UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Modeling and Real Deployment of C-ITS by Integrating Ground Vehicles and Unmanned Aerial Vehicles

Author: Seilendria A. Hadiwardoyo

Supervisors: Carlos T. Calafate
Juan-Carlos Cano

February 2019

"Modeling and Real Deployment of C-ITS by Integrating Ground Vehicles and Unmanned Aerial Vehicles"

A thesis submitted in partial fulfillment of the requirements for the Doctoral Degree in Informatics at Universitat Politécnica de Valéncia

Author (*Doctorando*):
Mtr. Seilendria Ardityarama Hadiwardoyo
ST.(UI, Indonesia), Maîtrise.(ULille, France), MSc.(UBS, France), DEA.(UMinho, Portugal)

Supervisors (*Directores*):
Prof. Carlos Tavares Calafate
Universitat Politècnica de València, Spain
Prof. Juan-Carlos Cano Escribá
Universitat Politècnica de València, Spain

Examiners (*Miembros del Tribunal*):
Dr. Francisco Martínez Domínguez
Universidad de Zaragoza, Spain
Prof. Manuel Perez Malumbres
Universitas Miguel Hernández de Elche, Spain
Dr. HDR. Houda Labiod
Télécom ParisTech, France

Reviewers (*Evaluadores Externos*):
Dr. HDR. Nadjib Achir
Université Paris 13, France
Dr. Joaquín Olivares Bueno
Universidad de Córdoba, Spain
Dott.ssa Ric. Claudia Campolo
Università di Reggio Calabria, Italy

Grupo de Redes de Computadores (GRC)
Department of Computer Engineering (DISCA)
Universitat Politècnica de València
Camí de Vera s/n, Edif. 1G
46022 - Valencia, Spain
Tel: (+34) 963 877 007 (Ext. 85720)
Web: `http://www.grc.upv.es/`

Valencia, España

February 2019

To my wife;
Issye Margaretha Kamal, B.A., M.Res.,
and my parents;
Prof. Dr. Ir. Sigit P. Hadiwardoyo, DEA
Dr. Ir. Riana H. Lumingkewas, M.T.

Acknowledgements

So many people have supported me during this long journey. I want to acknowledge them all, and return the support and kindliness that I received during this period.

I want to start by sincerely thanking my supervisors. I remember the first time I have emailed Prof. Carlos Calafate to inquire about the possibility to apply for the PhD position in the Smart@CarPhone project. He asked me to apply ASAP since the deadline was close! The next thing I know, I found myself packing my stuff and traveling to Spain, knocking on his door, and then introducing myself to the enthusiastic Prof. Juan-Carlos Cano. The rest is history, I have got my two best mentors during my PhD period. Thank you for the guidance, trust, and friendship that I will never forget!

It is also my pleasure to express my gratitude to the examiners, or the *tribunal* members, Dr. Francisco Martinez, Prof. Manuel Perez Malumbres, and Dr. Houda Labiod for sparing their invaluable time on my defence and examining the thesis. In the same way I want to thank the reviewers, or the *evaluadores*, Dott.ssa Ric. Claudia Campolo, Dr. Achir Nadjib, and Dr. Joaquin Olivares for having the patience to review the thesis and providing me with constructive comments and suggestions.

I want to show my gratitude to all GRC members. First to the GRC leader, Prof. Pietro Manzoni. Thank you for opening me the doors of GRC. Also, I want to thank Dr. Enrique Hernandez for his constructive advices during our monthly meetings, and helping me with my publications. A special thanks to

GRC lab members, Leo, Subha, los Jorges, Willian, Andres, Elsa, Francisco, Pablo, Oscar, Camilo, Chaker, Sergio, Miguel, Johann, Ali, Danilo. Thank you all for the solidarity and memories we experienced together in this group. I will never forget the coffee breaks we always share together.

I also want to thank the people I met during my internship period in Tokyo and in St. Petersburg. First, I would like to thank Prof. Yusheng Ji for being my *sensei* and receiving me at her lab, as well as her students for helping me with a lot of stuff during my stay in Japan. In addition, I would like to thank Dr. Dimitry Klionskiy for receiving me at SPbETU, as well as Dr. Kirill Krinkin and Dr. Elena Belova for giving me constructing advises during my research internship period in Russia.

To my friends in Spain, Kang Esa, Mbak Tessa, Mas Aryo, Abie, Anggoro, Yoga, Dhanis, and other friends at the Indonesian Students Association in Spain, Japan, and Russia, thank you for making my stay abroad so colorful and making me feel like home.

To my family, my brothers and sister (the Hadiwardoyos), and the in-laws (the Kamals), thank you for your abundant supports from kilometers away.

To my parents, my role models, that have initially and always encouraged me in doing the PhD since I was a kid, Prof. Dr. Ir. Sigit Pranowo Hadiwardoyo, DEA, and Dr. Ir. Riana Herlina Lumingkewas, MT, I can not thank them enough.

To my wife, Issye Margaretha Kamal, who is involved in my academic adventure, who knows my real ups and downs, thank you for the love, endless encouragements, and support.

