n + \gamma d_m \tag{4.13}$$

In Eq. (4.13), β represents the attenuation caused by the outer wall of a building and γ is an approximation of the internal structure of a building. These parameters are used to adjust the model to manage the influence of different kinds of buildings when setting urban scenarios.

4.3.4 Packet error models

The VANET research community has developed multiple models to mimic the effects of packet losses provoked by errors in packet decodification. These models have been implemented in network simulators in order to get more reliable results. In this section we briefly survey some of the strategies of packet error modeling, from the most classical, to the ones entirely devoted to VANETs.

Transmission and coverage range

In network simulators, nodes are configured with a coverage range parameter. In order to differentiate packets received with errors from packets received with no errors, a new effective coverage range is defined. All packets without errors are received in this range, which is called transmission range. This new radius is just a fixed bound, limited by a value of signal to noise ratio (SNR). If the SNR is higher than this bound, all packets are considered erroneous. This criterion does not consider that the interference level varies during the simulation and therefore the effective reception range could vary throughout time.

Packet error based on distance

The following two proposals, [76] and [9], are based only on the distance between two vehicles. In [76] the authors show results of experiments that consist of several measurements of the packet error rate (PER) along a varying distance between the sender and receiver (from zero to 500 m). The PER is estimated as the ratio of the number of unsuccessfully received packets with respect to the total number of sent packets. By using the collected data, several monotonically increasing functions were tested for the curve fitting process. Authors found that an acceptable trade-off between accuracy and execution time could be achieved using a fourth-order polynomial:

$$PER(X) = \begin{cases} 0 & \text{if} \quad x < 320m. \\ ax^4 + bx^3 + cx^2 + dx + e & \text{if} \quad 320m. \leq x \leq 400m. \\ 1 & \text{if} \quad x \geq 400m. \end{cases} \tag{4.14}$$

where PER is the Packet Error Rate and x is the Euclidean distance between vehicles. The values that were obtained through regression where: (a,b,c,d,e)=(5.29e-10,-3.37e-7,6.61e-5,-0.004,0.03).

Also, in [9] the authors propose Eq. (4.15) to compute the probability of successful message reception at a distance d given the intended communication range d_{CR} for given channel

parameters n (path loss exponent), s (standard deviation) and m (shape factor).

$$P_R(d,d_{CR}) \approx \frac{1}{2}\text{erfc}\left(\frac{10n\log(d/d_{CR})+\xi(\ln m-\Omega(m))}{\sqrt{2(s^2+\xi^2\zeta(2,m))}}\right) \tag{4.15}$$

where ω and ζ denote the digamma and the Hurwitz zeta functions [59] and $\xi = 10/\ln 10 \approx 4.343$.

Discrimination by Signal Interference Noise Relation (SINR)

This method was proposed in [23]. It calculates the SINR when a packet arrives, and determines if the value obtained is higher than a predefined score (which depends on the transmission rate). If the SINR of the packet is higher than this score, the packet is received correctly; if not, the packet is erroneous. This error is attributed to the physical layer if it does not exceed a threshold of 4dB, or to the memdium access control (MAC) layer, otherwise. This is a better approach than the previous one since the effective reception range of the packets varies depending on the channel conditions.

Theoretical estimation of Packet Error Probability (PEP)

A more sophisticated alternative is to calculate an error probability which is associated with each packet. This probability is calculated based on the SINR, the packet length and the codification type. The formulas and approximations used to calculate the error probability are described in [93], whereas implementation details can be found in [67]. Despite the great accuracy of this method, the authors of the formulas and approximations [93] suggest the use of estimates based on experimental models.

Packet Error Probability based on SINR

In [1] the authors adjusted the packet error probability (PEP) curve in terms of the SINR, the packet length l and the channel capacity R. Simulation results obtained from the Wireless Simulink [27] package were used to do this adjustment. The role of the PEP curve is obtained as following:

$$PEP_R(SINR,l) = \frac{1-tanh(a_R-b_R(SINR+10))}{2} \tag{4.16}$$

where a_R and n_R are coefficients (which depend on the capacity R) used to adjust the curve. The values obtained from the adjusted curve were validated with laboratory measurements in equipment that used IEEE 802.11p technology. The curves obtained with channel capacities of 3, 12 and 27 Mbps, with two different packet lengths, are depicted in Fig. 4.5. Notice that the SINR needed for a free error packet increases with the channel capacity. A channel capacity of 3Mbps requires only SINR$>$ 4 dB while the SINR must be higher than 24dB for a channel of 27 Mbps. The reason is that the 64-QAM coding used at 27 Mbps is less robust than for 3Mbps. Additionally, long packets require more SINR than short ones.

The integration in the functionality of the simulator is described in the following items:

- When a packet arrives, its SINR is calculated.
- If the $SINR < 4dB$ the packet is dropped, attributing the error to failures in the decoding process of PLCP header (Physical Layer Convergence Protocol).

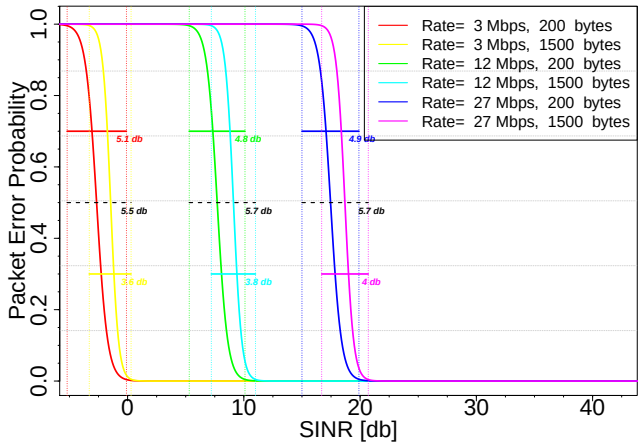


Figure 4.5: Packet Error Probability vs. SINR for channel capacities of 3, 12 and 27 Mbps, according to the PEP formula of [1].

- Otherwise, the physical layer obtains the packet error probability P_e through Eq. (4.16), by using the current SINR and a random number ϕ (generated from a uniform distribution between 0 and 1).
- If packet error probability $P_e < \phi$, the packet is correctly received; otherwise it is considered erroneous.

4.4 Comparison of propagation models

To compare the packet loss probabilities generated by different propagation models with a realistic packet error model we have used a simulation scenario that consists of a vehicle-to-vehicle transmission under the presence of different levels of interference represented by a single interfering node located at various distances, as it is shown in Fig. 4.6.

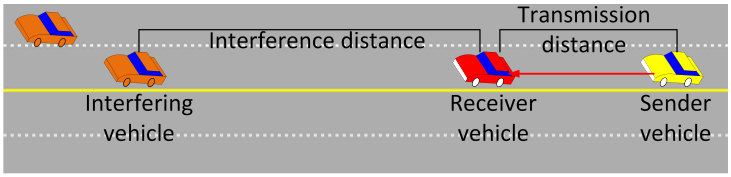
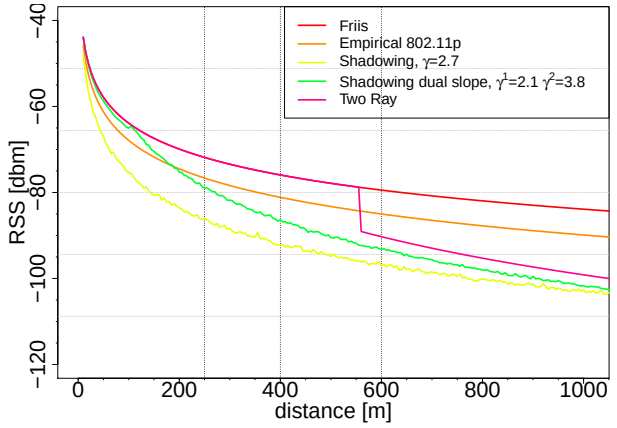


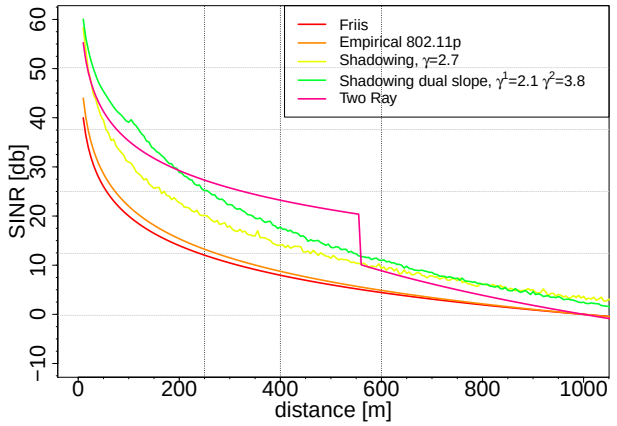
Figure 4.6: Received power as function of distance, according to four path loss models.

All tests were performed with the statistical software R [94], in which the propagation and fading models explained in Sec. 4.3 were implemented. We used the packet error model that computes an error probability based on the SINR [1], due to its high accuracy and simple implementation. Propagation models were configured with nodes having an approximate range of 400 m using a transmission power of 24 dBm. For each distance between vehicles, 1000 packets were sent to calculate the PEP associated with that distance. For each packet transmission, we recalculated the reception power for that distance. Also, we consider an interfering node located at a new random distance uniformly distributed between 10 and 2000 m and a new packet length between 200 and 1400 bytes. Three different channel capacities were used: 3, 12 and 27 Mbps.

The aim of our study is to determine how propagation models that predict different reception powers behave under a realistic packet error model that depends on the SINR. Note that if a propagation model attenuates the signal more rapidly than another propagation model, then its high attenuation rate will also modify the level of interference that affects communications. Hence, the SINR calculated with two different propagation models may be close to each other depending on the communication distance and the distance of the interfering node and thereby confer similar packet error probability (PEP). Fig. 4.7b shows how the SINR reverses the order seen in the signal attenuation of Fig. 4.7a in the presence of an interfering node located at 1000 m. So, aggressive signal path loss models such as log-normal Shadowing and Shadowing Dual Slope may get less or similar PEP than those more optimistic models as Friis and Empirical Inexpensive.



(a) Received power *vs.* distance, according to four path loss models.



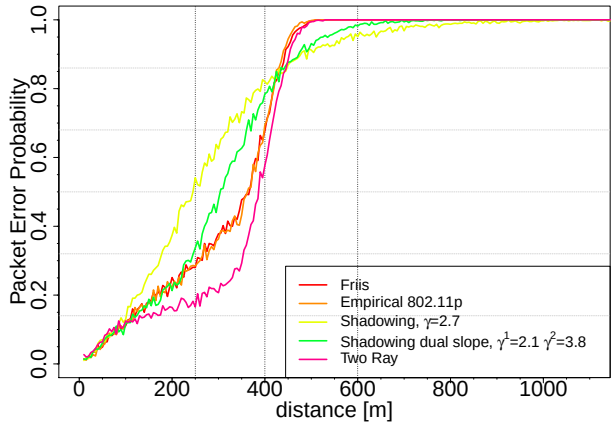
(b) SINR *vs.* distance, when there is a interfering node at 1000 m

Figure 4.7: Path loss models comparison

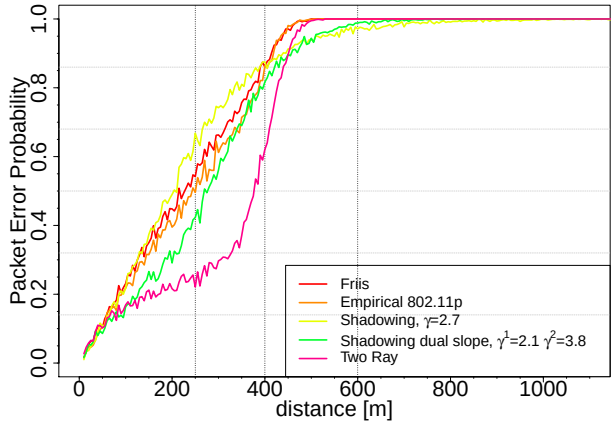
4.4.1 Packet error probability *vs.* distance

Fig. 4.8 shows the error probability of a packet based on the distance between two nodes for the channel capacities of 3, 12 and 27Mbps.

As it can be seen, Two-Ray Ground is the more optimistic propagation model (lowest PEP) for the three channel capacities. This model also has a steeper climb in all the channel capacities. The explanation lies in the behavior of this attenuation model which employs the Friis equation (Eq. (4.1)) up to the cutting distance of 560 m., from which the Two-Ray model uses Eq. (4.2), as it can be seen in Fig. (4.7a). This behavior causes that all transmissions that occur within the transmission range of the node, calculate their reception power with Eq. (4.1) and many of the interference signals are calculated with Eq. (4.2). Therefore, the resulting SINR are in average higher to that obtained with the other propagation models, which entails much lower PEP over short distances.

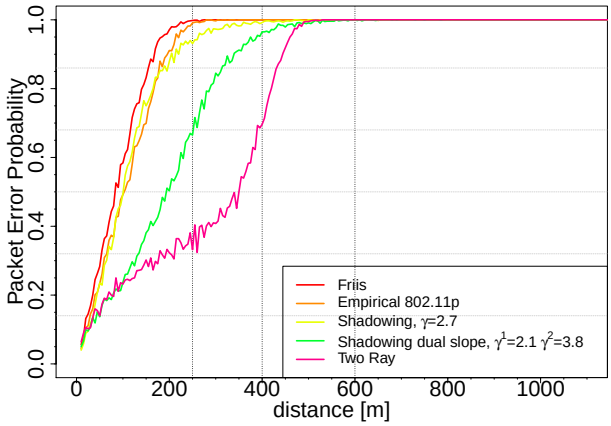


(a) Channel Capacity of 3 Mbps.



(b) Channel Capacity of 12 Mbps.

Fig. 4.8a shows the same behavior shown in Fig. 4.7a. The models that achieve a high PEP are those that attenuate more the signal. The log-normal Shadowing and Shadowing Dual Slope models produce weaker average received signal than the other models. More important to explain their high PEP for distances longer than 230 m is the variability introduced through its X_σ parameter in the received power in these models. This high variability allows that these



(c) Channel Capacity of 12 Mbps.

Figure 4.8: Packet Error Probability (PEP) *vs.* distance

two models produce SINR$\leq$4 dB in our simulation scenario, from which the probability of packet error starts to increase for the capacity channel of 3 Mbps (see Fig. 4.5). Regarding Empirical Inexpensive and Friis models, both of them obtain PEP values lower than with log-normal Shadowing and Shadowing Dual Slope. They have on average a lower SINR (see Fig. 4.7b) because they only introduce variability to their mean values through Rician fading, so they achieve low SINR values less often than the other models.

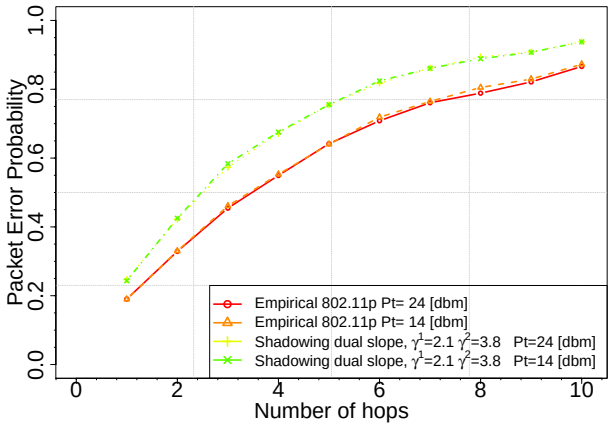
Fig. 4.8b allows the reader to appreciate that with the intermediate channel capacity of 12 Mbps, in which the PEP depends on SINR values between 7 and 13 dB, the path loss model behaves very similarly in all the range of distances. This similar behavior is because Shadowing and Shadowing Dual Slope can obtain the same SINR values of Friis and Empirical Inexpensive models thanks to the variability that the first two models can introduce to their power computation. It is worth to note that Inexpensive and Dual Slope Empirical models Shadowing (green and orange in Fig. 4.8b) present the most similar behavior. Both of them are empirical models proposed for VANETs through two independent works [104] and [140]. Dual Slope Shadowing has a lower PEP than Empirical Inexpensive because of its better SINR.

On the contrary, Fig. 4.8c shows that the packet error probability for a channel capacity of 27 Mbps has the same behavior exhibited by the SINR in Fig. 4.7b. This is because a packet needs a high SINR ($\geq$20 dB) to decrease its PEP. For this channel capacity the variability introduced by Dual Slope Shadowing is unable to compensate the low levels of SINR of Empirical Inexpensive model.

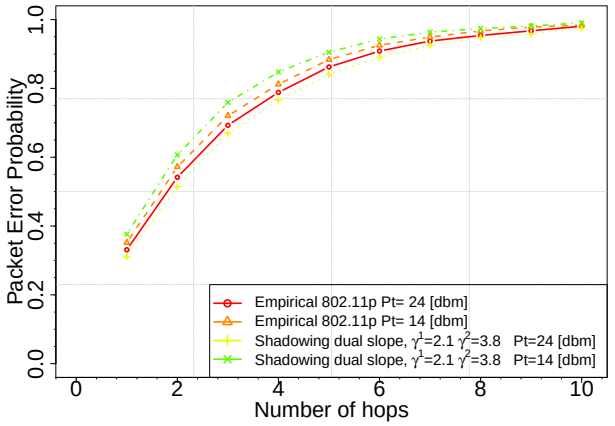
4.4.2 Packet error probability *vs.* path length

In this part of our study, we analyzed the packet error probability as a function of the number of hops in the path to reach destination. The results were obtained from 5000 combinations of random distances for each path length. The distance between two consecutive nodes in a path was chosen randomly between 10 to 400 meters in each one of the 5000 combinations.

At each hop, we included an interfering node as was explained in the previous section. In this group of simulations, we also compared the obtained PEP when nodes were configured with different values of transmission power but with the same coverage range. The results of this section are focused on Empirical Inexpensive and Dual Slope Shadowing models because they were designed for VANETs. Fig. 4.9 shows the results obtained for the channel capacities of 3, 12 and 27 Mbps.

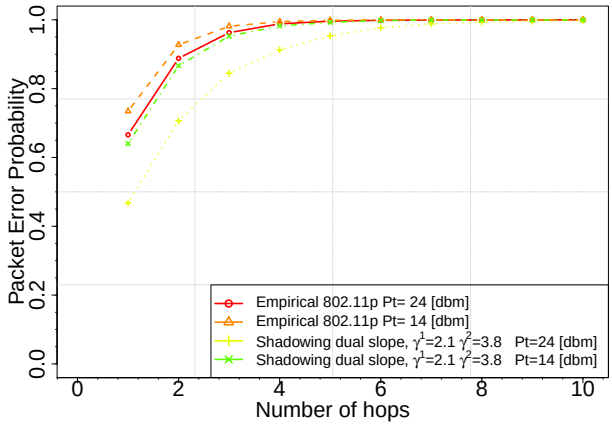


(a) Channel Capacity of 3 Mbps.



(b) Channel Capacity of 12 Mbps.

The dual Slope Shadowing model get a higher error probability that the Empirical Inexpensive model for the channel capacity of 3 Mbps, as it is depicted in Fig. 4.9a. The reason is the variability introduced by Dual Slope Shadowing and the SINR needed by a channel of 3 Mbps to ensure error-free packets as in the previous study. Our results indicate that the biggest difference between the two models is equal to 0.08 and its take place for path lengths between 2 and 5 hops. Fig. 4.9a also allows seeing that the PEP is independent of the transmission power configured in the nodes for this capacity.



(c) Channel Capacity of 27 Mbps.

Figure 4.9: Packet Error Probability (PEP) *vs.* path length

Fig. 4.9b shows the closest behavior between the two path loss models when the capacity of the channel is 12 Mbps. The variability introduced by Dual Slope Shadowing offsets the low level of SINR of Empirical Inexpensive. Hence, both model reach similar SINR levels and as a consequence very close PEP values for this channel capacity. The results show that the biggest difference between the two models (0.04) happens in paths lengths from 1 to 3 hops, and the transmission power is 24 dBm.

Fig. 4.9b also shows that low transmission powers entail higher PEP values, although the nodes have configured the same coverage range. The reason for this PEP-transmission power relationship lies on the interference computation to obtain the SINR. The total interference is the sum of the reception powers of all interfering packets plus the Nyquist noise. The received interference depends on the transmission power configured in the nodes. On the other hand, the Nyquist noise is always present (see Fig. 4.1) and only depends on the bandwidth of the transmission channel. So, if a node is configured to reach certain coverage range with low transmission power, the sensitivity of the antennas (see the green line in Fig. 4.1) will be closer to the level of Nyquist noise than if the node uses a higher transmission power. Therefore, the contribution of the constant Nyquist noise to the total interference is more important than with higher transmission powers in the case of low transmission power. This unavoidable contribution of the Nyquist noise leads to lower SINR, causing an increment in the PEP in the cases of nodes configured with low power transmissions.

Fig. 4.9c, where the channel has a capacity of 27 Mbps, depicts the highest PEP difference between propagation models as well as between the two transmission powers into each model. As we point out in the study of PEP as a function of distance, the variability of Dual Slope Shadowing does not compensate the low SINR of Empirical Inexpensive model for this channel capacity. Hence, the associated PEP behaves according to the average SINR of each path loss model (see Fig. 4.7b). The high PEP differences in each path loss model due to the power transmission configured in the nodes lies on the big importance of Nyquist noise in the SINR computation explained previously, and on the high SINR needed to keep a channel of 27 Mbps free of errors.

4.5 Evaluation of the Building Attenuation Model

The aim of this study is to provide a fair comparison among three different techniques to deal with the presence of obstacles in VANET simulation scenarios. Other research papers, such as [76], [43], show the importance of considering the effect of obstacles in the overall performance of VANET simulations. These works showed that using a channel propagation model that does not differentiate between LOS and NLOS situations leads to excessively optimistic performance evaluation.

4.5.1 Building models to be compared

In this paper, we evaluate three techniques of VANET channel modeling for simulation to consider the influence of obstacles in the communication between vehicles, known as the NLOS condition. The models that incorporate such an influence are: an empirically-computed attenuation factor, a complete blockage of the communication and a pre-computed attenuation factor.

Realistic VANET Channel Modeling

From the models surveyed in the last section, we chose the empirical and inexpensive radio shadowing model [104] as a realistic model that we use as a reference to compare the other two techniques, since this model is one of the most used for research in VANETs because of the following factors.

- It relies on real measurements taken from IEEE 802.11p devices.
- It is based on a simple modification of the free-space model, which captures the building attenuation with two easy-to-compute parameters.
- Its implementation in a simulator is straightforward, and it is alreadey preloaded in Vehicles in Network Simulation (VEINS), one of the most used VANET simulators.

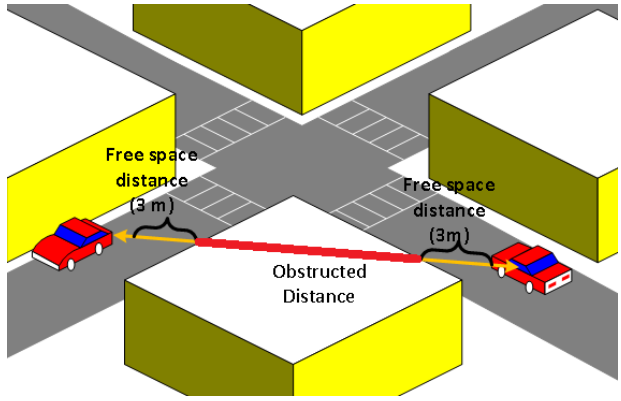
As we showed in [120], any realistic propagation model can be used as a reference to compare other proposals, since the comparison results are pretty invariable. The reason is that specific propagation models designed for VANETs provide very close results under intermediate channel capacities.

Total Blockage of Communications due to Obstacles

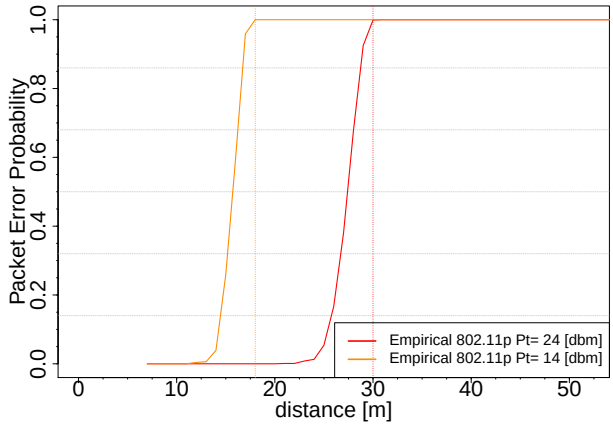
In this approach, considering the effects of obstacles in VANET simulations is done by assuming that communication between two nodes with an obstcle in the middle is completely blocked. Some papers, such as [113] and [73], use this approximation when evaluating their proposals.

To detect the presence of buildings in the LOS of a communication path, we used the same idea of [84], without having to compute all the intersection points. However, sometimes information about buildings is not available, and their influence is modeled in a conservative way, as it is done in [80]. Figure 4.10b shows how the packet error probability (PEP) varies when the obstructed distance between two nodes increases, as shown in the scenario of Figure 4.10a.

Figure 4.10b shows that the value of the PEP reaches one at around an obstructed distance of 30 m, even when a robust modulation scheme is used (as it happens in a low capacity channel). High power transmission and no interference are assumed. Thus, it would not be



(a) near-line-of-sight (nLoS) scenario at a corner.



(b) PEP *vs.* distance.

Figure 4.10: Packet error probability (PEP) behavior in an obstructed communication scenario between two vehicles near a corner. Empirical IEEE 802.11p channel model. Channel Capacity= 6 Mbps. Antenna sensitivity = -82 dbm.

surprising that most of communication fail. Assuming a total blockage of the signal due to obstacles, the overall performance would be close to that obtained using a more realistic channel model, as we will see in next Sec. 4.5.2.

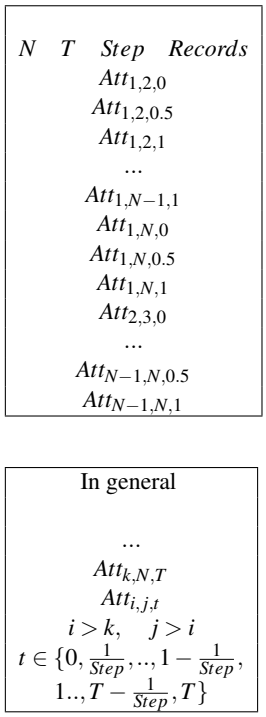
Pre-Computed Attenuation

The simulation of conventional channel models tends to be too slow, because of the great number of operations and searching algorithms implemented to determine an LOS or NLOS condition. As a result, some authors like [98] and [80] propose the use of an off-line file to store the values of the attenuation caused by the presence of buildings.

This approach requires the quantization of the movements on the streets in the simulation scenario. For this, a quantization pace is used to map vehicle positions from continuous time to positions in discrete time. In the same way, continuous positions on a street can be

replaced by discrete positions. The quantization error in this process depends on the size of the discretization step used. A small step would produce small errors, but also a large number of searching operations and, consequently, big files to store attenuation values. A more efficient strategy is performing discretization on the movements of vehicles when the simulation does not have to change the behavior of vehicles according to other events. Moreover, the map discretization is inevitable if the positions of the nodes change dynamically during the simulation. An issue with this map discretization is the need for vehicles to compute their discrete positions during the simulation. This might extend simulation times, since a searching algorithm is employed for this task.

For our work, we use a quantization process to determine the vehicles' positions in discrete moments. The pre-computed attenuation values should be stored in a fixed format to allow us an efficient retrieval of the nodes during simulation. We use an extended version of the output format file proposed in [80], which includes the length of the quantization step.



(a) Output format file

GETPOSITION(Src, Dst, t)

Require: Node IDs $1 \leq Src, Dst \leq N$ & $0 \leq t \leq T$
Ensure: Position *Pos* of pre-computed value for $Src \leftrightarrow Dst$ in t.

```
1: if Src > Dst then
2:     SWAP(Src, Dst)
3: (t_frac, t_int) ← SEPARATE(t)
4: i ← 1
5: while DISCRETE = false do
6:     if t_frac ≥ (Step−i)/Step then
7:         t_frac = (Step−i+1)/Step
8:         DISCRETE ← true
9:     i++
10: t ← t_frac + t_int
11: prePos ← 0
12: preNode ← 1
13: while preNode < Src do
14:     prePos ← prePos + (N − preNode) · Records
15:     preNode++
16: Pos ← prePos + (Dst − Src − 1) · Records + tStep
17: return Pos
```

(b) Position computation algorithm.

Figure 4.11: Pre-computed attenuation file format with its corresponding localization value algorithm used in this work.

The format for the output file that we have designed is depicted in Figure 4.11a, where the first field, N, is the number of nodes; T is the simulation time; $Step$ is the quantization step time; and *Records* is the number of discrete values for the two nodes in the simulation. The attenuation data is written in increasing order of source nodes, destination nodes and discrete

time. In the example, at the top of Fig. 4.11a, T is one and $Step$ is equal to two. For instance, the first entry of the array $Att_{1,2,0}$ stores the attenuation that a communication suffers between Nodes 1 and 2 at Time 0. The last entry $Att_{N-1,N,1}$ stores the attenuation factor between nodes $N-1$ and N at Time 1. The general procedure used to write the file can be found at the bottom of the same Fig. 4.11a. The output of this straightforward mechanism is an array of pre-computed values whose length can be quickly calculated with the four first elements, N, T, $Step$ and $Records$. In the file, $Att_{i,j,t}$ is the attenuation value due of the signal between node i and j at the time t. That attenuation value is expressed in dB. If this file is written using a binary format to make the file lightweight, it is possible for the whole array of values to be read, by the simulator, with a single operation. Furthermore, due to the fact that the structure of the file is known, there is no necessity to implement a searching algorithm to find a specific value; it is only needed to know the time and the IDs of the nodes in the communication.

The algorithm to compute the position in the array where the attenuation value (corresponding to the communication between nodes Src and Dst at time t) is located can be found in Figure 4.11b. This algorithm swaps the role of the source and destination nodes if their positions are not in increasing order. Then, the continuous time, t, is transformed to discrete time, based on the $Step$ employed in this process (Lines 6 to 13). After that, the initial position, $prePos$, is computed for the recorded values corresponding to node Src (Lines 16 to 19). Finally, (Line 20), the algorithm computes the offset value associated with the destination node Dst and the discrete time.

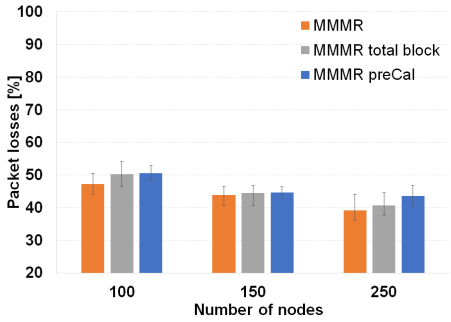
4.5.2 Comparison Results

We analyzed the performance of the three aforementioned attenuation models: realistic, total blockage of signal and pre-computed attenuation, in the multi-hop VANET simulation scenario extensively described in Chapter 3 , Sec. 3.4 using our Multi-Metric Map aware (MMMR) routing protocol [113]. The evaluation is focused on three extensively-used metrics applied to the performance analysis of VANET routing protocols. These metrics are the percentage of packet losses, the average packet delay and average number of hops. We perform the 3 steps procedure of statistical tests delineated in Chapter 3 , Sec. 3.5 to detect differences among the three attenuation techniques. Fig. 4.12 shows the results for three vehicle densities (100, 150 and 200 vehicles/km^2)

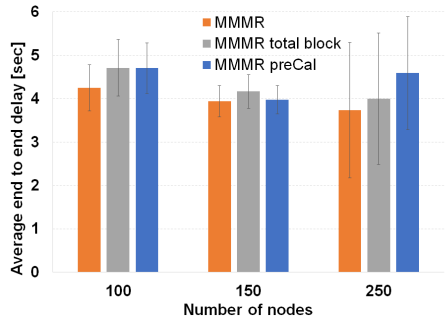
First, we check if the performance metrics differences among attenuation models depends on the vehicle density in the evaluated area (**Step 1** of the procedure of Sec. 3.5 from Chapter 3). There might be a relationship since a great number of nodes may generate more collisions and higher levels of interference. We can compare the results of building attenuation models without differentiating the vehicle density because there is not a significant building attenuation - vehicle density interaction since p-value$>$0.05. (p-value=0.193, Wilk's Λ = 0.508 and F(18,88)=1.34).

The test results to determine if there are differences in the performance metrics according to **Step 2** are shown in Table. 4.1. Recall that, we performed only one test per metric without differentiate the vehicle density employed during the simulation.

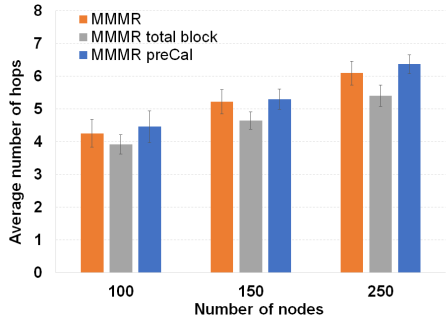
As the reader can notice, none of the p-values of Table 4.1 are higher than the significance threshold of 0.05. Hence, this means that there is a statistically significant difference among the performance metrics results obtained when the simulations use different building attenuation models.



(a) Percentage of packet losses



(b) Average end-to-end delay



(c) Average number of hops

Figure 4.12: Performance comparison of building attenuation model (CI 95%).

Performance Metric	**Wilk's Λ**	**F (2,35)**	***p*-Value 2 Sides**	**Is It Significant *p*-Value < 0.05?**
Packet losses	0.659	9.059	0.001	Yes
Average delay	0.651	9.386	0.001	Yes
Average No. of hops	0.253	51.628	0.0001	Yes

Table 4.1: MANOVA [60] results of testing difference in performance metrics among the effect of the building attenuation model. There is a significant difference when p-value < 0.05. This test does not need to differentiate the vehicle density used in the scenario.

We used a pairwise comparison to determine the models among which there exists a difference in terms of the performance metrics. Table 4.2 shows the p-values of the pairwise comparisons among building attenuation techniques (i.e., (Realistic, Total block), (Realistic, Pre-computed), (Total block, Pre-computed)) using the Wilcoxon statistical test [101]. These test corresponds to the **Step 3** of the analysis procedure of Sec. 3.5 from Chapter 3. Wilcoxon test [101] is highly robust for non-normal distributed data like the obtained from these simulations.

From Table 4.2, we can see that the results of the performance metrics are significantly different (see rows 1, 4 and 7) if we model the effects of building presence as the total absence of communication, compared with the results obtained with a realistic channel model, specifically designed for VANETs. See the lines "Realistic, total block". This behavior makes total sense, since this model involves nodes sensing fewer neighbors (vehicles cannot detect

Performance Metric	Pairwise (i , j)	p-Value 2 Sides	Is the Difference Significant (p-Value < 0.05)?	Median of Differences i - j
Packet losses	Realistic, total block	0.003	Yes	-2.3780%
	Realistic, pre-computed	0.0001	Yes	-3.0614%
	Total block, pre-computed	0.155	No	-0.91%
Average delay	Realistic, total block	0.0001	Yes	-0.5310 s
	Realistic, pre-computed	0.001	Yes	-0.4391 s
	Total block, pre-computed	0.572	No	0.0136 s
Average number of hops	Realistic, total block	0.0001	Yes	0.66 hops
	Realistic, pre-computed	0.119	No	-0.11 hops
	Total block, pre-computed	0.0001	Yes	-0.78 hops

Table 4.2: Pairwise comparison of the performance metrics. The column Median of differences refers to the difference between the first element (i) of the pair (i , j) minus the second element (j).

nodes behind obstacles in any case under the total-blockage attenuation model). Additionally, the absence of communications avoids the construction of paths that in a realistic approach may be feasible. Consequently, packets need to be stored in nodes for longer periods until finding a forwarding node, so the percentage of packet losses increases due to the timeouts. The communications between obstructed nodes is does not happen frequently, and most of them entail high error probabilities. As a consequence, the differences in the performance metrics are small in our simulation scenario, as it is shown in the column of the reported medians.

Regarding the comparison between the realistic channel model and the pre-computed attenuation approach,(see the lines "Realistic, pre-computed") it can be noticed (see row 8 in Table 4.2) that there is no statistically significant difference in the average number of hops. Nevertheless, there are discrepancies in the percentage of packet losses and in the average end-to-end delay (see rows 2 and 5 of Table 4.2). The median of the pre-computed attenuation performance results are not so far from the medians in the realistic scenario. The presence of differences between the aforementioned models is a consequence of the discretization process done in the pre-computed attenuation approach.

Lastly, total blockage and pre-computed building attenuation models (see lines Total block, precomputed) are compared in order to get an idea of the existing differences between these approaches and the realistic channel model. The reader can observe from Table 4.2 that there are only statistical differences in the average number of hops that a packet needs to reach the access point. Hence, the results obtained with these two models could be similar, at least in the percentage, to the packet losses and delay.

To conclude this section, we summarize the major features and results drawn from our performance evaluation to compare the 3 methods to model the presence of buildings.

- The results of statistical tests carried out with the performance metrics percentage of packet losses, end-to-end delay and average number of hops show differences when employing different attenuation models.
- A complete attenuation of the signal due to the presence of buildings, and pre-computed attenuation models in our simulations can be considered as pessimistic bounds for all the performance metrics.

- We did not find that the differences in the performance metrics are affected by the vehicle density employed during the simulation.

4.6 Evaluation of Packet Error Models

In this section we present an evaluation and comparison of the packet error models.

4.6.1 Packet Error models to compare

To evaluate the impact of using a realistic packet error model on the performance of VANET simulations, we use a basic error model dependent only on the antenna sensitivity and on the model of PEP curves based on SINR [1] that we described in the previous Sec. 4.3.4.

Realistic Packet Error Model

We chose the PEP model based on SINR as the Realistic Packet Error Model (Realistic-PEM) due to the following reasons:

- The model considers the SINR effect, the packet length and the type of codification used for the different bandwidth settings. These factors are the most relevant when estimating the error probability of a packet.
- Other experimental models do not consider the aforementioned aspects and are based on different criteria like the distance, the equipment characteristics and the configuration used in the models.
- Contrary to analytical models which require some effort to be incorporated in a network simulator, the PEP curves model based on SINR [1] implementation is extremely straightforward.

We would like to point out that all the packet error models for VANETs that we have surveyed [76] [9] [93] and [1] show a very close behavior in the curve of the packet error probability.

Basic Packet Error Model

The Basic Packet Error Model (Basic-PEM) that we use in this work, only considers the antenna sensitivity. This model is described by the following items.

- If the incoming packet has a power higher than the antenna sensitivity, the packet is successfully received and no error checking process is performed. If the power of the incoming packet is lower than the antenna sensitivity, the packet is discarded and, depending on the power level, interference is added.
- If another packet arrives with higher power than the antenna sensitivity, a collision occurs. If the power ratio of the two packets is lower than a fixed “collision threshold", both packets are discarded; otherwise, the packet with the highest power is received. The existence of collisions, then, is the only reason for which a packet with enough power can be erroneous.

It is worth to mention that in this basic packet error model, interference signals are not involved in the decision of whether a packet is erroneous or not. Interference signals are just used by the physical layer to sense the state of the wireless channel before the transmission of a packet.

Choosing the Antenna Sensitivity

Since the Basic-PEM explained previously relies on the antenna sensitivity, this parameter should be carefully selected in order to minimize the effect of not using a realistic model. In this work, we use the minimum sensitivity stated in receiver performance requirements for the Orthogonal Frequency Division Multiplexing (OFDM) physical layer in the IEEE 802.11 standard [55]. This sensitivity is obtained, according to the standard, as the minimum power level of incoming packets necessary to maintain the Packet Error Rate under the 10%. Table 4.3 presents the minimum sensitivity for IEEE 802.11p which uses channels with 10MHz of bandwidth.

Modulation	**Coding rate (R)**	**Minimum sensitivity (dBm) (10 MHz channel spacing)**
BPSK	1/2	-85
BPSK	3/4	-84
QPSK	1/2	-82
QPSK	3/4	-80
16-QAM	1/2	-77
16-QAM	3/4	-73
64-QAM	2/3	-69
64-QAM	3/4	-68

Table 4.3: Receiver Performance Requirements [55].

Theses sensitivity values lead to a very low packet error probability because their associated SNRs are very high.

Realistic vs Basic Packet Error Model

In this section we compare the two models, in terms of sensing channel, error operation and collision behavior.

Sensing the channel. For both models, the physical layer will sense the medium idle if the node is receiving or, transmitting packets or if the interference level is higher than the sensitivity of the antenna.

Error operation. This is where the two models differ more from each other. While Basic-PEM does not discard any packet with enough power due to errors in the decoding process, the Realistic-PEM follows the procedure described in [1] and summarized in Sec 4.3.4. Notice that the differences in this procedure do not affect the way in which the physical layer senses the channel.

Collision. While the Basic-PEM checks a fixed collision threshold to determine if one or both packets contain errors, the Realistic-PEM considers a packet (the one with the lowest power level) as interference from the other to calculate its error probability.

As it can be noted, the difference between the two models has essentially to do with the usage of interference signals in the realistic model, to determine the existence of errors. Considering the antenna sensitivity that we have explained in the previous section, packets that arrive according to the realistic packet error model will be correctly received with a very high probability even under the presence of interference signals, if they are not high powered signals.