Finally, I want to thank the Spanish government through the Ministry of Economy and Competitiveness (MINECO) and the European Union Commission through the European Social Fund (ESF) for co-financing and granting me the fellowship to fund my studies in Spain and my research stay in Russia. In addition, I would to thank the National Institute of Informatics for granting me the internship fund and the Japanese government through the Japan Society for the Promotion of Science (JSPS) for supporting my research work in Japan.

Seilendria A. Hadiwardoyo
Valencia, February 2019

Abstract

To provide a safer road traffic environment and make it more convenient, Intelligent Transport Systems (ITSs) are proposed as a solution endowed with cutting-edge technological advances. The integration of transportation elements like cars together with infrastructure elements like Road Side Units to achieve a networking environment offers new services in addition to Internet connectivity. This integration comes under the term Cooperative Intelligent Transport System (C-ITS).

Connecting cars with surrounding devices forming vehicular networks in Vehicle-to-Everything (V2X) open new deployments in C-ITS applications like safety-related ones. With the massive use of smartphones nowadays, and due to their flexibility and mobility, several efforts exist to integrate them with cars. In fact, with the right support from the vehicle's On-Board Unit (OBU), smartphones can be seamlessly integrated with vehicular networks. Hence, drivers can use their smartphones as a device to participate in C-ITS services for safety purposes, among others, which is a quite interesting research topic.

A significant problem arises when vehicular communications face signal obstructions caused by the environment. In fact, the impact of vegetation and buildings, whether in urban and rural areas, can result in a lower signal quality. One way to enhance vehicular communication networks is to deploy Unmanned Aerial Vehicles (UAVs) to act as relays for communication between cars, or ground vehicles. In fact, UAVs offer important deployment advantages, as they offer great flexibility in terms of mobility, in addition to an enhanced communications range.

To assess the quality of the communications, a set of measurements must take place. However, due to the cost of real deployments of UAVs and cars, real experiments might not be feasible for research activities with limited resources. Hence, simulation experiments become the preferred option to assess UAV-to-car communications.

Achieving correct and representative signal propagation models that can be imported to the simulation environments becomes crucial to obtain a higher degree of realism, especially for simulations involving UAVs moving anywhere throughout the 3D space. In particular, terrain elevation information must be taken into account when attempting to characterize signal propagation effects. In this research work, we propose both theoretical and empirical approaches to study the integration of vehicular networks combining cars and UAVs, and we study the impact of the surrounding environment on the communications quality. An application, a measurement framework, and a simulation model are presented in this thesis in an effort to model, develop, and deploy C-ITS services. More specifically, we propose a simulation model that takes into account 3D terrain features to achieve reliable UAV-to-car communication results.

Resumen

Para proporcionar un entorno de tráfico vial más seguro y eficiente, los sistemas ITS o Sistemas Inteligentes de Transporte representan como una solución dotada de avances tecnológicos de vanguardia. La integración de elementos de transporte como automóviles junto con elementos de infraestructura como RoadSide Units (RSUs) ubicados a lo largo de la vía de comunicación permiten ofrecer un entorno de red conectado con múltiples servicios, incluida conectividad a Internet. Esta integración se conoce con el término C-ITS o Sistemas Inteligentes de Transporte Cooperativos.

La conexión de automóviles con dispositivos de infraestructura permite crear redes vehiculares conectadas (V2X) vehículo a dispositivos, que ofrecen la posibilidad de nuevos despliegues en aplicaciones C-ITS como las relacionadas con la seguridad. Hoy en día, con el uso masivo de teléfonos inteligentes y debido a su flexibilidad y movilidad, existen varios esfuerzos para integrarlos con los automóviles. De hecho, con el soporte adecuado de unidad a bordo (OBU), los teléfonos inteligentes se pueden integrar perfectamente con las redes vehiculares, permitiendo a los conductores usar sus teléfonos inteligentes como dispositivos de bordo a que participan en los servicios C-ITS, con el objeto de mejorar la seguridad al volante entre otros. Tópico este, que hoy día representa un tema relevante de investigación.

Un problema a solucionar surge cuando las comunicaciones vehiculares sufren inferencias y bloqueos de la señal debidos al escenario. De hecho, el impacto de la vegetación y los edificios, ya sea en áreas urbanas y rurales, puede afectar a la calidad de la señal. Algunas estrategias para mejorar la comunicación

vehicular en este tipo de entorno consiste en desplegar UAVs o vehículo aéreo no tripulado (drones), los cuales actúan como enlaces de comunicación entre vehículos. De hecho, UAV ofrece importantes ventajas de implementación, ya que tienen una gran flexibilidad en términos de movilidad, además de un rango de comunicaciones mejorado.

Para evaluar la calidad de las comunicaciones, debe realizarse un conjunto de mediciones. Sin embargo, debido al costo de las implementaciones reales de UAV y automóviles, los experimentos reales podrían no ser factibles para actividades de investigación con recursos limitados. Por lo tanto, los experimentos de simulación se convierten en la opción preferida para evaluar las comunicaciones entre UAV y vehículos terrestres.

Lograr modelos de propagación de señal correctos y representativos que puedan importarse a los entornos de simulación se vuelve crucial para obtener un mayor grado de realismo, especialmente para simulaciones que involucran el movimiento de UAVs en cualquier lugar del espacio 3D. En particular, la información de elevación del terreno debe tenerse en cuenta al intentar caracterizar los efectos de propagación de la señal. En esta tesis doctoral, proponemos nuevos enfoques tanto teóricos como empíricos para estudiar la integración de redes vehiculares que combinan automóviles y UAVs, así mismo el impacto del entorno en la calidad de las comunicaciones. Esta tesis presenta una aplicación, una metodología de medición en escenarios reales y un nuevo modelo de simulación, los cuales contribuyen a modelar, desarrollar e implementar servicios C-ITS. Más específicamente, proponemos un modelo de simulación que tiene en cuenta las características del terreno en 3D, para lograr resultados confiables de comunicación entre UAV y vehículos terrestres.

Resum

Per a proporcionar un entorn de trànsit viari més segur i eficient, els sistemes ITS o Sistemes Intel·ligents de Transport representen una solució dotada d'avanços tecnològics d'avantguarda. La integració d'elements de transport com auto móvils juntament amb elements d'infraestructura com Road Side Units (RSUs) situats al llarg de lav via de comunicació permeten oferir un entorn de xarxa connectat amb multiples serveis, inclusa connectivitat a Internet. Aquesta integració es connex amb el terme C-ITS o Sistemes Intel·ligents de Transport Cooperatius , com ara els automòbils, amb elements d'infraestructura, com ara les road side units (RSU) o pals situats al llarg de la carretera, per a aconseguir un entorn de xarxa que oferisca nous serveis a més de connectivitat a Internet. Aquesta integració s'expressa amb el terme C-ITS, o sistemes intel·ligents de transport cooperatius.

La connexió d'automòbils amb dispositius d'infraestructura permet crear xarxes vehiculars connectades (V2X) vehicle a dispositiu, que ofreixen la possibilitat de nous desplegaments en aplicacions C-ITS, com ara les relacionades amb la seguretat. Avui dia, amb l'ús massiu dels telèfons intel·ligents, i a causa de la flexibilitat i mobilitat que presenten, es fan esforços per integrar-los amb els automòbils. De fet, amb el suport adequat d'unitat a bord (OBU), els telèfons intel·ligents es poden integrar perfectament amb les xarxes vehiculars, permetent als conductors usar els seus telèfons intel·ligents com a dispositius per a participar en els serveis de C-ITS, a fi de millorar la seguretat al volant entre altres. Tòpic est, que hui dia representa un tema rellevant d'investigació.

Un problema a solucionar sorgeix quan les comunicacions vehiculars ateixen inferències i bloquejos del senyal deguts a l'escenari. De fet, l'impacte de la vegetació i els edificis, tant en àrees urbanes com rurals, pot afectar la qualitat del senyal. Algunes estratègies de millorar la comunicació vehicular en aquest tipus d'entorn consisteix a desplegar UAVs o vehicles aeris no tripulats (drones), els quals actuen com a enllaços de comunicació entre vehicles. De fet, l'ús d'UAVs ofereix importants avantatges d'implementació, ja que tenen una gran flexibilitat en termes de mobilitat, a més d'un rang de comunicacions millorat.

Per a avaluar la qualitat de les comunicacions, s'han de realitzar mesures en escenaris reals. No obstant això, a causa del cost de les implementacions i desplegaments reals d'UAV i el seu ús combinat amb vehicles, aquests experiments reals podrien no ser factibles per a activitats d'investigació amb recursos limitats. Per tant, la metodologia basada en simulació es converteixen en l'opció preferida entre els investigadors per a avaluar les comunicacions entre UAV i vehicles terrestres.

Aconseguir models de propagació de senyal correctes i representatius que puguen importar-se als entorns de simulació resulta crucial per a obtenir un major grau de realisme, especialment per a simulacions que involucren el moviment d'UAV en qualsevol lloc de l'espai 3D. En particular, cal tenir en compte la informació d'elevació del terreny per a intentar caracteritzar els efectes de propagació del senyal. En aquesta tesi doctoral proposem enfocaments tant teòrics com empírics per a estudiar la integració de xarxes vehiculars que combinen automòbils i UAV, així com l'impacte de l'entorn en la qualitat de les comunicacions. Aquesta tesi presenta una aplicació, una metodología de mesurament en escenaris reals i un nou model de simulació, els quals contribueixen a modelar, desenvolupar i implementar serveis C-ITS. Més específicament, proposem un model de simulació que té en compte les característiques del terreny en 3D, per a aconseguir resultats fiables de comunicació entre UAV i vehicles terrestres.

Contents

Acknowledgements v

Abstract vii

Contents xiii

1 Introduction 1

1.1 Motivation . . . 1

1.2 Objectives and Methodology . . . 3

1.3 Structure of the Thesis . . . 4

2 An Overview of Cooperative Intelligent Transport Systems Applications 7

2.1 Introduction . . . 7

2.2 Cooperative Intelligent Transport Systems . . . 8

2.3 Vehicular Networks . . . 9

2.3.1 Vehicular Ad-hoc Networks . . . 11

2.3.2 Flying Ad-hoc Networks . . . 11

2.3.3 UAV Assisted Vehicular Networks . . . 12

2.4 Implementations of C-ITS . . . 13
2.4.1 Classification of C-ITS Applications . . . 13
2.4.2 Accelerating the Deployment of C-ITS . . . 16
2.5 Summary . . . 20