4.6.2 Comparison Results

The impact of error modeling for VANET simulations is evaluated in this section, and for this we used the MMMR protocol [113] in the simulation scenario described in Sec. 3.4. We evaluate the packet error models under three nodes' densities: 100, 150 and 250 vehicles. A high density of vehicles helps to avoid discarding packets since a suitable next forwarding hop would always be available; however, data transmissions will be more prone to be interfered. As stated in the previous section, we set receiving sensitivity values of -82, -77 and -68 dbm according to the receiver performance requirements of 6, 12 and 27 Mbps respectively, specified in the IEEE 802.11p standard [55]. We use these settings to adapt the Basic Error Model as a more realistic one.

Our evaluation is focused on three metrics widely used in the performance analysis of VANET routing protocols: packet losses, average packet delay and average number of hops. We perform the 3 steps procedure of statistical tests delineated in Sec. 3.5 to detect differences among the Basic and Realistic PEM for each data rate. Fig. 4.13 shows the results for the three vehicle densities.

First, we check for the three data rates (i.e., 6, 12 and 27 Mbps) if the performance metrics differences between the two packet error models depend on the vehicle density in the evaluated area (**Step 1** of the procedure of Sec. 3.5). This relationship is interesting due to the fact that the higher the number of nodes, the higher the interference generated. The results of the interaction tests in Table. 4.4 indicate that the differences between Basic-PEM and Realistic-PEM depend on the vehicle density for the medium and high channel capacities, 12 and 27 Mbps, respectively (the p-values$<$0.05 in Table 4.4). So, we only need to study the performance metrics results for each vehicle separately for the channel capacities of 12 and 27 Mbps. We do no have to distinguish among vehicle density for the lowest channel capacity analyzed (6 Mbps).

Channel Capacity Mbps	**Wilk's Λ**	**F (6,50)**	***p*-Value 2 Sides**	**Is the interaction significant ? *p*-Value $<$ 0.05?**
6	0.645	2.042	0.077	No
12	0.570	2.703	0.024	Yes
27	0.175	11.602	0.0001	Yes

Table 4.4: MANOVA [60] results for interaction test between Packet Error Models (PEM) and vehicle density for the three tested data rates.

Since we are only comparing two PEM, it is not necessary to perform Step 2 of the statistical procedure defined in Chapter 3, which is intended to test if there is any difference among three or more groups. Thus, we directly proceed with the pairwise comparison between Basic and Realistic PEM to find out if both models behave statistically similar or not. Table 4.5 shows the results of the pairwise comparison.

After carefully analyzing the p-values of Table 4.5 obtained from testing differences between both packet error models, the reader can realize the following facts:

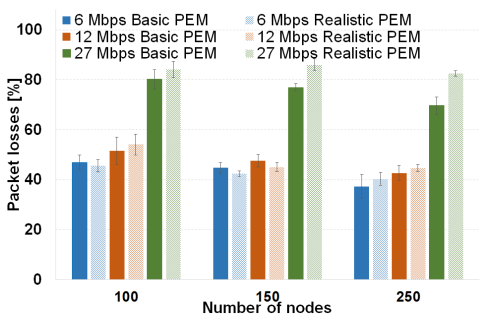
- For channel capacity values of 6 Mbps, there is no statistically significant difference between Basic and Realistic packet error models in any performance metric for the

Performance Metric	Channel Capacity	Vehicle Density	p-Value 2 Sides	Is the Difference Significant (p-Value < 0.05)?	Mean Differences (Basic - Realistic)
Percentage of Packet losses	6	All	0.810	No	0.218%
	12	100	0.187	No	-2.536%
		150	0.011	Yes	2.6%
		250	0.05	No	-1.936%
	27	100	0.02	Yes	-3.97 %
		150	0.0001	Yes	-8.99%
		250	0.0001	Yes	-12.2%
Average end-to-end delay	6	All	0.072	No	0.1 s
	12	100	0.456	No	-.287
		150	0.036	Yes	0.236 s
		250	0.328	No	-0.199
	27	100	0.469	No	-0.341 s
		150	0.0001	Yes	-3.79 s
		250	0.0001	Yes	-5.628 s
Average number of hops	6	All	0.109	No	0.120 hops
	12	100	0.692	No	-0.69 hops
		150	0.673	No	-0.118 hops
		250	0.016	Yes	-0.494 hops
	27	100	0.0001	Yes	2.41 hops
		150	0.005	Yes	1.95 hops
		250	0.094	Yes	0.5 hops

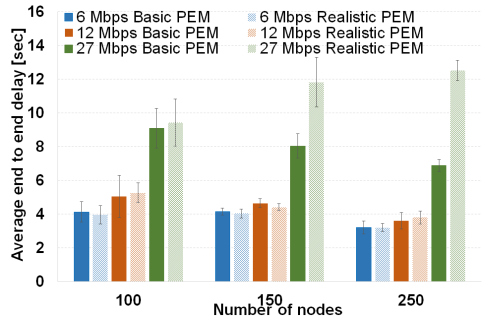
Table 4.5: Pairwise comparison of the performance metrics between Basic and Realistic packet error models (PEM).

analyzed channel capacity values. All of them have a p-value$>$0.05. Therefore, both models lead to very similar results in this channel capacity. This can be seen in Fig. 4.13.

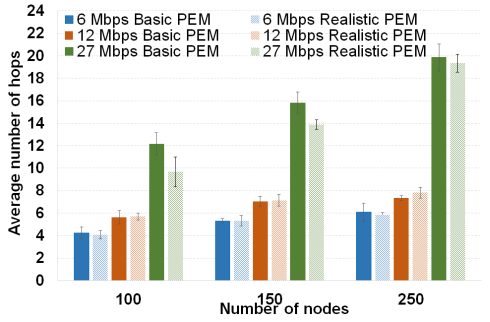
- The intermediate channel capacity of 12Mbps presents a mixed behavior. On the one hand, for low vehicle density (100 nodes), all the p-values in Table 4.5 are higher than 0.05. Hence, there is no significant difference between basic and realistic packet error models in this vehicle density. On the other hand, for the intermediate density (150 nodes), the basic error model is a conservative bound compared to the realistic one in the percentage of packet losses and average end-to-end delay in Table 4.5, whose p-values$<$0.05 indicate that these differences are significant. That is, the Basic PEM estimates more packet losses and delay than the Realistic PEM (the mean differences between Basic and Realistic PEM in these metrics are positive). Regarding the high vehicle density of 250 nodes, both models behave very similar and the only significant difference is the average number of hops, where the basic error model needs in average 0.5 hops fewer than the realistic model to reach destination.
- For the channel of 27 Mbps, there are statistically significant differences between the two error models in the three performance metrics and for the three vehicle densities that we evaluated (all p-values$<$0.05 except for average delay in low density). There is, in fact, a clear relationship between the number of nodes and the error model used in the simulation.



(a) Percentage of packet losses



(b) Average end-to-end delay



(c) Average number of hops

Figure 4.13: Performance comparison of Basic and Realistic Packet Error model for three channel capacities: 6, 12 and 27 Mbps. (CI 95%).

Figures 4.13a and 4.13b show that, as the number of nodes increases, the packet losses and the average delay decrease, when the basic error model is used for the channel capacity of 27 Mbps. Conversely, these metrics get increased when the realistic error model is employed. This is a consequence of the strength of interfering signals present between nodes that increases with the node density. Thus, in the Realistic Packet Error model more packets are cataloged as erroneous and, consequently, for this kind of simulation (realistic), nodes need to implement a longer backoff process to successfully transmit a packet. Regarding the average number of hops shown in Fig. 4.13a, the packets reaching their destinations in the Realistic Scenario tend to travel shorter paths than packets in the Basic Error Model, given that packets traveling through longer paths have fewer opportunities to successfully reach destination.

To conclude, the presence of packet errors in a realistic simulation environment may be mitigated by reducing the error probability. This can be achieved by setting the antenna sensitivity with a value that guarantees a high SNR. However, even when the antenna sensitivity is well set, the impact of a realistic error model over the simulation cannot be ignored, specially as the capacity channel increases.

4.7 Conclusions

A systematic study has been performed in this chapter about the simulation of multi-hop vehicular *ad hoc* networks at channel and physical layer levels. Particularly, on how the performance metrics of routing protocols could vary depending on the models utilized in the

different elements of a VANET communication. To do that, we have surveyed the models of all the elements currently implemented in a network simulator to mimic a vehicular wireless transmission. Our work starts by comparing the packet error probability through numerical simulations of different propagation models working with a realistic packet error model. Our results indicate that there is not a most conservative propagation model (i.e., a most pessimistic propagation model among the analyzed models). The selection of such conservative model depends on the channel capacity of the network. Moreover, the two models designed for vehicular communications, Empirical Inexpensive [104] and Dual Slope Shadowing [140], show very similar results.

Additionally, our tests verify that nodes configured with the same coverage range but different transmission power will create different results in a simulation. The reason is a more important role of the Nyquist noise in the interference process for a node with low transmission power. Therefore, an adequate description of simulation settings should include the transmission power used or sensitivity of the antennas and an expected coverage range.

Our study continues by analyzing three strategies to model the influence of buildings on the communication between vehicles. These strategies model this influence as the full attenuation of the communication signals, as a number of offline computed values of attenuation and as a straightforward and accurate realistic building attenuation scheme [104]. The results we obtained support that the performance metric scores depend on the building attenuation model used in the simulations. Hence, the research community should use a realistic propagation model when possible. Nevertheless, the performance differences between the realistic attenuation model and the other two models, are at the maximum 3% for the percentage of packet losses and 0.5 s for end-to-end delay in our simulation scenario.

Furthermore, we could not find any statistical relationship between the vehicle density in the scenario (which may include a higher data traffic load) and the building attenuation model used. Obstructed communications with higher capacity channels are less probable, and consequently, the resulting gap between a realistic building attenuation model and a total blockage of the signal should decrease when using higher capacity channels.

To conclude our study of the wireless channel modeling in a network simulator, we compared for three channel capacities (i.e.,6, 12 and 27 Mbps) a Realistic Packet Error model [1] with a Basic Error Model which is only based on the sensitivity of the antenna. The sensitivity of the antenna was configured according to the IEEE 802.11p [55] standard recommendation to reduce the packet error probability. Results indicate that there is not a statistically significant difference between the aforementioned packet error models for the low channel capacity and there are minimum differences for the medium-rate channel. However, there is a significant difference between the analyzed models for a high capacity channel (i.e. a channel of 27 Mbps) due to the fact that a high sensitivity cannot neglect the effects of interference signals even in low data traffic scenarios as the ones we simulated. In fact, the differences found in our tests increase according to the vehicle density for the high channel capacity.

Our results suggest that the antenna configurations with high sensitivity values in conjunction with basic Packet error models allow, in the best cases, would obtain reliable results from simulations, but those results are not guaranteed. This simple technique is based on the assumption that the interference levels will not exceed the SINR threshold when errors begin to appear. This condition may not be satisfied in all simulations, for instance simulations with very high traffic load.

5. Coherent, Automatic Address Resolution

In this chapter, the use of the Address Resolution (AR) procedures is studied for vehicular ad hoc networks (VANETs). We analyze the poor performance of AR transactions in such networks and we present a new proposal called Coherent, Automatic Address Resolution (CAAR). Our approach inhibits the use of AR transactions and instead increases the usefulness of routing signaling to automatically match IP and MAC addresses. Through extensive simulations, we compare our proposal CAAR to classical AR and to another of our proposals that enhances AR for mobile wireless networks, called AR+. In addition, we present a performance evaluation of the behavior of CAAR, AR and AR+ with traffic from a reporting service for VANETs. Results show that CAAR outperforms the other two solutions in terms of packet losses and furthermore, it does not introduce additional overhead.

5.1 Introduction

Due to the very dynamic topology of VANETs, their routing protocols need continuous broadcasting of signaling messages which are used to keep track of any changes in the network infrastructure. Next, an address resolution mechanism is traditionally required to set up a traffic flow when a new IP communication process starts. We claim that this address resolution task could be performed by the network-topology updating process accomplished by VANET routing protocols. In this way, we leverage the functionality of the routing signaling and avoid the address resolution traffic periodically interchanged among nodes.

We show the undesired effects of address resolution traffic in vehicular communications and highlight the advantages and drawbacks of our proposal in a multi-hop scenario for a unicast traffic service. We extend the analysis done for MANETs in [20] to VANET environments by using MMMR [113] traffic-aware routing protocol.

The main contributions of this chapter are: (a) an in-depth study of the address resolution behavior in VANETs, by showing its negative impact on the overall performance of such networks; (b) a proposal to suppress the common address resolution process in a VANET by adding layer-two addressing information to the signaling routing messages. This approach prevents collisions and facilitates the tasks of the protocol stack layers, such as the addition and subtraction of headers.

The rest of the chapter is organized as follows. Section 5.2 summarizes the Address Resolution procedure and particular its work flow in a wireless communication. Next, Section 5.3 summarizes related work to address resolution in mobile ad hoc networks. Afterwards, Section 5.4 describes our scenario, and shows how address resolution mechanisms can be enhanced to minimize the traffic requiered to set up a data communication in a multi-hop scenario. The section ends with an explanation of how the updating topology process embedded in routing mechanisms can carry out such address resolution. Then, Section 5.5 is entirely devoted to the evaluation of the different address resolution schemes according to the strategies we proposed. Finally, conclusions are drawn in Section 5.6.

5.2 Background

Our contribution is inspired by the Address Resolution (AR) work logic carried out by the Address Resolution Protocol (ARP) [91] in IPv4 and Neighbor Discovery (ND) [85] in IPv6 networks. Our proposal of an address resolution scheme is supported by the operation of the VANET routing protocols.

In general, the literature states that IPv6 is the most appropriate technology to support VANET communications [81], given the exhaustion of IPv4 addresses and the new features offered by IPv6. The IPv6's Neighbor Discovery Protocol (ND) includes an address resolution mechanism which is substantially similar to that of ARP [30]. Indeed, ND is more complex due to the fact that it combines address resolution and ICMP router discovery and redirect mechanisms in its operation, as it is described in the IPv6's RFC 4861 [85]. Since the IPv6's address resolution mechanism uses multicast communication, the traffic generated when requesting this service is theoretically much lower compared to that of ARP broadcast traffic. However, its wireless shared medium prevents VANETs from taking advantage of such multicast process since all neighboring nodes still receive the requests, even when those are not processed by all the nodes. Hence, a significant amount of traffic is generated anyway, due to the address resolution process. For the purposes of our analysis, both ARP and ND address resolution mechanisms have the same issues in terms of the amount of traffic they generate in a VANET scenario. Hence, we will use the parameters of both ARP and ND protocols when referring to the address resolution processes, provided that some simulation tools are still using the IPv4 stack.

In this section, we present a detailed study of the AR operation. Then, in Sec. 5.4.2 we propose a modified configuration of this address resolution mechanism to enhance its performance for mobile and dynamic topologies.

The link-layer address resolution process performed by a neighboring node was standardized in ARP [91] for IPv4 networks in 1982, and updated in Neighbor Discovery [85] for IPv6 networks in 2007. The operation of the AR mechanism is really simple. When a node A needs to send information (packets) to a node B, the AR daemon looks up in a translation table whether the B's hardware address exists or not. If A finds B's hardware address, then the AR daemon puts it in the MAC header of the packets sent to B. If A does not know B's hardware address, node A broadcasts (in ARP) or multicast (in IPv6) a message to request such information. When node B receives the request, it updates its address translation table and responds to node A with a reply message containing the hardware address of B.

Below we describe some AR operation features, which are identical in ARP an ND. We

highlight some issues in order to suggest improvements when applied to a VANET scenario.

- When a node uses the address information from the sender's ARP message to *update* an entry in the address translation table, it is called a *Free AR update*. This operation is done with every ARP packet whatever the destination of the message is. This operation can also be performed with ND with an Unsolicited Neigbor Advertisement process. These, *free updates* are possible in VANETs, since vehicles process messages in promiscuous mode due to the shared nature of the wireless channel.
- When a node *adds* addressing information obtained from a received ARP request to its address translation table, the process is known as *ARP learning*. ARP learning is performed in ND with Neighbor Solicitation Messages, which do the same job as ARP Requests.
- According to RFC 1122 [16] a node should keep a queue per destination to store at least one packet (to locate the last incoming packet) until its hardware address is resolved. Most of the current ARP implementations only use this really short buffer. ND [85], also keeps the use of a short queue. Nevertheless, this buffer size may not be large enough for mobile ad hoc networks where collisions occur frequently and, therefore, the buffer could drop packets because a request/reply was lost.
- The recommended maximum ARP request rate is 1 message per second per each destination node, in order to prevent ARP flooding in IPv4 [16]. In ND [85], *RetransTimer*, set the wait time before sending another Neighbor Solicitation. In addition, the maximum number of Neigbor Solicitation messages to a node can be done through `MAX_UNICAST_SOLICITS`.
- Another common implementation issue is that the address resolution process can only be completed with the reception of an AR reply message. In other words, node *A* will only try to send those packets stored in its buffer and addressed to node *B* when an ARP reply message (from *B* and addressed to *A*) is processed. Nevertheless, a hardware address might have been resolved by another AR message, such as an AR message (reply or request) where node *B* had participated, that node *A* could have heard from its neighborhood

The aforementioned AR features may provoke some issues on the VANET operation. In Section 5.4.2 we analyze some optimization procedures and we propose a modified configuration of AR, which we claim could be more suitable for VANETs.

5.3 Related Work

Issues with the Address Resolution (AR) operation in ad hoc networks have been reported in other works. Authors of a study about TCP performance over Mobile Ad Hoc Networks (MANETs) [50] found that a large amount of packet losses were due to the combination of outdated routes, unresolved MAC addresses for those next forwarding nodes and a very short ARP queue. In a measurement study of vehicular Internet access [18] using IEEE 802.11b networks, authors highlighted that the 1.5 out of 5 seconds necessary to initialize any data transmission, were due to ARP procedures. The AR process has been studied previously in a context of MANETs. In the performance comparison of routing protocols for multi-hop wireless ad hoc networks done in [17], authors noted a serious layer-integration problem between any on-demand MANET routing protocol and ARP. The problem arises when a

route for the stored packets is found. If there is a next-hop destination whose link-layer address is unknown, all the queued packets, except the last one sent from the routing protocol queue to the ARP queue, will be dropped. This is due to the short length of the ARP buffer (with room only for 1 packet). Broch and colleagues [17] solved the problem by fixing the rate at which packets are passed from the routing queue to the ARP queue. Also, they realized that the size of the ARP buffer could be increased to store more packets. Moreover, they suggested that the routing protocol could be aware that the ARP module already has the link-layer address of the next forwarding node.

Authors of [20] present the negative effects that result from the separation of neighbor discovery and link layer address resolution. They focused their attention on the conflict produced in the interaction between a *manycast* application and ARP. They observed the problems arose due to the dynamic topology in MANETs, which caused issues in the routing protocols because they need to use updated neighbor information. The conflict in *manycast* occurs when the route replies are propagated back to the *manycast* source. Carter and colleagues proposed in [20] an approach called Automatic Address Resolution (AAR) that is based on the maintenance of AR coupled with the neighbor discovery process carried out by the routing protocol through exchange of signaling messages needed to establish routes. They claimed that any routing protocol could easily perform automatic AR by recording the link-layer address when a neighbor discovery occurs. This is specially suitable for MANETs because this operation avoids the use of ARP messages. This ensures that the neighbor's link-layer address is already known when a unicast communication with that neighbor is needed. However, this proposal assumes that the link-layer source address is automatically accessible by the routing protocol and that AR does not use additional communication resources. We claim that this fact is not so evident and easy in real scenarios as it could seem. In any case, the results in [20] show that automatic AR performs better than ARP regardless the type of routing protocol (reactive or proactive).

Finally, in [26] a communication scheme is proposed to enable some IPv6 procedures without link-scope multicast. Since the support of link-scope multicast is difficult for VANET scenarios, the authors propose packet delivery mechanisms that take advantage of inherent location management functions. They support the proposal in the use of C2C architecture to perform IPv6 operations. The work comes up with the use of additional C2C and Layer 2 headers to encapsulate IPv6 packets in order for them to reach an IP next hop and a Layer 2 neighbor. By using beacon messages they propose to send the IEEE 802.11p MAC address (among other type of information) of the source node in a communication process. This is how the neighbor discovery process is carried out without depending on multicast.

5.4 Address Resolution proposals

This section presents the major contribution of our work about the improvement of the AR process. We name our approach as Coherent, Automatic Address Resolution (CAAR) for IP multi-hop vehicular ad hoc networks. First of all, we examine in Sec. 5.4.1 the vehicular network scenario that we have used in this work. Next, in Sec. 5.4.2 we review some straightforward modifications to the address resolution procedure in order to improve its performance. Finally, Sec. 5.4.3 specifies the characteristics of our novel address resolution approach.

5.4.1 Application Scenario

We focus our analysis on a multi-hop vehicular ad hoc network that performs routing operations at the network layer. This approach has the advantage of preserving the original functionality of the different layers in the stack of protocols. In the following, we describe the communication procedure performed in our application scenario.

Communication Procedure

For the application scenario that we have considered, the following assumptions were done:

- Two vehicles that are not in communication range of each other need a forwarding node able to connect those two nodes. Different from traditional wired systems, the packet forwarding operation occurs in VANETs even when nodes are in the same network addressing (within the same “IP site"). For ad hoc networks, the concept of forwarding is more related to the physical distance between nodes rather than a logical hierarchy of network addressing.
- A node needs to have a valid non-link local IPv6 address since nodes cannot forward packets using link-local addresses [48]. In particular, for this work we assume that vehicles use a unique local address. A unique local address [49] enables packet routing only within a site, so packets are not expected to be routable through the Internet.

The following 4 steps were performed to enable communication in the scenario:

1. A node obtains the IP configuration parameters from the IP site it belongs to. A vehicle in a multi-hop scenario has two ways to do this:
   a) If vehicles have direct communication to an access point, the Road Side Units (RSUs) provide them with IP parameters for self configuration. The RSUs periodically broadcast the IP configuration (to one-hop neighbors) in the acknowledgement frame of the Wireless Access Vehicular Environment (WAVE) service, which is defined in the IEEE 1609.3 standard [53].
   b) If a vehicle is located several hops from a RSU, it is unable to receive a service acknowledgement frame sent by the RSU. In such case, vehicles have to use a multihop modification of the Router Discovery (RD) Process . The RD process is one of the functions of the Neighbor Discovery Protocol, and consists of two messages: Router Solicitation and Router Advertisement. The latter carries the IP parameters (network prefix, MTU, etc) of the vehicle network configuration.
   A modification of multihop RD is presented in [131]. It proposes the use of Router request and reply messages to encapsulate the Router Solicitation and Advertisement messages. The router request is sent to a predefined multicast gateway address. The gateway will respond using a unicast Router Advertisement. A unicast temporal IP address is used by the initiator node to receive the router reply. This initial and temporal IP address is built from a specific prefix called `MANET_INITIAL_PREFIX`. Alternatively, this initial IP address can be the valid IP used for the previous associated IP site.
2. A vehicle auto-configures a unique-valid local IP address. After a node receives its IP configuration, it generates for itself a unique Interface ID (IID) to be used on the stateless IP address auto configuration process. The IID is appended to the network prefix in order to generate a unique local IP address. This unique ID can be generated using a random generator as proposed in [131]. This process has to do duplicate detection. Another option

is using the Vehicle Identification number (VIN) [56], which may avoid the duplicate detection phase.
Other IP configuration mechanisms for mobile ad hoc networks have been surveyed in [10] and all of them can be applied to our VANET scenario.

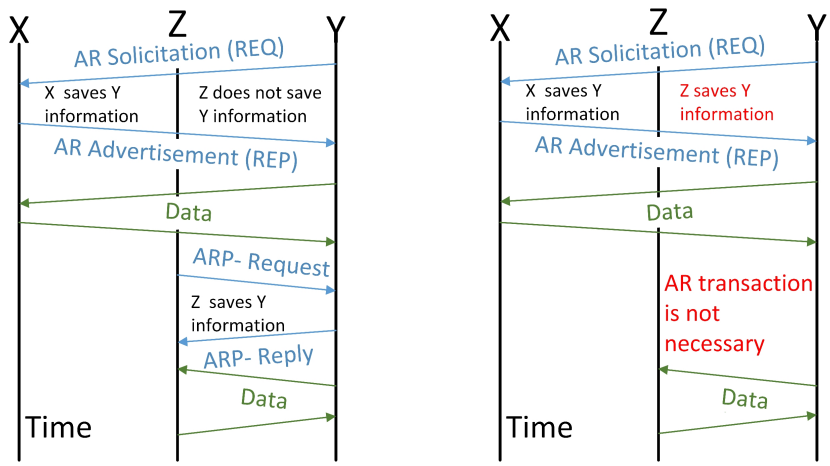
3. Once a vehicle is configured with a local-unicast IPv6 address, it can establish unihop or multi-hop IP communications, either V2V or V2I. In any case, both parts of a communication process will need to perform link-layer address resolution in order to reach the next forwarding hop before sending data packets.
4. When a node leaves the influence area of the IP site to which the vehicle is associated, it will receive new IP configuration parameters announced by the WAVE service acknowledgement frame of another RSU in the new IP site or through multihop procedure. A node could be aware that it is not in the range of its associated IP site, when:
   a) A vehicle receives hello messages from neighbors with different IP range.
   b) A node is able to know, based on its position, whether it is in the geographical area covered by its current IP Site. This information can be sent during the IP configuration process or by other means such as an IP Site bounds layer in the preloaded maps of the vehicle.
   c) Besides, during the change from one IP site to another, the node could use its neighbors from the previous IP site still reachable to send information through them. Once a node obtains the new IP configuration and begins to discover neighbors in the new site, it should erase all the obsolete associations.

These steps show possible ways that enable vehicles to keep connected with their surrounding peers and other infrastructure services. In this work, we assume that address auto-configuration has already been accomplished, and that all the nodes are in the same IP site using stable and valid configuration. Below we explain the characteristics of the traffic we used for the VANET scenario.

5.4.2 Improvements to the Address Resolution Procedure for Ad hoc Networks (AR+)

In vehicular networks, every node listens to the communications coming from the nodes within the coverage area (its neighbors). Also, the address resolution (AR) request messages are always received by the node regardless the recipient of the messages, so AR messages could be easily processed by all nodes that anyway receive them to take advantage of that alredy available information. Based on those two facts, we propose AR+, which includes the following straightforward modifications of the AR operation to enhance the overall performance of the AR procedure.

- A node must perform the *AR learning operation*, as described in Sec. **??**, using all the AR request packets that it hears, and not only using those packets addressed to that node. This change is based on the fact that all the AR packets that a node receives come from any of its neighbors, which might become next forwarding nodes for other packets in the near future. If that is the case, new AR transaction might not be necessary since the nodes have already proactively saved the MAC addresses of possible next hops. In simple terms, a node adds or updates an entry of its AR table for any kind of AR message received, regardless of whether the node needs that entry for its current routes or not. Figure 5.1 shows the advantage of this scheme. By applying the proposed modification, the AR process would sometimes not be necessary since the address would already be resolved.



(a) Traditional AR Procedure.

(b) Extension of the AR learning

Figure 5.1: Extension of the AR Learning operation

- A node will maintain an AR queue at least as large as the routing protocol queue. This prevents a packet in the AR buffer from being dropped when a new packet comes from the routing queue if the address resolution process is not completed yet. That is, there is a packet in the AR queue waiting for the address resolution process to be completed. If the AR queue is very short (typically 1 packet) then a new incoming packet from the routing queue may drop that stored packet because the AR process does not communicate with routing protocol. This problem can be mitigated by just making both queues of the same length.
- According to the previous modification, since the AR buffer is able to store more than 1 packet (its length is a parameter of design), the AR process could send as many requests as stored packets in the AR module, but without exceeding the maximum AR request rate of 1 message per second per destination. In this way, a node has more than one chance to request an address resolution.
- A node will perform a proactive dequeuing of packets. Its AR module will not only try to send stored packets when an AR reply is received. In addition, it will try to dequeue packets when any address resolution message is heard. That is, given two neighboring nodes (let us say A and B), node A might receive a request message coming from node B and going to any node (not necessarily to A) before the reply message from B arrives to A since, for example, the reply was lost. In this way, A can learn the MAC address of B even if there are losses in their own request-reply procedure.
- A final modification we propose to include in the AR procedure relies on the coherence of the services among layers. We claim that the “validity ARP entry time" should be equal to the “valid time of a route" in the case of topology-based protocols (e.g., AODV) and to the “neighbor‘s lifetime" in geographical-based protocols (e.g., GPSR). The objective is to prevent a new AR transaction for a neighbor whose entry was previously deleted from the AR table (because its timer expired) while that node is still valid for the routing level (its entry in the routing table is still valid).

All the previous changes were implemented in our enhanced AR process for VANETs seeking to improve the vanilla AR in terms of losses and delay as well as to decrease the traffic necessary to perform the AR task. The improvements proposed to the AR resolution process, called AR+, might be interesting as strategies to enhance its performance without significantly changing the protocol.

5.4.3 Coherent, Automatic Address Resolution (CAAR)

As we mentioned in the introduction of this chapter, our work extends the Automatic Address Resolution (AAR) proposed in [20] for MANETs. We called our proposal CAAR Coherent AAR since its operation is in concordance with the different tasks assigned to each communication layer. CAAR is straightforward to be implemented, avoiding interactions between routing and MAC layers to perform the AR process. In this section we show the advantages of our approach by means of a detailed description of the process flow of AAR and CAAR.

Automatic Address Resolution Procedure

Carter et al. [20] assumed that the source MAC address field of the MAC header is transparently available for the routing protocol daemon after receiving routing signaling messages. However, the implementation of a mechanism to make the MAC address available to the routing protocol is not a trivial task. The main reason lies in the extra work that the link layer needs to perform. We propose and show in Fig. 5.2 two possible paths that a packet may follow when applying the approach of [20]. In both cases the link layer obtains the MAC header, reads the source MAC address and dispatches the frame payload to the IP layer. A first possible implementation is:

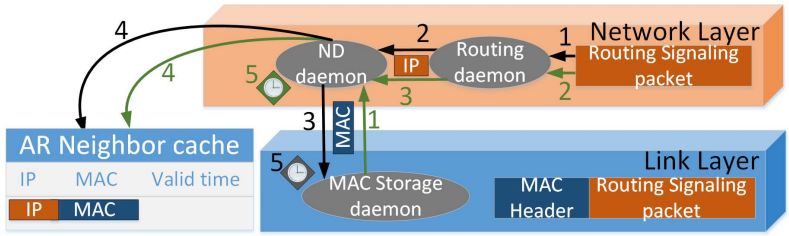


Figure 5.2: Possible packet flows in the Automatic Address Resolution

[**Green path** in Fig. 5.2]. The link layer pushes the Source MAC Address to the AR process which starts a timer (step 1) that waits an IP address to shape a Neighbor entry to be added in the neighbors' table. Next, the signaling packet is processed by the routing daemon (step 2), which extracts the IP source address and sends it to the AR daemon (step 3). Finally, a proper neighbor entry is added to the AR table (step 4). If no IP address is provided to the AR process before the timer expires (step 5), then the MAC address is discarded.

An alternatively procedure differs from the previous one in the implementation of a timer, as follows:

[**Black path** in Fig. 5.2]. The link layer saves the source MAC address and initializes a timer. The signaling packet is treated by the routing daemon (step 1) which extracts the source IP address and sends it to the AR process (step 2). The AR daemon retrieves the source MAC address saved by the link layer (step 3) and writes the new information in the neighbors' table (step 4). As in the first process flow, if the timer initialized by the link layer expires (step 5), the MAC address is discarded.

For both alternatives, if a new packet arrives before the previous AR entry (corresponding to the previous packet) is created, both MAC addresses (the one of the previous packet and the new one of the current packet) are discarded. We would like to point out that there are many options or extra steps that could be performed over the mechanism to do the AR, as explained before. Nevertheless, we think that all these improvements entail too much extra work to the link and routing layers and, furthermore, they are not necessary to carry out the AR.

Process flow of CAAR

CAAR is really straightforward. Every node copies its MAC address, available for instance from its MIB (Management Information Base), into the signaling routing messages as a new field of them. This action will slightly increase the length of those signaling messages in 6 bytes with the purpose of not augmenting the MAC layer complexity or cross layering cooperation between MAC and routing layers. The scheme of our proposal, depicted in Fig. 5.3 and its algorithm (in Fig. 5.1), works only at the network layer. This is explained below.

[**CAAR Path** in Fig. 5.3]. When a routing signaling packet arrives to a node, it is processed by the routing daemon (step 1) which extracts the source IP and MAC addresses (step 2, ROUTING-PROCESS algorithm) from the IP header and the signaling message, respectively. We do not need to use timers, since the match is directly done when an IP Packet encapsulated in a routing message is received. Such IP packets, as depicted in Fig. 5.4, contain both IP and MAC addresses at the same communication layer. Next, the pair of IP and MAC addresses is sent to the Address Resolution process (step 3), which adds or updates an entry in the AR table.

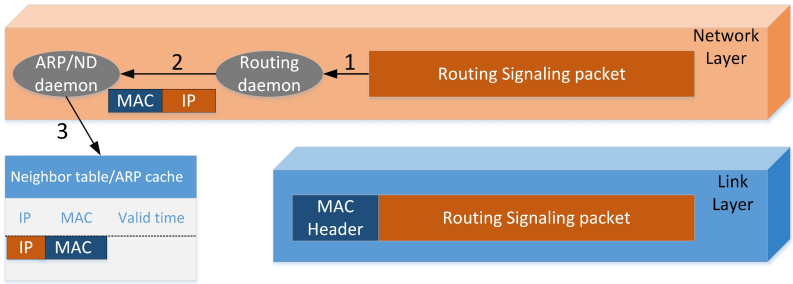


Figure 5.3: Scheme of our CAAR proposal

The AR-PROCESS algorithm in CAAR is extremely simple compared to the tasks assigned in the AAR alternatives. The main advantage of our approach is the different minimal extra work introduced in the routing process, whose goal is to attain proper addresses tuples

ROUTING-PROCESS(P)

Require: A packet P
Ensure: MAC_{src} **and** IP_{src} of P to AR-PROCESS.
1: $MAC_{src} \leftarrow$ GETSOURCEMAC(P)
2: $IP_{src} \leftarrow$ GETSOURCEIP(P)
3: **if** ISROUTINGMESSAGE(IP_{src})$= TRUE$ **then**
4: AR-PROCESS($MAC_{src}P$, $IP_{src}P$)
5: **return**

AR-PROCESS($MAC_{src}P$, $IP_{src}P$)

Require: MAC_{src} or $IP_{src}$of P
Ensure: AR-ENTRY(MAC_{src} , IP_{src}) to TableARP.
1: $TEMP_{MAC} = MAC_{src}$
2: $TEMP_{IP} = IP_{src}$
3: Entry=AR-ENTRY($TEMP_{MAC}$, $TEMP_{IP}$)
4: **return** Success

Algorithm 5.1: CAAR algorithms.

(MAC, IP) to be used by the AR process. Notice that the operation of the link layer remains invariable. Moreover, the CAAR is *automatic* because it matches IP and MAC addresses without using any AR signaling. Notice that we have eliminated every exchange of messages of the traditional AR signaling, since they are not necessary in CAAR. This way we increase the performance of the VANET as a consequence.

5.4.4 CAAR Implementation in VANET routing protocols

We have explained the advantages in the implementation of CAAR at the cost of minimal overhead in the routing signaling message. Since the AR process is triggered by the reception of a routing message, the AR procedure is done for all the neighbors, which are the candidates as possible next forwarding nodes. This inhibits the exchange of Neighbor Solicitation and the Neighbor Advertisement messages of AR signaling.

Our proposal is *coherent* in the implementation with VANET routing protocols. First, it only adds the MAC address in a new field and performs AR within the routing messages used to create or maintain routes, which are common tasks for all VANET routing protocols. Fig. 5.4 shows the IP encapsulation for a hello message structure. Notice that the new MAC field is added at the end of the signaling message to facilitate its processing by the routing daemon. This is suggested in case other type of MAC address different from Ethernet is used. In such cases, the MAC field should include the type and length sub-fields to distinguish the MAC protocol. This approach is the same implemented by ND to deal with different Source/Target Link-layer Address [85]. Nevertheless, the use of a different MAC address format seems unlikely because the WAVE architecture only supports IEEE MAC 802.11p [83]. In this work we just encapsulate the MAC address into the MAC field of the routing signaling messages.

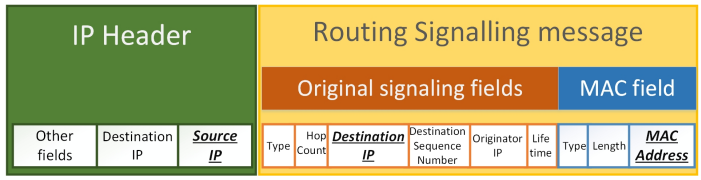


Figure 5.4: An IP Packet carrying a VANET routing signaling message.

In **topology-based protocols**, the MAC field should be added in the Route request (RREQ) and Route reply (RREP) during the route discovery process and in the hello messages part of the maintenance process. Fig. 5.5 depicts the operation of CAAR during the creation of a route. Notice that while the MAC address is obtained from the routing signaling message, the source IP address is read from the IP header, since this IP address corresponds to the previous hop. The destination and originator's IP addresses from the routing signaling message are part of the end-to-end path (e.g., nodes A and C in Fig. 5.5) and do not change during the path construction. Obtaining information from different parts of the IP packet (header and payload) is a task already implemented by the routing protocol and it is done to create the backward path. Therefore, this task does not really entail to much extra work. An intermediate node in the path (node B in Fig. 5.5) forwards its MAC and IP addresses in routing message (IP packet payload) and IP header, respectively. These read/write operations on the routing signaling message should not be considered as an additional task due to the fact that a forwarding node must anyway update information in the signaling message, like number of hops. Hello messages, are one-hop route reply messages that only inform about the presence of a node. Thus, the corresponding work in CAAR in this matter is receiving a hello message and not forwarding it. CAAR does not use Error notification messages, which inform about invalid routes that are not going to be used anymore. Thus, it is unnecessary to perform address resolution for routes that are no longer used. This might represent important savings in bandwidth, taking into account that VANETs have a very dynamic topology where routes get broken frequently.

Since the messages we use to implement CAAR are needed for every topological protocol (no matter which routing criteria is implemented) in the building process route, such messages can be leveraged to transport additional information by adding fields in them. More important, CAAR is independent of the routing criteria used by the node to create the path since the MAC is carried in all the routing signaling packets relevant to these tasks.

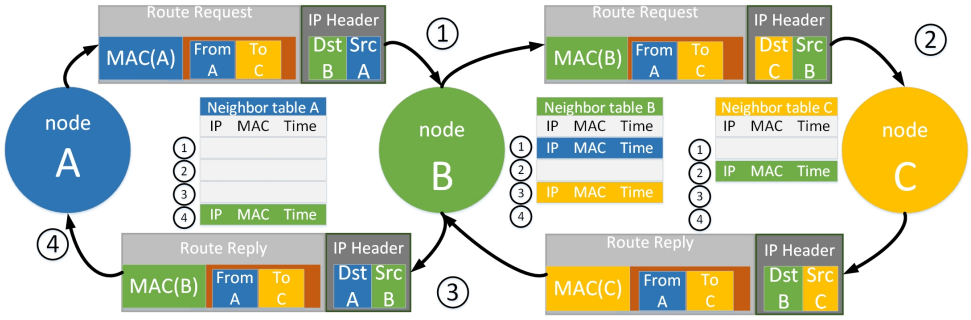


Figure 5.5: An IP Packet carrying a VANET Routing signaling message.

Geographical-based protocols are usually associated with a greedy approach because they make the forwarding decision at the arrival of each packet, based on local information only. This means that this category of routing protocols only need information coming from nodes at a distance of one hop. Therefore, most of the routing protocols implement only hello signaling messages with a number of fields depending on the information that the protocol uses to forward packets. Since hello messages can be considered as one-hop route replies, its destination field (see Fig. 5.4) contains the IP address of the originating node, which is also the source IP address in the IP header. Thus, the AR process can be performed without using any information of the IP header.

Finally, CAAR is *coherent* in the management of expiration timers. We make the lifetime of ND cache entries equal to the route lifetime in topology-based protocols or equal to the neighbor lifetime in position-based protocols. It means that when the route or neighbor entry is created, the ND entry is also created with the same timer, and both timers are updated by the reception of a hello message. This simple action prevents unnecessary ND transactions since an ND timer with a timeout shorter than the route/neighbor timer's causes a ND entry to be deleted (its timer expires) even when the corresponding node is still a valid next forwarding node (its timer is still alive). But then, a ND timer with a timeout longer than the route/neighbor timer entails an unnecessary larger ND cache with entries useless for the routing daemon.

5.5 Performance Evaluation

We analyze the performance of our proposed CAAR scheme, which includes the MAC address in the routing signaling messages. We compare CAAR to the traditional AR named TAR (using the default operation) and AR+ (our AR improvement that modifies some parts of the traditional AR). We use MMMR [113] as the VANET routing protocol to evaluate the behavior of our proposal. We use the simulation scenario and settings described in Sec.3.4 from Chapter3 with three different number of vehicles: 60, 100, 150 which correspond to densities of 40, 67 and 100 veh/km^2, respectively.