3 An Overview of Modeling Vehicular Networks Communications 21
3.1 Introduction . . . 21
3.2 Wireless Channel Characteristics . . . 22
3.3 Radio Propagation Models . . . 23
3.3.1 Models by Propagation Scale . . . 23
3.3.2 Modeling Approaches . . . 25
3.4 Channel Models for Vehicular Environments . . . 27
3.4.1 Vehicular Networks Channel Measurement Campaigns . . . 27
3.4.2 Car-to-Car Communications Modeling . . . 27
3.4.3 Car-to-Infrastructure Communications Modeling . . . 30
3.4.4 UAV-to-Infrastructure Communications Modeling . . . 31
3.4.5 UAV-to-Car Communications Modeling . . . 31
3.5 Simulation Tools for Vehicular Communications . . . 32
3.5.1 Network Simulation Tools . . . 32
3.5.2 Traffic Simulation Tools . . . 33
3.5.3 Interlinking Tools . . . 34
3.6 Summary . . . 36

4 Car-to-Car Communications Application and Modeling 37
4.1 Introduction . . . 37
4.2 Smartphone-Based Driving Safety ITS Application for Vehicular Networks . . . 37
4.2.1 Application Features . . . 38
4.2.2 Implementation Details . . . 41
4.2.3 Deployment . . . 45
4.3 Empirical Evaluation of V2V Communications at Intersections . . . 46
4.3.1 General Overview . . . 47
4.3.2 Testing Tool and Data Analysis . . . 48
4.3.3 Selection of Target Intersections . . . 49

4.4 Experimental Results 52
4.4.1 Packet Delivery Ratio at Intersections. 53
4.4.2 Event Notification Effectiveness on Intersections 56
4.5 Intersection Modeling 58
4.5.1 Modeling procedure 58
4.5.2 Fitting results 59
4.5.3 Model Applicability to Simulation Environments 63
4.6 Applicability on the ITS Application in Relation to Intersection Modeling. . . . 64
4.6.1 Functionality Test 64
4.6.2 Network Performance Test. 65
4.7 Summary 68
5 Experimental Characterization of UAV-to-car Communications 71
5.1 Introduction 71
5.2 Architecture Overview 72
5.2.1 General Overview 72
5.2.2 UAV-to-Car Communications 72
5.2.3 GRCBox Use and Drone's Interface for Communications 74
5.3 Experimental Settings. 74
5.3.1 Experimental environment. 74
5.3.2 Experimental Tools 77
5.3.3 Data Analysis 79
5.4 Experimental Results 81
5.4.1 100m altitude. 83
5.4.2 40m altitude 85
5.5 Experimental Modeling. 86
5.6 Model Applicability and Comparison Against Existing Models. 89
5.6.1 Comparison with Existing Models 89
5.6.2 Applicability of the proposed Model in Simulation 92
5.7 Summary 92
6 Simulation Modeling of UAV-to-car communications 95
6.1 Introduction 95

6.2 3D Model for UAV-to-Car Communications 96
6.2.1 Elevation Model 96
6.2.2 Propagation Model 98
6.2.3 Implementation of the proposed model in simulation. 105
6.3 Simulation Setup 106
6.3.1 Simulation Architecture 107
6.3.2 Simulation Settings 108
6.3.3 Data Analysis 109
6.4 Simulation Results 110
6.5 Impact of Digital Elevation Models on Performance 114
6.5.1 Impact of Digital Elevation Models on Simulation Time. 115
6.5.2 Altitudes Recorded by the Different Elevation Models 117
6.5.3 Performance Using Different Elevation Models 117
6.6 Impact of Diffraction Models on Performance 119
6.6.1 Performance when using different Path Loss Models 120
6.6.2 Impact of Path Loss Models on Simulation Time 122
6.7 Summary 122

7 Conclusions, Publications, and Future Works 125

7.1 Conclusions. 125
7.2 Publications Related to the Thesis 127
7.2.1 Journal Articles 127
7.2.2 Conference Proceedings 128
7.3 Funding and Projects 128
7.4 Future work 129

Acronyms 131

Bibliography 135

List of Figures

2.1 ITS: the big picture [1]. 8

2.2 Example of an use case scenario for UAV-enabled ITS: UAVs can be used as flying Road Side Units (RSUs) to capture video recordings of an accident, and then relay data to a mobile services center [MGAUKT17]. 10

2.3 GRCBox hardware module connected to a VANET with three different nodes [TPCCM15]. 19

3.1 Multipath Propagation Scenario [Pat01]. 23

3.2 Path Loss, Shadowing, and Multipath effects versus Distance [Gol05]. 24

3.3 The architecture of Veins [SGD11]. 35

4.1 Device working in "Civil Mode". 40

4.2 Device working in the "Administrative Mode". 40

4.3 Architecture of the Messiah Application. 41

4.4 Information contained in the message generated by the Messiah Application. 42

4.5 Flow Chart for Sending Packets. 43

4.6 Flow Chart for Receiving Packets. 44

4.7 Location and the trajectory for Scenario 1 (open) 50

4.8 View of the vehicle parked at Intersection 1. 51

4.9 Location and trajectory for Scenario 2 (buildings). 51

4.10 View of the vehicle parked at Intersection 2. 52

4.11 Location and trajectory for Scenario 3 (trees). 53

4.12 View of the vehicle parked at Intersection 3. 54

4.13 Heat Maps for the different Scenarios. Each plot shows the packet delivery ratio depending on the sender position, scenario (open, building and trees) and antenna location (dashboard, rooftop). 55

4.14 Delivery ratio at the three intersections for different GPS error ranges. 57

4.15 Curve Fittings for Delivery Ratio vs. Distance. 60

4.16 Differences among the fitted models and data. 62

4.17 Photo taken from within the car. 64

4.18 Structure of the Ad-Hoc Network. 65

4.19 Distance vs Packet Loss. 67

4.20 Cumulative Distribution Function for the Inter-Packet Arrival Time. 68

4.21 Traffic Overhead based on Route Size 69

5.1 UAVs acting as Mobile RSUs. 72

5.2 Flow of Packets from UAV (Drone) to Car. 73

5.3 Satellite view of the target road near Casinos. 75

5.4 Topographic map of Casinos. 76

5.5 Elevation profiles relevant to our experiments. 77

5.6 Real view of the UAV-to-Car communications scenario in Casinos. 78

5.7 UAV-to-Car communications in our experimental scenario. . . . 79

5.8 View of the UAV antenna orientation. 80

5.9 Views of the car antenna location. 80

5.10 Heat Maps for the different scenarios where the drone's altitude is of 100 meters. Each plot shows the packet delivery ratio depending on the sender's antenna orientation (Vertical, Horizontal), and the receiver's antenna location (Inside, outside). 82

5.11 Heat Maps for the different scenarios where the drone's altitude is of 40 meters. Each plot shows the packet delivery ratio depending on the sender's antenna orientation (Vertical, Horizontal) and the receiver's antenna location (inside, outside). . 84

5.12 Curve Fittings of Delivery Ratio vs. Distance, with the antenna installed Inside or Outside the vehicle. 86

5.13 Comparison of results obtained from simulations and real experiments. 91

6.1 SRTM DEM Data Structure. 98

6.2 Set of slices that divide the LOS into different inspection points. 99

6.3 Simulation Architecture. 107

6.4 Comparison of Heatmaps from Real Experiments and Simulation.111

6.5 Packet Delivery Ratio vs. Distance for both scenarios. 113

6.6 Received Signal Strength indicator vs. Distance for both scenarios.114

6.7 Execution Time for the different Elevation Models. 116

6.8 Comparison of Altitudes Recorded. 118

6.9 Comparison of Elevation Models on the Packet Delivery Ratio. 119

6.10 Comparison of Elevation Models on RSSI. 120
6.11 Comparison of Diffraction Models on the Packet Delivery Ratio. 120
6.12 Comparison of Diffraction Models on the RSSI. 121
6.13 Execution time for the different diffraction models. 122

List of Tables

4.1 Experiment Parameters for Car-to-Car Communications. 48

4.2 c parameter and χ^2 error values for each scenario and antenna position. 61

5.1 Experiment Parameters for UAV-to-Car Communications. . . . 75

5.2 Scenario Categories 81

5.3 Value of parameters a, b and c for each scenario with its R^2 value . 88

5.4 Simulation Parameters for Comparability Test. 90

6.1 Comparison of diffraction methods. 100

6.2 Simulation Parameters for UAV-to-Car Communications. 108

6.3 Summary of Results Based on Methods Used in the Simulation. 123

Chapter 1

Introduction

1.1 Motivation

In an effort to make the traffic on our roads more efficient, convenient, and safe, Intelligent Transport System (ITS) applications are deployed as one of the key solutions to address these challenges. In particular, ITS address traffic-related issues in future Smart Cities, such as helping emergency services, or changing the way of driving to reduce accidents, fuel consumption, and contaminant emissions. The deployment of ITS will eventually minimize transportation problems like congested roads, as well as promote road safety and help at making cities more sustainable.

As part of the ITS concept, communications between surrounding nodes in the road traffic environment play an important role when distributing relevant information. This cooperation, known under the term C-ITS, allows the dissemination of messages containing critical information to be more efficient, which can be useful for safety applications. Communications that involve vehicles in C-ITS fall under the Vehicle-to-Everything (V2X) paradigm, which in turn breaks down into the exchange of data (i) between cars (Vehicle-to-Vehicle (V2V)), and (ii) towards infrastructure elements (V2I, or Vehicle-to-Infrastructure); the latter act as relays towards a wider network, or the Internet itself, becoming thus essential to provide ITS services and applications. Regarding V2V communications, these can be combined with the ad-hoc networking paradigm, giving rise to the so-called Vehicular Ad-hoc Networks (VANETs).

Safety issues are one of the main concerns in transportation. Safety-related applications are developed in an effort to reduce the probability of traffic accidents by leveraging the advantage of inter-vehicular communications, or, in this case, VANETs. Since vehicles can communicate seamlessly, VANETs allow us to easily exchange traffic-related information. Applications that, e.g., deliver emergency alerts, can help the driver to become aware of an accident that has just occurred. Thus, they allow the driver to react adequately when facing such an emergency situation. The fact that the use of the smartphone is wider nowadays, helps to accelerate the applicability of VANET-enabled applications as smartphones can connect to the vehicle's On-Board Unit (OBU) to deploy V2V communications.