All the figures are presented with confidence intervals (CI) of 95% obtained from ten simulations per point using independent mobility seeds per simulation. Regarding the AR parameters, Table 5.1 shows the values set in the simulations for the three AR schemes. These parameters are the ones discussed on Sec. 5.2, Sec. 5.4.2 and Sec. 5.4.3 for TAR, AR+ and CAAR, respectively. The first row of this table, identify the types of messages used by each AR scheme to update information. The next three rows indicate the length of the AR buffer per destination, the rate of AR request messages and the maximum number of AR request messages. The two last rows are the lifetime of the AR entries and the period to check any change in the validity of the Ar entries.

We have measured four metrics in our performance evaluation: percentage of packet losses, delay, average number of hops and incurred AR signaling. For a deep statistical analysis of the results and because we noticed a trend among the three AR mechanism, we have performed the Jonckheere-Terpstra test [101] (called J-T test henceforth) to check if the three AR schemes show the same behavior (known as *null hypothesis*) or if the results indicate a certain trend (known as *alternative hypothesis*) in relation to a given ordering among the AR schemes. The order used in this work to arrange the AR schemes and carry out the J-T test was TAR, AR+ and CAAR.

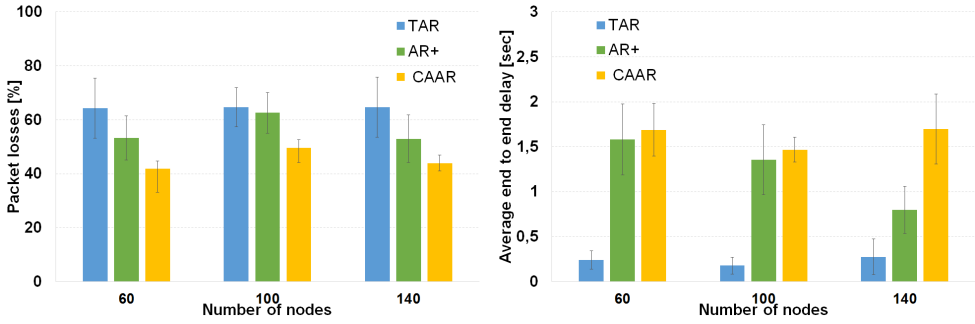
Parameter	AR Schemes		
	TAR	**AR+**	**CAAR**
Update entry process	AR pkts to node	AR pkts to anyone	routing pkts
AR buffer ($pkts$ x dst)	1	30 = routing buffer	1
AR Request rate ($pkts/s$)	1	1	-
Max AR-Req	3	= AR queued pkts	-
Lifetime AR entry (s)	3	2 (MMMR neighbor lifetime)	
Table checker timer (s)	1	0.300 same as routing checker	

Table 5.1: Parameter settings of Address Resolution.

In the cases in which the J-T test stated that the results among the AR schemes were statistically the same (and therefore the test could not reject the Null Hypothesis), we carried out the Kruskal-Wallis test [101] (called K-W test henceforth) to assess whether the three results followed the same distribution (*null hypothesis*), or if there was any difference in the results among TAR, AR+ and CAAR (*alternative hypothesis*) regardless that the difference among those results did not follow a clear tendency in the J-T test. These tests are summarized in Table 5.2. In the results of J-T tests we included the value of the Standardized Test Statistic (STS), which can be seen as a measure of the correlation existing between the results assessed and the AR scheme implemented. The sign of STS indicates if the result increases (positive sign) or decreases (negative sign) when we change the AR scheme following the evaluating order established between schemas (i.e., TAR, AR+ and CAAR). The magnitude of the STS gives a notion of how much strong is the result's change among the AR schemes. Additionally, we included for all the cases where there is a difference among the results of the AR schemes (i.e., the null hypothesis is rejected), a pairwise comparison between the AR schemes to visualize the tendency in the J-T test. These pairwise results are presented in Table 5.3.

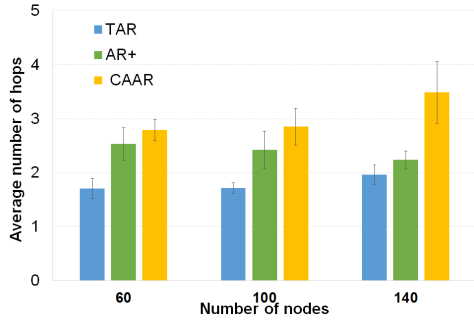
From the packet losses point of view, drawn in Fig. 5.6a we can notice that losses decrease as we included improvements in the AR operation, i.e., AR+ improves TAR, whereas CAAR further improves AR+. Losses-AR dependability decreases with the number of nodes (see how the STS negative value of the first row in Table 5.2 increases in magnitude as N increases). CAAR always performs significantly better than TAR in terms of losses even in the low density scenario (see Fig. 5.6a). This behavior is due to the hop-to-hop forwarding done in MMMR for each packet that requires AR procedures in each hop. At the moment of taking the forwarding decision, the use of a buffer with AR+ and the no necessity of any AR transaction with CAAR alleviates both the packet discarding using AR+ and the collisions using CAAR.

In the case of delay, see Fig. 5.6b, MMMR increases the delay as the AR operation changes from TAR to CAAR, i.e AR+ produces delays higher than AR and CAAR shows delays higher than AR+. See the STS values in the second row of Table 5.2 in which the positive STS values indicate an increasing trend in the delay as the AR scheme changes. This is because



(a) Percentage of packet losses.

(b) Average end-to-end delay.



(c) Average number of hops.

Figure 5.6: Performance evaluation to compare TAR, AR+ and CAAR using the MMMR routing protocol [113] (CI 95%).

		$N = 60$		$N = 100$		$N = 140$	
Parameter Hypothesis	**Test 60/100/140**	**STS**	***p*-value**	**STS**	***p*-value**	**STS**	***p*-value**
Losses	J-T/J-T/J-T	-3.15	0.002	-3.23	0	-4.53	0
Delay	J-T/J-T/J-T	4.11	0	4.53	0	4.72	0
Hops	J-T/J-T/J-T	4.37	0	4.87	0	4.64	0
AR Signaling	K-W/K-W/K-W		0.585		0.769		0.495

Table 5.2: Hypothesis Test Summary to test the effect of the AR mechanism. J-T = Jonckheere-Terpstra tend test. K-W = Kruskal-Wallis test. STS = Standardized Test Statistic. We performed J-T test for all the evaluated metrics. When the J-T test retains the null hypothesis (i.e., p-value$\geq$0.05), we performed K-W to look for any difference. Null hypothesis: The distribution of the parameter's result is the same across ARP, ARP+ and CAAR. J-T Alternative hypothesis: The distribution of the result follows an order across ARP, ARP+ and CAAR. K-W Alternative hypothesis: There is at least one scheme for which its distribution is different from the other schemes. In all the tests, we rejected the null hypothesis when the p-value is lower than the **significance level of 0.05**.

the improvement in packet losses produces that packets from farther positions can reach destination. This occurs with AR+ and CAAR (see Fig. 5.6a), so the average delay in those schemes increases (packets coming from farther sources take longer to reach destination).

			p-value		
# row	**# Nodes**	**Parameter**	**CAAR-TAR**	**CAAR-AR+**	**TAR-AR+**
1	60	Losses	**0.01**	**0.029**	0.05
2	60	Delay	**0**	0.272	**0**
3	60	Hops	**0**	0.075	**0.001**
4	100	Losses	**0.004**	**0.006**	0.298
5	100	Delay	**0**	0.076	**0**
6	100	Hops	**0**	**0.016**	**0**
7	140	Losses	**0**	**0.01**	**0.019**
8	140	Delay	**0**	**0.004**	**0.012**
9	140	Hops	**0**	**0.001**	**0.025**

Table 5.3: Pairwise comparison of performance metrics. The comparisons were performed using the Mann–Whitney test. Null hypothesis: The results of the two AR schemes come from the same distribution. Alternative hypothesis: The distributions of results of the two AR schemes are different. In all the tests, we rejected the null hypothesis when the *p*-value is lower than the **significance level of 0.05.**

The difference of delay between AR+ and CAAR is only statistically significant for the high density scenario. This can be seen in the 8th row of Table 5.3, for the delay with N=140 nodes. For CAAR-AR+ with N=140, the p-value below 0.05 means that the delays between both AR schemes behave statistically differently; while for CAAR-AR+ for $N = 60$ (2nd row) and N=100 (5th row), the p-value is above 0.05, meaning that they behave similarly. The reason is that CAAR maintains the collisions low while AR+ cannot avoid them due to the high number of nodes, which allows more successful transmissions. Notice that, the delay with MMMR increases very much for long distances (which are much possible using AR+ or CAAR than using TAR) because it performs a conservative next hop selection, storing packets in a buffer instead of dropping them when there is no proper next forwarding node. This feature improves packet losses (see Fig. 5.6a) but increases the average end-to-end delay (see Fig. 5.6b).

Regarding the number of hops, depicted in Fig. 5.6c, they are completely related with the delay and the packet losses. With AR+ and CAAR, the destination node can receive packets from farther sources, generating longer paths with higher number of hops. Looking at Table 5.3, we can see a significant difference in the average number of hops between AR+ and CAAR when there is also a significant difference in the delay. For instance, the aforementioned significant difference between CAAR-AR+ in delay for N=140 is also present in the behavior of hops, as it can be seen in rows 8th and 9th of Table 5.3 with a p-value$<$0.05. In opposition, for low densities there is no significant difference (p-value$>$ 0.05) as the reader can see in Table 5.3 in rows 2th and 3th for N=60. For the middle case of N=100 nodes, we can see in 5th and 6th rows that both AR produce the same delay but CAAR has a higher average number of hops because if helps to successful transmit more packets.

We also included an analysis of the Address Resolution signaling in our evaluation. The incurred AR signaling was calculated in the case of CAAR as the extra 6 bytes added to the hello routing messages to carry the MAC address. On the other hand, the TAR and AR+ signaling were computed using the total number of REQ/REP messages of AR transactions.

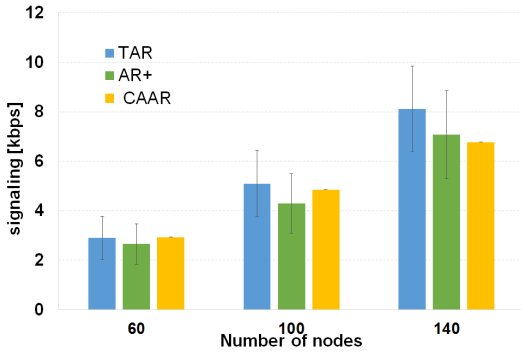


Figure 5.7: Signaling traffic incurred only by the AR process using the three Address Resolution schemes (CI 95%). AR signaling in Traditional AR (TAR) and AR+ involves all AR REQ/REP messages. AR signaling in CAAR only consists of 6 bytes added to hello routing message in MMMR to carry the MAC address.

Fig. 5.7 presents the results with the three vehicles densities evaluated. There is no significant difference among the signaling traffic introduced by the three AR resolution schemes (see the p-values in the last row of Table 5.2, which are above the significance level 0.05 meaning that the three results follow the same distribution).

Based on these results analyzed so far, we can conclude that CAAR improves the packet-losses performance thanks to a coherent and straightforward mechanism to perform the AR process without introducing extra signaling overhead.

5.6 Conclusions

The main contribution of our work in this chapter is a new scheme to carry out the Address Resolution (AR) process in Ad Hoc networks, which we called Coherent, Automatic Address Resolution (CAAR). CAAR improves the traditional AR in vehicular scenarios. CAAR encapsulated the MAC address in the periodic, already available signaling messages of the routing protocols.

Our simulations show that the performance evaluation results of the routing protocol under study (MMMR), have a statistically significant dependence with the AR scheme implemented. CAAR improves the percentage of packet losses compared to the other schemes we evaluated, i.e. traditional AR (TAR) and AR+. As consequence, the average number of hops and the average packet delay increases. CAAR obtains the highest values of delay (around 1.8 sec instead of 0.8 sec for the other schemes). The reason is related to the lower packet losses (around 20% fewer losses), which is due to the absence of AR signaling avoiding collisions, which increases the number of packets using longer paths to arrive at the destination. However, packets coming through long distances could be stored in the routing buffer longer than packets coming through short distances until suitable forwarding nodes are found. Finally, our statistical analysis shows that CAAR in any case introduces more overhead, keeping it on the same level as classical AR over MMMR.

6. Mitigation of packet duplication in VANET routing

Vehicular ad hoc networks (VANETs) are self-organized networks designed to improve the drivers' safety and the efficiency of vehicular traffic management. In this kind of networks, packet losses are common in the path from source to destination, mainly due to the high mobility of the nodes (i.e., vehicles). Reliability mechanisms at different layers are used to mitigate the impact of this issue. In this chapter, we explain how the loss of acknowledgment frames (ACK) at the medium access control (MAC) layer with the recovery procedure of VANET routing protocols lead to an unexpected generation of duplicate packets. We propose two incoming filters with different levels of detection of duplicate packets at routing layer to prevent the propagation of unnecessary packet copies, so that the available bandwidth will increase as consequence. We carried out simulations with three vehicle densities in an urban scenario and with two different mechanisms to adapt the size of the contention window (CW), which is an important factor in VANETs to avoid collisions. The extensive statistical analysis of the results show that our approaches are effective to reduce the traffic load in terms of routing operations, idle time, duplicate packets and end-to-end delay. However, we also observed an expected minimum degradation of the packet delivery ratio, thus the benefits of the proposal are clear.

6.1 Introduction

Vehicular *ad hoc* networks (VANETs) [47] are temporal, self-organized networks where vehicles send and receive information to other vehicles or to fixed infrastructure points. These communications might help to improve the driver's safety and the efficiency of the vehicular traffic management. VANETs have to face fast topology changes, a low link lifetime and a potentially high number of nodes taking part in the network, among other issues. Particularly, the high mobility is considered a challenge because it may cause scattering of nodes, preventing a correct data delivery. Thus, the IEEE 802.11 MAC layer uses acknowledgment frames (ACK) to ensure the correct reception of a frame and routing protocols resend packets for which the MAC layer did not receive the ACK in a given time.

Most of the research work on VANETs has been focused on improving the routing mechanisms to increase their performance under such dynamic environments. In this chapter

we study the occurrence of undetected duplicate packets due to the interaction of the reliability mechanisms implemented at the MAC and routing protocol layers. Moreover, we propose two strategies to deal with propagation of undesired packet copies to increase the availability of the common wireless channel for new packet transmissions.

The rest of the chapter is organized as follows: Sec. 6.2 summarizes similar studies about the management of packet copies. Then, Sec. 6.3 describes the reliability mechanism employed in a unicast VANET communication and why duplicate packets appear in unicast communications in VANETs. Afterwards, Sec 6.4 presents the two mechanisms proposed in this work to control the number of duplicates. After that, Sec. 6.5 is devoted to describe the evaluation of our two proposals operating in the selected representative routing protocols and the results obtained from the statistical tests. Finally, conclusions are drawn in Sec. 6.6.

6.2 Related work

There are several proposals for wireless and mobile network technologies that try to improve the availability of resources, maximize the throughput capacity and reduce the network latency. In [103] the authors focus on the energy-delay and the storage-delay trade-offs on a mobile sensor network using the Shared Wireless Infostation Model (SWIM) [102]. The SWIM scheme can lead to a large number of redundant packet copies in the network. The storage in the node is released by eliminating the packets that have at least one copy already offloaded. Authors present five possible removal packet methods: JUST TTL, FULL ERASE, IMMUNE, IMMUNE TX and VACCINE. Each of these methods corresponds to a different storage-delay trade-off in the system. On one hand, with JUST_TTL all packets remain in the system until T seconds have elapsed from the original packet creation. On the other hand, VACCINE (the most severe method) erases the packet from a node when it is transmitted to the next node and shares an identifier of the packet to avoid further propagation of the copies.

The work proposed in [139] studies the performance of various epidemic style routing schemes by means of ordinary differential equation (ODE) models. The authors conclude that once a node delivers a packet to destination, it should delete the packet from its buffer to save storage space and handle already-delivered packets using any of the aforementioned removal packet methods. The authors present three schemes:

- K-hop forwarding. A packet can traverse at most K hops to reach destination.
- Probabilistic forwarding. Under this scheme when two nodes meet, each node accepts a relay packet with probability p. When p=0, the probabilistic forwarding degenerates to direct source-destination delivery, and when p=1, epidemic routing is performed. Varying p in the range (0,1) allows a trade-off between storage/transmission requirements and delivery delay.
- Limited-time forwarding. When a node accepts a packet copy, it triggers a timer. When the timer expires, the copy is deleted from the buffer. The choice of the timeout value allows trading-off the delivery delay against storage and number of transmissions.

The proposed model [139] assumes that when two nodes move into the transmission range of each other, they exchange: their identifications and information of packets already forwarded, which is known as anti-packet. Also, nodes use a global timer associated with each packet to handle the lifetime of anti-packets. This is done to avoid an excessive signaling overhead in epidemic routing.

These mitigation mechanisms to avoid flooding of the ad hoc network with copies have been largely studied for dissemination-purposes protocols. Nonetheless, to the best of our knowledge there is not similar proposals or evaluations for unicast transmissions because it was not expected to have packet copies in such cases.

6.3 Background

Before explaining our contribution to prune duplicate packets in unicast transmissions in VANETs, in this section we provide information about the reliability mechanisms at MAC and routing layers used in multi-hop unicast VANET communications to improve the packet delivery. Also, we discuss how packet replicas appear in unicast communication in a VANET due to the use of those reliability mechanisms.

6.3.1 Reliability mechanisms in a VANET communication

Reliability mechanisms ensure the reception of messages. In VANETs packet losses take place in the path between source and destination usually because of collisions with other packet transmissions, fast movement of the nodes and limited communication range of vehicles. Therefore, reliability mechanisms are of paramount importance for the performance of VANET routing protocols. Since the routing operation is done at each hop, reliability mechanisms are performed hop-by-hop. In unicast VANET routing protocols, reliability mechanisms usually perform at both the MAC and the routing layers.

Acknowledgment MAC procedure

A frame transmission may be unsuccessful for a variety of reasons, e.g. collisions, unrecoverable bit errors induced by fading, interference, etc. A typical example is when a node moves into a neighborhood and it becomes a hidden terminal that can provoke collisions with an ongoing transmission.

To guarantee that a frame was received successfully the IEEE 802.11 [55] standard establishes the sent of acknowledgment frames (ACK frames). According to the standard each time a node receives a unicast MAC frame addressed to it, the node has to reply with an ACK frame to the sender node. If the ACK frame arrives within the ACK timeout interval then the node sets its contention window (CW) to the minimum value and a new frame transmission process can begin. Otherwise, the sender node doubles its current maximum CW size and starts the retransmission of the frame. The retransmission process is carried out a predefined maximum number of times (`MAX_RETRY`).

The IEEE 802.11 standard prevents the possibility that a frame may be received more than once because of the loss of an ACK frame. This is achieved by means of a cache of tuples (address, sequence number) from recently received frames in the receiver node. If a tuple from an incoming frame matches with an entry of the cache then the node shall reject that frame as a duplicate. After the maximum number of retries is reached, the MAC layer informs the routing protocol that the packet could not be sent and returns the packet to it.

Local recovery on VANET routing protocols

When a routing protocol receives a notification from the MAC layer informing that a packet could not be delivered to the selected next forwarding node, the routing protocol starts a

local recovery process, which can proceed in the two following ways:

1. The routing process can retry the transmission of the packet to the same node. It can be performed a predefined maximum number of times (`MAX_ROUTING_RETRY`). Notice that for each transmission retry from the routing layer, the MAC layer would perform (`MAX_RETRY`) retransmission attempts. Using or not the same node to forward a packet can be decided based on the freshness of that neighbor entry. The objective of this is to reduce the process' complexity and not to discard routes or neighbors because of temporary disconnection between nodes.
2. Once the number of attempts of routing retransmissions reaches its maximum, the routing protocol uses the MAC feedback to update its information as follows:
   - In the case of topological approach protocols, the routing protocol invalidates those route entries that have as next hop an unreachable node.
   - In the geographical-oriented protocols, the protocol deletes the entry of the unreachable node from the table of neighbors.
   - Finally, the recovery mechanism behaves differently depending on the protocol. For instance, iAODV tries to repair the broken route. Geographical protocols like GPSR or MMMR look for another neighbor to forward the packet. In the meanwhile, the packet is stored in a buffer while the local recovery process takes place. If this mechanism fails after a certain time, then the packet is discarded.

6.3.2 Appearance of packet replicas

Replicas of unicast packets are produced due to the lack of communication of the recovery processes between MAC and routing layers. Basically, undesired copies of packets take place when a retransmission at the routing layer is performed when a frame was successfully received but its corresponding ACK frame was lost. A general packet flow that generates the processing of copies by the routing protocol is depicted in Fig. 6.1, where the routing process of node A chooses B to forward a packet.

At step 1, the packet is encapsulated into a MAC frame with a sequence number X. The MAC layer tries to send this frame without success until the Retry $n-1$, in case the frame collides or contains errors. Thus, node A does not receive an ACK frame. At step 2, the MAC retry n is successfully received by node B, which sends the corresponding ACK frame to node A. In addition, node B checks if the frame is a duplicate of a previous one. Since it is not, the payload of the frame (the packet) is passed to the routing layer. At this point, the routing protocol of node B can try to forward the packet to destination or delivery it to upper layer if it is the destination. In step 3, let us suppose that the ACK frame sent by B was lost and A retries the frame sent until it reaches the maximum allowed number of attempts without success. Then in step 4, the MAC layer of node A pushes back the packet to the routing protocol because the transmission failed.

Since the routing protocol of A receives a packet from a failed MAC transmission, it starts the recovery process. Due to the first notification of failure, in step 5 node A decides to retry the packet forwarding to node B. Therefore, the MAC layer of A assigns a new sequence number Y to the packet because it starts a new frame transmission. Next at step 6, this new frame is successfully received by node B in the retry m. B sends the ACK frame to A and checks if the frame is a duplicate of a previous one. Due to the fact that this frame has a different sequence number from the used in step 1 (because it is a new frame), the MAC filtering cache does not detect any duplicate and delivers the packet to the routing protocol.

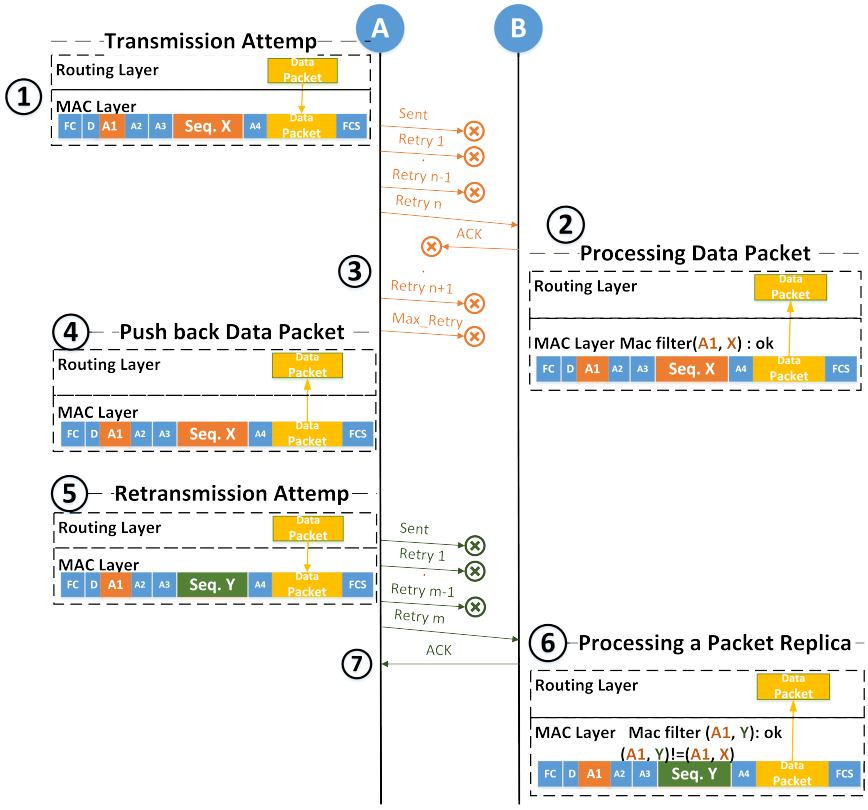


Figure 6.1: Packet copies in unicast transmissions. The routing local recovery process tries to forward the packet to the same node, but the MAC layer treats it as a new transmission attempt and assigns a new sequence number. Therefore, the duplicate of the packet can not be detected by the MAC layer.

Finally in step 7, the ACK is successfully received by node *A* and the 1-hop transmission of the packet is complete.

Notice that, the packet received by the routing protocol of node *B* at step 6 is a copy of the previous one processed at step 2. Hence, this new incoming copy should be discarded at routing level. However, the routing protocol is not aware about the presence of a duplicate packet and therefore node *B* will forward a copy of a packet already sent previously. Copies of packets can appear at any hop in the path to destination, and they will be transmitted through the rest of the path. Thus, the packet replicas could become a significant portion of the total traffic in multi-hop, unicast data transmissions of VANETs. This would affect the available bandwidth in the wireless channel to send new packets instead of unnecessary copies.

6.4 Mitigation of packet copies by caching

Mitigation mechanisms to deal with packet copies have not been studied for unicast transmissions, in which this phenomenon is not evident as we already explained in Sec. 6.3.2.

In this section, we propose straightforward techniques to mitigate the propagation of packet copies without damaging the performance of the VANET routing protocols.

One the one hand, the unintentional generation of duplicate packets explained in the previous section could increase the level of reliability to the end-to-end communication. On the other hand, the lack of a mechanism to control the forwarding of packet copies could lead to a high use of nodes' resources (e.g, buffer room, processing time) and the available bandwidth in the common wireless channel. In this section, we present two alternative ways to control the propagation of packet copies in a unicast multi-hop communication. The main idea is to implement a memory of the processed packets similar to the established in the IEEE 802.11 standard and amendments to detect incoming duplicate frames.

Failed Frames Cache at the MAC layer

In this first alternative to cope with unnecessary duplicate packets, all the modifications have been done at the MAC layer. To tackle the issue of unnecessary duplicate packets, the MAC layer records information of the payload (i.e., the packet) of every frame that failed to transmit after reaching the maximum number of retries and before returning the packet to the upper layer. The procedure implementation is detailed in Alg. 6.1.

OUTGOINGMAC-FILTER(P)

Require: A frame F, FAILED-CACHE
Ensure: Consistent SqN to F carrying copies of P .
1: $MAC_{nexHop} \leftarrow$ GETDESTMAC(F)
2: $P \leftarrow$ GETFRAMEPAYLOAD(F)
3: $ID_p \leftarrow$ GETPACKETID(P)
4: $SqN_{store} \leftarrow$ RETRIEVECACHE ($MAC_{nextHop}$, ID_p)
5: **if** $SqN_{store} = NULL$ **then**
6: SETSQNUMBER (F , NewSqN)
7: **else**
8: SETSQNUMBER (F , SqN_{store})
9: **return**

STOREINCACHE ($MAC_{nextHop}$, ID_p, NewSqN)

Require: $MAC_{nextHop}$, ID_p, NewSqN, FAILED-CACHE
Ensure: FAILED-CACHE with recent sent packets.
1: $SqN_{store} \leftarrow$ RETRIEVECACHE ($MAC_{nextHop}$, ID_p)
2: **if** $SqN_{store} = NULL$ **then**
3: **if** ISFULL (FAILED-CACHE) =*True* **then**
4: DELETELASTRECORD()
5: PUSH_FRONT($MAC_{nextHop}$, ID_p, NewSqN)
6: **return**

Algorithm 6.1: Detection of packets duplicate. MAC filtering alternative.

The Alg. 6.1 consists of two functions:

OUTGOINGMAC-FILTER searches a match of the new frame F in a cache of outgoing frames information previous to its transmission. A match (line 4) in the cache is based

on the MAC address of the next forwarding hop and the packet ID, obtained from the payload of the frame (line 3). If a match is not found (line 5), then the function will return a *NULL* value and a new sequence number is used to transmit the frame F. Otherwise, the search function will return the sequence number used previously to transmit that packet . This sequence number will be used again to retransmit the packet. In this way, since the new MAC transmission (second attempt onward at routing layer) of the packet at step 5 in Fig. 6.1 will use an unique MAC sequence number, the duplicate filter of the MAC layer in node *B* at step 6 will detect the duplicate frame and discard it. Hence, the routing layer will not process a duplicate of a packet. Also, notice that the task of OUTGOINGMAC-FILTERof Alg. 6.1 is not performed in the MAC retransmission process of the frames.

STOREINCACHE This procedure is triggered every time a MAC transmission fails and a packet is returned to the routing protocol. It saves the destination MAC (i.e., the next hop), the packet ID, and the sequence number used in the MAC transmission (line 5) if this information is not stored in the cache (line 2). The algorithm deletes the oldest entry if the cache is full (line 3).

Processed Packets Cache in the Routing protocol

Another way to deal with unnecessary copies of packets is to modify the operation of the routing protocol. This approach has the advantage of keeping the MAC layer without changes. Moreover, if some transmissions require redundancy which means forwarding of packet copies, then the MAC layer could not provide such services because it is not able to understand packet formats, or it would need to be in continuous communication with the routing layer. So, it is more efficient that a packet filter is managed by the routing layer. In this way, the routing layer can decide when to use or not certain filter. In our proposal, the routing protocol employs a filter, similar to the used at the MAC layer, to validate if an incoming packet is a duplicate of a previous one. This proposal also employs two procedures, one for filtering purposes and another to feed the cache of processed packets.

ROUTINGFILTER is a function that checks whether a packet is a duplicate or not. For that aim we propose two filters with different levels of discarding duplicate packets.

- *Based on the Packet ID and on the Previous hop ID.* In this filter, an incoming packet is searched in the ROUTING-CACHE by its ID and the address of the previous hop that forwarded the packet. (line 3 of ROUTINGFILTER function in Fig. 6.2). If both parameters match with a record in the cache (line 4), the packet is discarded and therefore it will not be routed again or pushed to upper layers. This filter requires that the ID of the previous hop will be transmitted as a field of the packet header, which means an additional overhead in the packet transmission (e.g., the IPv6 address of the previous node). It also needs that every node that forwards a packet has to update this information.
- *Based on the Packet ID* This is a more aggressive level of discarding duplicates of packets. This filter drops a packet based only on a match of the packet ID in the ROUTING-CACHE. Since this condition is simpler to meet than the above one, more packet copies will be discarded. Additionally, this filter does not need that packets carry information about the previous hop.

The aforementioned filters are applied to new incoming packets, which might sometimes be unnecessary packets. As we stated previously, some applications may

welcome some redundancy, so the first filter (i.e., base on packet ID and previous node ID) could be applied in those situations. Even more, if the channel is idle then the routing protocol could adapt among these filters and not to use any of them, because more packet copies in such situations could help to guarantee the delivery of packets.

STOREINCACHE This procedure feeds the cache of processed packets. It is triggered every time that a packet is forwarded by the routing protocol to the next hop. Alg. 6.2 shows the function that stores the ID of the previous node and the ID of the packet for the first filter. As in the MAC procedure, the cache saves the newest entry at the top of the list (line 5) if the entry is not already in the cache (line 2).

ROUTINGFILTER(P)

Require: A packet P, ROUTING-CACHE
Ensure: Drop duplicate packets .
1: $ID_{prevHop} \leftarrow$ GETPREDECESSOR(P)
2: $ID_p \leftarrow$ GETPACKETID(P)
3: Found$\leftarrow$ ISINCACHE($ID_{prevHop}$, ID_p)
4: **if** Found $= True$ **then**
5: DROP(P)
6: **return**

STOREINCACHE ($ID_{prevHop}$, ID_p)

Require: $ID_{prevHop}$, ID_p, ROUTING-CACHE
Ensure: ROUTING-CACHE with recent sent packets.
1: Found$\leftarrow$ ISINCACHE($ID_{prevHop}$, ID_p)
2: **if** Found $= True$ **then**
3: **if** ISFULL (ROUTING-CACHE) $=True$ **then**
4: DELETELASTRECORD()
5: PUSH_FRONT($ID_{prevHop}$, ID_p)
6: **return**

Algorithm 6.2: Detection of duplicate packets. Routing layer alternative.

Computing a Packet Identifier

The key aspects of our proposals to mitigate the event of packet copies are:

1. To guarantee a unique identifier for each packet.
2. The ID packet generator should be fast to introduce the least possible delay.

Since there is not such packet ID considered by network layer, standards or routing protocols in the format of the packets, we propose to use non-cryptographic hash functions on the packet's payload to get a unique ID. This kind of hash functions meet the requirements aforesaid. They are designed for looking up tasks like the proposed here to be done by our filters in our proposal. Non-cryptographic hash functions are resistant from collisions (i.e., two different inputs that produce the same has value). Moreover, since they are not designed for security purposes, non-cryptographic hash functions are fast and can supply different hash length values depending on the processor and time constraints. Examples of these functions are Murmur [6], cityHash [41] and xx-hash [28]. If no hash function could be

implemented in a node, Check Redundancy Codes (CRC) like the used in the MAC layer to detect transmission errors in the data could be obtained from the packet payload as packet identifier. Hash functions are faster than CRC, and they should be preferred to avoid extra delays.

It is worth noting that each node can choose a different function to compute a packet ID because its use is completely local. Thus, the packet ID does not need to be carried in the packet, avoiding adding overhead to the packet transmission.

6.5 Performance Evaluation

The cache of failed frames at MAC layer and the duplicate packet filter at routing level described in the previous section, will provide the same results because both mechanisms operate very equivalently although at different layers. To carry out our performance evaluation we preferred to implement the cache of processed packets in the routing layer to avoid modifying the operation of the MAC layer. More important, a routing cache keeps independent the functions of both MAC and routing layers.

We used the simulation scenario for a reporting service and the setting described in Sec.3.4 from chapter 3 to analyze the impact of our two duplicate packet filters (PF) at routing layer (i.e., based on packet ID and node ID or based only on packet ID). We used MMMR [113] to evaluate the behavior of our proposal and incorporate the coherent, automatic address resolution proposed in the previous chapter. We performed the simulations using two mechanisms to adapt the size of the contention window (CW): the default included in the IEEE 802.11p MAC specifications and an alternative proposed in [8] to adapt the CW in a smoother way especially designed for VANETs to improve the congestion control performance of the network. Henceforth, we refer to these two CW mechanisms as DCW and CCCW, respectively, in the notation of figures and tables. We compared the impact of the routing filter on six different metrics. These metrics are the packet delivery ratio (PDR), the average end-to-end delay, the average number of hops (previously explained in Ch. 3 Sec. 3.5) and three more metrics that are interesting for this study:

- *End-to-end repeated packets.* This is the percentage of duplicate packets that arrive to the AP with respect to the total number of different received packets. This metric represents the percentage of the routing effort that does not provide utility to the end-to-end communication.
- *Percentage of idle time.* It is the average of the idle time sensed by a node measured in 1 second. A node senses the channel idle when it is not transmitting nor receiving a packet and given that the interference level is below the antenna sensitivity. Notice that a higher idle time sensed by the nodes leads to more bandwidth available in the channel to transmit more information. An estimation of the available bandwidth derived from the idle time measure could be done using the model proposed by [87] or [112].
- *Average number of collisions.* It is the ratio between the total number of collisions in the network and the total number of received packets excluding the reception of repeated packets. This metric is related to the previous one and it provides information of the interference in the communications. We decided to include this metric because it was used in [8] to evaluate the advantage of using other CW mechanism in order to decrease collisions and improve the availability of the channel.

Following the three steps procedure described in Sec. 3.5 from chapter 3 we have to analyze the results for each density separately because there is a significant three-way interaction (**Step 1**) with a p-value = 0.001 (Wilk's Λ = 0.088 and F(24,32)=3.16).

Table 6.1 shows the results of the further interaction test between *CW* and *PF* for the MMMR protocol (**Step 1**). First, we perform the so-called "all together" test in which we evaluate the interaction between the two factors considering the correlation among the six performance metrics. If the p-value < 0.05 for this test, we further carry separate interaction tests for each metric. In table 6.1 we can see that for the medium and the high density scenarios, the "all together" test have a p-value under the threshold, so independent interaction test per metric need to be performed in those cases. The results of the interaction tests for each metric indicate that the seek of performance differences in the routing filtering for the average number of hops, percentage of idle time and the collisions ratio need to further differentiate the CW mechanism used during the simulation in the high density scenario (250 vehicles) because p-value < 0.05 for these metrics. There is also this significant interaction (i.e., $PF \times CW$) for the evaluation of collisions ratio in the scenario of 150 nodes.

Number of vehicles	Performance Metric	Wilk's Λ	F (2,8)	p-Value
100	All metrics together*	0.743	0.346	0.971
150	All metrics together*	0.115	4.223	0.001
	Packet Delivery Ratio	0.471	4.488	0.050
	Average delay	0.869	0.603	0.570
	Average no. of hops	0.628	2.367	0.156
	% of repeated packets	0.457	4.747	0.05
	% of idle time	0.588	2.808	0.119
	Collisions ratio	**0.340**	**7.772**	**0.013**
250	All metrics together*	0.110	4.373	0.001
	Packet Delivery Ratio	0.934	0.284	0.760
	Average delay	0.549	3.285	0.091
	Average no. of hops	**0.300**	**9.333**	**0.008**
	% of repeated packets	0.896	0.463	0.646
	% of idle time	**0.285**	**10.01**	**0.007**
	Collisions ratio	**0.100**	**36.16**	**0.0005**

Table 6.1: MANOVA [60] results for interaction test between packet filtering (*PF*) and CW mechanism for MMMR protocol. *Degrees of freedom in F statistic are 12 and 26. If there is a significant interaction (p-value<0.05) in the "All metrics together" test, then interaction tests per metric need to be performed.

The test results to determine if there are differences in the performance metrics, according to **Step** 2 of the statistical procedure of Ch. 3, are shown in Table. 6.2. For this analysis, the data of the metrics were grouped according to the results of the interaction tests analyzed previously. This means that in most of the cases, we performed only one test per metric in each density without differentiate the CW mechanism employed during the simulation. We labeled these cases as "together" in the CW column. All the p-values in this table are lower than 0.05, excluding the collisions ratio in the low density scenario (100 vehicles). Hence,

the use of different packet filters (i.e., PF0, PF1, PF2) produce a statistical significant change among them in all performance metrics and therefore they require pairwise comparisons to analyze the differences. The only exception for this additional test is the aforementioned average number of collisions in the low density scenario for which there is not difference among the packet filtering techniques.

Metric	**Number of vehicles**	**CW**	**Wilk's Λ**	**F (2,27)**	***p* Value**
Packet Delivery Ratio	100	together	0.223	13.903	0.002
	150	together	0.223	13.941	0.002
	250	together	0.049	77.438	0.0005
Average delay	100	together	0.080	46.031	0.0005
	150	together	0.056	67.452	0.0005
	250	together	0.008	498.754	0.0005
Average number of hops	100	together	0.029	131.68	0.0005
	150	together	0.019	208.410	0.0005
	250	DCW	0.010	403.034	0.0005
	250	CCCW	0.016	250.362	0.0005
% of repeated packets	100	together	0.020	200.22	0.0005
	150	together	0.007	531.151	0.0005
	250	together	0.004	1033.26	0.0005
% of idle time	100	together	0.055	69.079	0.0005
	150	together	0.025	158.304	0.0005
	250	Default	0.027	145.73	0.0005
	250	CCCW	0.041	92.951	0.0005
Collisions ratio	**100**	**together**	**0.667**	**2.001**	**0.197**
	150	Default	0.104	34.30	0.0005
	150	CCCW	0.386	6.375	0.022
	250	Default	0.036	107.395	0.0005
	250	CCCW	0.048	79.478	0.0005

Table 6.2: MANOVA [60] results of testing difference in performance metrics among routing filtering techniques for MMMR protocol. There is an significant difference when p-value $<$ 0.05. “Together" means that to apply the test it is not needed to differentiate between CW mechanisms.

Table. 6.3 shows the results of the pairwise comparisons among packet filters (i.e., (PF0, PF1), (PF0, PF2), (PF1, PF2)) following the **Step 3** of the analysis. The is table only shows the cases in which the differences between packet filters are not statistical significant for a particular metric (i.e., p-value$\geq$0.05). So, the average values of the metrics are very similar and can be considered statistically the same. The rest of the results of the pairwise comparison tests, e.g. average delay, average number of hops and % of idle time comparisons obtained p-values $<$ 0.05 and we do not include them in the Table. 6.3. Those p-values indicate that choice of the packet filtering technique (i.e., PF0, PF1 or PF2) has an impact in the values of the metrics. The comparison of average values of the six performance

Metric	Number of vehicles	CW	Pairwise	*p* Value
Packet Delivery Ratio	100	together	(PF0, PF1)	0.099
	100	together	(PF1, PF2)	0.178
	150	together	(PF0, PF1)	0.267
	250	together	(PF0, PF1)	0.351
% of repeated packets	100	together	(PF0, PF1)	0.063
	150	together	(PF0, PF1)	0.078
Collisions ratio	150	CCCW	(PF0, PF1)	0.100

Table 6.3: Pairwise comparison of the performance metrics in which there is a difference among routing filtering techniques for MMMR protocol (i.e., p-value < 0.05 in Table 6.2). The table only shows the results for metrics and pairs of packet filters with absence of statistical significant differences (i.e., p-value ≥ 0.05). “Together" means that to apply the test it is not needed to differentiate between CW mechanisms.

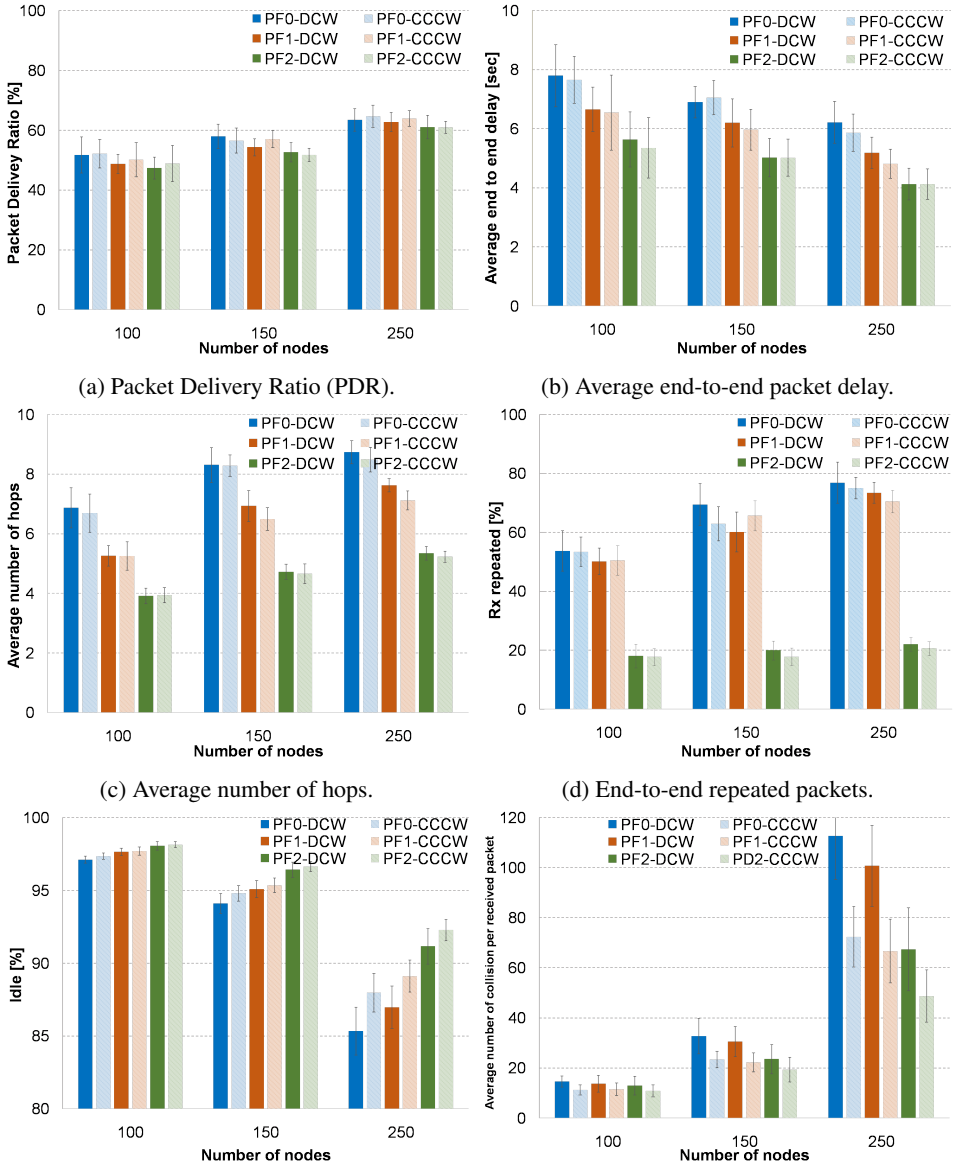
metrics for the MMMR routing protocol, are depicted in Fig 6.2. Considering the previous statistical analysis we continue analyzing the behavior of packet filters in the metrics.

As it can be seen from Fig. 6.2a, when the routing filter based on packet ID (PF2) is applied, the PDR slightly decreases for every density compared to the default operation (PF0). In fact, this difference is never higher than 5% in the worst case, which is the scenario with 150 vehicles. The PDR decrements because there are fewer chances that a packet arrives at destination. This is due to the fact that each packet has a lower number of copies in the network thanks to the routing filters. However, the PDR is the same between PF0 and PF1 (see the first PDR row in table 6.3). The reason is that PF1 (i.e. based on the packet ID and on the ID of the previous node) is less severe than PF2 for discarding packets.

Fig. 6.2b shows that the delay decreases when the routing protocol uses a more severe filter to decrease the chance of duplicate packets. The reason for this is that links between nodes are more available to transmit packets. Those links have a higher percentage of idle time because already received packets are not forwarded by the routing protocol in every node. From Fig. 6.2c, we can realize that applying a filter to detect duplicates of packets in each node, the number forwarding operations per each packet decreases. As a consequence, the hop count decreases as well because nodes do not route duplicate packets. In both delay and hop count metrics, the packet filter is more severe, the reduction of the value in these metric are higher.

From Fig. 6.2d the percentage of repeated packets for PF1 goes from 40% in the low vehicle density to almost 70% in high vehicle scenario. The number of end-to-end repeated packets is still high even when the routing filter PF1 is used and it is moderate for PF2. Moreover, in the low and intermediate density the difference between PF1 and the default operation of not using a filter (PF0) is not significant (see % of repeated packets in table 6.3 with p-value$\geq$0.05). Nevertheless, the soft filter PF1 decreases the end-to-end copies in high density scenario because of the higher number of transmissions and generation of duplicates present in this case. For the case of PF2, i.e., the most severe filter to detect duplicates, the percentage of duplicate packets decreases considerably to a value around 20% in all the

densities. The hard filter PF2 discards much more duplicate packets than the soft filter PF1. Nevertheless, PF2 does not discard all the packet copies because the routing filters avoid the forwarding of packet copies but it cannot prevent the generation of its own duplicates (see



(a) Packet Delivery Ratio (PDR).

(b) Average end-to-end packet delay.

(c) Average number of hops.

(d) End-to-end repeated packets.

(e) Percentage idle time.

(f) Average number of collisions.

Figure 6.2: Performance evaluation of our proposals for the geographical routing protocol MMMR. [113]

Fig. 6.1). Generation of duplicates can occur at any hop including the last hop to destination, therefore copies created in this last hop will always be received by the destination because the proposed filters works at the reception of packets in the forwarding process and does not in the routing of the own packets. In addition, it has to be considered that the buffer cache of the filters is limited and some packet copies could be forwarded because their filtering records were deleted.

Regarding the average number of collisions per received packet depicted in Fig. 6.2f, the routing filters alleviate the occurrence of collisions, specially in high density scenarios. The nodes forward a lower number of packets (copies), which decreases the collision probability. In the low density scenario the use of filters barely modifies the ratio of collisions because the data traffic load is not considerable. On the other hand, in the medium density scenario, both filters reduce this metric when the default CW mechanism was employed. However, the collisions ratio does not change between PF0 and PF1 for the the CW designed for congestion control (CCCW). This fact reveals that the CCCW mechanism adapts better under higher traffic load. Notice that a low number of the collisions will help to avoid failed MAC transmissions that cause the creation of packet copies and consequently the number of duplicate packets will decrease. The reduction of duplicate packets and collisions is the reason for the higher percentage of idle time (see Fig. 6.2e) in all densities when the duplicate packet filters are used compared to the default operation.

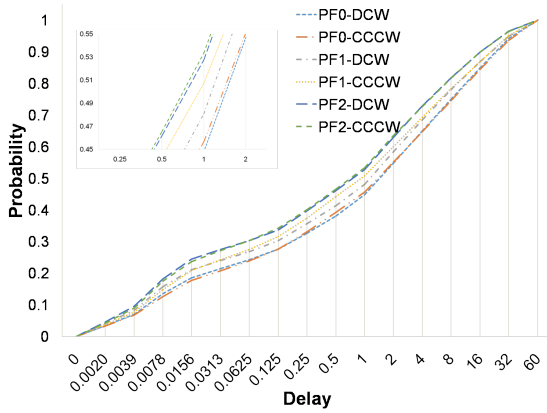
To provide a better picture on how our proposals of routing filters modify the performance results, we have included in Fig. 6.3 the cumulative distributions function (CDF) of delay, number of hops and percentage of idle time in a node.

The shape of the three CDFs are roughly the same regardless the packet filtering mechanism and the CW mechanism used in the scenario. The main difference is the value for which a certain probability is reached. For instance, in Fig. 6.3a when PF2 (i.e., filter based only on the packet ID) is used the 50% of the packets are received with a delay lower than 1 s, while when PF0 is used (i.e., no filter is applied) the same percentage of received packets have a delay of 2 s. The same behavior takes place in the distribution of the number of hops (including packet copies) in Fig. 6.3b in which PF2 has the 90% of the packets that used 8 hops or fewer to reach destination and only 74% of the packets without any packet filtering took the same hop count or less. The previous differences make that nodes with PF2 are more idle than the default operation. Specifically, 25% more nodes sense the channel idle 94% of the time or higher.

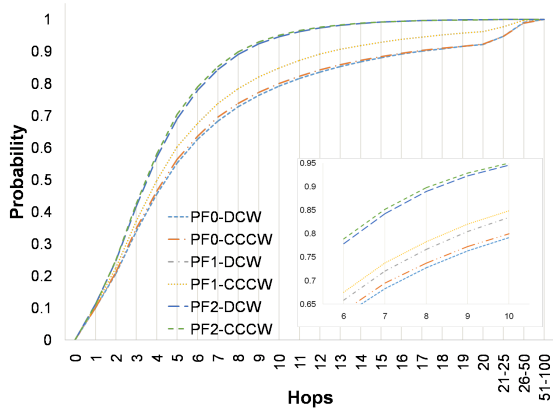
Study of Contention Windows mechanism

Despite of the fact that our intention of employing two CW mechanism, i.e., the default CW (DCW) and a CW that adapts in a better way to control the congestion in the channel [8] (CCCW), was to check if packet duplicates took place in the two mechanisms, we have included in this section a brief comparison between these two options.

The reader can realize that CCCW provides the same or better results than the default CW proposed in the IEEE 802.11 standard by checking Fig. 6.2. Particularly, we observe an appreciable reduction of collisions per received packet ratio in both routing protocols and especially in high density areas (see Fig. 6.2f). This is a similar result than the obtained in [8] in which the CCCW was presented.



(a) CDF of End-to-end delay.

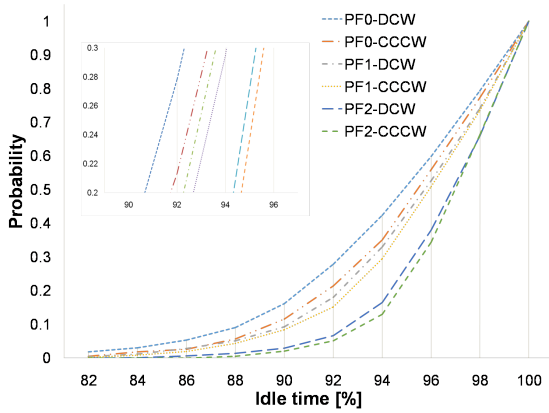


(b) CDF of number of hops.

Another important result is that the use of CCCW makes that nodes sense the channel idle more time than with the default CW, specially at high vehicle density, as it can be seen in Fig. 6.2e. This allows that the network be able to support a higher data traffic than using DCW.

6.6 Conclusions

In this chapter we have explained how unintentional copies of packets take place in unicast communication due to the recovery mechanisms employed by the routing protocols that seek to improve the reliability when there is a link breakage. We have proposed the implementation of two different filters in the routing protocol to prevent the unexpected propagation of duplicate packets in a VANET. The first proposed filter is based on the tuple of packet ID and node ID (called PF1) and the other filter only uses the packet ID (called PF2). Therefore, the second filter is more strict and discards more packet copies than PF1. Nodes do not need to manage any packet ID of global validity because its use is only local. Hence, the



(c) CDF of % of idle time in a node.

Figure 6.3: Cumulative Distribution Functions (CDF) for 150-vehicle scenario with MMMR routing protocol. [113]

implementation of the second filter does not introduce any overhead to the communication. The performance evaluation shows that our filtering proposals improve the overall network performance compared to the free propagation of the duplicate packets (default operation PF0). These improvements include the number of routing operations (i.e., the number of hops) and the percentage of idle time. These advantages come at a minimum packet delivery ratio decrement.

7. Heuristic Methods in Geographical routing protocols

This chapter proposes different local-search heuristics to improve the performance of geographical routing protocols in VANET networks, which typically rely the selection of the next node only on the best candidate. The presented algorithms are based on modifications of the well-known Simulated Annealing and Tabu meta-heuristics [135]. We have divided our heuristics according to their operation in forwarding and recovery algorithms. Moreover, we combine them in a generic Geographical Heuristic Routing (GHR) protocol to improve the overall performance of the system. We also propose the 2-hops Geographical Anycast Routing (2hGAR) protocol that uses information of nodes located two hops away to take the forwarding decisions. Simulation results of our routing proposals show that some features of GHR and 2hGAR behave better than other proposals depending on the vehicle densities. In all the cases, the use of Tabu-search shows a significant increment in the packet delivery ratio.

7.1 Introduction

VANETs face particular challenges such as fast topology changes, low link lifetime or a potentially high number of nodes taking part in the network, among others. Particularly, the two former issues have encouraged researches to propose geographical routing protocols for VANETs that make their routing decision based only on local information and that do not need to construct end-to-end paths.

Geographical routing protocols have evolved from considering only the geographical distance between nodes in their forwarding criteria until including other additional metrics like speed and direction of the vehicle. However, the routing decision of the next hop is based mainly on the *best next forwarding hop* according to a given metric. The selection of the *best node* is the predominant criterion in the routing decision among most of geographical routing protocols. On the other hand, the successful hop-by-hop forwarding strategy of geographical VANET protocols can be seen as an application of a well known family of heuristic protocols called "local-search algorithms", which are widely and successfully used in discrete optimization. In this chapter we review Tabu-search and Simulated Annealing meta-heuristic designed to improve the results obtained with local-search algorithms in discrete optimization problems.

These techniques could also be applied in geographical routing protocols for VANETs. The contribution of this chapter is the adaption of these general techniques for operating in the forwarding and recovery phases of delay tolerant geographical VANET routing protocols based on local information to improve their delivery performance.

Moreover, we design a generic Geographical Heuristic Routing (GHR) protocol that combines all the proposed adaptations. We also propose the 2-hops Geographical Anycast Routing (2hGAR) protocol, which is inspired in the 2-opt heuristic used in the Traveling Salesman Problem [88] to improve the results obtained by local-search. 2hGAR uses information of nodes located two hops away from the current node, to take the forwarding decision and is specifically designed for anycast applications. We found that the best settings for GHR and 2hGAR depends of the vehicle densities.

The rest of the chapter is organized as follows: Sec. 7.2 introduces the notation used along this chapter. This section also presents the motivation of this work by showing how a well known protocol like GPSR [62] is a particular application of the greedy heuristic optimization method. In addition, Sec. 7.2 overviews general heuristic optimization techniques applied in discrete optimization. Then, Sec. 7.3 summarizes some other works that uses optimization techniques in wireless networks. Next, Sec. 7.4 presents adaptations of the reviewed heuristics for their application in geographical routing protocols based on local-search. After that, Sec. 7.5 describes the algorithms of our two heuristic protocol proposals. Sec. 7.6 is devoted to the performance evaluation of our contributions. Finally, conclusions are drawn in Section 7.7.

7.2 Background

In this section, we give definitions and notation used along this chapter. Also, we motivate the study of heuristic methods in the development of geographical routing protocols. We conclude this section reviewing two heuristics proposed in discrete optimization field to solve NP-hard problems.

7.2.1 Notation and definitions

We introduce some notation for the elements that define a local-search heuristic. We are going to use them in the following sections. The goal is to keep the notation succinct and easy to understand without loosing rigor in our definitions.

- First of all, a discrete problem is characterized by a set of states $s \in States$.
- The objective function f, that we want to minimize. Let $f(s)$ denotes the value of the objective function in state s.
- N(s) defines the neighborhood of the current node s. In a local search, the next state from s is a neighbor of s.
- L(N(s), s) denotes the set of legal neighbors of s. Legal neighbors are the states that meet some requirements to be considered as a possible next state towards the solution.
- Finally, S(L(N(s), s), s) is the notation for the selection function applied on the legal set of neighbors of node s. The greedy algorithm selects the neighbor from the legal neighbors that minimizes the objective function. That neighbor is the best neighbor.

7.2.2 Motivation

In this section we start explaining the motivation to look for new enhanced heuristics for geographic (position-based) routing protocols in VANETs by describing the logic of one of these protocols. Greedy Perimeter Stateless Routing (GPSR) [62] is one of the most extended proposals of geographical routing. The protocol uses greedy forwarding, in which the neighbor geographically closest to the packet's destination is chosen as the packet's next hop. Following the notation introduced previously GPSR operates as follows:

- In GPSR, a state refers to a node of the network. We refer as the current state s to the node currently holding a packet.
- The objective function in the case of GPSR is the distance to destination. The goal is to minimize such distance.
- A neighbor of s in GPSR is any node $v \in \mathrm{N}(s)$ that is within the maximum transmission range from s.
- A neighbor $v \in \mathrm{N}(s)$ is a legal neighbor $v \in \mathrm{L}(\mathrm{N}(s), s)$ in GPSR if v is closer to the destination's position than the current node s.
- Finally, GPSR applies a greedy approach and selects the neighbor geographically closest to the packet's destination from the list of legal neighbors, this is $\mathrm{S}(\mathrm{L}(\mathrm{N}(s), s), s)$.

The wide acceptance of GPSR motivated that other researchers proposed new greedy protocols to improve the GPSR performance. The reader can realize that any geographical routing protocol that works with a local-search scheme follows the exact process described for GPSR, but modifying the requirements and criteria employed by each function at the different steps (i.e., determination of legal neighbors through function L and selection of next hop by function S).

Notice that a key step in local-search heuristic is the way a node selects one of the legal neighbors to be the next hop. However, all the position-based protocols surveyed in the state of the art ([22], [68] [100] [29] and [72]) follow a greedy approach in the selection step.

Algorithm 7.1 presents the general strategy to select the next hop, which summarizes the process of selecting the next hop (function S) over the suitable candidates obtained through function L among the neighbors of s (stated as $\mathrm{N}(s)$). When there is not a suitable next hop, routing protocols use an alternative recovery procedure, being one of the most popular the *carry and forwarding* mechanism.

LOCALSEARCH(s, P)

Require: initial state s := source node, a packet P
Ensure: Forward packet P to the closest s to destination d.
1: **while** $s \neq d \wedge L(N(s), s) \neq \emptyset$ **do**
2: s := S(L(N(s),s),s)
3: **return**

Algorithm 7.1: Summarized iterative process Local Search algorithm according to the notation introduced in Sec. 7.2.1.

7.2.3 Meta-Heuristics overview

Geographical protocols that take routing decisions at each hop or junction use local information to select the "best" next hop. Nonetheless, it is well-known from heuristic optimization that such greedy approach could lead to get stuck in a non optimal point known as *local-minimum* (e.g. a node that is not the final destination) in which no further improvement is possible.

In this section we introduce the general ideas of a well known set of algorithms employed to improve the "local search", covered by the name of "metaheuristics" [135]. Based on these algorithms, in Sec. 7.4 we propose some modifications for the geographical routing protocols in VANETs.

Tabu Search

Tabu can be combined with any other heuristic. The key idea is to keep track of the states already visited and forbid these movements, which are considered "Tabu". Typical implementations of Tabu include to forbid:

1. The last k states already visited by the packet.
2. The states visited more than k times.

Optionally, the Tabu list could be erased periodically. Using Tabu technique we are able of avoiding loops and achieve optimality more frequently. Alg. 7.2 shows the pseudo-code of a general local search heuristic that includes a Tabu list. Note that the function L, that determines which neighbors are good candidates for being the next state, now includes a new parameter: the Tabu list τ. Every new hop is included in the list of forbidden hops if they are not good solutions.

TABULOCALSEARCH(s, τ)

Require: initial state $s := s_{init}$, Tabu list τ
Ensure: Do not select a node $s \in \tau$.
1: **while** NOTSOLUTION(s) **do**
2: τ := concat(τ, s)
3: s := S(L(N(s),τ, s), s)
4: **return**

Algorithm 7.2: Tabu list implementation in a local search algorithm. Possible legal neighbors of s are also compared against the forbidden list τ of previous visited states.

Simulated Annealing (SA)

Simulated Annealing is a meta-heuristic inspired on a physical principle of the formation of a glass. In this process, the materials are heated and then cooled slowly. If the cooling occurs rapidly, the crystal is imperfect. Then the temperature of crystal cooling T must be tuned to decrease gradually and get an optimal crystal.

The general strategy of Simulated Annealing (SA) consists of allowing at each iteration with a given probability p to go to a state s that worsens the value of the objective function. This probability decreases with the number of iterations. The analogy between probability and crystal temperature of cooling, tells us how fast and greedy an algorithm is being to reach a

solution and to obtain a crystal, respectively. Accepting worse solutions allows us a more extensive search among the set of states s to reach the optimal solution.

Alg. 7.3 shows the pseudo code of Simulated Annealing using the notation of Sec. 7.2.1 and defining T as the temperature (greediness parameter), the reduction factor $r \in [0, 1]$, p as the probability of worsen, f(s) as the objective function value in the current state and f(s') as the value in the next state.

SIMULATEDANNEALING(s_{init}, T_{init})

Require: initial state $s := s_{init}$ and $T := T_{init}$
Ensure: Search among more states to find a state solution s

```
 1: while NOTSOLUTION(s) do
 2:     for k=0; k < loops; k++ do
 3:         s' := S(L(N(s), s), s) (At random)
 4:         Δ = f(s') - f(s)
 5:         if d < 0 then
 6:             s := s'
 7:         else
 8:             random := U[0, 1]
 9:             p = e^(-Δ/T)
10:             if p < random then
11:                 s := s'
12:     T := r T
13: return
```

Algorithm 7.3: Procedure of Simulated Annealing (SA).

Simulated Annealing selects between one of the possible legal neighbors at random and computes the difference in objective function value, called d, between the next hop and the current state. Then, a negative value of d means that we are approaching the target; otherwise we are worsening and we only move with probability $p = e^{-d/T}$. The probability formula relates the greediness parameter T and the worsen probability p. It also tells us that the higher the difference d, the less probable is to move to the next hop. In addition, the greater T (temperature) the most probable is to move to a worse state. Because we are reducing T every k iterations, we have that as the algorithm progresses, the probability of worsening becomes lower each time and the algorithm behaves in a greedier way each time.

For Simulated Annealing it is highly recommended to test different values of T to determine which greediness values work best. A bad decision of the T value could lead to a excessive number of iterations to reach the solution. In the particular case of routing protocols that means that a bad selection of the greediness factor T can produce a potentially higher number of packet jumps between nodes before reaching the destination point. This is very important because every forwarding operation has associated a packet loss probability. A higher number of packet jumps could increase the packet loss ratio considerably.

This meta-heuristic could be implemented in geographical routing protocols, if a suitable metric is used as the greediness parameter T, as we will do in next Sec. 7.4.

7.3 Related work

Currently, there are several proposals to optimize the routing process in wireless ad-hoc networks. In [99] the authors propose a new routing protocol called Tabu Search base Routing Protocol (TSRP) which introduces a mechanism of the meta-heuristic Tabu Search. The idea is to record all nodes that forwarded the data from the sensor that has sensed the event to the sink. Thus, when a sensor senses an event, the choice of the next sensor is based on maximizing the cost function which considers the energy and the visibility of this sensor compared to the sink avoiding previous visited nodes. Simulation results showed that TSRP prolongs the network lifetime longer than the protocols Gossiping [5] and MFR [107].

The authors in [58] propose a routing optimization algorithm to minimize the route cost from a source to a destination within a reasonable time. The proposed algorithm is designed by using a Tabu search mechanism. The Tabu search algorithm carries out two neighborhood generating operations in order to determine an optimal path and minimize algorithm execution time. The proposed Tabu search algorithm was compared to genetic algorithm and the Simulated Annealing, in terms of routing cost and algorithm execution time. The comparison results show that the proposed Tabu search algorithm outperforms the other algorithms and that it is suitable for routing optimization problem.

The optimal parameter setting of the optimized link state routing (OLSR) was analyzed in [109], also as a optimization of routing in ad hoc networks. The authors use a series of representative meta-heuristic algorithms (particle swarm optimization, differential evolution, genetic algorithm, and Simulated Annealing) in order to find automatically optimal configurations of this routing protocol. This study used realistic VANET scenarios based on the city of Malaga. Simulations show better results than that standard OLSR configuration.

In [2] the authors focus their study on the fact that the measured data used in a VANET will always be corrupted by noise and other factors. The authors used Simulated Annealing optimization technique for finding lower bounds on how much improvement is possible given the inaccuracies in the measurements. They use two algorithms previously proposed (VLOCI [4] and VLOCI2 [3]) to improve the GPS coordinates.

7.4 Heuristics for Geographical routing protocols

In this section, we present some heuristics for geographical routing protocols that incorporate the meta-heuristics explained in the previous section. First, we propose an implementation of a Tabu list. After that, we propose new algorithms classified according to their operation in forwarding and backwarding (or recovery). That is, the forwarding heuristics are the ones that are used when an improvement is possible. In other words, these heuristics are used when the set of legal nodes is not empty. On the other hand, we propose a modification of the *carry and forwarding* recovery mechanism when no better next hop is found and we do not want to wait for too long until one node that improves the current solution is found.

7.4.1 Tabu Search implementation

As we explained in Sec. 7.2.3, Tabu search needs to update a list of forbidden states. In geographical routing protocols, a forbidden state means a node that must not be considered to forward a packet.

We propose to incorporate the Tabu list of forbidden nodes in the routing header of each packet. Hence, there will be a trade-off between the length of the list that provides information about the path followed by a packet and the overhead introduced by the list.

ROUTE(P)

Require: a packet P, set of neighbors N(s) of the current node s
Ensure: not to forward to nodes in τ.
1: τ:= GETTABULIST(P)
2: **for all** $n \in$ N(s) **do**
3: **if** $n \notin \tau$ **then**
4: ADDTO(N$_\tau$(s), n)
5: $s_{\text{next\_hop}}$:=S(L((N$_\tau$(s)), s), s)
6: UPDATETABULIST (τ)
7: FORWARD($P, s_{\text{next\_hop}}$)
8: **return**

UPDATETABULIST (τ, s, P)

Require: Tabu list τ,ID of current node s and packet P.
Ensure: updated τ in P.
1: **if** ISFULL (τ) =*True* **then**
2: DELETELASTID(τ)
3: WRITE_FRONT(τ, s)
4: PUTTABULIST(P, τ)
5: **return**

Algorithm 7.4: Tabu search in geographical routing algorithm.

The routing procedure of a node that implements Tabu is depicted in Alg. 7.4. The algorithm obtains the previous nodes visited by the packet (line 1) and discards them as forwarding nodes (loop from line 2 to 6). The selection procedure in the protocol remains identical. After that, the routing protocol selects the next hop (line 5). A key step takes place when the current node updates the Tabu list by adding itself to that list (line 6). This update task includes to delete the oldest entry and puts the ID of the current node at the beginning of the list.

This routing procedure repeats at every hop and because of that, different nodes are evaluated to avoid getting stuck in local minimums and loops. Notice that the Tabu list of a packet could be seen as a dynamic filter to determine legal nodes in addition to the static requirements established by the routing protocol.

7.4.2 Forwarding. (There are candidates to be next forwarding nodes)

In this section we include some possible modifications to choose the next forwarding node when there is a set of legal neighbors. i.e., nodes that meet requirements like high lifetime, low SINR, etc.

Selecting the first legal neighbor from the list

The selection of the best neighbor has the drawback of having to traverse the entire list every time the protocol has to forward a packet. An alternative to greedy approach is to select the first neighbor that meets the forwarding requirements (" a legal neighbor") from an ordered list of neighbors.

We propose to order the Table of neighbors according to the age of the information. Every time a node receives a 1-hop signaling message from a neighbor, the node updates the corresponding entry and puts it on the top of the list. Hence, when the node visits the list and finds the first legal neighbor it will forward the packet to the most recent updated neighbor.

The main idea is that nodes in the top of the list of the neighbors will be more likely to be in a reliable communication range. Additionally, the computational cost could be considerably reduced when a node experiments a high data traffic rate.

Random selection

This heuristic selects between one of the legal neighbors at random. If n is the number of possible neighbors, the probability of selecting a neighbor is $p = 1/n$.

In comparison to other heuristics, randomizing the selection makes the algorithm more balanced. For instance, sending a packet to the legal neighbor geographically closest to the packet's destination has the drawback that transmission might fail because the signal might be poor. Conversely, sending it to the closest neighbor has the disadvantage that packets will need more hops to reach destination.

This heuristic could be improved following a different criterion. For instance a greedy choice (best neighbor), with a probability $p \in (0,1)$ and randomize the selection with probability $1-p$. The next heuristic implements this idea taking advantage of the Simulated Annealing (SA) heuristic explained in the previous section.

Modified Simulated Annealing

In this heuristic, we made some modifications to the traditional Simulated Annealing described in Sec. 7.2.3. The algorithm depicted in Alg. 7.5 works as follows:

- Instead of selecting a neighbor at random, the protocol selects the best node from the legal neighbors (line 3). *Best node* can mean the closest node to destination as in GPSR, or the one with the highest value of a combination of metrics like in traffic aware routing protocols.
- As a first step, We propose to use the distance from a node n to destination, namely $D(n)$, as the metric to evaluate the improvement on selecting that node. If the selected neighbor is not closer to destination than the current node s, then the routing protocol applies a recovery method such as *carry and forwarding* (line 6).
- The distance of the current node to destination $D(s)$ plays the role of temperature T (greediness factor) in Simulated Annealing. The "temperature" in our proposal (i.e., $D(s)$) is not reduced every k iterations. Instead, it tailors to the geographical position of each node. As a packet jumps, it should be approaching to the objective node, therefore $D(s)$ should decrease. Hence, the effect produced by using $D(s)$ as T will be the same as with traditional Simulated Annealing.

MODIFIEDSIMULATEDANNEALING(P, s_{init})

Require: initial state $s := s_{init}$, a packet P
Ensure: Forward packet P to the closest s to destination d.

```
1:  while s ≠ d ∧ L(N(s),s) ≠ ∅ do
2:      s' := S(L(N(s),s),s) (select best node)
3:      d_s = D(s)
4:      d_{s'-s} = D(s')-d_s
5:      if d_{s'-s} > 0 then
6:          RECOVERY PROCESS (s, s')
7:      else
8:          p = e^{α.(d_{s'-s}/d_s)}
9:          random := U[0, 1]
10:         if p < random then
11:             s := s' (best node)
12:         else
13:             s := S(L(N(s),s),s) (select at random)
14:     FORWARD(P, s)
15: return
```

Algorithm 7.5: Modified Simulated Annealing in the forwarding operation of a geographical routing protocol.

- If the best node is closer to destination (i.e., $d_{s'-s} < 0$), the routing protocol operates according to Eq. (7.1).

$$p = e^{\alpha . \frac{d_{s'-s}}{d_s}} \tag{7.1}$$

  If the best candidate makes a huge improvement in $d_{s'-s}$ or/and the current node is close to the objective (d_s small), the probability of selecting the best node is high. Whereas, if we are making little improvement or/and we are far from the objective, the probability of selecting at random is high. The α in Eq. (7.1) is a tuning parameter to balance deterministic and randomness behavior.

It is worth noting that the aforementioned modifications prune the high number of hops needed by Simulated Annealing by means of introducing randomness only among closer neighbors. On the other hand, this proposal differentiates from the random greedy because it does not always choose randomly.

7.4.3 Backwarding (there are no candidates as next forwarding node)

Recovery mechanisms are used when there is no neighbor that meets the forwarding criteria and that reduces the distance to destination. In this section, we describe an algorithm based on Simulated Annealing to be used with a *carry and forwarding* approach.

Simulated Annealing Recovery

The idea of the algorithm is that if the current node does not find any legal neighbor that improves the current solution (reduce distance to destination), then the node could worsen

the distance to destination with a certain probability p. This is, the node forwards the packet to a neighbor, even when the selected node increases the distance to the destination node. Otherwise, with probability $(1-p)$ the node keeps the packet in a buffer. Alg. 7.6 describes this approach.

SIMULATEDANNEALINGRECOVERY(P, s, s')

Require: s current node, s' best node
Ensure: Forward P to a farther node s with prob. p
1: d_s = D(s)
2: $d_{s'-s}$ = D(s') -d_s
3: $p = e^{-\beta \frac{d_{s'-s}}{d_s}}$
4: random := U[0, 1]
5: **if** p > random **then**
6: s := s'
7: **else**
8: **while** Expired(t_{out})=*False* **do**
9: Keep the packet in the buffer
10: **return**

Algorithm 7.6: Modified Simulated Annealing applied to *carry and forwarding* process of a geographical routing protocol.

Notice that the equation in line 3 used in Alg. 7.6 is identical to Eq. (7.1) except that $d_{s'-s} > 0$ (because the best node of s is farther to destination than s), therefore the negative sign is needed in Eq. (7.2).

$$p = e^{-\beta . \frac{d_{s'-s}}{d_s}} \tag{7.2}$$

From the algorithm we see how the probability of *worsening* (i.e., of selecting a worse next forwarding node) becomes higher for two factors:

- Farther the node s is from destination, the more likely to forward the packet to s', which increments the distance to destination (worsen the solution).
- The closer are nodes s and s', the more likely to worsen the current solution (i.e., to forward the packet to s').

While the packet is in the buffer, the node searches periodically in its neighbor list if there is a suitable next forwarding node. After a packet has been stored in the buffer during an interval of time higher than t_{out}, it is discarded.

7.5 Geographical Heuristic routing protocol proposals

In this section, we present our two proposals of geographical routing protocols: Geographical Heuristic Routing (GHR) protocol and 2-hops Geographical Anycast Routing (2hGAR) protocol. They follow the DTN approach of carrying the packet when there is not a suitable next forwarding node. Moreover, both routing protocols can use any routing criteria to select the next forwarding node to take forwarding decisions at each hop.

On the one hand, the GHR protocol combines all the adaptations presented in the previous Sec. 7.4 for the forwarding and recovery phases of the protocol, which are based on Tabu search and Simulated Annealing (SA). On the other hand, the 2hGAR protocol employs information of nodes located two hops away from the current node. This allows nodes to increase their knowledge of the network state and not only rely on the local information (i.e., data from the neighbors). The latter approach is specifically designed for routing anycast data in which, information has to be delivered to any member of the destination set. An example of such anycast traffic are the traffic generated by reporting services from vehicles to authority centers, in which data packets need to reach any of the access point of the authorities.

Alg. 7.7 shows the generic routing process of a packet in a node that employs a geographical routing protocol with *carry and forwarding* as recovery strategy. The algorithm is written for an anycast routing protocol. Nonetheless, the algorithm works exactly the same for a unicast routing. In that particular case, the destination set *Dst* used as input of the algorithm will have a single element that is the destination node of the packet.

ROUTE(P)

Require: a packet P, Destination set Dst, current vehicle v
Ensure: Forward P to a vehicle neighbor v_{nh} to reach a member Dst or save P in buffer.

```
1:  Dst ← GETDESTINATIONSET(P)
2:  if v ∈ Dest then
3:      GOTOUPPERLAYER(P)
4:      return
5:  if ∃ n ∈N(v) ∧ n ∈ Dest ∧ ISLEGAL(n) = True then
6:      v_nh ← n
7:  else
8:      v_nh ← DOROUTING(P, v, Dst)
9:      if v_nh ≠ v then
10:         FORWARD(P, v_nh)
11:     else
12:         BUFFERING(P)
13: return
```

Algorithm 7.7: General routing process of a DTN geographical protocol.

When a packet arrives to a node, the first step is to determine the destination set of that packet (line 1 in Alg. 7.7). The information of the destination set can be retrieved with the anycast IP address of the packet. A list of destination nodes is associated with each anycast IP address. Every node carries this mapping Table with it. So, the look-up task of the destination set does not need any online search. Moreover, since the typical destination set will be fixed network points like RSUs, then their positions also can be a priori known. The associations anycast IP address-destination and RSU position do not need to be constantly updated. A vehicle can refresh this information where there are available internet connections for instance when the vehicles are parked at home or office.

If the current node v is member of the destination set Dst, then the packet is processed by the upper layer (line 3 in Alg. 7.7) and the routing process ends. When the current node

is not a destination member, it searches if it has a member n of the destination set among its neighbors N(v), which in addition is considered a legal node (See. line 5). A "legal node" refers to a neighbor that meets a certain list of characteristics which depend on the routing protocol. For instance, in the case of our MMMR [113] protocol used in this thesis, a legal neighbor has to be in LOS and an estimated power reception has to be higher than its antenna sensibility plus 1dB. If the current node v finds such neighbor then it chooses as next forwarding node that neighbor n (line 6). Otherwise, a selection of the next forwarding hop v_{nh} is performed (line 8). Next, if the selected node v_{nh} is not the current node v (line 9 in Alg. 7.7) then the routing protocol forwards the packet to v_{nh}. Else, the packet is stored in the buffer until a suitable node appears. The next sections explain in detail two routing proposals for the function DOROUTING, which is in charge of selecting the next forwarding node.

7.5.1 Geographical Heuristic Routing protocol (GHR)

As we anticipated in the introduction of this section, Geographical Heuristic Routing protocol (GHR) combines the forwarding and recovery heuristics presented in the previous Sec. 7.4, which are based on Tabu-search and Simulated Annealing (SA). GHR does not need to add any information to the hello messages. Alg. 7.8 shows the procedure of our proposal GHR. First, it needs five input parameters:

- The packet P to extract its Tabu list τ.
- The set *Dst* of possible destination nodes according to the anycast address of destination.
- The boolean variable *Tabu* that indicates if Tabu routing has to be used. This means that a node cannot forward packets to the nodes in the Tabu list τ, as was explained in Sec. 7.4.1.
- The boolean variable *First* to select the first neighbor that meets the routing conditions.
- The forwarding factor α affects the probability of selecting a random legal neighbor, which is closer to destination than the current node v to forward the packet. If $\alpha = 0$ then the next forwarding node will be selected randomly among the legal neighbors (i.e., Random forwarding). On the contrary, if this factor $\alpha \to +\infty$ then the best neighbor closer to destination than the current node will always be selected (i.e., Best forwarding). For $0 < \alpha < +\infty$, we obtain the Simulated Annealing (SA) forwarding explained in Sec. 7.4.2.
- A recovery factor β tunes the probability of avoiding "*carry and forwarding*" approach in favor of forwarding a packet to a legal neighbor, which is farther from destination than the current node. When $\beta = 0$ the routing protocol will always select a legal neighbor, if there is any, as the next forwarding node. On the other hand, when $\beta \to \infty$, *carry and forwarding* is always applied. Similar to the forwarding factor, if $0 < \alpha < +\infty$ then the GHR protocol uses the recovery Simulated Annealing, which forwards the packet to a farther node to destination with a probability p, computed according to Eq. (7.2).

The first operations performed by the GHR are the following:

1. Extraction of the Tabu list τ from the packet P and
2. To set the initial value of the decision variables v_{nh}, v_f and v_r to the current node ID (line 1 and 2 of Alg. 7.8).