Critical applications that disseminate real-time safety information should work by broadcasting messages that are reliable and time-bounded. In addition, distributing such messages through V2V networks in critical safety applications is expected to have a very low transmission delay. If a vehicle in a V2V network is facing a particular emergency situation, it should be warned as soon as possible. However, if the transmission delay is too high, the message can lose its relevance. Thus, the environment itself becomes relevant as it can directly affect this transmission. Vegetation and buildings in the roadside can be obstacles to the dissemination of messages in a traffic road, especially at intersections. Thus, obstacles to signal propagation may arise whether communications take place in urban or rural areas.

Another issue to consider in C-ITS is its deployment when the coverage area lacks infrastructure support. Even though V2X communications can be supported by various radio access technologies, such as using 4G LTE technology in areas with limited infrastructure (e.g. rural areas), major problems typically arise when the communications take place in areas that have no infrastructure support at all.

One of the solutions for such scenarios is the deployment of UAVs as support nodes to ground vehicles. The cooperation between UAVs and ground vehicles can improve the data exchange, and this approach offers benefits to multiple ITS applications like remote sensing and rescue, or disaster assistance operations. When deploying such applications, multiple UAVs can conform a network between themselves, creating what is known as a Flying Ad-hoc Network (FANET). As a subclass of VANETs, FANETs differ from the former as they are characterized by highly mobile nodes moving freely in the 3D space; on the contrary, VANETs are restricted to 2D movements along streets/roads.

Assessing the performance of communications between UAVs and cars is a challenge since a real deployment involves many resources and, in some case, one even needs to deploy UAVs in restricted flight areas. A feasible method to analyze this type of communication is using a simulation framework. However, simulation tools require an appropriate radio propagation model for results to be meaningful. Most simulation tools only consider the communication in two dimensions. However, when communications involve UAVs, the simulation environment should account for all three dimensions, and take into account obstacles such as hills or mountains that can affect the communication signal. Thus, it is necessary to implement an appropriate model that enables the network simulation to have a high degree of realism.

In this regard, to provide C-ITS services, it is desirable to assess the quality of the communication since, in some critical applications, the communication should be reliable and timely in order to be useful. For this reason, a thorough study of the communications quality by creating an application, followed by a real deployment and simulation experiments, are required to create and validate a realistic communications model to enable further theoretical and empirical studies that are both reliable and meaningful.

1.2 Objectives and Methodology

The main objective of this thesis is to provide an implementation of vehicular communication systems, including cars and UAVs, in a real context, as well as its modeling based on empirical studies. Specifically, we will analyze the applicability of C-ITS, investigate the impact of the environment on C-ITS applications, and find an appropriate model for communications, in addition to providing solutions involving UAVs for V2X communications. In fact, not only will we assess the communications performance in real testbeds, but we will also validate them through simulation by developing an appropriate simulation framework.

In order to accomplish the main goal of this thesis, the following specific objectives were defined:

- A thorough study on the state of the art on vehicular communications, its implementations for C-ITS, measurement techniques, and characterization.

- Creation of an application based on smartphones that uses V2V communications to provide C-ITS services.
- Evaluation of the V2V communications performance through real experiments, analyzing the impact of the environment on the packet delivery ratio.
- Investigating the feasibility of using flying nodes, or UAVs, to support V2V communications, and assessing performance by developing a communications architecture combining cars and UAVs.
- Bringing the UAV-to-car communications results into our simulation environment, and propose an extension to the simulation framework so as to achieve an acceptable degree of realism by including three dimensional terrain features in the signal propagation model.

Regarding the methodology used in this thesis, we adopted a pragmatic approach. First, we reviewed the state of the art in order to find the most relevant contributions in the area of knowledge related to vehicular communications in the context of Cooperative ITS. Next, we focus on a real implementation of V2V communications, where we propose a smartphone-based application that represents a real usage case of C-ITS, as well as measurements focusing on the impact of intersections on communications. Finally, we emphasize on the use of UAVs as a support to vehicular communications. We focus on the modeling part of the communication in real experiments, as well as on simulation.

1.3 Structure of the Thesis

The thesis follows a methodology encompassing both theoretical and practical issues. Hence, on the one hand, we perform a fundamental research study (theoretical) based on empirical data. On the other hand, we do applied research, developing applications and undertaking experimental studies.

The thesis is organized in 7 chapters. Below, we briefly describe the contents of each part:

- **Chapter 1. Introduction:** we present the purpose, structure, and motivation of this research work.
- **Chapter 2. An Overview of Cooperative Intelligent Transport Systems Applications:** we provide a review of C-ITS, its technologies, and applications.

- **Chapter 3. An Overview of Modeling Vehicular Networks. Communications:** we present a review of research works focusing on the characterization of the vehicular communications channel.

- **Chapter 4. Car-to-Car Communications Application and Modeling:** we perform an implementation of C-ITS by proposing an application for safety purposes that makes use of VANETs, and we analyze the effectiveness of V2V communications on intersections, detailing how the environment affects the quality of communications, which in turn defines the performance of the C-ITS application.

- **Chapter 5. Experimental Characterization of UAV-to-Car Communications:** we investigate the use of UAVs to communicate with cars, and characterize the communications quality through real experiments.