GEOGRAPHICALHEURISTICROUTING ($P, Dst, Tabu, First, \alpha, \beta$)

Require: a packet P, destination set Dst, use of $Tabu$, use of $First$ node, forwarding factor α, recovery factor β, current vehicle v
Ensure: Select the best neighbor v_{nh} to reach a member of the destination set v_{dst}.
1: $\tau \leftarrow$ GETTABULIST($P, state$) {if $Tabu = False$ then $\tau = \emptyset$}
2: $v_{nh} \leftarrow v$, $v_f \leftarrow v$, $v_r \leftarrow v$ {Initializing next hop variables}
3: $s_{v_f} \leftarrow -\infty$, $s_{v_r} \leftarrow -\infty$
4: $\theta \leftarrow$ U[0, 1]
5: L(N(v),v) $\leftarrow \emptyset$ {Initial empty set of legal neighbors of v closer to v_{dst}}
6: **for all** $v_{dst} \in Dst$ **do**
7: $d_v \leftarrow$ D(v, v_{dst})
8: **for all** $n \in$ N(v) **do**
9: **if** $n \notin \tau$ **then**
10: **if** ISLEGAL(n) = $True$ **then**
11: $d_{v-n} \leftarrow d_v$ - D(n, v_{dst})
12: $s_n \leftarrow$ COMPUTEMETRIC(n)
13: **if** $d_{v-n} < 0$ **then** {Forwarding phase}
14: L(N(v),v) = L(N(v),v) $\cup$ n
15: **if** $s_n > s_{v_f}$ **then**
16: $v_f \leftarrow n$, $s_{v_f} \leftarrow s_n$, $d_{v-v_f} \leftarrow d_{v-n}$ {Set the current best vehicle as forwarding vehicle}
17: **if** First=$True$ **then**
18: $v_{nh} \leftarrow v_f$
19: **goto End**
20: **else** {Revocery phase: To use a farther vehicle to destination than the current v}
21: **if** $s_n > s_{v_r}$ **then**
22: $v_b \leftarrow n$, $s_{v_r} \leftarrow s_n$, $d_{v-v_b} \leftarrow d_{v-n}$ {Set the best legal vehicle is not closer to v_{dst} as recovery vehicle}
23: **end for**
24: **end for**
25: **if** L(N(v),v)$\neq \emptyset$ **then** {Forwarding phase}
26: $p = e^{\alpha d_{v-v_f}/d_v}$
27: **if** p $> \theta$ **then**
28: $v_f :=$ S$_{\text{Random}}$(L((Nv),v)) {Set a random legal neighbor as forwarding vehicle}
29: v_{nh}=v_f
30: **else** {Recovery phase}
31: **if** $v_r \neq v$ **then**
32: $p = e^{-\beta d_{v-v_b}/d_v}$
33: **if** p $> \theta$ **then**
34: v_{nh}=v_r
35: **End:**
36: **if** $v_{nh} \neq v \wedge Tabu = True$ **then** {The packet will be forwarded}
37: UPDATETABULIST (τ, v, P)
38: **return** v_{nh}

Algorithm 7.8: Our proposal Geographical Heuristic Routing (GHR) protocol.

The Tabu list will be empty if the *Tabu* option is not enabled and therefore there will not be any forbidden neighbor as next hop. Regarding v_{nh}, it stores the next forwarding node. It can be equal to the selected node v_f of the forwarding phase, to the selected backup node v_r in the recovery phase or it can store the initial value v because no neighbor was chosen to forward the packet P.

Secondly, GHR initializes to a very low value the best scores s_{v_f} and s_{v_r} of the selected node in the forwarding and recovery phases, respectively (line 3). Then, a random uniform value between 0 and 1 is stored in the variable θ (line 4). The probability of random or recovery forwarding is compared to the value θ to decide if any of these options is performed. Next, in line 5 of Alg. 7.8, the set of legal neighbors L(N(v), v) is initialized as an empty set. L(N(v), v) stores the legal neighbors n which are closer to a destination member $v_{\text{dst}} \in Dst$ than the current node v.

GHR searches the best forwarding v_f and recovery v_r nodes considering all the members of the destination set Dst. To do that, GHR searches for each destination member v_{dst} (see the "for" loop from line 9 to 24 of Alg. 7.8) if any of the neighbors in $N(v)$ (see the "for" loop between from line 8 to 23 of Alg. 7.8) improves the current scores s_{v_f} and s_{v_r} of v_f and v_r, respectively. In the selection process the first step is to obtain the distance from the current node to the destination member (line 7). Then, GHR checks if the neighbor $n \in N(v)$ is not in the Tabu list τ (line 9) and that it is legal (line 10). Then the algorithm considers neighbor n as a possible next forwarding node. Otherwise, the node is discarded by the selection process.

After verifying the eligibility of a neighbor n, the GHR protocol calculates the distance difference d_{v-n} to the destination member v_{dst} between the current node v and the candidate node n (line 11). In addition, the algorithm computes the routing metric score of the neighbor n (line 12). If $\text{d}_{v-n} \geq 0$ in line 13, it means that n is closer to v_{dst} than the current node v and therefore n is a candidate to be the next forwarding node v_f. Otherwise (i.e., $\text{d}_{v-n} < 0$), the neighbor n is considered as a possible recovery node. In the forwarding phase, a neighbor n is added to the set of legal neighbors L(N(v), v) (line 14).

Next, if the score of neighbor n is higher than the current best score s_{v_f} then the neighbor n becomes the best forwarding node v_f and the best score and the distance to destination d_{v-v_f} are updated (line 16). When the *first* option is set to true, the first legal forwarding node v_f is selected as the next forwarding node v_{nh} (lines 17 and 18) and the searching stops and goes to the final step of the algorithm. When the neighbor n is considered to the recovery phase (line 20), if the neighbor score s_n is higher than the current best recovery score s_{v_r} then n is the new best recovery node v_r and the corresponding score and distance are updated (line 22).

After the GHR protocol searches the best forwarding node considering all the members of the destination set, it checks whether the set L(N(v), v) is empty (line 25 of Alg. 7.8). If it is not empty, a probability p is obtained as a function of the forwarding factor α, the distance difference d_{v-n} and the distance from the current node to destination d_v. The purpose of these factors was explained in the forwarding Simulated Annealing of Sec. 7.4.2. If $p > \theta$, then the next forwarding node is chosen randomly from the set of legal neighbors (line 28); otherwise the next forwarding node is the best neighbor previously stored in v_f. When there is no legal neighbor closer to destination (i.e., L(N(v),v)=$\emptyset$) but there is a legal recovery node ($v_r \neq v$ in line 32) then a probability p, computed alike as in the forwarding case and

explained in Sec. 7.4.3, decides if the packet P is forwarded to v_r ($p > \theta$ in lines 33 and 34) or on the contrary the packet is stored in a buffer.

Finally, if *Tabu* routing is being used (i.e., $Tabu = True$) and the packet will be forwarded to some neighbor (i.e., $v_{\text{nh}} \neq v$) then the current node v is added to the Tabu list τ according to the procedure described in Alg. 7.4.

7.5.2 2-hops Geographical Anycast Routing protocol (2hGAR)

Our next proposal called Geographical Anycast Routing protocol (2hGAR) is inspired on the 2-opt heuristic used in the Traveling Salesman Problem [88] to improve the result obtained by local-search.

Other routing proposals in the literature use information of nodes located at 2-hops as [97] [7] by adding all the identifiers of the neighbors of a node in the hello message that it sends periodically. As we said, 2hGAR is specifically targeted for anycast applications because each node only adds information about the current “best" neighbor to reach a member of the destination group to the hello message, instead of the whole list of neighbors. In this way, every time a node receives a hello message, that node will have updated routing information of 2-hops away from its current position. This 2-hops knowledge of the network topology can help the node to take a better forwarding decision and avoid voids, which entails the use of recovery strategies

Additional information in the Hello message

A geographical routing protocol needs to add the additional fields shown in Fig. 7.1 to its hello message to route packets according to our 2hGAR approach. These fields are:

- **Destination member ID.** This field carries the ID member of the destination set *Dst* for which the metric score and distance are calculated. The length of this field depends on the ID type. It could be an IPv4 address (32 bits), an IPv6 address (128 bits) or a unique identifier used among the members of the destination group. With this field, the receiver of the hello message can compute its distance to that member of the destination set.
- **Nexthop node ID.** It is the identifier of the best next forwarding hop that the sender of the hello message would use to forward the packet. This field is required to avoid loops or bad forwarding nodes as we will explain in the next section. If the sender of the hello message does not have any better forwarding hop than itself, then it will record its own identifier in this field.
- **Distance to destination.** It stores the distance from the next hop of the hello message sender (Nexthop node ID) to the destination member indicated in the first field (Destination member ID).
- **Metric score.** This is the score obtained by the best next hop of the hello message sender computed according to the criteria of the routing protocol.

The routing algorithm

The routing procedure used by our proposal 2-hops geographical anycast routing protocol (2hGAR) is depicted in Alg. 7.9. As in our previous proposal GHR, the 2hGAR protocol extracts the Tabu list τ of the packet. Recall that the list is empty if the *Tabu* option is not enabled. Also, the protocol initializes the next hop ID v_f to the current node v and the initial score s_f to $-\infty$ (lines 1 and 2).

Original Hello Message	Destination (member ID)	Next hop node ID	Distance to destination	Metric score
variable size	32-128 bits	32-128 bits	12 bits	20 bits

Figure 7.1: Additional fields in the hello message of 2-hops Geographical Anycast Routing protocol (2hGAR)

2HGEOGRAPHICALANYCASTROUTING ($P, Dst, Tabu$)

Require: a packet P, destination set Dst, use of $Tabu$
Ensure: Select the best neighbor v_{nh} to reach a member of the destination set v_{dst}.

1: $\tau \leftarrow$ GETTABULIST($P, Tabu$) {if $Tabu = False$ then $\tau = \emptyset$}
2: $v_f \leftarrow v$ {Initializing next hop variables}
3: $s_{v_f} \leftarrow -\infty$ {Initializing scores of next hop }
4: **for all** n ∈ N(v) **do**
5: **if** $n \notin \tau$ **then**
6: **if** $(n_{\text{nh}} \notin N(v) \vee n_{\text{nh}} = n) \wedge n_{\text{nh}} \neq v$ **then**
7: $\text{d}_v \leftarrow \text{D(v,n}_{\text{dst}})$
8: **if** ISLEGAL$(n) = True \wedge \text{d}_{n_{nh}} < d_v$ **then**
9: **if** $\text{s}_{n_{\text{nh}}} > \text{s}_{v_f}$ **then**
10: $\text{v}_f \leftarrow n$, $\text{s}_{v_f} \leftarrow \text{s}_{n_{nh}}$ {Set the current best vehicle as forwarding vehicle}
11: **if** $\text{v}_f \neq v \wedge Tabu = True$ **then** {There is a better next hop than v}
12: UPDATETABULIST (τ, v, P)
13: **return** v_f

Algorithm 7.9: Our proposal 2-hops Geographical Anycast Routing (2hGAR) protocol.

Regarding the selection of the next forwarding node, the algorithm checks all the neighbors $n \in N(v)$ of the current node ("for" loop from line 4 to 10 of Alg. 7.9) to find the best next hop. Three conditions must be fulfilled by a neighbor node n to be considered as a candidate next forwarding node:

- Firstly, the algorithm checks that the current neighbor n is not in the Tabu list τ.
- Secondly, 2hGAR makes an important validation that prevents bad forwarding selection and loops through the condition of line 6. The first part of this condition (i.e., $(n_{\text{nh}} \notin N(v) \vee n_{\text{nh}} = n)$) guarantees that the current node v will not choose any neighbor n whose next hop n_{nh} is within the neighborhood of the current node because the current node v could forward the packet to n_{nh} directly without passing through n. The exception $n_{\text{nh}} = n$ means that the node n does not have any forwarding node better than itself, therefore it writes its own ID in the field of *Next hop ID* in the hello message. The second part of the condition of line 6 states $n_{\text{nh}} \neq v$, that is the next hop n_{nh} of the neighbor n cannot be the current node v, because that case would create the loop $v \rightarrow n \rightarrow v$.
- Finally, the routing protocol checks if neighbor n fulfills other requirements (ISLEGAL(n) in line 8), which depend on the routing protocol. Additionally, the routing protocol in the condition of line 8 of Alg. 7.9 requires that the distance of the candidate next hop (located 2-hops away from v) to destination $d_{n_{\text{nh}}}$ will be shorter than the distance d_v from the current node.

If a neighbor n meets the three previous conditions (i.e., Tabu in line 5, Free loops in line 6 and Eligibility in line 8) then the neighbor n can be considered as a suitable next forwarding node v_f. Otherwise, the neighbor n is discarded to forward the packet.

To every neighbor n that is a suitable next forwarding, its metric score $s_{n_{nh}}$ is compared to the best current score s_{v_f}. If $s_{n_{nh}} > s_{v_f}$ then the neighbor n is the new next forwarding node v_f and its corresponding score is stored to be compared with the remaining neighbors (line 10 of Alg. 7.9). When the routing protocol has finished the checking of all the neighbors $N(v)$, v_f will store the ID of the next forwarding node. If $v_f = v$ then there is not a suitable next forwarding node then *carry and forwarding* is used as recovery method. Otherwise, if Tabu routing is used ($Tabu = True$ in line 11) then the Tabu list τ is updated by adding v.

7.6 Performance evaluation

In this section, we present the performance evaluation of our two routing proposals GHR and 2hGAR. First, we study the forwarding and recovery options of GHR in order to find the most suitable configuration for this protocol. After that, we compare the best configurations of GHR with 2hGAR to see the advantages of each of them in the performence metrics. To carry out our performance evaluation we use the simulation scenario for a reporting service and the setting described in Sec.3.4 from chapter 3. We use MMMR [113] in the core of our routing proposals to score the neighbors and choose the best one among them. Additionally, we have incorporated the coherent, automatic address resolution explained in Chapter 5 and the packet filter based on packet ID proposed in Chapter 6. For the evaluation of our proposals we run 20 simulations per each vehicle density using different movement traces to present the figures with a confidence interval of 95%. We analyze our proposal on four different metrics:

- The percentage of packet losses.
- The average end-to-end delay.
- The average number of hops (these three were explained previously in chapter 3 Sec. 3.5)
- The percentage of idle time sensed by a node measured in 1 second. This metric provides us an idea of the bandwidth consumption done by the routing mechanism. Notice that a higher idle time sensed by the nodes leads to more bandwidth available in the channel to transmit more information. An estimation of the available bandwidth derived from the idle time measure could be done using the model proposed by [87] or [112].

7.6.1 Evaluation of the forwarding phase in GHR

To evaluate the forwarding phase of our Geographical Heuristic Routing (GHR) protocol, we distinguish three factors that could affect the performance of the GHR. These factors are:

1. The vehicle density of the scenario (VD).
2. The use of a Tabu list in the routing (T).
3. The forwarding technique (FT).

We compare the four different ways that GHR uses to select the next forwarding node: the best legal neighbor ($\alpha \to \infty$), a random legal neighbor ($\alpha = 0$), randomly according to Simulated Annealing (SA) ($\alpha = 3$) or the use of first legal neighbor. Following the three

steps procedure described in Sec. 3.5 from chapter 3 we have to analyze the results for each density separately because there is a significant three-way interaction $FT \times T \times VD$ (**Step 1**) with a p-value = 0.001 (Wilk's Λ = 0.377 and F(24,92)=2.41).

Table 7.1 shows the results of the further interaction test between the use of Tabu T and FT for GHR protocol (**Step 1**). First, we perform the so-called "all together" test in which we evaluate the interaction between the two factors considering the correlation among the six performance metrics. If the p-value < 0.05 for this test, we further carry separate interaction tests for each metric. In Table 7.1 we can see that for the medium and the high density scenarios, the "all together" test has a p-value under the threshold, so independent interaction test per metric need to be performed in those cases. The results of the interaction tests for each metric indicate that the seek of performance differences in the forwarding technique has to differentiate if tabu was used or not in the routing in the medium and high density scenarios (150 and 250 vehicles, respectively) to compare the average number of hops. There is also this significant interaction (i.e., $FT \times T$) for the evaluation of percentage of idle time in the scenario of 250 nodes.

Number of vehicles	Performance Metric	Wilk's Λ	F (3,17)	p-Value
100	All metrics together*	0.284	1.684	0.234
150	All metrics together*	0.103	5.834	0.009
	% of packet losses	0.858	0.941	0.442
	Average delay	0.661	2.910	0.065
	Average no. of hops	**0.335**	**11.272**	**0.0001**
	% of idle time	0.666	2.840	0.069
250	All metrics together*	0.113	5.235	0.013
	% of packet losses	**0.502**	**0.284**	**0.007**
	Average delay	0.665	2.858	0.068
	Average no. of hops	**0.369**	**9.691**	**0.001**
	% of idle time	**0.549**	**4.653**	**0.015**

Table 7.1: MANOVA [60] results for interaction test between forwarding technique (FT) and the use of Tabu (T) for forwarding phase of GHR. *Degrees of freedom in F statistic are 12 and 8. If there is a significant interaction (p-value<0.05) in the "All metrics together" test then interaction tests per metric need to be performed.

The test results to determine if there are differences in the performance metrics, according to **Step** 2 of the statistical procedure of Chapter 3, are shown in Table 7.2. For this analysis, the data of the metrics were grouped according to the results of the interaction tests analyzed previously. This means that in most of the cases, we performed only one test per metric in each density without differentiating whether Tabu is enabled during the simulation. We labeled these cases as "together" in the Tabu column. All the p-values in this table are lower than 0.05, excluding the percentage of packet losses in the medium density scenario (150 vehicles) and the average end-to-end delay for the three vehicle densities. Hence, the use of different forwarding techniques (i.e., $\alpha \to \infty$, $\alpha = 0$, $\alpha = 3$ or *First*) produce a statistical significant change among them in all performance metrics and therefore they require pairwise

comparisons to analyze the differences. These pairwise comparisons are not needed for the average delay metric in all densities and for the percentage of packet losses in the medium density scenario because there are not difference among the forwarding techniques in these cases.

Metric	Number of Density	Tabu	Wilk's Λ	F (3,17)	p Value
% of Packet Losses	100	together	0.516	5.306	0.009
	150	**together**	**0.666**	**2.846**	**0.068**
	250	No	0.237	18.222	0.0005
	250	Yes	0.456	6.747	0.003
Average delay	**100**	**together**	**0.894**	**0.671**	**0.582**
	150	**together**	**0.975**	**0.145**	**0.932**
	250	**together**	**0.662**	**2.890**	**0.066**
Average number of hops	100	together	0.163	29.14	0.0005
	150	No	0.198	22.901	0.0005
	150	Yes	0.081	63.96	0.0005
	250	No	0.046	118.0	0.0005
	250	Yes	0.022	253.14	0.0005
% of idle time	100	together	0.332	69.079	0.0005
	150	together	0.203	22.22	0.0005
	250	No	0.137	35.77	0.0005
	250	Yes	0.155	30.979	0.0005

Table 7.2: MANOVA [60] results of testing difference in performance metrics among routing forwarding techniques (FT). There is an significant difference when p-value < 0.05. “Together" means that to apply the test it is not needed to differentiate the use of Tabu in the forwarding techniques.

Following the **Step 3** of the statistical analysis, Table 7.3 shows the results of the pairwise comparisons among the forwarding techniques (i.e., ($\alpha \to \infty$, $\alpha = 3$), ($\alpha \to \infty$, $\alpha = 0$), ($\alpha \to \infty$, *First*), ($\alpha = 3$, $\alpha = 0$), ($\alpha = 3$, *First*) and ($\alpha = 0$, *First*)) in which there are not statistical significant for a particular metric (i.e., p-value$\geq$0.05). In these cases, the average values of the metrics are very similar and can be considered statistically the same. The rest of the results of the pairwise comparison tests (not included in Table 7.3), e.g. the percentage of packet losses for the high density scenario when Tabu is not enabled, obtained p-values $<$ 0.05. Those p-values indicate that the forwarding technique (i.e., Best node, SA, Random or First) has an impact in the values of the metrics.

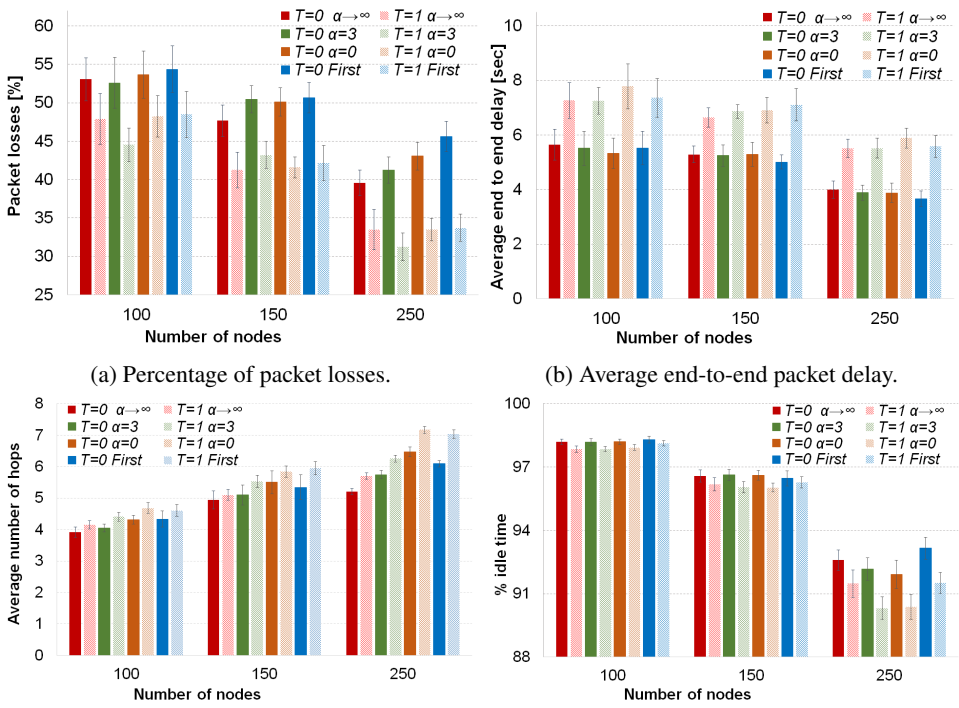
The comparison of average values of the four performance metrics with the different forwarding techniques are depicted in Fig 7.2. Considering the previous statistical analysis we continue analyzing the behavior of forwarding techniques in the metrics.

Firstly, the use of our Tabu list, which consists on the three last nodes in the packet path, decreases the percentage of packet losses considerably. This descent in the packet losses goes from 6% in low density scenario till around 10% with high vehicle density as it can be seen in Fig. 7.2a. The reason for this improvement is that our Tabu list provides memory

Metric	Number of Density	CW	Pairwise	*p* Value
% of packet losses	100	together	$(\alpha \to \infty, \alpha = 0)$	0.317
	100	together	$(\alpha \to \infty, First)$	0.276
	100	together	$(\alpha = 0, First)$	0.579
	250	Yes	$(\alpha \to \infty, \alpha = 0)$	0.98
	250	Yes	$(\alpha \to \infty, First)$	0.855
	250	Yes	$(\alpha = 0, First)$	0.741
Average number of hops	100	together	$(\alpha = 0, First)$	0.584
	150	Yes	$(\alpha = 0, First)$	0.142
	250	Yes	$(\alpha = 0, First)$	0.05
% of idle time	100	together	$(\alpha \to \infty, \alpha = 0)$	0.121
	100	together	$(\alpha \to \infty, First)$	0.09
	100	together	$(\alpha = 3, \alpha = 0)$	0.067
	150	together	$(\alpha \to \infty, \alpha = 3)$	0.584
	150	together	$(\alpha \to \infty, \alpha = 0)$	0.444
	150	together	$(\alpha = 3, \alpha = 0)$	0.820
	250	No	$(\alpha = 3, \alpha = 0)$	0.096
	250	Yes	$(\alpha \to \infty, First)$	0.096
	250	Yes	$(\alpha = 3, \alpha = 0)$	0.724

Table 7.3: Pairwise comparison of the performance metrics in which there is a difference among forwarding techniques for GHR protocol (i.e., p-value < 0.05 in Table 7.2). The Table only shows the results for metrics and pairs of forwarding techniques with absence of statistical significant differences (i.e., p-value ≥ 0.05). “Together" means that it is not needed to differentiate whether Tabu was enabled or not to apply the test.

to the routing decision. This memory helps to avoid neighbors of the current node already visited by the packet that otherwise they could be selected again to forward it. Therefore, the Tabu list avoids loops and helps to consider other possible next forwarding nodes. On the other hand, the use of our Tabu routing proposal increases the average number of hops (See. Fig. 7.2c) around 0.6 hops in average. Forbid nodes as next hops, forces nodes to search other path, that might be longer. Moreover, the average end-to-end delay increases around 2 s for low vehicle density scenario and 1.5 s for the high density scenario. The higher delay is because of the longer paths. In addition, since the amount of possible next forwarding nodes decreases due to the list of prohibited nodes (i.e., Tabu list) the *carry and forwarding* procedure is used more often, increasing the average delay of packets. Finally, it is important to notice that the use of a Tabu list is overhead to be carried by each packet until reaching destination. This means more bandwidth utilization for a packet transmission. Hence, the % of idle time sensed by a node decreases when our Tabu routing is enabled. In Fig. 7.2d, it can be seen that as the vehicle density increases the difference between to use or not to use Tabu in the percentage of idle time increases as well, reaching 2% in the scenario with 250 vehicles. Nevertheless, The slight higher delay and lower idle time are not so important compared to the noticeable lower losses achieved when the Tabu list is used. Thus, this trade-off clearly shows benefits in favor of the Tabu list.



(a) Percentage of packet losses.

(b) Average end-to-end packet delay.

(c) Average number of hops.

(d) Percentage idle time.

Figure 7.2: Performance evaluation of the forwarding techniques (FT) included in our proposal GHR. $T = 1$ means that our Tabu routing is used. The four FT are: the best node selection ($\alpha \to \infty$), the Simulated Annealing ($\alpha = 3$), random forwarding ($\alpha = 0$) and the first legal node (*First*).

Regarding the behavior the forwarding techniques, as it can be seen from Fig. 7.2a, in low vehicle density the forwarding inspired in Simulated Annealing ($\alpha = 3$) has a slightly and statistically significant improvement (around 2%), compared to the other three approaches (i.e, best node, random and first legal). They behave similarly according to our statistical analysis (see p-values$>$0.05 for packet losses and 100 vehicles in Table 7.3). For the medium density, there is not a statistically significant difference among the forwarding techniques neither when the Tabu is used, neither when this option is disabled (see second row in Table 7.2). In high density scenario the behavior is different and depends if Tabu routing is being used or not. When Tabu is not used, it is clear from Fig. 7.2a that the selection of the best neighbor gets the lowest percentage of packet losses. On the other hand, a complete random selection or the selection first legal neighbor as next forwarding node have the worst % of packet losses. The reason lies on the high number of hops that these two forwarding techniques use to reach destination. When Tabu routing is used in the high density scenario, the behavior of the routing techniques change completely. The degree of randomness given by the SA forwarding ($T = 1$, $\alpha = 3$ in Fig. 7.2a), gets the best results. The advantage of this approach is based on not always selecting the best node, which could avoid collisions or

links saturation, and on avoiding already visited node by using the Tabu list. The other three approaches (i.e., $\alpha \to \infty$, $\alpha = 0$ and *First*) have statistically the same results thanks to the use of Tabu.

As it can be seen from Fig. 7.2b, the delay among the four different techniques is the same regardless the use Tabu routing for the three vehicle densities. This was confirmed by our statistical analysis, whose results (p-values$\geq$0.05) are shown in the average delay section of Table 7.2. From Fig. 7.2c, we can realize that applying a forwarding factor $\alpha < \infty$, the average number of hops increases as it is expected because the randomness in the forwarding increases too. So, the highest number of hops is always obtained by a complete random selection ($\alpha = 0$) and the lowest hop count is of the best selection ($\alpha = \infty$). When Tabu is used or in low vehicle density, the selection of the First legal node needs as many hops as a random selection. (see the p-value$\geq$0.05 of the pairs ($\alpha = 0$, *First*) in average number of hops in Table 7.2).

The percentage of idle time depicted in Fig. 7.2d depends on the forwarding mechanism. The selection of the first legal neighbor has a high percentage of idle time for the three densities. This reveals a better use of the available bandwidth of this forwarding mechanism in spite of the high number of hops that *First* strategy needs to operate. The reason is that *First* strategy prefers to use recent updated neighbors, which have the most stable links, and therefore will have a higher number of successful transmissions at the first attempt than other approaches like best node or random selection. In fact, this technique reaches the highest value for medium density scenario (150 nodes) while the other techniques obtain p-values$\geq$0.05 in the pairwise comparisons of idle time, see Table 7.3. In the high density scenario, the *First* strategy has the same high level of idle time as the best node selection ($\alpha \to \infty$) while random ($\alpha = 0$) and SA ($\alpha = 3$) have the lowest level of idle time because they perform more number of hops than the classical selection of the best node. However, it is worth noting that Simulated Annealing forwarding obtains the lowest percentage of packet losses with Tabu for this high vehicle density.

7.6.2 Evaluation of the recovery phase in GHR

In this section, we evaluate the performance of the recovery factor β. This factor allows that the routing protocol forwards a packet to a farther node to destination than the current node instead of keeping the packet in the buffer until a better next forwarding node appears. For this evaluation, we consider only the forwarding techniques best neighbor ($\alpha \to \infty$) and the Simulated Annealing forwarding ($\alpha = 3$), both with Tabu routing enabled. The reason to only use these two strategies in the tests of the recovery phase is that SA forwarding has the lowest percentage of packet losses in the three densities of vehicles and the selection based on the best neighbor is the classical approach in the geographical routing. Moreover, the other two techniques, i.e., random forwarding ($\alpha = 0$) and first legal neighbor, behave similar to the best node selection. They do not outperform the criterion of the best node in terms of packet losses and delay.

For this part of our study we work with three factors that could affect the performance of the recovery phase of GHR. These factors are:

1. The vehicle density of the scenario (VD).
2. The forwarding technique (FT).
3. The recovery technique (RT), which depends on β factor. We consider three different

values for the β. They are: $\beta \to \infty$ which is the default *carry and forwarding* approach, $\beta = 3$ to use a recovery SA and $\beta = 0$, which always selects a next forwarding node if the current node has some legal neighbor.

We do not evaluate the use of the recovery factor β for the forwarding techniques without Tabu because the packet losses increase significantly for those cases, as it can be seen in Fig. 7.3 for the forwarding technique based on the selection of the best neighbor ($\alpha \to \infty$). The values of $\beta = 3$ or $\beta = 0$ performs bad because the recovery mechanism creates loops without Tabu. The recovery mechanism selects in most of the cases the previous neighbor that forwarded the packet to the current node. This creates loops and nodes can not avoid those nodes because they do not have a track of previous path followed by the packet.

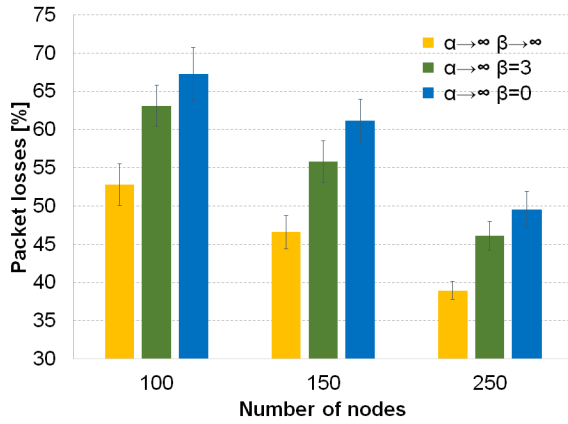


Figure 7.3: Percentage of packet losses for different values of the recovery factor β, using the best node criterion at the forwarding phase without Tabu.

We have to analyze the results for each density separately because there is a significant three-way interaction $RT \times FT \times VD$ (**Step 1**) with a p-value = 0.0001 (Wilk's Λ = 3.41 and F(16,100)=4.47).

Table 7.4 shows the results of the further interaction test between the FT and RT for GHR protocol (**Step 1**). We can see that for the medium and the high density scenarios, the "all together" tests have a p-value under the threshold, so an independent interaction test per metric needs to be performed in those cases. The p-value < 0.05 in the interaction tests for each metric indicate that the seek of performance differences in the recovery technique should be done for each forwarding technique (i.e., best neighbor and SA forwarding) separately.

The test results to determine if there are differences in the performance metrics, according to **Step 2** of the statistical procedure of Chapter 3, are shown in Table 7.5. All the p-values in this table are lower than 0.05, excluding the percentage of packet losses in the high density scenario (250 vehicles) for the forwarding factor $\alpha \to \infty$ (select the best neighbor). Hence, the use of different values for the recovery factor β produce a statistical significant change among them in all performance metrics and therefore they require pairwise comparisons to analyze the differences.

Table 7.6 shows the results of the pairwise comparisons among the forwarding techniques (i.e., ($\beta \to \infty$, $\beta = 3$), ($\beta \to \infty$, $\beta = 0$), ($\beta = 3$, $\beta = 0$)) following the **Step 3** of the analysis,

Number of vehicles	Performance Metric	Wilk's Λ	F (2,18)	*p*-Value
100	All metrics together*	0.358	1.784	0.06
150	All metrics together*	0.228	5.07	0.006
	% of packet losses	**0.658**	**4.771**	**0.022**
	Average delay	0.907	0.919	0.095
	Average no. of hops	**0.514**	**8.51**	**0.003**
	% of idle time	0.761	3.866	0.055
250	All metrics together*	0.334	2.996	0.043
	% of packet losses	**0.706**	**3.05**	**0.041**
	Average delay	**0.548**	**7.415**	**0.004**
	Average no. of hops	0.798	2.275	0.202
	% of idle time	**0.508**	**8.718**	**0.002**

Table 7.4: MANOVA [60] results for interaction test among recovery techniques (RT) and the forwarding technique (FT) for recovery phase of GHR. *Degrees of freedom in F statistic are 8 and 12. If there is a significant interaction (p-value$<$0.05) in the "All metrics together" test then interaction tests per metric need to be performed.

Metric	Number of Density	Forwarding factor	Wilk's Λ	F (2,18)	*p* Value
% of Packet Losses	100	∞ & 3	0.357	16.210	0.0001
	150	∞	0.662	4.592	0.024
	150	3	0.679	4.261	0.031
	250	∞	**0.894**	**1.067**	**0.365**
	250	3	0.427	12.076	0.001
Average delay	100	∞ & 3	0.204	35.17	0.0001
	150	∞ & 3	0.169	44.11	0.0001
	250	∞	0.167	44.99	0.0001
	250	3	0.65	129.924	0.0001
Average number of hops	100	∞ & 3	0.217	32.486	0.0001
	150	∞	0.198	36.398	0.0001
	150	3	0.081	63.96	0.0005
	250	∞ & 3	0.241	28.373	0.0005
% of idle time	100	∞ & 3	0.110	72.73	0.0001
	150	∞ & 3	0.219	32.05	0.0001
	250	∞	0.234	29.478	0.0001
	250	3	0.493	9.251	0.002

Table 7.5: MANOVA [60] results of testing difference in performance metrics among routing recovery techniques (RT). There is an significant difference when p-value < 0.05.

in which differences between β values are not statistical significant for a particular metric (i.e., p-value$\geq$0.05). In these cases, the average values of the metrics are very similar and can be considered statistically the same. The rest of the results of the pairwise comparison tests, e.g. the average end-to-end delay in the three vehicle densities obtained p-values $<$

0.05 and we do not include them in the Table 7.3. Those p-values indicate that the value of the recovery factor β has an impact in the values of the metrics.

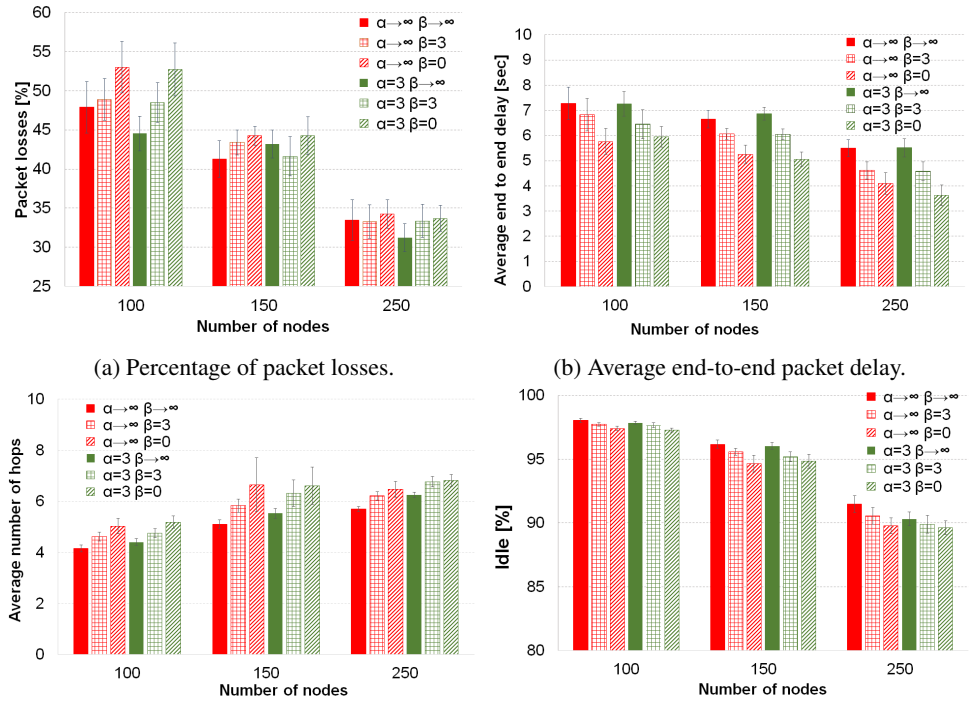
Metric	Number of Density	Forwarding Factor	Pairwise	p Value
%	150	∞	($\beta = 3$, $\beta = 0$)	0.279
packet	150	3	($\beta \to \infty$, $\beta = 0$)	0.336
losses	250	3	($\beta = 3$, $\beta = 0$)	0.562
Average number of hops	150	∞	($\beta = 3$, $\beta = 0$)	0.106
% of idle time	150	3	($\beta = 3$, $\beta = 0$)	0.068

Table 7.6: Pairwise comparison of the performance metrics in which there is a difference among recovery techniques for GHR protocol (i.e., p-value < 0.05 in Table 7.5). The Table only shows the results for metrics and pairs of recovery values with absence of statistical significant differences (i.e., p-value ≥ 0.05).

The comparison of average values of the four performance metrics with the different forwarding techniques are depicted in Fig 7.4.

The behavior of the recovery techniques in terms of packet losses can be seen in Fig. 7.4a. The *carry and forwarding* strategy ($\beta \to \infty$) is the best option for the low vehicle density. As the value of β increases, the percentage of packet losses becomes significantly higher in this density. In the intermediate density, for the classical forwarding to the best node ($\alpha \to \infty$), SA recovery and the Aggressive Recovery factor $\beta = 0$ reach the same level of packet losses (see the first row in Table 7.6) between them, but they do not improve the default *carry and forwarding* ($\beta \to \infty$). Only the simulated annealing (SA) recovery ($\beta = 3$) has a better performance than the default *carry and forwarding* for SA annealing forwarding ($\alpha = 3$) in the intermediate density. Moreover, in this scenario (150 vehicles) this approach ($\alpha = 3$, $\beta = 3$) has the same packet losses than the default selection of the best node with *carry and forwarding* ($\alpha \to \infty$, $\beta \to \infty$). In the high vehicle density, the three recovery mechanisms behave very similar with the classical forwarding ($\alpha \to \infty$). On the other hand, for the SA forwarding process ($\alpha = 3$), *carry and forwarding* ($\beta \to \infty$) is again the best strategy when SA is used. However, contrary to the low density scenario, the other two recovery values of β, which are similar between them (see the third row in Table 7.6) are close to percentage of packet losses of SA forwarding with $\beta \to \infty$.

The average end-to-end delay is completed related with the value of the recovery factor β. As it can be seen from Fig. 7.4b, when the β value decreases, the average end-to-end delay decreases as well for the two forwarding techniques. The reason is that low β values use the buffer less often than *carry and forwarding*, which is the main cause of high delays. $\beta = 0$ has the lowest delay because it only uses the buffer when there is not any legal neighbor. This Aggressive Recovery leads to a decrease around 2 seconds with respect to the *carry and forwarding* technique in the three vehicle densities. More important, this decrement comes with none or very little degradation in the percentage of packet losses for intermediate and



(a) Percentage of packet losses.

(b) Average end-to-end packet delay.

(c) Average number of hops.

(d) Percentage idle time.

Figure 7.4: Performance evaluation of the recovery techniques (RT) included in our proposal GHR. The three RT are: *Carry and forwarding* ($\beta \to \infty$), Simulated Annealing (SA) ($\beta = 3$) and Aggressive Recovery ($\beta = 0$). We compare them with two forwarding techniques (FT): best node selection ($\alpha \to \infty$) and SA ($\alpha = 3$)

high densities, respectively.

From Fig. 7.4c, we can realize that applying a recovery factor $\beta < \infty$, the average number of hops increases because the packets are forwarded more times than with *carry and forwarding*. When β decreases, the increment in the average of hops is statistically significant (see p-values$<$0.05 in the average number of hops section in Table 7.5). In fact, the only two β values that reach the same number of hops are the SA recovery ($\beta = 3$) and the Aggressive Recovery ($\beta = 0$) for classical forwarding ($\alpha \to \infty$) in the intermediate density. Notice that the differences in the average number of hops between the classical approach ($\beta \to \infty$) and the other β values examined in this chapter is up to 1.5 hops at maximum.

Regarding the percentage of idle time depicted in Fig. 7.4d, it follows exactly the same behavior described for the average number of hops. This is, while the probability to forward a packet instead of keeping it in the buffer (low β values) increases, the percentage of idle time decreases. Only in the high density scenario and for SA forwarding ($\alpha = 3$), the idle time sensed by SA recovery ($\beta = 3$) and Aggressive Recovery ($\beta = 0$) are the same. Nevertheless, the maximum difference between the conservative *carry and forwarding* and the other mechanisms is at maximum 2% for the three vehicle densities.

To summarize the performance evaluation of our Geographical Heuristic Routing (GHR) protocol, we have found that SA forwarding with *carry and forwarding* ($\alpha = 3$, $\beta \rightarrow \infty$) is the best option for low density areas because it shows the best percentage of packet losses in this vehicle density. In the intermediate density, SA forwarding and recovery ($\alpha = 3$, $\beta = 3$) obtains the same level of packet losses than traditional forwarding ($\alpha = \rightarrow \infty$, $\beta \rightarrow \infty$) with a lower delay. This makes that we choose this configuration for medium vehicle density. Finally, we consider that SA forwarding with Aggressive Recovery ($\alpha = 3$, $\beta = 0$) is appropriate for high vehicle density scenarios. This GHR setup gets the lowest average delay and only decrements around 2% the best packet delivery ratio in this density, achieved by SA forwarding ($\alpha = 3$, $\beta \rightarrow \infty$). We will use these specific configurations, which depend on the vehicle density, to compare both GHR and 2hGAR routing protocols in the next section.

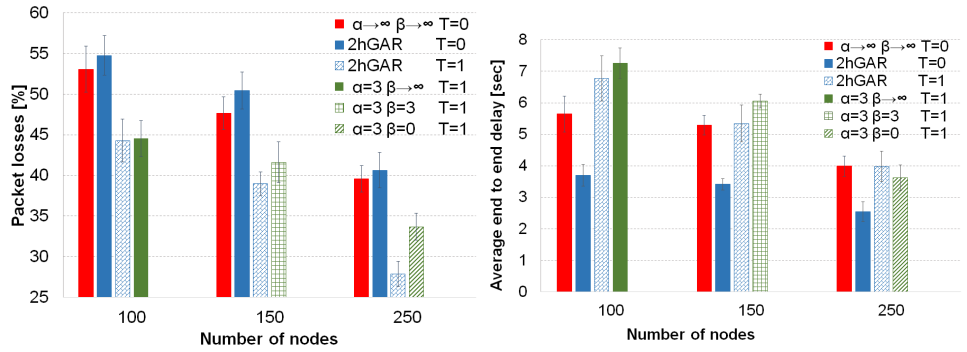
7.6.3 Performance comparison between 2hGAR and GHR

In this section we present a comparison between the best configurations of GHR for the three vehicle densities and our proposal 2-hops Geographical Anycast Routing (2hGAR). We tested 2hGAR with and without Tabu-routing enabled. Fig. 7.5 shows the performance results in the four different metrics. We have included the default operation of our routing protocol GHR (red column), which is how MMMR [113] works, to see how our proposals (i.e., GHR and 2hGAR) can improve its performance. MMMR always selects the best neighbor and applies *carry and forwarding* ($\alpha \rightarrow \infty$, $\alpha \rightarrow \infty$, no Tabu).

Since we use different configurations of GHR in the three vehicle densities, we perform our statistical analysis per vehicle density. Moreover, we have decided to perform directly the two following pairwise comparisons: (1) default configuration ($\alpha \rightarrow \infty$, $\beta \rightarrow \infty$, T=0) with 2hGAR, and (2) the best configurations of GHR with 2hGAR with Tabu routing. The reason to simplify our analysis is that 2hGAR behaves significantly different with and without Tabu in the four metrics, especially in percentage of packet losses and delay, as it can be seen in Fig. 7.5. There is also evident the difference between 2hGAR without Tabu and the best configurations of GHR. Table 7.7 show the results of the pairwise comparisons.

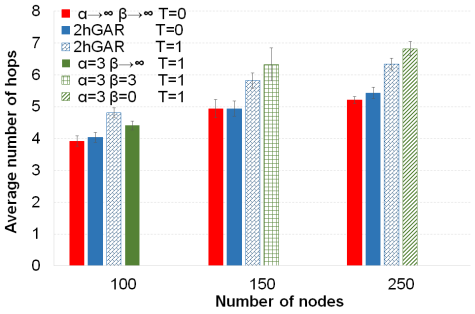
Our statistical results indicate that there is no difference in terms of packet losses between the default operation of GHR ($\alpha \rightarrow \infty$, $\beta \rightarrow \infty$, T=0) and 2hGAR without Tabu (see the three first rows of Table 7.7, where p-value $\geq$0.05). Nevertheless, there is a big improvement in the average end-to-end delay when 2hGAR without Tabu is used, around 1.5 seconds in all densities respect to default GHR (see Fig. 7.5b). Notice that this improvement is achieved even when 2hGAR and default GHR have *carry and forwarding* as recovery technique, which should increase the average delay considerably. However, the main advantage of 2hGAR is that the node is aware of a bigger surrounding area (i.e., 2-hops away from it). Hence, 2hGAR can find more suitable next forwarding nodes more often than the default operation of GHR. Then, 2hGAR does not have to store packets as many times as in the default GHR case ($\alpha \rightarrow \infty$, $\beta \rightarrow \infty$, T=0), which justifies its lower delay (see Fig. 7.5b).

Regarding the average number of hops between default GHR ($\alpha \rightarrow \infty$, $\beta \rightarrow \infty$, T=0) and 2hGAR without Tabu (2hGAR T=0), there is only a statistically significant increment of 0.25 hops in the high density scenario when 2hGAR is used (see blue and red column in Fig. 7.5c for 250 nodes). For the other two vehicle densities, both protocols use the same number of hops (see the average number of hops in Table 7.7). Nevertheless, the percentage of idle time sensed by nodes with 2hGAR with Tabu disabled is statistically significant lower

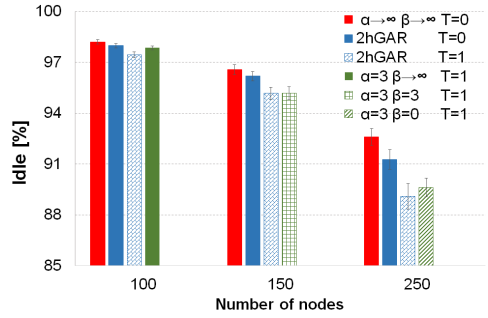


(a) Percentage of packet losses

(b) Average end-to-end packet delay.



(c) Average number of hops.



(d) Percentage idle time.

Figure 7.5: Performance comparison between the best configuration of GHR per each vehicle density and 2hGAR. The two forwarding techniques (FT) are: Best Node selection ($\alpha \to \infty$) and Simulated Annealing (SA) ($\alpha = 3$) The three recovery techniques RT are: *Carry and forwarding* ($\beta \to \infty$), the SA ($\beta = 3$) and Aggressive Recovery ($\beta = 0$). $T = 1$ indicates that Tabu routing is enabled.

than with default GHR ($\alpha \to \infty$, $\beta \to \infty$, T=0). The maximum difference in this metric is 3% for the high density scenario (250 vehicles in Fig. 7.5d). The reason is the additional overhead in the hello messages that 2hGAR needs to operate.

On the other hand, 2hGAR with Tabu (2hGAR T=1) outperforms in terms of packet losses, the best configuration of GHR in intermediate and high vehicle densities (see Fig. 7.5a). In the scenario with low vehicle density, both protocols 2hGAR and GHR ($\alpha \to \infty$, $\beta \to \infty$) with Tabu, obtain the same level of packet losses (see p-value$\geq$0.05 in Table 7.6). Particularly, in the high vehicle density, 2hGAR with Tabu reaches 27% of packet losses, which is an improvement of 7% respect to GHR with SA forwarding ($\alpha = 3$) and Aggressive Recovery ($\beta = 0$). Despite the better performance in terms of packet losses of Tabu 2hGAR, the average end-to-end delay is statistically the same as the best configurations of GHR in high vehicle density (see average delay in Table 7.6). Moreover, 2hGAR Tabu has a significant lower delay than GHR configurations in low and intermediate vehicle densities, as it can be seen in Fig. 7.5b. The larger knowledge that 2hGAR has of the network topology compared to GHR is the reason for this better performance, in packet losses and especially in average delay.

Metric	Number of Density	Tabu Factor	Pairwise	*p* Value
%	100	No	(($\alpha \to \infty$ $\beta \to \infty$), 2hGAR)	0.130
packet	150	No	(($\alpha \to \infty$ $\beta \to \infty$), 2hGAR)	0.063
losses	250	No	(($\alpha \to \infty$ $\beta \to \infty$), 2hGAR)	0.241
losses	100	Yes	(($\alpha = 3$ $\beta \to \infty$), 2hGAR)	0.852
average end-to-end delay	250	Yes	(($\alpha = 3$ $\beta = 0$), 2hGAR)	0.263
Average	100	No	(($\alpha \to \infty$ $\beta \to \infty$), 2hGAR)	0.084
number	150	No	(($\alpha \to \infty$ $\beta \to \infty$), 2hGAR)	0.995
of hops	150	Yes	(($\alpha = 3$ $\beta = 3$), 2hGAR)	0.100
% of	150	Yes	(($\alpha = 3$ $\beta = 3$), 2hGAR)	0.961
idle	250	Yes	(($\alpha = 3$ $\beta = 0$), 2hGAR)	0.198
time				

Table 7.7: Pairwise comparison of the performance metrics between GHR and 2hGAR protocols. The two FT are: best node selection ($\alpha \to \infty$) and Simulated Annealing (SA) ($\alpha = 3$) The three RT are: *carry and forwarding* ($\beta \to \infty$), the SA ($\beta = 3$) and Aggressive Recovery ($\beta = 0$). The Table only shows the results for metrics and pairs of recovery values with absence of statistical significant differences (i.e., p-value ≥ 0.05).

The behavior of the average number of hops between our proposals 2hGAR and GHR, depicted in Fig. 7.5c, depends on to the density of vehicles in the scenario. In the scenario with 100 nodes, 2hGAR needs more hops than GHR to reach destination. 2hGAR forwards the packets to other nodes instead of storing them in a buffer as GHR does because its limited awareness of the scenario. On the contrary, in the high vehicle density (250 vehicles), 2hGAR uses the information of its 2-hops neighborhood to find shorter routes than GHR. Therefore, 2hGAR becomes more efficient in terms of hops and requires 0.5 hops in average fewer than GHR ($\alpha = 3$, $\beta = 0$ T=1). The two protocols, 2hGAR and GHR tie in the number of hops in the intermediate density (see the last row of average number of hops in Table 7.6). In addition, the recovery techniques configured in GHR for the intermediate and high densities contribute to their high average number of hops.

Regarding the percentage of idle time sensed by a node, 2hGAR maintains a similar behavior than GHR. 2hGAR compensates its signaling overhead with a lower number of hops in scenarios with intermediate and high densities. Only in low vehicle density, in which GHR uses *carry and forwarding* and a low number of hops, 2hGAR uses more time the channel than GHR.

To summarize our performance evaluation, Table 7.8 shows our recommended protocol configuration depending on the vehicle density with their corresponding expected percentage of packet losses and average end-to-end delay. In our selection we have prioritized the performance of the protocols in the percentage of packet losses because our objective is a traffic reporting service, which is a kind of delay tolerant application. Nevertheless, when two protocols have very similar packet losses, we prefer those configurations that introduce lower delay in the data transmission.

Number of Density	Type of Traffic	Support of Tabu	Recomended Protocol	% Packet Losses	Average delay (s)
Low 100	Unicast	No	GHR ($\alpha = 3$ $\beta \to \infty$)	52.6	5.5
		Yes		44.5	7.25
	Anycast	No	GHR ($\alpha = 3$ $\beta \to \infty$)	52.6	5.5
		Yes	2hGAR	44.3	6.7
Medium 150	Unicast	No	GHR ($\alpha \to \infty$ $\beta \to \infty$)	47.6	5.3
		Yes	GHR ($\alpha = 3$ $\beta = 3$)	41.6	6
	Anycast	No	2hGAR	49.6	3.4
		Yes		39	5.3
High 250	Unicast	No	GHR ($\alpha \to \infty$ $\beta \to \infty$)	39.6	4
		Yes	GHR ($\alpha = 3$ $\beta = 0$)	33.5	3.6
	Anycast	No	2hGAR	40.6	2.5
		Yes		27.8	4

Table 7.8: Recommended protocol configurations of GHR and 2hGAR for different vehicle density. The two forwarding techniques FT are: Best node selection ($\alpha \to \infty$) and Simulated Annealing (SA) ($\alpha = 3$) The two recovery techniques RT are: *Carry and forwarding* ($\beta \to \infty$), the SA ($\beta = 3$). 2hGAR only uses selection based on best node and *carry and forwarding*.

Notice that for anycast applications like the reporting services in a smart city, the use of 2hGAR is preferred in intermediate and high density scenarios because of the good packet delivery ratio and low delay. 2hGAR does not reach the same level of packet losses than GHR in low vehicle density. Hence, we suggest the use of GHR in this vehicle density although the difference is not big (around 4%). If it is available, the use of Tabu mechanism is recommended for a delay tolerant application because Tabu improves the percentage of packet losses.

7.7 Conclusions

In this chapter we have shown how the greedy approach of geographical routing protocols for VANET is a direct application of the local-search heuristic, widely used in discrete optimization. In addition, we have reviewed some well known techniques in the optimization field designed to improve the behavior of greedy search, which are named as meta-heuristics [135].

Inspired on the general concept of meta-heuristics, we have proposed straightforward modifications for a generic geographical routing protocol to improve its performance. We divided our proposed heuristics according to their operation in forwarding and recovery. The forwarding heuristics are used when an improvement to reach destination is feasible. Recovery heuristic are thought for being used with *carry and forwarding* approach. Furthermore, we suggest an implementation of the Tabu method based on the recording of partial paths of packets for avoiding get into traps local minimums and loops.

We integrated the forwarding and recovery heuristics in a generic Geographical Heuristic Routing (GHR) protocol. GHR can adapt its forwarding criterion depending on the application requirements. We use the traffic-aware MMMR [113] protocol in the core of our proposal GHR to validate and score the neighbors of the current node in the forwarding process. Nonetheless, GHR could use any other routing protocol for the validation and scoring tasks.

Furthermore, in this chapter we also present our proposal called 2-hops Geographical Anycast Routing (2hGAR) protocol. 2hGAR implements a simple strategy with low signaling overhead to get information of a 2-hops neighborhood to select the best neighbor to route traffic to an anycast IP address.

An extensive performance evaluation of our both proposals GHR and 2hGAR, indicates that the use of a Tabu list contributes to improve the packet delivery ratio in around 5 to 10%. However, this better performance come at the price of additional delay (2 s) because of the more restrictive selection process of the next forwarding node. On the other hand, we show that the classical selection of the best node to forward a packet (i.e., the node with the best metric) and the *carry and forwarding* recovery are only adequate when the use of Tabu is not possible. On the contrary, if the routing process enables a Tabu list then a forwarding strategy selection based on Simulated Annealing and a recovery procedure that does not use the buffer frequently are preferred.

Finally, the main advantage of using a two hops neighborhood of 2hGAR is a considerable decrement of the average delay compared to the best GHR configurations for the different vehicle densities regardless whether Tabu routing is used or not. Moreover, 2hGAR presents a significant better packet delivery ratio in high density scenarios.

8. Optimization models for efficient RSU deployment

This chapter proposes two stochastic, mixed-integer linear optimization models to select the best positions locate road site units (RSUs)in the deployment of VANETs' fixed infrastructure. Both models take advantage of the inherent stochasticity provided by the vehicles' movements by using mobility traces to determine which are the best positions to place RSUs to maximize connectivity in a multi-hop VANET scenario while keeping the number of RSUs as low as possible. The first model mimics the routing behavior of such network and takes into account the maximum bandwidth capacity of the nodes and gateways. Since this model can be intractable by traditional optimization solver depending on size of data or routing possibilities, we include a procedure for a suboptimal solution that is close to the optimal one in terms of the number of required gateways or the shared selected gateways. The second model simplifies the first one by using pre-computed multihop connectivity information. Our simulation results validate that the solutions offered by our second model are accurate enough for the purpose of RSUs location.

8.1 Introduction

Under normal conditions vehicular ad-hoc networks need to reach fixed network infrastructure to access public information services, which make the deployment of Road Side Units (RSUs) a key factor for the operation of these networks. One of the characteristics that make the study of vehicular ad-hoc networks (VANETs) really challenging is the stochasticity introduced by the mobility of the vehicles.

In this chapter, we propose two stochastic, mixed-integer linear optimization models (SOM) [11] for the optimal placement of Road Side Units (RSUs) over a geographical area. The aim of our models is to choose the minimum number of RSU to be deployed in a specific area such that all the moving vehicles can reach some fixed infrastructure point in a multi-hop fashion regardless their position. To do that, our model does not rely in any deterministic (particular) vehicle distribution. Instead, our models use a representative set of different positions of vehicles along the time that can be extracted from real vehicle movements traces as [118], which are more trustful and are becoming more popular among the research community to test their proposals. Our models take uncertainty into account by

considering a whole set of vehicles' movements in the area to provide a solution that is the best for different vehicles's movements and more reliable than only using a deterministic vehicle distribution in the area.

Our first model emphasizes in a realistic routing behavior and in the control of the bandwidth capacity of the nodes and gateways. This optimization model has to obtain the best routing paths for each vehicle in each vehicle distribution in such way that the number of RSUs keeps low. Hence, this model can be easily untractable because the number of possible paths to be analized grows fast especially when the network is highly connected. A suboptimal procedure is included for those cases in which the model cannot be solved in a reasonable time due to the size of the solution space.

The second model avoids the routing issue of the first approach. It uses pre-computed connectivity information between vehicles and RSUs, which makes it easier to solve and able to deal with larger areas than the former approach. Nonetheless, in the second model we cannot control the bandwidth capacity of the nodes.

The rest of the chapter is organized as follows: Sec. 8.2 surveys related work. Then, Sec. 8.3 explains the first model and the alternative procedure to get a suboptimal solution. Next, Sec. 8.4 compares the optimal and suboptimal solutions of this first model. After that, Sec. 8.5 describes the second model in detail and the process to obtain the multi-hop connectivity information. Next, Sec. 8.6 presents results obtained with a solution provided by our second model in a realistic scenario. Finally, conclusions and future work are drawn in Sec. 8.7

8.2 Related work

Typically, the deployment of gateways is seen as a coverage area problem. In wireless sensor networks (WSNs) this problem is narrowly related to energy saving, connectivity, network reconfiguration and quality of service. Therefore, maximizing coverage using resource constrained nodes is a non-trivial problem. The coverage problem for WSNs has been studied extensively in recent years. In [15] the authors present a fully sponsored sensor discovery scheme called the intersection point method (IPM), which works under irregular sensing range and can efficiently increase the accuracy of the discovery method through a unit circle test. By adjusting the radius of this unit circle test, the scheme can be made tolerant to holes of a certain size, making the solution flexible when the degree of accuracy must be controlled. Hence, this solution is suitable to maintain a high coverage rate in WSN under an irregular polygon sensing range. The works [32] [61] are focused on wireless mesh networks (WMN), where [32] first formulates the Internet gateway placement as an integer linear programming incorporating QoS considerations. In [61] the authors use the normal logarithm distribution model of the shadow effect to design a weighted objective function to guarantee the node's connectivity and coverage. They also consider an heuristic tree-set partition algorithm based on the number of hops to achieve that nodes with high throughput and better connectivity acts as gateways.

On the other hand, it is well-known that the roadside infrastructure, which serves as gateway, is an important part of the VANETs assisting in tasks such as connectivity or routing. This is required in such vehicular communications, since vehicle networks frequently suffer uncertain connectivity changes. Currently, there are several studies on vehicular networks

that try to design efficient roadside infrastructure deployments, most of them focused on maximizing coverage and reducing implementation costs. For instance, in [70] the authors propose an optimization framework for road side unit (RSU) deployment and configuration. The objective in that work is to minimize the total cost to deploy and maintain the RSUs that participate in the network, playing with constraint of covering streets and maximum number of hops. Other studies take into account the maximum transmission delay. The proposal of [136] considers the movement of the vehicles and the multi-hop forwarding to study the spatial propagation of information in VANETs to model a placement strategy of the RSU with bounded delays. The authors in [33] study the effect of the position of fixed infrastructure in vehicular networks obtaining an increase in coverage of 15% if the RSU is placed in the center of intersections instead of in the corners. In [24] the authors present a geometry-based coverage strategy to solve the maximum coverage in vehicle-to-infrastructure communications in urban environments. They take into account the shape and area of road segments. Their solution uses a genetic algorithm that provides a global optimal solution for irregular regions. In [116] and [115] the authors formulate a model to deploy Dissemination Points (DP) as a maximum coverage problem (MCP). The formulation maximizes vehicle-to-DP contacts. In [21] the problem of maximizing the number of vehicles covered by the RSUs deployed in the city is modeled using a maximum coverage with time threshold problem (MCTTP). The same model was used in [117] where greedy algorithms were used to improve it. The authors conclude that the vehicular mobility is the main factor in achieving an optimal deployment of RSUs. To the best of our knowledge, no previous work have proposed a model that considers the randomness of the vehicles' movements to choose the best location for RSUs installation. which is the aim of our proposal.

8.3 A highly realistic model for RSU deployment

Our proposed linear model try to choose the minimum integer number of gateways that should be deployed in a specific area such that most of the vehicles in different movement snapshots can reach a gateway in a multi-hop VANET scenario.

We propose a two-stages stochastic optimization model with recourse [11] to deploy in an optimal way the RSUs over an area. In our problem, the first stage is represented by the subset of RSU that has to be selected prior to know the distribution of vehicles in the area. On the other hand, the best association between vehicles and the chosen RSUs is done in the second stage after the distribution of vehicles is known (when the stochasticity is disclosed). This first model emphasizes the realistic multi-hop behavior, in which nodes employ greedy approaches to reach the closest gateway. Besides, our model considers an approximation of the effective capacity of the wireless channel due to the multi-hop transmission. Moreover, our model takes into account the maximum demand that nodes and RSUs can serve. In the following subsections, the different parts of the model are explained.

8.3.1 Data Sets

Our proposal uses connectivity information between vehicles and RSU as input parameter. The following data sets and parameters are required by the model.

- **R** is the set of candidate RSUs among which our model chooses the most valuable to maximize the packet delivery ratio from vehicles to the RSU deployed. The set R

includes an artificial RSU named r_∞ to which every node can connect. If in the solution of the model a vehicle connects to this RSU, it means that this vehicle is disconnected. The use of this artificial RSU simplifies the model. The candidate gateways are already located over the map area.

- **S** set of observations of vehicles' positions S in order to consider the randomness of this factor. Each observation $s \in S$ is a snapshot of vehicles located at different positions obtained from movement traces.
- **V** the set of vehicles considered in the model. In particular V_s is the subset of vehicles which appear in the scenario s. Its cardinality is $|V|$.
- $\mathbf{L_{V_s}}$ is the set of average traffic load associated with each node $v \in V_s$. This data set is useful to test different traffic loads among nodes, for instance, the fleet of buses in the city. In this work we will use the same traffic load for all nodes and in all movements snapshots $s \in S$.
- **H** represents the set of path lengths allowed by the model to connect nodes with a RSU. In the model the maximum route length is denoted by $h_{\max} = |H|$, that is the path length from all the vehicles to the artificial RSU. No other RSU is connected to a vehicle by a path of length $h_{\max}$.
- $\mathbf{P_H}$ is a set of penalty factors associated with the path length H. Traffic loads sent through longer paths will use more bandwidth resources than through a one-hop path. In this work, we use as penalty factor the mean number of times that a message should be sent to get one successful reception as a function of the number of hops. This penalty factor follows a geometric distribution. The probabilities of a successful message reception for different path lengths were obtained from [120]. $P_{h_{\max}}$ is big enough to penalize the fact that a vehicle is not connected to a real RSU.
- **CVV** is the set of tuples $\langle s, v, v \rangle$ that represents the adjacency matrix information among client nodes $v \in V$ at each movement snapshot $s \in S$. The presence of the tuple $\langle s, v, v \rangle$ in this set means that node $j \in V$ is aware of the presence of node $k \in V$ in the movement snapshot $s \in S$.
- **CVR** is also an adjacency matrix, but between client nodes $v \in V$ and gateways $r \in R$ at each movement snapshot $s \in S$.
- $\mathbf{C_R}$ is the set of traffic load capacities associated with each candidate RSU $e \in R$. In this work we set the capacity for all the gateways to half of the channel bit rate to consider the time spent by the backoff process and ACK transmission.
- $\mathbf{Cost_R}$ is the set of installation cost for each candidate RSU $e \in R$. .

8.3.2 Variables of the model

Our model uses the following variables to determinate which gateways should be selected.

- $\mathbf{S_R}$ is a boolean variable that indicates if a gateway $g \in GW$ is chosen in the solution ($S_g = 1$) of the model or not ($S_g = 0$). The set S is the first stage decision variables in the structure of our stochastic problem.
- $\mathbf{D_{M \times N}}$ If a node $n \in N$ cannot reach a gateway in the movement snapshot $m \in M$, then $D_{m,n} = 1$. Disconnected nodes in the model are due to different reasons, for instance: nodes that do not have any node or gateway around them, or nodes that cannot be connected through the maximum established path length.
- $\mathbf{VNH_{|H-2| \times CVV}}$ This boolean variable stores the information about the next hop in

the path of a node. If the variable $VNH_{h,\langle s,j,k\rangle} = 1$ then the node k which is located at $h \in H$ hops from a gateway, is the next hop of node $j \in N$ for the movement snapshot $s \in S$.

- $\mathbf{VR_{H\times V_s\times R}}$ This variable associates a node with a selected RSU. The aim is to know how much traffic receives each gateway. For instance, $VR_{h,j,s,r} = 1$ indicates that the node $j \in V$ in the movement snapshot $s \in S$ is connected with RSU $r \in R$ at $h \in H$ hops from it. Notice that VR stores information about the path length of a node to reach the selected gateway.
- $\mathbf{VH_{H\times V_s}}$ is a boolean variable that indicates if in the solution of the model a node $v \in V$ in the movement snapshot $s \in S$ has a path length of $h \in H$ hops to reach any selected gateway $r \in RSU$. This means that $S_r = 1$.

8.3.3 The model formulation

As we mentioned, the aim of our model is to maximize the connectivity between nodes and gateways but maintaining the number of deployed RSU in a vehicular ad-hoc network as low as possible.

In the objective function, shown in Eq. (8.1), the first term adds the cost of the selected gateways. On the other hand, the second term adds the traffics of the nodes across all the movements' snapshots, taking into account the penalty factors P_h for the traffic sent to a gateway from paths oh h hops.

$$\min_{S,Rts} \quad \sum_{r\in R} S_r Cost_r + \sum_{\substack{v\in V_s \\ h\in H}} VH_{h,v} P_h L_{s,v} \tag{8.1}$$

The objective function will minimize the disconnected nodes with larger demands but using the lowest number of candidate gateways. The model also tries to connect vehicles with RSUs by employing short paths. It is worth to mention that the model will not disconnect a node to avoid the use of a gateway because the penalty factor of disconnected nodes is much higher than the capacity of a gateway. Consequently, the solution will prefer to activate gateways instead of to disconnect nodes because the latter increases the value of the objective function. Moreover, if the specific interest of the user is to detect the best positions to install the RSUs, regardless the installation cost, then the value $Cost_r$ must be the same for the whole set of candidate RSUs.

Constraints

Our proposal aims to accurately model the routing behavior of ad-hoc networks. In particular, our model considers the following aspects:

- Nodes try to connect to the closest gateway with respect a routing metric. In this proposal, the routing metric used is the number of hops, which is directly related with distance and delay.
- We assume that nodes keep a single route to a destination, i.e. in our case a gateway. This implies that all traffic that a node receives from other nodes will be forwarded to the same gateway.

The last is a strong consideration of the model, because none of the nodes can be used as a smart router that balances the traffic of its neighbors among different gateways.

Eq. (8.2) establishes that a node j can be associated with only one gateway r in every movement $s \in S$ and the path that connects both j with r has a length of h hops. This first condition includes the fact that a node can be disconnected.

$$\sum_{h \in H,\ r \in R} VR_{h,s,j,r} = 1, \qquad \forall s \in S, \forall j \in V_s \tag{8.2}$$

Eq. (8.3) allows knowing which is the path length of a node without using the four index variable VR, that also gives information about the associated gateway. This constraint binds VH variable with the value of RSU and path length selected by a node in the VR variable.

$$VR_{h,s,j,r} \leq VH_{h,s,j}, \qquad \forall s \in S, \forall j \in V_s, \forall h \in H, \forall r \in R \tag{8.3}$$

A second condition intends to allocate nodes providing a next forwarding node to each one of them except for those disconnected. In restriction of Eq. (8.4), where $A \setminus B$ indicates the set A without the elements of set B, states that if a node j is located at h hops of a gateway, then it must select only one nexthop among their neighbors k such that it must be at $h-1$ hops from a gateway. In the case of a node located at one hop ($h = 1$) from a gateway, Eq. (8.5) guarantees that those nodes must be connected to only one gateway i in the solution.

$$VH_{h,s,j} = \sum_{\substack{k \in V_s: \\ \langle s,j,k \rangle \in CVV}} VNH_{h-1,s,j,k}, \qquad \begin{array}{l} \forall h \in H \setminus \{1, h_{\max}\}, \forall s \in S, \\ \forall j \in V_s, \end{array} \tag{8.4}$$

$$VH_{1,s,j} = \sum_{\substack{r \in R: \\ \langle s,j,r \rangle \in CVR}} VR_{1,s,j,r}, \qquad \forall s \in S, \forall j \in V_s \tag{8.5}$$

Two additional constraints are needed for a proper fulfillment of the next forwarding node conditions. Eq. (8.6) states that if a node j is selected as next forwarding node by a node k which is located at $h+1$ hops, then node j must be at h hops in the movement snapshot s. Similarly, in Eq. (8.7) if a node j is located at one hop from a gateway and it selects the gateway i $\left(VR_{1,s,j,r} = 1\right)$ then RSU r must be in the set of gateways of the solution $(S_r = 1)$. This constraint guarantees that the selected RSUs in the solution are the gateways which fulfill all the other constraints.

$$VNH_{h,\langle s,k,j \rangle} \leq VH_{h,s,j}, \qquad \begin{array}{l} \forall h \in H \setminus \{h_{\max-1},\ h_{\max}\}, \\ \forall \langle s,k,j \rangle \in CVV \end{array} \tag{8.6}$$

$$VR_{s,j,1,r} \leq S_r, \qquad \forall s \in S, \forall j \in V_s, \forall r \in R \tag{8.7}$$

Constraints that mimic the behavior of routing protocols is achieved through Eq. (8.8). This condition that guarantees a node does not forward traffic of their neighbors to other gateways apart from its own gateway, avoiding a balance of traffic load. More precisely, Eq. (8.8) binds nodes j and k, which is the next hop of node j, to have the same gateway. This is because when node j is associated with gateway r $\left(VR_{h,s,j,r} = 1\right)$ if node k would be attached to another gateway g which means $VR_{h-1,s,k,g} = 1$, then the inequality of Eq. (8.8) would be violated since Eq. (8.2) forces a node to be associated with only one gateway.

$$VNH_{h-1,\langle s,j,k \rangle} + VR_{h,s,j,i} \leq VR_{h-1,s,k,i} + 1, \quad \begin{array}{l} \langle s,j,k \rangle \in CVV, \\ \forall h \in H \setminus \{1, h_{\max}\} \end{array} \tag{8.8}$$

Another important feature written in Eq. (8.9) of the proposed model is that it allows us to impose a maximum capacity load to each candidate gateway $r \in R \setminus \{r_\infty\}$, so that we can guarantee that no gateway in the solution will receive an unmanageable traffic load. This also means that there is not node v with a saturated bandwidth capacity because all transmissions that use node v as relay point, also share the same RSU. Finally, Eq. (8.10) sets the maximum number of RSUs (Max_R) that can be activated in the solution.

$$\sum_{\substack{j \in V_s, \\ h \in H \setminus \{h_{\max}\}}} VR_{h,s,j,r} P_h L_{s,j} \leq C_i, \qquad \forall s \in S, \forall r \in R \setminus \{r_\infty\} \tag{8.9}$$

$$\sum_{r \in R \setminus \{r_\infty\}} S_r \leq \text{Max}_R \tag{8.10}$$

8.3.4 A fast suboptimal solution

The proposal of this paper provides an optimal solution considering a whole set of movements snapshots for an area. However, the solution of this model can become infeasible to be obtained by an optimization solver depending on the range of the values of the data and depending on the number of variables and constraints that it has. In this section, we propose a procedure to get a suboptimal solution in a fast way when an optimization solver cannot provide a solution for our discrete network model.

Instead of getting the solution of the whole set of movements' snapshots, we propose to use the optimal solutions of the independent subsets of movements as follows:

1. To get the subsets of movements $X \in \binom{M}{k}$ obtained from all $k-$combinations of the set of movements M.
2. To solve the discrete network model for all subsets $X \in \binom{M}{k}$. Each subset X provides a solution set of gateways S with its corresponding capacity utilization U.
3. Next, we use the optimal solution of each subset X. A good and fast solution for the whole set of movements can be found in the join of all set of selected gateways S in each solution. For the cases where there is a constraint in the maximum number of gateways to $MaxGW$, the capacity utilization U of all subsets is added and the top $MaxGW$ gateways in decreasing order from joined capacity utilization, are the selected gateways in the solution.

The idea behind this procedure is similar to the idea of a greedy algorithm, in the sense that we use "local solution" (independent solutions for each subset $X \in \binom{M}{k}$) to get a global solution (for the whole set of movements snapshots M). This procedure is fast because the addition of the times required by a solver for each subset will be generally shorter than the time needed to solve the entire set M, especially for large cardinality sets, for which this approach was specially targeted. Nevertheless, the following procedure is suboptimal because, as it can be seen in Sec 8.4, it will use more gateways than needed or will not employ the same set of gateways than the optimal solution when a constraint of the maximum number of gateways is set.

We want to point out that the cardinality k of the subsets X should be as high as possible to be solved in a reasonable time by an optimization solver. If each subset X has a high number of movements snapshots $m \in M$, less "local" the solution will be. This is because each of these "local" (independent) solutions will take into account several movements' snapshots.

8.4 Comparing solutions for the realistic model

This section presents the solutions obtained by our highly realistic model and by the proposed fast and suboptimal procedure for a VANET scenario deployed over a real map layout in which vehicles reach an RSU in one hop or through other vehicles using some routing protocol.

The model were programmed and solved for the two scenarios using the IBM ILOG CPLEX Optimization Studio software [51]. We ran CPLEX over a workstation with a processor Intel core i7 at 2.8GHz with 16GB of RAM memory.

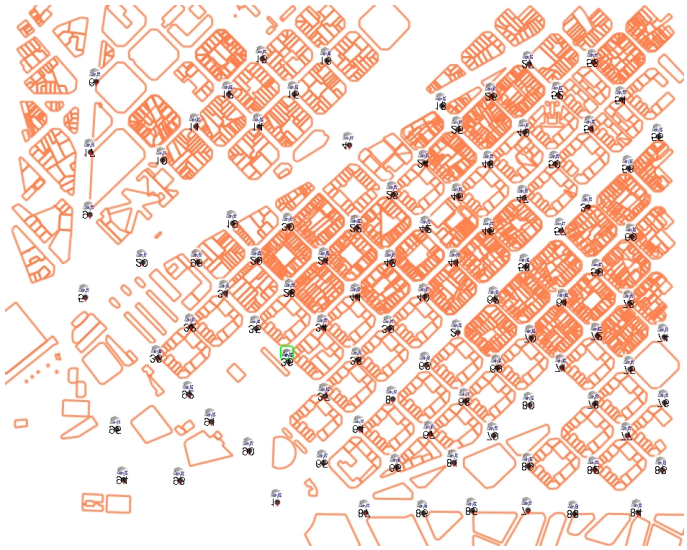


Figure 8.1: VANET 1.5 km^2 scenario. Eixample district of Barcelona. Candidate positions to locate the RSUs.

We consider 150 vehicles, which corresponds to a density of 100 vehicles per km^2. There are 95 RSUs (i.e. GWs) placed in the intersections of the streets (see Fig. 8.1). Simulation Simulations were carried out using the parameters' configuration of 3.4. We obtain the adjacency matrix in the network for the different movement snapshots from the Neighbor list of MMMR routing protocol [113]. We use the highest data rate of 27 Mbps for IEEE 802.11p [55] in the vehicles of this simulation and set their receiving sensing to -68 dbm according to the receiver performance requirement of the standard. The aforementioned characteristics lead with the highest number of RSUs required to cover an specific area because the effective transmission range of a node decreases considerately (in this case between 100 to 200m depending on the interference) compared to the low rates modulations. Hence, the adjacency matrix of the VANET scenario is very spare. There are many vehicles (around of 20%) that do not have connectivity with any other vehicle or RSU. Notice that we constrained the maximum number of gateways to be used in the solution to 30, and the maximum number of hops in a route to 5. Besides, for the solution of the model in this scenario, we use the constraint of the maximum number of gateways allowed to be used in the solution and the maximum number of hops in a route is set to 5. Taking into account the

low connectivity and the restricted number of RSUs to be deployed, the solutions provided by the “global" and “local" approaches share the following characteristics:

1. Both solutions locate the RSUs in positions with high degree of connectivity to reach as many vehicles as possible through direct connection with a gateway or through multi-hop forwarding. Notice that these positions (Active RSU in Fig. 8.2) match with the streets of more vehicular traffic flow like avenues (wider streets in Fig. 8.1).
2. A big amount of vehicles are connected through direct link to a gateway and maintain similar number of vehicles connected using the different path lengths. This is depicted in Fig. 8.3 and is consequence of the low connectivity of the network. Also, it can be seen that both solutions have roughly the same number of disconnected nodes.

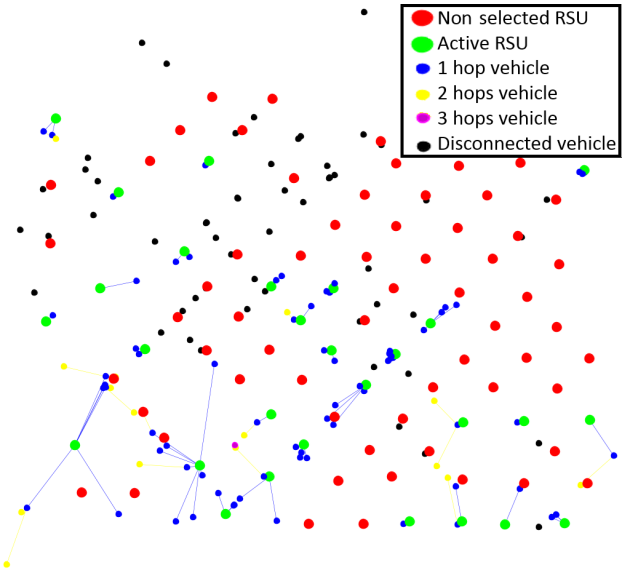
It is expected that “optimal" and “suboptimal" solutions will provide close results, given the close arrangement of vehicles among the path lengths, depicted in Fig. 8.3. Table 8.1 shows the results of this comparison. In fact, both solutions use the maximum number of RSUs and they have in common 26 out of 30 RSUs. We also include the results for an even more reduced number of allowed RSUs, when only 8 out of 15 RSUs are the same in both solutions.

Table 8.1: RSUs comparison between optimal and suboptimal solutions for the VANET scenario.

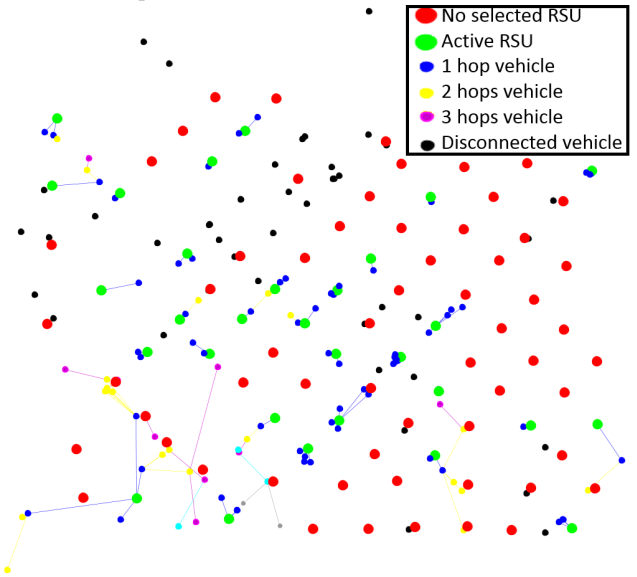
	Number of gateways in the solution		
Allowed No. of Gateways	optimal	fast suboptimal	Common
30	30	30	26
15	15	15	8
	Execution Time (hh:mm:ss)		
30	00:03:15	00:01:48	
15	00:03:24	00:01:41	

A close look to Fig. 8.4a allows us to realize that the suboptimal solution puts its “non-common" RSUs (light-blue color) in positions with more connectivity and near to other selected RSUs, while the optimal solution locates its non-shared RSUs in areas that are no covered by the suboptimal solution. The fact that the suboptimal solution covers first the crowed areas, which is more evident in Fig. 8.4b, is because the local optimal solution picks one RSU among all the set that can cover an area, and the chosen RSU is not the same in all the independent solutions. Nevertheless, this effect should be less important when the size of the subsets of movements increases.

To conclude, notice that the execution times needed to solve the model, shown in Table 8.1 depends on the scenario characteristics, more specifically in the number of routing alternatives. For instance, a scenario with a low number of nodes but highly connected might spend more time than a scenario with a higher number of nodes but with lower connections. In conclusion, there is not a fixed subset size that guarantees its solution by the model and such size depends on the trade-off between connectivity complexity and number of total nodes.



(a) Global optimal solution for the VANET scenario for the 8th movement snapshot.



(b) Local optimal solution for the VANET scenario for the 8th movement snapshot.