- **Chapter 6. Simulation Modeling of UAV-to-Car Communications:** we develop a framework architecture and a model suited for simulation with its validation that conforms to real experimental results.

- **Chapter 7. Conclusions, Publications, and Future Works:** we conclude this thesis, present the publications related to the thesis, as well as a list of future research lines.

The final part of the document includes a list of acronyms and the bibliography.

Chapter 2

An Overview of Cooperative Intelligent Transport Systems Applications

2.1 Introduction

In this chapter, we present a literature review addressing the fundamentals of Cooperative Intelligent Transport System (C-ITS) and its applications. We start with the understanding of ITS solutions, vehicular networks, and finally the chapter ends with an overview of the implementation of C-ITS solutions using smartphone-based applications. In general, we will cover the whole picture of C-ITS that involves vehicles to provide different kinds of services.

2.2 Cooperative Intelligent Transport Systems

ITS are a basic element supporting the future Smart Cities by addressing traffic-related issues. In particular, it is the combination of advanced Information and Communication Technology (ICT) systems and a better transportation infrastructure that, together, pave the way for providing novel services in ITS environments [HMVMGPGP11]. ITS aims at making traffic more efficient, convenient, and safe [XSRC12], addressing noble goals like helping emergency services or changing the way we drive to reduce accidents, fuel consumption, and contaminant emissions. This will eventually minimize transportation problems such as congested roads, will promote road safety, and will in general help at making cities more sustainable [BMSMI12].

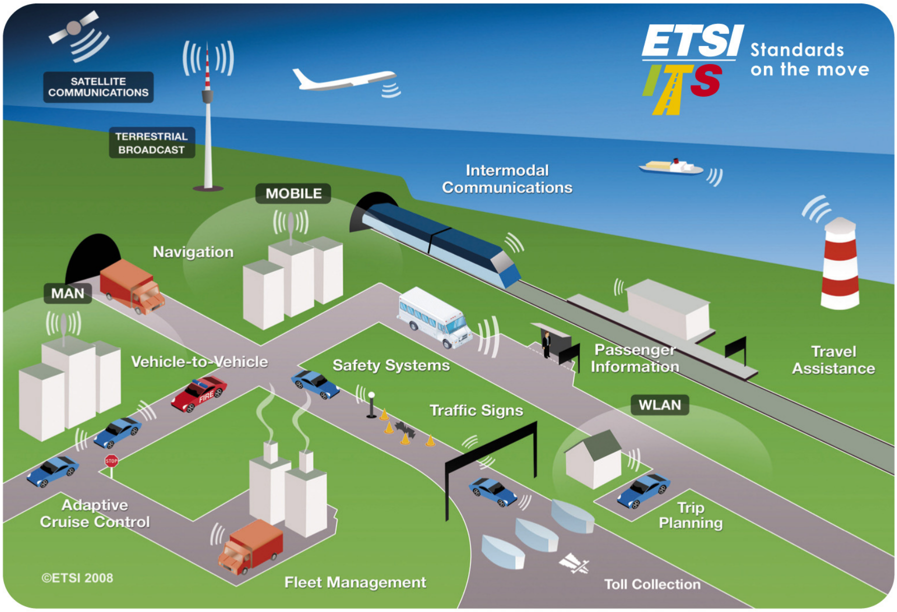


Figure 2.1: ITS: the big picture [1].

At first, ITS solutions were standalone, being unable to share data and cooperate. A newer paradigm, called C-ITS, enables the communication and sharing of information in order to facilitate actions and improve safety, sustainability, efficiency and comfort. In particular, C-ITS solutions cover the scope beyond stand-alone systems by promoting communication and cooperation between its

[1] Source: https://www.etsi.org/

participants [AFF16]. Examples of scenarios where the information exchange can be useful includes traffic jams, accidents, and road hazards, among others.

The real implementation of C-ITS, such as helping drivers to be more aware of other vehicles, disseminate warnings, and provide real-time information about traffic conditions, rely on connectivity among vehicles in the environment, as well as with the environment itself. In other words, C-ITS relies on both V2V and Vehicle-to-Infrastructure (V2I) communications, which are jointly known as V2X communications, to exchange critical information for ITS services, including Internet access and location-based services like parking management, or point-of-interest notification. To enable such interaction, technologies such as Dedicated Short Range Communications (DSRC) are used under special-purpose devices deployed within vehicles or in physical infrastructures. In addition, a dedicated variant of IEEE 802.11 operating in an allocated frequency band of 5.9 GHz, has emerged has an enabling technology to support C-ITS [Fes14].

Moreover, under the V2X paradigm, C-ITS applications can also make use of mobile infrastructure elements through the deployment of Unmanned Aerial Vehicles (UAVs). As an example, by using UAVs, a road support team can rely on these devices, flying them around the location of an accident to provide immediate basic support or to merely report the situation. In addition, UAV-enabled C-ITS applications also aim at providing efficient information when communicating with vehicles underground about traffic conditions. This opens up new opportunities for further development of C-ITS, as multiple UAVs can act as mobile Road Side Units (RSUs), flying together, collaborating, and co-ordinating to execute a specific mission to support the delivery of ITS services (see Fig. 2.2).