Figure 8.2: Comparison between global and local optimal solutions for the VANET scenario. Allowed number of RSU $\leq$ 30. Allowed number of hops$\leq$5.

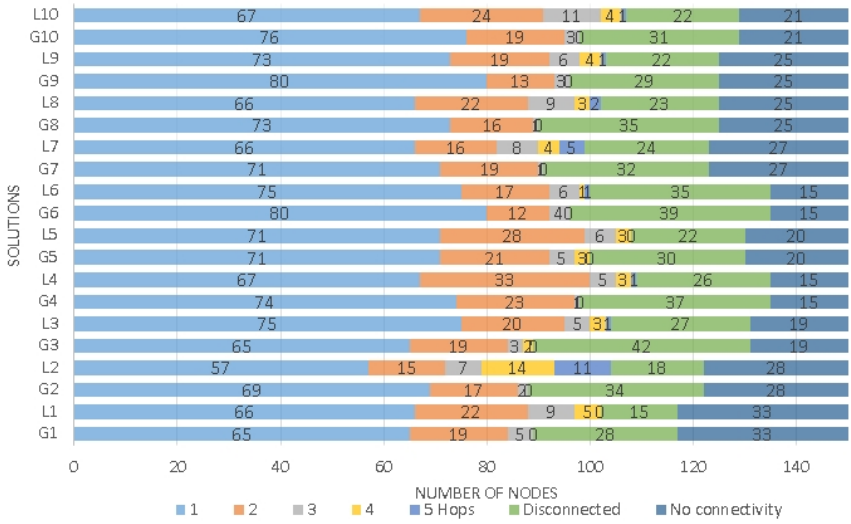
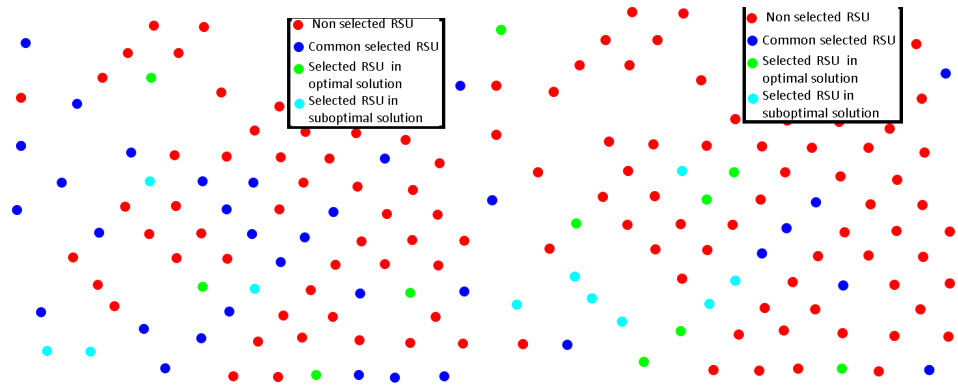


Figure 8.3: Comparison between global and local optimal solutions for the VANET scenario. Max. No. of GWs = 30. Max. No. of hops = 5.



(a) Max. No. of RSUs = 30.

(b) Max. No. of RSUs = 15.

Figure 8.4: Selected RSU comparison between optimal and suboptimal solutions for the VANET scenario. Max. No. of hops = 5.

8.5 A scalable model for the RSU deployment

Our first proposal could be too difficult to solve by traditional solvers, especially in large areas because it is necessary to use integer variables to mimic the routing behavior. In this second proposal we let the routing task out of the model solution, which makes it more scalable than the first approach presented in Sec. 8.3. On the other hand, we continue using a two-stages stochastic optimization model with recourse [11] to deploy in an optimal way the RSUs over an area. The results of our model are based on the multi-hop connectivity

information computed. In addition, the model continues considering an approximation of the effective capacity of the wireless channel due to the multi-hop transmission and takes into account the maximum traffic demand that an RSU can serve.

8.5.1 Data Set and Variables of the model

In this formulation we define CVR as the set of tuples $\langle s,h,v,r\rangle$ that provides the information about the multi-hop connectivity between vehicles (V) V and the set of candidate RSU R. The presence of the tuple $\langle s,h,v,r\rangle$ in the set CVR means that vehicle v can reach RSU r in the scenario s through h hops. Notice that $\langle s,h_{\max},v,r_\infty\rangle$ for all $v \in V_s$ are always present in the set because we consider that all nodes can reach the artificial RSU.

This second model uses only the two following variables to determine which gateways should be selected.

- **S** is a boolean variable that indicates if an RSU $r \in R$ is chosen for the solution $(S_r = 1)$ of the model. The set S is the first stage decision variables in the structure of our stochastic problem.
- **Rts** is a set of variables in the $[0,1]$ domain that associates a portion of the traffic load of a vehicle to an RSU with which it has connectivity. For instance, $Rts_{\,s,h,v,r} = 0.8$ indicates that the 80% of the traffic load that belongs to vehicle v can be received by RSU r through a route of h hops in the scenario s. Consequently, Rts plays the role of second-stage variables in the stochastic problem, which are decided for each scenario and after that the RSU will be selected.

8.5.2 The model formulation

The goal of the proposed model is to select the minimum number of RSUs to maximize the muti-hop connectivity between nodes and fixed infrastructure points. The objective function is shown in Eq. (8.11). The first term remains the same as in the first model, so the model will try to use the minimum number of RSU. On the other hand, the second term adds the whole traffic generated in the network. The model tries to connect vehicles with RSUs by employing short paths because we are imposing increasing penalty factors as a function of path lengths. Hence, the solution of the model will select RSUs easily reachable from a high number of nodes using the minimum number of hops in the different scenarios.

$$\min_{S,Rts} \quad \sum_{r\in R} S_r Cost_r + \sum_{\langle s,h,v,r\rangle \in CVR} Rts_{s,h,v,r} P_h L_{s,v} \tag{8.11}$$

The first condition in Eq. (8.12) states that the traffic load of every vehicle v of the scenarios in S has to be served by some subset of candidate RSUs reachable from the vehicle through multi-hop routing. Notice that in this subset the artificial RSU r_∞ can be included, which is reachable for all vehicles at the maximum number of hops $h_{\max}$. In this case, only the portion of the traffic served by r_∞ will be highly penalized. Also, notice that any $Rts_{s,h,v,r} = 1$ means that the whole traffic of v can be served by a unique RSU r, and this is the closest solution to the real behavior of a VANET, in which balance of traffic loads (fractional values of $Rts_{s,h,v,r}$) is unlikely.

$$\sum_{\substack{h\in H, r\in R:\\ \langle s,h,v,r\rangle \in CVR}} Rts_{s,h,v,r} = 1, \qquad \forall v \in V_s, s \in S \tag{8.12}$$

The constraint of Eq. (8.13) is related to the previous constraint and it basically establishes that if a portion of the traffic load of vehicle v is served by the RSU r (i.e., $Rts_{s,h,v,r} > 0$) then the RSU r must be included in the solution ($S_r = 1$). This is the condition that forces the model to activate RSUs in the solution and search from the best ones. Best RSUs are those that can receive as much traffic load as possible. Additionally, the reader can realize that this is the condition that binds the first stage problem (selection of RSU) with the second stage problem (maximize data transfer between vehicles and RSUs).

$$Rts_{s,h,v,r} \leq S_r, \qquad \langle s,h,v,r \rangle \in CVR \tag{8.13}$$

An important constraint of the proposed model provided the realism that it adds to the solution, is written in Eq. (8.14). This condition imposes that the maximum capacity load of each candidate RSU $r \in R$ can serve, will not be exceeded by the connected vehicles to them. This constraint does not apply to the artificial RSU used by the unserved traffic loads. However, this constraint does not deal with link saturation in a node like in the first model because it is possible that a node participates in some of the routes described in multi-hop connectivity matrix CVR.

$$\sum_{\substack{v \in V, h \in H: \\ \langle s,h,v,r \rangle \in CVR}} Rts_{s,h,v,r} P_h L_{s,v} \leq C_r, \qquad \forall s \in S, \forall r \in R \setminus \{r_\infty\} \tag{8.14}$$

The last restriction, Eq. (8.15) sets the maximum number of RSUs (Max_R) that the solution can have. If such limitation is not at stake, it can be removed of the model.

$$\sum_{r \in R \setminus \{r_\infty\}} S_r \leq \text{Max}_R \tag{8.15}$$

8.5.3 Connectivity information

In this section, we describe how to obtain the input information about multi-hop connectivity through the boolean matrix multiplication of the adjacency matrix among vehicles A_s and the adjacency matrix between vehicles and candidate RSU notated as B_s. Theses matrices represent the connectivity at 1 hop in the network. A non-zero position in this kind of matrices represents that the nodes involved can communicate between them. In particular B_s stores the information on which vehicles can communicate with RSUs directly. The same information for h hops, called $B_{s,h}$, is computed as follows:

$$B_{s,h} = A_s^{h-1} B_s \tag{8.16}$$

Notice that, $B_{s,h}$ contains information about vehicles that can connect to RSUs using from 1 to h hops. $B_{s,h}$ is the most expensive step in the process with a complexity of $O(n^3 + n^2 m)$ for each hop in each of the scenario, where n is the number of vehicles and m the number of RSUs. The connectivity matrix $C_{s,h}$, which tells us which are the vehicles that are been connected to a RSU using h hops, is obtained as:

$$C_{s,h} = B_{s,h} - B_{s,h-1} \tag{8.17}$$

Therefore, the position $C_{s,h,v,r}$ of this matrix, which indicates if the vehicle v can reach RSU r, will be 1 only the first time that it can communicate with that RSU, and 0 otherwise. The set of tuples of the CVR parameter are constructed from the non-zero positions of $C_{s,h}$ matrices. Notice that $C_{s,1} = B_s$ for each scenario $s \in S$.

8.6 Results of the scalable model

We use a synthetic movement trace generated by C4R [38] to determinate which is the best position to locate one RSU among the five candidate positions shown in Fig. 8.5 within an urban area of Barcelona. Once the model provides a solution, we remove the chosen RSU's position and solve the model again with the remaining set of candidate RSUs until this set is empty. The optimization solver that we use is CPLEX [51]. To test how well the solutions of our model behave, we compare them to simulation results from ten simulations for each one of the candidate RSUs. The simulation settings are the same described in Sec. 3.4 from Chapter 3 and use 100 and 150 vehicles. The configuration of the model is depicted in Table 8.2.

Parameter	Value
Area	1.5 km x 1 km
Nº of nodes / RSUs	150 / 5
RSUs	5
Nº hops in model	5 Hops
Simulation time	300 sec
Nº scenarios in model	20 scn, every 15 s
Transmission range	∼400 m (LOS)
Mobility generator	SUMO [65] / C4R [38]
Mobility model	Krauss modified [66, 64]
Routing protocol	MMMR [113]

Table 8.2: Simulation settings.

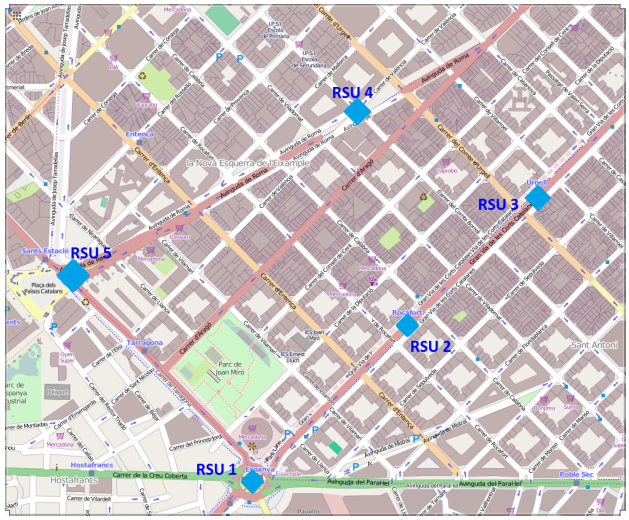


Figure 8.5: Considered scenario of Barcelona, from OpenStreetMap. There are 5 candidate locations to set the RSUs.

The locations suggested by our stochastic model to install one RSU among the candidate set depicted in Fig. 8.5 is shown in Table 8.3.

Best RSU	2nd RSU	3rd RSU	4th RSU	5th RSU
RSU 1	RSU 5	RSU 2	RSU 4	RSU 3

Table 8.3: Locations suggested by our stochastic model.

In fact, our model gives a draw between RSU 2 and RSU 4 (the value of the objective function is the same activating these RSUs). However, Table shows the best order revealed by the simulation results.

Best RSU	2nd RSU	3rd RSU	4th RSU	5th RSU
RSU 1	RSU 5	RSU 2	RSU 4	RSU 3

Table 8.4: Locations according to the simulation results.

As can be seen from Tables 8.3 and 8.4, the order suggested for our model agree with the one obtained form the simulation results. The real order is clearly manifest in both vehicle densities if we look at the performance of the packet delivery ratio (PDR) in Fig. 8.6a and the average delay in Fig. 8.6b. On the other hand, the performance difference in the average number of hops, in which our model relies, is not so clear, especially between the results provided by RSU 2 and RSU 4.

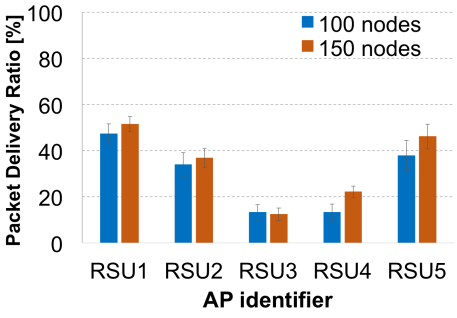
The results presented in this section validate the reliability of the solutions of our stochastic model to detect the most suitable locations to install RSUs in a city. Additionally, the results show that badly chosen positions could lead to a very poor PDR and high delays.

8.7 Conclusions

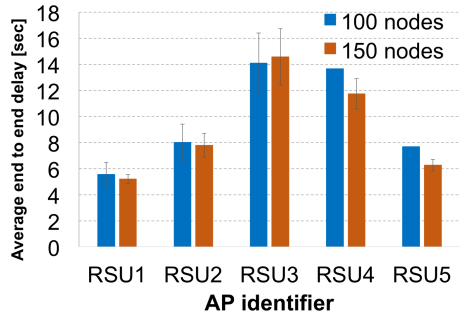
In this chapter we have proposed two stochastic, mixed-integer linear optimization models for the efficient deployment of RSUs in an area. The goal of the models are to minimize the integer number of gateways but providing connectivity to as many users as possible based on some constraints. The proposed models use different vehicles' distribution, which can be obtained from different realistic movement traces. [118].

The first model takes into account the specific routing criteria employed in a VANET to provide a realistic solution. On the other hand, these efforts come at the price of more complexity to get the solution of the model. This solution could become infeasible to obtain depending on the size of data and adjacency matrix of the nodes. To tackle this issue, we also have proposed a suboptimal procedure based on combinations of solutions of subsets of the original data. Results over a urban area of Barcelona show that the model prefers to cover first the crowded areas when there is a low connectivity among nodes. Regarding to the optimal and suboptimal solution, the results show that the suboptimal solution uses more gateways than the optimal to cover an area or that the suboptimal solution does not cover the sparest areas when there is a bound in the number of gateways to be used.

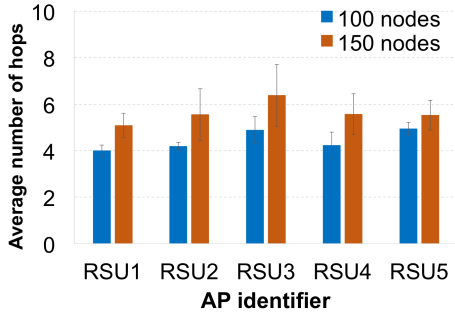
The second model lets the routing task out of the formulation. Instead, it is fed by the multi-hop connectivity information offline. Our tests suggest that our model accurate enough to detect correctly the most important positions to locate RSUs.



(a) Packet Delivery Ratio (PDR)



(b) Average end-to-end packet delay.



(c) Average number of hops.

Figure 8.6: Performance metrics results.

9. Conclusions and Future Work

This chapter presents the main conclusions reached in this Ph.D. thesis. It also includes a list of the research papers in which the results of our work have been published. Finally, we point out some future research work.

9.1 Conclusions

The main objective of this thesis was to make a contribution in the design of VANET routing protocols for urban environments. In the scenario that we consider, routing protocols in VANETs are in charge of forwarding the information generated by the vehicles to monitoring centers through multi-hop vehicular communications. All the proposals presented in this thesis aim to improve the operation and the overall performance of VANET routing protocols for reporting services.

We started our research by analyzing several important simulation aspects of a VANET to guarantee trustworthy results using our realistic simulation framework. One of our main contribution is the adaptation of heuristic techniques from discrete optimization in the decision process of geographical VANET routing protocols with successful improvements in the packet delivery ratio and end-to-end delay. In addition, we have dealt with an enhancement procedure for the address resolution necessary to initialize data transmissions. Furthermore, we have increased the available bandwidth and reduced the channel congestion by tackling the unexpected generation of duplicate packets during the data forwarding process. Finally, we have proposed models for an efficient placement of RSUs, which are the sinks that gather the vehicles' reports. We can list the main contributions of this research work as follows:

- Firstly, we have studied the geographical routing protocols in VANETs, which take forwarding decision hop-by-hop. Geographical protocols are the most suitable kind of routing schemes for VANETs. These protocols adapt to the fast topology changes and low link lifetime of VANETs by using local information and low signaling overhead. In this sense, we have proposed our Multi-Metric Map aware Routing (MMMR) protocol [113]. MMMR takes advantage of all the local information available and considers four metrics (distance, trajectory, vehicle density and available bandwidth) in the forwarding decision. This makes our MMMR adapt easier than other proposals

to the changes in the network topology and thereby MMMR improves the performance of the multi-hop VANET communication.

- Secondly, we have studied the elements that mimic the wireless channel in a vehicular communication. These components are: propagation, building attenuation and packet error models. Their correct modeling is of paramount importance to get reliable simulation results. We have compared different propagation models widely used in the VANET research. We have found that propagation models specifically designed for VANETs get close results among them and that also they are different from the results obtained by general wireless propagation models. Moreover, we have verified the relevance of the configured transmission power in nodes for the VANET simulation results despite of the maximum coverage range set in nodes.
  We have ascertained that modeling the influence of buildings on the communication between vehicles as full attenuation of the signals leads to conservative results compared to realistic models such as [104] in terms of packet losses and delay. The total block of the signal outside the road should be used when there is not available building information to be used in simulation. We also have verified little difference (around 3% in packet loses) in the performance results when pre-computed attenuation files are used in VANET simulations. Offline attenuation files reduce considerably the simulation time at the cost of using approximate attenuation values. The use of this technique is highly recommended in early research stages, where a big number of simulations need to be performed. More important, both approximate methods, i.e., total block of signal and attenuation files, can be used well regardless the vehicle density.
  Additionally, we have found that realistic packet error models affect considerably the performance results and its incorporation in network simulators should not be ignored in favor of a basic model based only on the antenna sensitivity. Our results indicate that only in low channel capacities a well set basic model and a realistic packet error get similar results. However, differences in the performance results are unavoidable in high channel capacities and these differences tend to increase with the vehicle density.

- Since traditional communication protocols were mainly thought for wired networks, most of their operations or default configurations are not suitable for very dynamic environments like VANETs. This is the case of the Address Resolution (AR) procedure. We have proposed Coherent Automatic Address Resolution (CAAR) [126] to increase the utility of hello messages of geographical routing protocols by adding the layer two address at the end of the message. This allows nodes to know at the same time MAC and IP addresses of a node and can begin a data transmission without the need to exchange AR messages. CAAR provides a significant 15% increase in the packet delivery ratio independently of the vehicle densities. Furthermore, our proposal CAAR introduces an overhead similar to the expected by the AR signaling.

- We have faced the generation of duplicate packets due to the interaction of the independent reliability mechanisms of MAC and routing layers. We have proposed two filters that can be implemented in each node independently [128]. The first and most strict filter is based on the packet ID, whereas the second filter requires the ID

of the previous hop in addition to the packet ID. The performance evaluation shows that our filtering proposals improve the overall network behavior compared to the free propagation of the duplicate packets and with a minimum degradation of the packet delivery ratio. The proposed packet filters can be applied in each vehicular node in a progressive way according to the network load changes. That is, no packet filter should be used when the link is stable and with low data traffic because a higher number of packet copies deal a better packet delivery ratio for this kind of protocols. On the other hand, when the node experiments congestion, our packet filters alleviate it.

- We have incorporated Simulated Annealing and Tabu [135] heuristics from discrete optimization to the forwarding process of our proposal called Geographical Heuristic Routing (GHR) [123]. GHR uses in the core of the routing process our Multi-Metric Map aware algorithm to qualify and discard neighbors. However, GHR is a generic strategy that could implement any other algorithm to choose the next forwarding node. Our results indicate that a forwarding scheme based on the selection of the best node and on carry and forwarding is not the best alternative for geographical routing protocols despite of its good performance. Conversely, a certain level of randomness in the forwarding and recovery phase of the routing operation lead to better performance in percentage of packet losses and average end-to-end delay.
  We have also proposed our 2-hops Geographical Anycast Routing (2hGAR) protocol. 2hGAR implements a lightweight signaling procedure to gather routing information of a 2-hops neighborhood. The wider knowledge of 2hGAR helps to take better forwarding decision compared to the simple case of using only local (1 hop) information. Our results indicate that 2hGAR presents lower delays than GHR and an increase in the packet delivery ratio, especially in high density scenarios. It is worth noting that the use of a Tabu list increases the packet delivery ratio for both protocols. Tabu adds information to the forwarding decision that helps to avoid loops.

- Finally, we have proposed two mixed-integer linear optimization models to find the best positions to install the Road Side Units (RSUs) in charge of gathering all the reporting information generated by vehicles. Both models suggest the positions where RSUs should be located considering the inherent stochasticity provided by the vehicular movement. Simulation results show that our second optimization model detects the most valuable positions to locate RSUs.

9.2 Publications derived from this research work

This thesis has been developed within the framework of several Spanish R&D projects, in particular, TEC2010-20572-C02-02 Continuity of Service, Security and QoS for Transportation Systems “CONSEQUENCE”, TEC2013-47665-C4-1-R EMergency Response In Smart COmmunities. Privacy and QoS “EMRISCO” and TEC2014-54335-C4-1-R INcident monitoRing In Smart COmmunities. QoS and Privacy. “INRISCO”. Also, this research was supported by the SENESCYT of Republic of Ecuador under contract 217-2012. Most of the research results presented in this dissertation have been published in journals and conferences.

In this section, we list the publications that have been generated from the research work done in this thesis. Also, the code developed is available at:

`http://www.lfurquiza.com/research/`

JCR Journal Publications

- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba, Isabel Martín & Mónica Aguilar, "Propagation and Packet Error models in VANET simulations", Latin America Transactions, IEEE (Revista IEEE America Latina) 12(3), pp. 499–507 (May 2014), `http://dx.doi.org/10.1109/TLA.2014.6827879`. [120]
- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba, David Rebollo-Monedero, Ahmad Mohammed Mezher, Mónica Aguilar-Igartua & Jordi Forné, "Coherent, Automatic Address Resolution for Vehicular Ad Hoc Networks", Int. J. Ad Hoc Ubiquitous Comput. (*In press*), pp. 1–18 (Accepted in February 2015), `http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijahuc`. [126]
- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba & Ángel Romero Muir, "Mitigation of packet duplication in VANET unicast protocols", Ad Hoc Networks, pp. 1–21 (January 2016), *Submitted*. [128]
- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba & Mónica Aguilar Igartua, "Performance Evaluation of a Geographical Heuristic Routing Protocols for VANETs in urban scenarios", Journal of Network and Computer Applications, pp. 1–24 (January 2016), *In preparation*. [123]
- Carolina Tripp-Barba, **Luis Urquiza-Aguiar**, Mónica Aguilar Igartua, David Rebollo-Monedero, Luis de la Cruz Llopis, Ahmad Mohammed Mezher & José Aguilar-Calderón, "A multimetric, map-aware routing protocol for VANETs in urban areas", Sensors (Basel, Switzerland) 14(2), pp. 199–224 (January 2014), `http://dx.doi.org/10.3390/s140202199`. [113]
- Carolina Tripp Barba, **Luis Urquiza Aguiar** & Mónica Aguilar Igartua: "Design and evaluation of GBSR-B, an improvement of GPSR for VANETs", IEEE Latin America Transactions 11(4), pp. 1083 – 1089 (June 2013), `http://dx.doi.org/10.1109/TLA.2013.6601753`. [111]

ESCI Journal Publication

- **Urquiza-Aguiar**, Carolina Tripp-Barba, José Estrada-Jiménez & Mónica Aguilar Igartua, "On the Impact of Building Attenuation Models in VANET Simulations of Urban Scenarios", Electronics 4(1), pp. 37–58 (January 2015), `http://dx.doi.org/10.3390/electronics4010037`. [125]

Refereed Conferences

- **Luis Urquiza-Aguiar**, Andrés Vázquez-Rodas, Carolina Tripp-Barba, Ahmad Mohammed Mezher, Mónica Aguilar Igartua & Luis de la Cruz Llopis, "Efficient Deployment of Gateways in Multi-hop Ad-hoc Wireless Networks", In: Proceedings

of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN '14), pp. 93–100. ACM Press, Montreal, QC, Canada (September 2014), http://doi.acm.org/10.1145/2653481.2653487. [130]

- Carolina Tripp-Barba, **Luis Urquiza-Aguiar**, José Estrada, José Aguilar-Calderon, Anibal Zaldivar-Colado & Mónica Aguilar Igartua, "Impact of packet error modeling in VANET simulations", In: 2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST '14), pp. 1–7. IEEE, Ota, Nigeria (October 2014), http://dx.doi.org/10.1109/ICASTECH.2014.7068133. [114]
- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba, José Estrada-Jiménez & Mónica Aguilar Igartua, "Empirical Analysis of the Minkowski Distance Order in Geographical Routing Protocols for VANETs", In: Wired/Wireless Internet Communications SE - 24, Lecture Notes in Computer Science, vol. 9071, pp. 327–340. Springer International Publishing, Málaga, Spain (May 2015), http://dx.doi.org/10.1007/978-3-319-22572-2_24. [124]
- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba & Ángel Romero, "Reducing Duplicate Packets in Unicast VANET Communications", In: Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN '15), pp. 1–8. ACM Press, Cancun, Mexico (November 2015), http://doi.acm.org/10.1145/2810379.2810387. [127]
- **Luis Urquiza-Aguiar**, Daniel Almeida, Carolina Tripp-Barba & Mónica Aguilar Igartua, "Heuristic Methods in Geographical Routing Protocols for VANETs", In: Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN '15), pp. 41–48. ACM Press, Cancun, Mexico (November 2015), http://doi.acm.org/10.1145/2810379.2810394. [121]
- **Luis Urquiza-Aguiar**, Carolina Tripp-Barba & Mónica Aguilar Igartua, "A Stochastic Optimization Model for the Placement of Road Site Units", In: EAI International Conference on Smart Objects and Technologies for Social Good, pp. 1–6. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer International Publishing, Rome, Italy (2016), *Publication pending*. [122]
- **Luis Urquiza-Aguiar**, Andrés Vazquez-Rodas, Carolina Tripp-Barba, Mónica Aguilar Igartua, Luis de la Cruz Llopis & Emilio Sanvicente Gargallo, "MAX-MIN based buffer allocation for VANETs", In: Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on, pp. 1–5. Vancouver. BC, Canada (September 2014), http://dx.doi.org/10.1109/WIVEC.2014.6953231. [129]
- Ahmad Mohammed Mezher, Juan Jurado Oltra, **Luis Urquiza-Aguiar**, Cristhian Iza Paredes, Carolina Tripp Barba & Mónica Aguilar Igartua, "Realistic environment for VANET simulations to detect the presence of obstacles in vehicular ad hoc networks", In: Proceedings of the 11th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks (PE-WASUN '14), pp. 77–84. ACM Press, Montreal, QC, Canada (September 2014), http://doi.acm.org/10.1145/2653481.2653488 [80].

9.3 Future work

Research in Vehicular Ad hoc Networks has experimented a continuous and important increase during the last years. During the development of the contributions that compose this thesis, several issues attracted our attention and now they constitute future research lines.

- Our proposal 2-hops Geographical Heuristic Routing (2hGAR) protocol [123] needs further research regarding the valid time of the hello messages. Since hello messages contain information of 2 hops, that topological information could change easily and therefore the nodes could use obsolete information in their forwarding decisions. We will investigate strategies that can extended the valid period of the information, including a variable generation time of hello messages.

- Anycast groups is one of the main features of IPv6 and its usefulness in vehicular reporting services is clear. However, how a node should select the specific member of the anycast group to which it would forward the information requires further investigation. This could result in a significant improvement in the performance of anycast applications.

- All our work assumes a VANET which is already operating. That is, we assume that all the vehicles are properly configured with a unique IP addresses or other equivalent identifier. However, to the best of our knowledge there are only few works regarding the nodes' configuration process. Future work may be oriented to do a comparative study of the auto-configuration address mechanisms by using a network simulator. Such study would include an analysis of parameters like average time to obtain a valid configuration, number of auto-configuration attempts, amount of signaling traffic introduced by the mechanism, among others. Also, we are interested in the impact of the IP configuration changes in the performance of routing protocols.

- There are many proposals about power control in ad hoc networks. It would be interesting to evaluate the performance of our proposals in conjunction with such mechanisms. An expected advantage is a more available channel because there would be lower collisions. However, power control algorithms should be fast enough to adapt to the rapid topology changes inherent in vehicular environments.

- A recent work [63] explores the use of multiple channel, supported by WAVE [54], to reduce channel utilization and maintain the throughput of data dissemination. We are planning to carry a similar analysis for routing protocols.

Appendix A. Evaluation of the Minkowski distance

We carry out an empirical study of how distance between two nodes is measured and its impact in the performance of a geographical routing protocol for VANETs. The distance equations used in this work are obtained by setting the order parameter of the Minkowski distance function. Simulation results from the topology of a real city indicate that the use of dominant distance could improve some classical performance metrics like the packet delivery ratio, average number of hops or end-to-end packet delay. Nevertheless, these results are quite close to the ones obtained with the classical and widely accepted Euclidean distance.

A.1 Introduction

Vehicular *ad hoc* networks (VANETs) [47] face particular challenges compared to MANETs, such as faster topology changes, lower link lifetimes or a potentially greater number of nodes taking part in the network, among others. Particularly, the two former features have encouraged researches to propose new routing protocol for VANETs that do not need to construct end-to-end paths and that make their routing decision based only on local information.

Geographical routing protocols have emerged as an alternative to the classical topological routing approach. This kind of routing protocols considers the geographical (Euclidean measured) distance between nodes in their forwarding criteria. On the other hand, the Minkowski distance function [14] provides a general equation to measure the level of dissimilarity between two points. The Euclidean distance is a particular case of the Minkowski distance function. In this work, we analyze the performance impact of a distance-based VANET routing protocol, when it employs distance equations (obtained from the Minkowski distance function) different from the Euclidean one in the computation of its routing metric, when selecting the next hop.

The rest of this work is organized as follows: Sec. A.2 introduces the Minkowski distance family. Then, Sec. A.3 describes how using a different distance function changes the forwarding decision made by a distance-based VANET routing protocol. Next, Sec. A.4 is devoted to the evaluation of different Minkowski distances in the GBSR [111] and the results

obtained from the statistical tests. Finally, conclusions are drawn in Sec. A.5.

A.2 Minkowski distance

A distance function δ for two n-dimensional points x and y measures how far they are from each other. This is also known as the level of dissimilarity between these two points. A distance measure $\delta(x,y)$ satisfies:

$$\delta(x,y) = \delta(y,x) \tag{A.1a}$$
$$\delta(x,y) \geq 0 \tag{A.1b}$$
$$\delta(x,x) = 0 \tag{A.1c}$$

The Minkowski distance [14] of order r between the points x and y is defined as:

$$\delta_r(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{1/r} \tag{A.2}$$

If $r < 0$, the Minkowski distance function δ (A.2) can be seen as a similarity measure instead of quantifying how different are two points. Particular cases of the Minkowski distance family are the Manhattan and Euclidean distances. These distances are obtained with the order r equal to 1 and 2, respectively, in the Minkowski distance function δ. When the order $r \to \infty$ the Minkowski distance function is:

$$\delta_r(x,y) = \lim_{p \to +\infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = \max_{i=1}^{n} |x_i - y_i| \tag{A.3}$$

This distance is called “dominant" because its value is equal to the maximum of the absolute value of the differences between their components x and y.

Figure A.1 shows all the points that are at a distance of 1 from the center, which is the definition of a circle in Euclidean distance ($r = 2$). Notice how that “circle” grows progressively until reaching the square form in the infinity ($r \to +\infty$). This is because when r increases, the influence of the highest component $|x_i - y_i|^r$ in Eq. A.2 increases notably compared to the other components in the distance computation. On the other hand, when $r < 2$, the area defined by the perimeter is smaller than the Euclidean case ($r = 2$).

A.3 Minkowski distance in geographical distance routing metric

Geographical routing protocols mostly base their forwarding decision on the geographical distance from their neighbors to destination. On the other hand, the Minkowski distance function provides a whole family of distances to measure the dissimilarity between two points. This section provides a short explanation of how the forwarding decision is affected by considering alternative ways to measure the distance between two points.

The use of order $r \neq 2$ (Euclidean distance) in the Minkowski distance function will affect the operation of a geographical routing protocol in the following parameters:

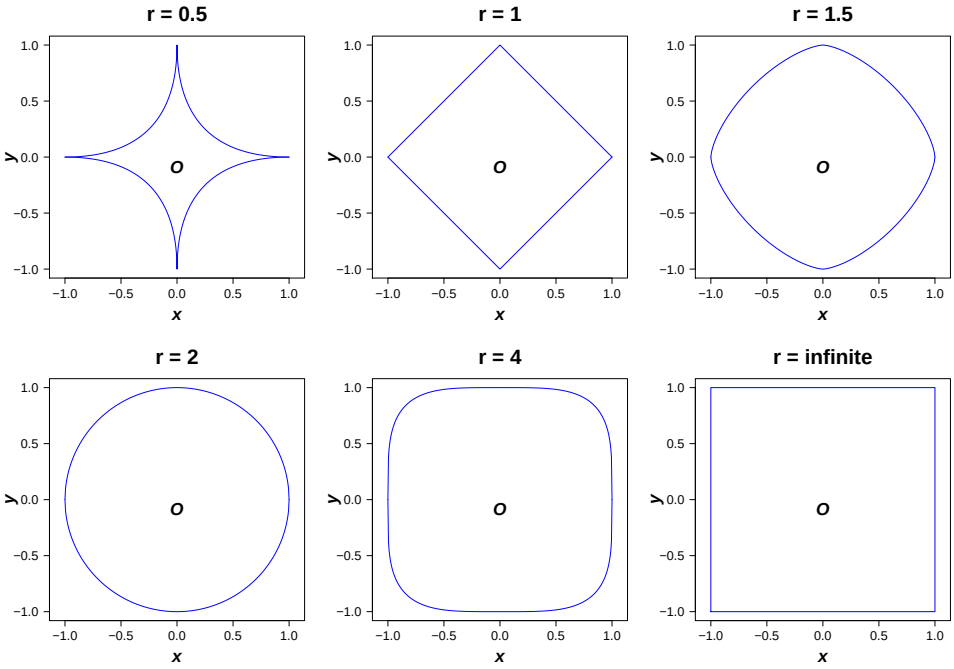


Figure A.1: Points in a 2-dimensional space at a distance of 1 from the center (O) using the Minkowski distance function with different values of the order r. Special cases are: Manhattan distance (r=*1*), Euclidean distance (r=*2*) and Dominant distance ($r \rightarrow +\infty$).

1. The size and form of the searching area to find the next forwarding node.
2. The decision of which neighbor is the closest to destination.

To give an example, Figs. A.2 and A.3 show a comparison among the use of Euclidean distance ($r = 2$), Manhattan ($r = 1$) and dominant distance ($r \rightarrow +\infty$), respectively.

As it can be seen in both figures, the Euclidean distance ($r = 2$) between source and destination is the radius of the circle that contains all the nodes closer than the source to the destination, which are called next-hop candidates. For the case of the Manhattan distance ($r = 1$), this area is a diamond. For dominant distance ($r \rightarrow +\infty$) the area has a square shape. The selected next forwarding node will be in the intersection area that contains the next hop candidates that are also within the coverage area of the source (pink circle around source node in Figs. A.2 and A.3). Notice that this area changes depending on the Minkowski order r. For instance, in the aforementioned figures, red areas mean searching zones only valid if we consider Manhattan distance (Fig. A.2) or dominant distance (Fig. A.3), but not for Euclidean distance. Conversely, the blue areas, are searching regions only valid for the Euclidean distance.

In both Figs. A.2 and A.3, node a is closer than node b to destination, according to the Euclidean distance ($r = 2$). Nonetheless, it is just the opposite if the routing protocol employs the other Minkowski distance orders ($r = 1$ or $r \rightarrow \infty$) to make the routing decision. Summarizing, the previous two examples show how the use of other Minkowski distance

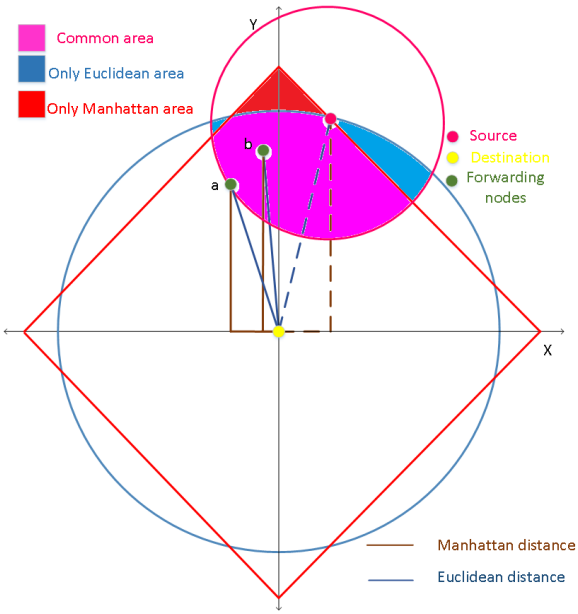


Figure A.2: Euclidean distance ($r = 2$) *vs.* Manhattan distance ($r = 1$).

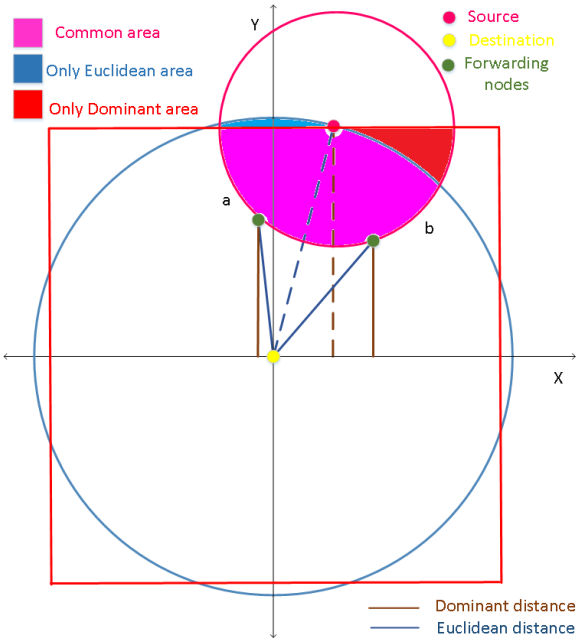


Figure A.3: Euclidean distance ($r = 2$) *vs.* dominant distance ($r \to +\infty$).

order values r, will affect the influence area where the next hop is selected. Moreover, the selected next forwarding node will not be the same as the chosen one with the Euclidean distance in most of the cases.

A.4 Empirical Analysis

The simulation scenario consists of a multi-hop VANET, where we analyzed the impact of the order parameter r of the Minkowski distance function in the routing operation of our distance-based proposal GBSR [111]. To do this, we use the scenario and simulation settings described in Sec.3.4 from chapter 3. We use an area of 1.5 km^2 and 100 and 150 vehicles, which correspond to vehicle densities of 67 and 100 veh/km^2.

All the figures are presented with confidence intervals (CI) of 95%, obtained from 20 simulations per each density value and order parameter r using different movement traces per each simulation.

Figures A.4, A.5 and A.6 depict the results of the percentage of packet losses, the average packet delay and the average number of hops for the packets to reach destination, respectively.

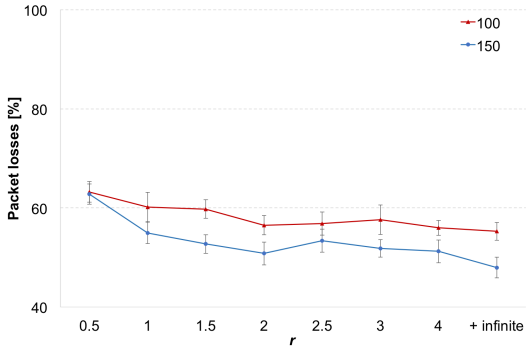


Figure A.4: Percentage of packet losses.

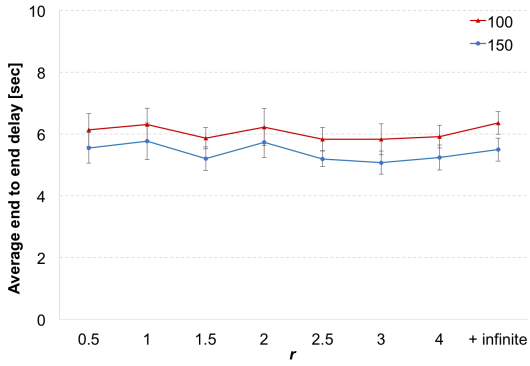


Figure A.5: Average end-to-end packet delay.

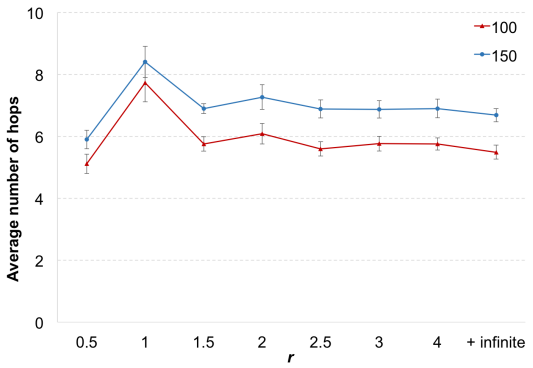


Figure A.6: Average number of hops.

As it can be seen from Fig. A.4, when the routing protocol uses the Minkowski distance function (see. Eq. A.2) with $r < 2$, the increase in the percentage of packet losses is considerable compared to the losses when $r = 2$ for both vehicle densities. Notice that small r values implies that each $|x_i - y_i|^r$ component has a similar contribution in the computation of the Minkowski distance $\delta_r(x,y)$. Notice also that, for these values of r (i.e., $r \in \{0.5, 1, 1.5\}$), the average end-to-end delay in Fig. A.5 are similar among them. However, the averages number of hops are very different (see Fig. A.6) in the three cases. The measure of distance with $r = 1.5$ produces slightly shorter routes than the ones obtained with Euclidean distance ($r = 2$). Manhattan distance $r = 1$ has by far the longest average number of hops in a route. This fact explains the high percentage of packet losses for Manhattan distance, because more hops entails higher chances of packet collision or packet error reception. On the contrary, the Minkowski distance with $r = 0.5$ has the shortest average number of hops, at the cost of the highest packet losses. These facts could be explained due to the shape of the searching area for $r = 0.5$ (see Fig. A.1), which in most of the cases will differ more than any other searching area from the Euclidean circle.

To analyze the performance metrics for $r \geq 2$, we employ statistical tests because the relationship of the results among the distances is not so evident as when $r < 2$. For each density of vehicles we will use the pairwise Wilcoxon statistical test [101] to check whether the differences between the results obtained with all the $r \geq 2$ and those coming from the simulation with Euclidean distances are statistically significant. Tables A.1, A.2 and A.3 summarize the results of this test for three performance measures (i.e., percentage of packet losses, average end-to-end delay, and average number of hops in a path), and grouped by the two vehicle density. The outcome of a statistical test is a probability called the p-value (fourth column of tables. A.1, A.2 and A.3), which is compared with a threshold named the significance level. If the p-value is lower than the significance level, then the difference between both performance metrics are statistically significant (fifth column in the tables). We employed a significance level on each test of 0.025, to obtain an overall error probability among the four pairwise comparison per metric of 0.10 (i.e., 0.025×4). The results of the statistical tests indicate that when there is a low vehicle density scenario (100 vehicles), there is no improvement or degradation on the percentage of packet losses and average end-to-end packet delay compared with the results obtained using the Euclidean distance (see

tables. A.1 and A.2). However, there is a statistically significant difference in the average path length when the routing protocol GBSR employs a Minkowski distance with order $r = 2.5$ or the dominant distance ($r \to +\infty$), as it can be seen in the fifth and eighth rows of table A.3. It means that the use of $r \in \{2.5, +\infty\}$ in our routing protocol GBSR for our low-density scenario provides the same quality of service (packet losses and delay) through shorter paths.

Vehicle Density	**Pairwise (*r*,2)**	**Standardized Test Statistic**	***p*-Value 1 Side**	**Is the Difference Significant (*p*-Value $<$ 0.025)?**	**Median of Differences**
100	2.5	-0.336	0.378	No	1.088 %
	3	-0.896	0.194	No	2.52%
	4	-0.485	0.324	No	0.158 %
	$+\infty$	-1.381	0.088	No	-1.441%
150	2.5	-2.091	0.018	Yes	2.549%
	3	-0.971	0.174	No	1.024%
	4	-0.299	0.392	No	1,738 %
	$+\infty$	-2.24	0.012	Yes	-3.096 %

Table A.1: *p*-values of Wilcoxon signed rank test for a pairwise comparison of the effect of the Minkowski distance order *r* for the packet losses metric.

Vehicle Density	**Pairwise (*r*,2)**	**Standardized Test Statistic**	***p*-Value 1 Side**	**Is the Difference Significant (*p*-Value $<$ 0.025)?**	**Median of Differences**
100	2.5	-1.232	0.115	No	-0.483 s
	3	-1.307	0.101	No	-0.341 s
	4	-1.083	0.147	No	-0.158 s
	$+\infty$	-0.859	0.205	No	0.037 s
150	2.5	-2.427	0.007	Yes	-0.500 s
	3	-2.763	0.002	Yes	-0.584 s
	4	-2.203	0.013	Yes	-0.409 s
	$+\infty$	-1.269	0.108	No	-0.186 s

Table A.2: *p*-values of Wilcoxon signed rank test for a pairwise comparison of the effect of the Minkowski distance order *r* for the average end-to-end packet delay metric.

For the intermediate vehicle density scenario (150 vehicles), our results indicate that only when GBSR used the dominant distance to make the forwarding decision, the percentage of packet losses shows a statistically significant reduction (around 3%, see table A.1) compared with GBSR employing classical Euclidean distance. This packet losses reduction is obtained without increasing the average end-to-end delay, which remains statistically equal (eighth row in table A.2). Moreover, this reduction of packet losses and similar delay, using the dominant distance comes with the use of shorter paths than the created by GBSR using the Euclidean distance (around 0.18 hops, see table A.3). In the cases of $r = 3$ and $r = 4$, both distances do not have noticeable differences with the Euclidean in the percentage of packet losses. Nonetheless, regarding to the average end-to-end and path length, there are statistically significant improvements as shown in rows sixth and seventh in tables A.2 and A.3. The

Vehicle Density	Pairwise (r,2)	Standardized Test Statistic	p-Value 1 Side	Is the Difference Significant (p-Value < 0.025)?	Median of Differences
100	2.5	-2.837	0.002	Yes	-0.367 hops
	3	-1.680	0.049	No	-0.1635 hops
	4	-1.867	0.03	No	-0.228 hops
	$+\infty$	-3.173	0.0005	Yes	-0.515 hops
150	2.5	-1.829	0.035	No	-0.337 hops
	3	-2.165	0.015	Yes	-0.0136 hops
	4	-1.979	0.024	Yes	-0.66 hops
	$+\infty$	-3.323	0.0005	Yes	-0.11 hops

Table A.3: p-values of Wilcoxon signed rank test for a pairwise comparison of the effect of the Minkowski distance order r for the average number of hops metric.

Minkowski's order $r = 2.5$ in the intermediate vehicle density for the percentage of packet losses provides worse marks than the obtained with the classical Euclidean distance and only improves the average end-to-end packet delay.

To conclude with, the simulation results show that it is possible to improve some performance metrics of our geographical routing protocol GBSR, if a different distance equation is used by the routing protocol. Particularly, the dominant distance ($r \to +\infty$) equation outperforms the traditional results ($r = 2$) in terms of packet losses and average number of hops for intermediate vehicle density and only the latter in low density areas. However, the result differences in the performance metrics between Euclidean and other distances are low.

A.5 Conclusions and Future work

In this work we have tested the impact of using the equations of different distance definitions in the forwarding decision of a geographical routing protocol for VANETs. The distance equations were obtained through the Minkowski distance function, modifying the value of its order parameter. Our results in a realistic urban scenario, indicate that for low and intermediate vehicle densities, the use of the dominant distance ($r \to +\infty$) in the routing decision leads to the creation of paths that are shorter than the ones obtained by the routing protocol employing the traditional Euclidean distance ($r = 2$). Moreover, for the intermediate density (150 vehicles), the routing protocol improves its packet delivery ratio without increasing the packet delay. Also, for intermediate densities, the Minkowski's orders 3 and 4 are able to reduce the end-to-end packet delay keeping the same level of packet losses. Nevertheless, our results show that the performance of Euclidean distance and the best ones obtained by other Minkowski r values are quite similar (around 3% of difference). Thus, we can conclude that the traditional Euclidean distance can be used safety instead of other definitions of distances.

Bibliography

[1] Abrate, F., Vesco, A., Scopigno, R.: An Analytical Packet Error Rate Model for WAVE Receivers. In: 2011 IEEE Vehicular Technology Conference (VTC Fall). pp. 1–5. IEEE (Sep 2011), http://dx.doi.org/10.1109/VETECF.2011.60930939

[2] Ahammed, F., Taheri, J., Zomaya, A.: Using simulated annealing to find lower bounds of localization with noisy measurements. In: Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International. pp. 601–608 (May 2012), http://dx.doi.org/10.1109/IPDPSW.2012.75

[3] Ahammed, F., Taheri, J., Zomaya, A., Ott, M.: Vloci2: Improving 2d location coordinates using distance measurements in gps-equipped vanets. In: Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. pp. 317–322. MSWiM '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2068897.2068952

[4] Ahammed, F., Taheri, J., Zomaya, A., Ott, M.: Vloci: Using distance measurements to improve the accuracy of location coordinates in gps-equipped vanets. In: Proceedings of the Int. ICST Conf. on Mobile and Ubiquitous Systems. pp. 149–161 (2012)

[5] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38, 393–422 (2002)

[6] Appleby, A.: Murmur hash, https://sites.google.com/site/murmurhash/, [Last Accessed: 2016-01-07]

[7] Baguena, M., Calafate, C., Cano, J.C., Manzoni, P.: An adaptive anycasting solution for crowd sensing in vehicular environments. Industrial Electronics, IEEE Transactions on 62(12), 7911–7919 (Dec 2015)

[8] Balador, A., Calafate, C., Cano, J.C., Manzoni, P.: Congestion Control for Vehicular Environments by Adjusting IEEE 802.11 Contention Window Size. In: Aversa, R., Kołodziej, J., Zhang, J., Amato, F., Fortino, G. (eds.) Algorithms Archit. Parallel

Process. SE - 30, Lecture Notes in Computer Science, vol. 8286, pp. 259–266. Springer International Publishing (2013), http://dx.doi.org/10.1007/978-3-319-03889-6_30

[9] Baltzis, K.B.: On the Effect of Channel Impairments on VANETs Performance. RadioEngineering 19(4), 689–694 (2010)

[10] Bernardos, C., Calderón, M., Moustafa, H.: Survey of IP address autoconfiguration mechanisms for MANETs. Internet-Draft, RFC Editor (June 2010), http://tools.ietf.org/html/draft-bernardos-manet-autoconf-survey-05

[11] Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering, Springer (1997)

[12] Bitam, S., Mellouk, A., Zeadally, S.: HyBR: A Hybrid Bio-inspired Bee Swarm Routing Protocol for Safety Applications in Vehicular Ad Hoc Networks (VANETs). Journal of Systems Architecture 59(10, Part B), 953 – 967 (2013), http://dx.doi.org/10.1016/j.sysarc.2013.04.004

[13] BMW: BMW Technology Guide: Car-to-car communication, http://www.bmw.com/com/en/insights/technology/technology{_}guide/articles/cartocar{_}communication.html, [Last Accessed: 2016-01-29]

[14] Borg, I., Groenen, P.: Modern Multidimensional Scaling - Theory and Applications. Springer New York, New York, second edn. (2005)

[15] Boukerche, A., Fei, X., Araujo, R.B.: A coverage-preserving scheme for wireless sensor network with irregular sensing range. Ad Hoc Networks 5(8), 1303 – 1316 (2007), recent Research Directions in Wireless Ad Hoc Networking

[16] Braden, R.: Requirements for Internet Hosts – Communication Layers Status. RFC 1122, RFC Editor (October 1989), http://www.rfc-editor.org/rfc/rfc1122.txt

[17] Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols. In: Proceedings of the 4th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking. pp. 85–97. MobiCom '98, ACM, New York, NY, USA (1998), http://doi.acm.org/10.1145/288235.288256

[18] Bychkovsky, V., Hull, B., Miu, A., Balakrishnan, H., Madden, S.: A Measurement Study of Vehicular Internet Access Using in Situ Wi-Fi Networks. In: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking. pp. 50–61. MobiCom '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1161089.1161097

[19] Caloca, C., Garcia Macias, J.A.: Adaptive Solutions in Multihop Communication Protocols for Vehicular Ad Hoc Networks. In: Watfa, M. (ed.) Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, chap. Adaptive S, pp. 301–322. IGI Global (May 2010), http://www.igi-global.com/chapter/adaptive-solutions-multihop-communication-protocols/43176/

[20] Carter, C., Yi, S., Kravets, R.: ARP considered harmful: manycast transactions in ad hoc networks. In: Wireless Communications and Networking. WCNC '03 IEEE. vol. 3, pp. 1801–1806 (2003)

[21] Cavalcante, E.S., Aquino, A.L., Pappa, G.L., Loureiro, A.A.: Roadside unit deployment for information dissemination in a vanet: An evolutionary approach. In: 14th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO). pp. 27–34. New York, NY, USA (2012), http://doi.acm.org/10.1145/2330784.2330789

[22] Cha, S.H.: A Survey of Greedy Routing Protocols for Vehicular Ad Hoc Networks. The Smart Computing Review 2(2), 125–137 (Apr 2002), http://dx.doi.org/10.6029/smartcr.2012.02.003

[23] Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., Hartenstein, H.: Overhaul of IEEE 802.11 Modeling and Simulation in Ns-2. In: Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems. pp. 159–168. MSWiM '07, ACM, New York, NY, USA (2007)

[24] Cheng, H., Fei, X., Boukerche, A., Mammeri, A., Almulla, M.: A geometry-based coverage strategy over urban vanets. In: 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN). pp. 121–128. New York, NY, USA (2013), http://doi.acm.org/10.1145/2507248.2507250

[25] Chia, S.T.S., Snow, P.: Characterising radio-wave propagation behaviour at 1700 MHz for urban and highway microcells. In: Micro-Cellular Propag. Model. IEE Colloq. pp. 11/1–11/4 (Nov 1992)

[26] Choi, J., Khaled, Y., Tsukada, M., Ernst, T.: IPv6 support for VANET with geographical routing. In: 2008 8th International Conference on ITS Telecommunications. pp. 222–227. IEEE (Oct 2008), http://dx.doi.org/10.1109/ITST.2008.4740261

[27] Clark, M.: IEEE 802.11a WLAN model, http://www.mathworks.com/matlabcentral/fileexchange/3540-ieee-802-11a-wlan-model, [Last Accessed: 2015-12-07]

[28] Collet, Y.: xxhash, http://cyan4973.github.io/xxHash/, [Last Accessed: 2016-01-07]

[29] Darwish, T., Abu Bakar, K.: Traffic aware routing in vehicular ad hoc networks: characteristics and challenges. Telecommunication Systems (Mar 2015), http://dx.doi.org/10.1007/s11235-015-0008-7

[30] Davies, J.: Understanding IPv6. Microsoft Press, 3rd edn. (2012)

[31] Department of Economic and Social Affairs: World Urbanization Prospects: The 2014 Revision, Highlights (ST/ESA/SER.A/352). Nations, United (2014), http://dx.doi.org/10.4054/DemRes.2005.12.9

[32] Ding, J., Xu, J., Zheng, Z.: Gateway deployment optimization in wireless mesh network: A case study in china. In: IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics (SOLI). pp. 300–305 (Jul 2009)

[33] Dubey, B.B., Chauhan, N., Pant, S.: Effect of position of fixed infrastructure on data dissemination in vanets. IJRRCS 2(2), 482–486 (2011)

[34] Estinet-Technologies: EstiNet 8 Network Simulator and Emulator, http://www.estinet.com/products.php?lv1=13&sn=15, [Last Accessed: 2016-01-07]

[35] European Commission: eCall in all new cars from April 2018 | Digital Agenda for Europe, http://ec.europa.eu/digital-agenda/en/news/ecall-all-new-cars-april-2018, [Last Accessed: 2016-01-15]

[36] European Commission: FP 7 Information and Communication Technologies, http://cordis.europa.eu/fp7/ict/, [Last Accessed: 2016-01-15]

[37] Ferreiro-Lage, J.A., Gestoso, C.P., Rubinos, O., Agelet, F.A.: Analysis of Unicast Routing Protocols for VANETs. In: Networking and Services, 2009. ICNS '09. Fifth International Conference on. pp. 518–521 (2009), http://dx.doi.org/10.1109/ICNS.2009.96

[38] Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: A realistic simulation framework for vehicular networks. In: Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques. pp. 37–46. SIMUTOOLS '12, ICST, Brussels, Belgium (2012), http://dl.acm.org/citation.cfm?id=2263019.2263025

[39] Friis, H.T.: A Note on a Simple Transmission Formula. Proceedings of the IRE 34(5), 254–256 (may 1946)

[40] Gerla, M., Kleinrock, L.: Vehicular networks and the future of the mobile internet. Computer Networks 55(2), 457–469 (Feb 2011), http://dx.doi.org/10.1016/j.comnet.2010.10.015

[41] Google Inc.: Cityhash, https://code.google.com/p/cityhash/, [Last Accessed: 2016-01-07]

[42] Gorrieri, A., Ferrari, G.: Irresponsible AODV routing. Vehicular Communications 2(1), 47–57 (Jan 2015)

[43] Gozalvez, J., Sepulcre, M., Bauza, R.: Impact of the radio channel modelling on the performance of VANET communication protocols. Telecommunication Systems 50(3), 149–167 (Dec 2010)

[44] Granelli, F., Boato, G., Kliazovich, D., Vernazza, G.: Enhanced GPSR Routing in Multi-Hop Vehicular Communications through Movement Awareness. IEEE Communications Letters 11(10), 781–783 (oct 2007), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4389785

[45] H. Moustafa, S.M Senouci, M.J.: Introduction to Vehicular Networks. In: Vehicular Networks: Techniques, Standards and Applications. Auerbach Publications – CRC Press (Taylor & Francis Group) (2008), http://www.intechopen.com/source/pdfs/12877/InTech-Communications_in_vehicular_ad_hoc_networks.pdf

[46] Haerri, J. and Filali, F. and Bonnet, C.: Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns. Proceedings of the 5th IFIP Mediterranean Ad-Hoc Networking Workshop pp. 14–17 (2006), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.3391&rep=rep1&type=pdf

[47] Hartenstein, H., Laberteaux, K., Ebrary, I.: VANET : Vehicular Applications and Inter-Networking Technologies. Wiley Online Library (2010), http://onlinelibrary.wiley.com/doi/10.1002/9780470740637.fmatter/summary

[48] Hinden, R., Deering, S.: IP Version 6 Addressing Architecture. RFC 4291, RFC Editor (February 2006), http://www.rfc-editor.org/rfc/rfc4291.txt

[49] Hinden, R., Haberman, B.: Unique Local IPv6 Unicast Addresses. RFC 4193, RFC Editor (October 2005), http://www.rfc-editor.org/rfc/rfc4193.txt

[50] Holland, G., Vaidya, N.: Analysis of TCP Performance over Mobile Ad Hoc Networks. In: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking. pp. 219–230. MobiCom '99, ACM, New York, NY, USA (1999), http://doi.acm.org/10.1145/313451.313540

[51] IBM ®: Ilog cplex optimization studio v 12.5, http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud, [Last Accessed: 2015-07-20]

[52] IBM ®: SPSS statistics, http://www-01.ibm.com/software/analytics/spss/, [Last Accessed: 2015-07-20]

[53] Wireless Access in Vehicular Environments (WAVE) - Networking Services. IEEE Std 1609.3-2010 (Revision of IEEE Std 1609.3-2007) pp. 1–144 (2010), http://dx.doi.org/10.1109/10.1109/IEEESTD.2010.5680697

[54] Ieee standard for wireless access in vehicular environments (wave)–multi-channel operation. IEEE Std 1609.4-2010 (Revision of IEEE Std 1609.4-2006) pp. 1–89 (Feb 2011), http://dx.doi.org/10.1109/IEEESTD.2011.5712769

[55] IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2012 pp. 1–2793 (2012), http://dx.doi.org/10.1109/IEEESTD.2012.6178212

[56] Imadali, S., Petrescu, A., Janneteau, C.: Vehicle Identification Number-Based Unique Local IPv6 Unicast Addresses (VULA). Internet-Draft , RFC Editor (February 2013), https://tools.ietf.org/html/draft-imadali-its-vinipv6-vula-00

[57] ISO/IEC JTC 1, Information technology: Information technology Smart cities. Tech. rep., International Organization for Standardization, Vienna, Austria (2014), http://www.iso.org/iso/smart{_}cities{_}report-jtc1.pdf

[58] Jang, K.W.: A tabu search algorithm for routing optimization in mobile ad-hoc networks. Telecommun. Syst. 51(2-3), 177–191 (Nov 2012), http://dx.doi.org/10.1007/s11235-011-9428-1

[59] Jeffrey, A., Zwillinger, D.: Table of Integrals, Series, and Products. Academic Press, 7th edn. (2007)

[60] Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis. Pearson, 6th edn. (Apr 2007)

[61] Jun, P., QiangQiang, Z.: Gateways placement optimization in wireless mesh networks. In: International Conference on Networking and Digital Society. pp. 221–226 (2009)

[62] Karp, B., Kung, H.T.: GPSR Greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00. pp. 243–254. ACM Press, New York, New York, USA (Aug 2000), http://dl.acm.org/citation.cfm?id=345910.345953

[63] Klingler, F., Dressler, F., Cao, J., Sommer, C.: MCB – A multi-channel beaconing protocol. Ad Hoc Networks 36, 258–269 (jan 2016), http://dx.doi.org/10.1016/j.adhoc.2015.08.002

[64] Krajzewicz, D., Hertkorn, G., Rossel, C., Wagner, P.: SUMO - Simulation of Urban MObility: An open-source traffic simulation. In: Proc. 4th Middle East Symposium on Simulation and Modelling (MESM2002). pp. 183–187 (September 2002)

[65] Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of SUMO - Simulation of Urban MObility. International Journal On Advances in Systems and Measurements 5(3&4), 128–138 (Dec 2012)

[66] Krauss, S., Wagner, P., Gawron, C.: Metastable states in a microscopic model of traffic flow. Phys. Rev. E 55(5), 5597–5602 (1997), http://link.aps.org/doi/10.1103/PhysRevE.55.5597

[67] Lacage, M., Henderson, T.R.: Yet Another Network Simulator. In: Proceeding from the 2006 Workshop on Ns-2: The IP Network Simulator. WNS2 '06, ACM, New York, NY, USA (2006)

[68] Lee, K., Lee, U., Gerla, M.: Survey of Routing Protocols in Vehicular Ad Hoc Networks. In: Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, pp. 149–170. Information Science Reference (2009)

[69] Li, F., Wang, Y.: Routing in vehicular ad hoc networks: A survey. Vehicular Technology Magazine, IEEE 2(2), 12–22 (2007)

[70] Liang, Y., Liu, H., Rajan, D.: Optimal placement and configuration of roadside units in vehicular networks. In: IEEE 75th Vehicular Technology Conference (VTC Spring). pp. 1–6 (May 2012)

[71] Linnartz, J.P.: Wireless Communication website, http://www.wirelesscommunication.nl/reference/contents.htm, [Last Accessed: 2016-01-07]

[72] Liu, J., Wan, J., Wang, Q., Deng, P., Zhou, K., Qiao, Y.: A survey on position-based routing for vehicular ad hoc networks. Telecommunication Systems (2015), http://link.springer.com/10.1007/s11235-015-9979-7

[73] Lochert, C., Mauve, M.: Geographic routing in city scenarios. ACM SIGMOBILE Mobile Computing and Communications Review 9, 2005 (2005)

[74] Lochert, C., Mauve, M., Fü\ssler, H., Hartenstein, H.: Geographic Routing in City Scenarios. SIGMOBILE Mob. Comput. Commun. Rev. 9(1), 69–72 (2005), http://doi.acm.org/10.1145/1055959.1055970

[75] Martínez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: Citymob: A Mobility Model Pattern Generator for VANETs. In: Proc. IEEE International Conference on Communications ICC Workshops. pp. 370–374 (May 2008)

[76] Martinez, F.J., Fogue, M., Toh, C.K., Cano, J.C., Calafate, C.T., Manzoni, P.: Computer Simulations of VANETs Using Realistic City Topologies. Wireless Personal Communications pp. 1–25 (Mar 2012), http://www.springerlink.com/content/95hk2n700v635064/

[77] McNamara, J.: GPS for Dummies. John Wiley & Sons (2008)

[78] Menouar, H., Lenardi, M.: Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks. Vehicular Technology pp. 2101–2105 (2007), http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4350090

[79] Mercedes-Benz: Car-to-X communication., https://www.mercedes-benz.com/en/mercedes-benz/innovation/car-to-x-communication/, [Last Accessed: 2016-01-29]

[80] Mezher, A.M., Jurado Oltra, J., Urquiza-Aguiar, L., Iza Paredes, C., Tripp Barba, C., Aguilar Igartua, M.: Realistic environment for VANET simulations to detect the presence of obstacles in vehicular ad hoc networks. In: Proceedings of the 11th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks - PE-WASUN '14. pp. 77–84. PE-WASUN '14, ACM Press, Montreal, QC, Canada (2014), http://doi.acm.org/10.1145/2653481.2653488

[81] Mohammad, S., Rasheed, A., Qayyum, A.: VANET Architectures and Protocol Stacks: A Survey. In: Strang, T., Festag, A., Vinel, A., Mehmood, R., Rico Garcia, C., Röckl, M. (eds.) Communication Technologies for Vehicles SE - 9, Lecture Notes in Computer Science, vol. 6596, pp. 95–105. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-19786-4_9

[82] Mohr, D; Muller, N; Krieg, A; Gao, P; Kaas, H W; Krieger, A; Hensley, R.: The road to 2020 and beyond: What's driving the global automotive industry? Tech. rep., McKinsey & Company, Inc (2013), http://www.mckinsey.com/client{_}service/automotive{_}and{_}assembly/latest{_}thinking

[83] Morgan, Y.L.: Notes on DSRC & WAVE Standards Suite: Its Architecture, Design, and Characteristics. IEEE Communications Surveys & Tutorials 12(4), 504–518 (2010), https://doi.org/10.1109/SURV.2010.033010.00024

[84] Nagel, R., Eichler, S.: Efficient and realistic mobility and channel modeling for vanet scenarios using omnet++ and inet-framework. In: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. pp. 89:1–89:8. Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2008), http://dl.acm.org/citation.cfm?id=1416222.1416323

[85] Narten, T., Nordmark, E., Soliman, H., Simpson, W.: Neighbor Discovery for IP version 6 (IPv6). RFC 4861, RFC Editor (September 2007), http://www.rfc-editor.org/rfc/rfc4861.txt

[86] Naumov, V., Baumann, R., Gross, T.: An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In: 7th ACM international symposium on Mobile ad hoc networking and computing. pp. 108–119. ACM Press, New York, USA (2006), http://dl.acm.org/citation.cfm?id=1132905.1132918

[87] Nguyen, N.V., Guerin-Lassous, I., Moraru, V., Sarr, C.: Retransmission-based available bandwidth estimation in IEEE 802.11-based multihop wireless networks. In: Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems - MSWiM '11. p. 377. ACM Press, New York, New York, USA (oct 2011), http://dl.acm.org/citation.cfm?id=2068897.2068961

[88] Nilsson, C.: Heuristics for the traveling salesman problem. Tech. rep.

[89] OpenStreetMap Project, http://wiki.openstreetmap.org/wiki/Main_Page, [Last Accessed: 2016-01-07]

[90] Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, RFC Editor (2003), http://www.rfc-editor.org/rfc/rfc3561.txt

[91] Plummer, D.: Request For Comments: 826 An Ethernet Address Resolution Protocol. Internet Engineering Task Force - Network Working Group p. 11 (1982)

[92] Punnoose, R.J., Nikitin, P.V., Stancil, D.D.: Efficient simulation of Ricean fading within a packet simulator. In: Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd. vol. 2, pp. 764–767 vol.2 (2000)

[93] Pursley, M., Taipale, D.: Error Probabilities for Spread-Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding. IEEE Transactions on Communications 35(1), 1–12 (Jan 1987)

[94] R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013), http://www.R-project.org/

[95] Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edn. (2001)

[96] Sarr, C., Chaudet, C., Chelius, G., Guérin-Lassous, I.: Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks. IEEE Transactions on Mobile Computing 7(10), 1228–1241 (October 2008)

[97] Schnaufer, S., Effelsberg, W.: Position-based unicast routing for city scenarios. In: World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a. pp. 1–8 (2008)

[98] Scopigno, R., Cozzetti, H.: Signal shadowing in simulation of urban vehicular communications. In: Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on. pp. 131–138 (Sept 2010)

[99] Semchedine, F., Bouallouche-Medjkoune, L., Bennacer, L., Aber, N., Aïssani, D.: Routing protocol based on tabu search for wireless sensor networks. Wirel. Pers. Commun. 67(2), 105–112 (Nov 2012), http://dx.doi.org/10.1007/s11277-011-0367-7

[100] Sharef, B.T., Alsaqour, R.a., Ismail, M.: Vehicular communication ad hoc routing protocols: A survey. Journal of Network and Computer Applications 40(1), 363–396 (Apr 2014), https://doi.org/10.1016/j.jnca.2013.09.008

[101] Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. CHAPMAN & HALL/CRC, Boca Raton, second edn. (2000)

[102] Small, T., Haas, Z.J.: The shared wireless infostation model: A new ad hoc networking paradigm (or where there is a whale, there is a way). In: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing. pp. 233–244. MobiHoc '03, ACM, New York, NY, USA (2003), http://doi.acm.org/10.1145/778415.778443

[103] Small, T., Haas, Z.J.: Resource and performance tradeoffs in delay-tolerant wireless networks. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking. pp. 260–267. WDTN '05, ACM, New York, NY, USA (2005)

[104] Sommer, C., Eckhoff, D., German, R., Dressler, F.: A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments. In: Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on. pp. 84–90 (Jan 2011)

[105] Sommer, C., Dressler, F.: Vehicular Networking. Cambridge University Press (2014), http://dx.doi.org/10.1017/CBO9781107110649

[106] Spaho, E., Ikeda, M., Barolli, L., Xhafa, F.: Performance Comparison of OLSR and AODV Protocols in a VANET Crossroad Scenario. In: Park, J.J.J.H., Barolli, L., Xhafa, F., Jeong, H.Y. (eds.) Information Technology Convergence SE - 5, Lecture Notes in Electrical Engineering, vol. 253, pp. 37–45. Springer Netherlands (2013), http://dx.doi.org/10.1007/978-94-007-6996-0_5

[107] Takagi, H., Kleinrock, L.: Optimal transmission ranges for randomly distributed packet radio terminals. IEEE Transactions on Communications COM-32(3), 246–257 (March 1984), (Also "Multiple Access Communications, Foundations for Emerging Technologies", Norman Abramson (Ed.), IEEE Press, 1992, pp.342-353.)

[108] THE EUROPEAN PARLIAMENT AND OF THE COUNCIL: DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the framework for the deployment of Intelligent Transport Systems. Official Journal of the European Union 53(6), 1–13 (2015), http://dx.doi.org/10.1016/10.3000/17252555.L{_}2010.207.eng

[109] Toutouh, J., Garcia-Nieto, J., Alba, E.: Intelligent olsr routing protocol optimization for vanets. Vehicular Technology, IEEE Transactions on 61(4), 1884–1894 (2012), http://dx.doi.org/10.1109/TVT.2012.2188552

[110] Toyota Global Site: Vehicle-infrastructure Cooperative Systems, http://www.toyota-global.com/innovation/intelligent{_}transport{_}systems/infrastructure/, [Last Accessed: 2016-01-29]

[111] Tripp Barba, C., Urquiza Aguiar, L., Aguilar Igartua, M.: Design and evaluation of GBSR-B, an improvement of GPSR for VANETs. IEEE Latin America Transactions 11(4), 1083 – 1089 (Jun 2013), http://dx.doi.org/10.1109/TLA.2013.6601753

[112] Tripp-Barba, C., Aguilar Igartua, M., Urquiza Aguiar, L., Mezher, A.M., Zaldívar-Colado, A., Guérin-Lassous, I.: Available bandwidth estimation in GPSR for VANETs. In: Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications - DIVANet '13. pp. 1–8. ACM Press, Barcelona, España (Nov 2013), http://dx.doi.org/10.1145/2512921.2516961

[113] Tripp-Barba, C., Urquiza-Aguiar, L., Aguilar Igartua, M., Rebollo-Monedero, D., de la Cruz Llopis, L.J., Mezher, A.M., Aguilar-Calderón, J.A.: A multimetric, map-aware routing protocol for VANETs in urban areas. Sensors (Basel, Switzerland) 14(2), 2199–224 (Jan 2014), http://dx.doi.org/10.3390/s140202199

[114] Tripp-Barba, C., Urquiza-Aguiar, L., Estrada, J., Aguilar-Calderon, J.A., Zaldivar-Colado, A., Igartua, M.A.: Impact of packet error modeling in VANET simulations. In: 2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST). pp. 1–7. IEEE, Ota (oct 2014), http://dx.doi.org/10.1109/ICASTECH.2014.7068133

[115] Trullols, O., Barcelo-Ordinas, J., Fiore, M., Casetti, C., Chiasserini, C.F.: Information dissemination in vanets: deployment strategies for maximizing coverage. In: 6th Euro-NF Workshop on Wireless and Mobility in the Network of the Future. pp. 1–6 (2009)

[116] Trullols, O., Barcelo-Ordinas, J., Fiore, M., Casetti, C., Chiasserini, C.F.: A max coverage formulation for information dissemination in vehicular networks. In: IEEE Int. Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB). pp. 154–160 (Oct 2009)

[117] Trullols, O., Fiore, M., Casetti, C., Chiasserini, C., Ordinas, J.B.: Planning roadside infrastructure for information dissemination in intelligent transportation systems. Computer Communications 33(4), 432–442 (2010), http://dx.doi.org/10.1016/j.comcom.2009.11.021

[118] Uppoor, S., Trullols-Cruces, O., Fiore, M., Barcelo-Ordinas, J.M.: Generation and Analysis of a Large-Scale Urban Vehicular Mobility Dataset. Mobile Computing, IEEE Transactions on 13(5), 1061–1075 (May 2014), http://dx.doi.org/10.1109/TMC.2013.27

[119] Urquiza, L.: Developed code for Estinet simulator, http://www.lfurquiza.com/research/estinet, [Last Accessed: 2016-01-07]

[120] Urquiza-Aguiar, L., Tripp-Barba, C., Martin, I., Aguilar, M.: Propagation and Packet Error models in VANET simulations. Latin America Transactions, IEEE (Revista IEEE America Latina) 12(3), 499–507 (May 2014), http://dx.doi.org/10.1109/TLA.2014.6827879

[121] Urquiza-Aguiar, L., Almeida, D., Tripp-Barba, C., Aguilar Igartua, M.: Heuristic Methods in Geographical Routing Protocols for VANETs. In: Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks - PE-WASUN '15. pp. 41–48. ACM Press, Cancun, Mexico (nov 2015), http://doi.acm.org/10.1145/2810379.2810394

[122] Urquiza-Aguiar, L., Tripp-Barba, C., Aguilar Igartua, M.: A Stochastic Optimization Model for the Placement of Road Site Units. In: Gaggi, O., Manzoni, P., Palazzi, C. (eds.) EAI International Conference on Smart Objects and Technologies for Social Good, pp. 1–6. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer International Publishing, Rome, Italy (2016), publication Pending

[123] Urquiza-Aguiar, L., Tripp-Barba, C., Aguilar Igartua, M.: Performance Evaluation of a Geographical Heuristic Routing Protocols for VANETs in urban scenarios. Journal of Network and Computer Applications pp. 1–24 (January 2016), in preparation

[124] Urquiza-Aguiar, L., Tripp-Barba, C., Estrada-Jiménez, J., Aguilar Igartua, M.: Empirical Analysis of the Minkowski Distance Order in Geographical Routing Protocols for VANETs. In: Aguayo-Torres, M.C., Gómez, G., Poncela, J. (eds.) Wired/Wireless Internet Communications SE - 24, Lecture Notes in Computer Science, vol. 9071, pp. 327–340. Springer International Publishing, Malaga, Spain (2015), http://dx.doi.org/10.1007/978-3-319-22572-2{_}24

[125] Urquiza-Aguiar, L., Tripp-Barba, C., Estrada-Jiménez, J., Igartua, M.A.: On the Impact of Building Attenuation Models in VANET Simulations of Urban Scenarios. Electronics 4(1), 37–58 (2015), http://dx.doi.org/10.3390/electronics4010037

[126] Urquiza-Aguiar, L., Tripp-Barba, C., Rebollo-Monedero, D., Mezher, A.M., Aguilar-Igartua, M., Forné, J.: Coherent, Automatic Address Resolution for Vehicular Ad Hoc Networks. Int. J. Ad Hoc Ubiquitous Comput. (***In press***), 1–18 (Accpeted

in 2015), https://www.inderscience.com/admin/ospeers/getInProduction.php?id=48381&fid=116&fromonsusy=yes

[127] Urquiza-Aguiar, L., Tripp-Barba, C., Romero, A.: Reducing Duplicate Packets in Unicast VANET Communications. In: Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks - PE-WASUN '15. pp. 1–8. ACM Press, Cancun, Mexico (nov 2015), http://doi.acm.org/10.1145/2810379.2810387

[128] Urquiza-Aguiar, L., Tripp-Barba, C., Romero Muir, Á.: Mitigation of packet duplication in VANET unicast protocols. Journal of Network and Computer Applications pp. 1–21 (January 2016), submitted

[129] Urquiza-Aguiar, L., Vazquez-Rodas, A., Tripp-Barba, C., Igartua, M.A., de la Cruz Llopis, L.J., Gargallo, E.S.: MAX-MIN based buffer allocation for VANETs. In: Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on. pp. 1–5. Vancouver. BC, Canada (2014), http://dx.doi.org/10.1109/WIVEC.2014.6953231

[130] Urquiza-Aguiar, L., Vázquez-Rodas, A., Tripp-Barba, C., Mezher, A.M., Aguilar Igartua, M., de la Cruz Llopis, L.J.: Efficient Deployment of Gateways in Multi-hop Ad-hoc Wireless Networks. In: Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks. pp. 93–100. PE-WASUN '14, ACM, Montreal, QC, Canada (2014), http://doi.acm.org/10.1145/2653481.2653487

[131] Wakikawa, R., Malinen, J., Perkins, C., Tuominen, A.: Global connectivity for IPv6 Mobile Ad Hoc Networks. Internet-Draft, RFC Editor (March 2006), https://tools.ietf.org/html/draft-wakikawa-manet-globalv6-05

[132] Wang, S.Y., Chou, C.L.: NCTUns tool for wireless vehicular communication network researches. Simulation Modelling Practice and Theory 17(7), 1211–1226 (August 2009)

[133] Wang, S.Y., Chou, C.L., Huang, C.H., Hwang, C.C., Yang, Z.M., Chiou, C.C., Lin, C.C.: The design and implementation of the NCTUns 1.0 network simulator. Computer Networks 42(2), 175–197 (June 2003)

[134] Wang, S.Y., Wang, P.F., Li, Y.W., Lau, L.C.: Design and implementation of a more realistic radio propagation model for wireless vehicular networks over the nctuns network simulator. In: Wireless Communications and Networking Conference (WCNC), 2011 IEEE. pp. 1937–1942 (March 2011)

[135] Wolsey, L.A.: Integer Programming. Wiley Series in Discrete Mathematics and Optimization, Wiley, New York (1998)

[136] Wu, H., Fujimoto, R., Riley, G., Hunter, M.: Spatial propagation of information in vehicular networks. IEEE Transactions on Vehicular Technology 58(1), 420–431 (Jan 2009)

[137] Xiao, D., Peng, L., Ogugua Asogwa, C., Huang, L.: An Improved GPSR Routing Protocol. International Journal of Advancements in Computing Technology 3(5), 132–139 (jun 2011), http://www.aicit.org/ijact/paper_detail.html?q=219

[138] Yang, X.L., Liao, D., Sun, G., Lu, C., Yu, H.F.: GPCR-D: A Topology and Position Based Routing Protocol in VANET. Adv. Mater. Res. 846-847, 858–863 (nov 2013), http://dx.doi.org/10.4028/www.scientific.net/AMR.846-847.858

[139] Zhang, X., Neglia, G., Kurose, J., Towsley, D.: Performance modeling of epidemic routing. Computer Networks 51(10), 2867–2891 (Jul 2007)

[140] Zhao, X., Kivinen, J., Vainikainen, P., Skog, K.: Propagation characteristics for wideband outdoor mobile communications at 5.3 GHz. Selected Areas in Communications, IEEE Journal on 20(3), 507–514 (2002)