2.3 Vehicular Networks

The Intelligent Transportation Systems (ITS) paradigm includes Vehicular Networks (VNs) that cover services like traffic and mobility management, in addition to safe driving. VNs are wireless communication networks that support enhanced driving and communications among vehicles. Communications with the traffic environment under the V2X paradigm includes both V2V and V2I communications, making vehicles act as relays to support cooperative driving among communicating vehicles on the road [KAEHJLW11].

VNs make use of two different technologies depending on the application: ad-hoc communications through the 802.11p standard, and infrastructure net-

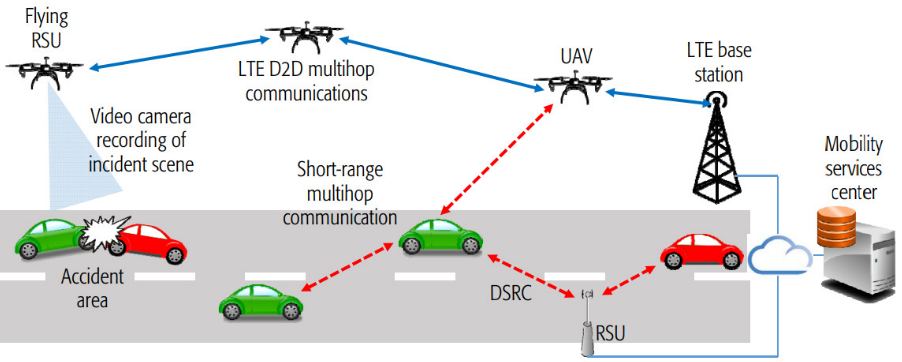

Flying RSU
LTE D2D multihop communications
UAV
LTE base station
Video camera recording of incident scene
Short-range multihop communication
Mobility services center
Accident area
DSRC
RSU

Figure 2.2: Example of an use case scenario for UAV-enabled ITS: UAVs can be used as flying RSUs to capture video recordings of an accident, and then relay data to a mobile services center [MGAUKT17].

works that use cellular networks or Wi-Fi. Using infrastructure networks requires a base station to be deployed, and depends on a centralized network topology. As for ad-hoc communication, it relies on direct links between vehicles or mobile stations, which in turn create a VANET, which is in fact a subset of Mobile Ad-hoc Network (MANET).

The V2X paradigm encompasses the exchange of data between cars (V2V), and between cars and infrastructure elements (V2I), which act as relays towards a wider network or the Internet itself, being essential to provide ITS services and applications [GK11]. However, since vehicular networks are quite dynamic, the challenge comes when disseminating messages containing critical information that needs to be delivered timely, and as fast as possible, especially in emergency and safety scenarios [ZGFZZ16; KAMA13].

Another challenge to consider in V2X communications is when the coverage area lacks infrastructure support. Although vehicular communications can rely on various radio access technologies [HCLMMWY10], e.g. using 4G LTE technology to support communications in areas with limited infrastructure like rural areas [NRSG17], major problems typically arise when the communications take place in areas that have no infrastructure support at all.

Ad-hoc communications is the preferred networking approach in vehicular scenarios, since the distribution of information should involve multiple vehicles. In the case of using Wi-Fi direct, although the deployment is fast and easy, its usage is limited to a single hop communications. Thus, C-ITS services cannot be fully deployed using this technology.

2.3.1 Vehicular Ad-hoc Networks

As part of the ITS concept, communications between vehicles play an important role in distributing relevant information. This kind of communication can be enabled through V2V communications which, combined with the ad-hoc networking paradigm, give rise to VANETs. VANETs create a network of vehicles that communicate with each other. These networks allow transportation-related applications to be deployed [PDLFEBC09].

VANETs differs from other MANET subsets as they have unique characteristics. In fact, VANETs have predictable mobility, as vehicles are constrained by road topology and layout. In addition, VANETs are characterized by a variable network density since the network depends on the vehicle density, which can vary due to, e.g., traffic jams. Moreover, the most prominent characteristic is the rapid change rate of the network topology. This is associated to the high speeds of vehicles (especially in highways) [JJUR17], and the presence of obstacles.

Many C-ITS applications nowadays benefit from the use of VANET-related technologies. C-ITS applications that use such technologies will be explained in the section 2.4.

2.3.2 Flying Ad-hoc Networks

UAVs, or drones, are currently becoming an emerging solution for critical situations, i.e. disaster response like Search And Rescue (SAR) [WT10] and fire fighting [YZL15]. In addition, compared to terrestrial communications, the adoption of UAVs not only offers a quick and flexible deployment, but also the chances of having Line-of-Sight (LOS) with the receiver increases due to their higher altitude [BST13]. Recently, thorough studies analyzed the capabilities of UAVs as communication agents, and their usefulness in several application scenarios [JMAJAZ17; BCCMPR17].

UAVs can also cooperate with ground vehicles in a particular network, allowing to improve the data exchanges between them. This approach offers benefits to multiple ITS applications [CCGI+05; MGAUKT17], like rescue and disaster assistance operations [YQYRHB17], and remote sensing [DDLW09]. In such cases, we typically rely on multiple UAVs to conform a network between themselves, creating what is known as a Flying Ad-Hoc NETwork (FANET). As a subclass of Vehicular Ad hoc NETworks (VANETs), FANETs differ from standard VANETs since they are characterized by highly mobile nodes moving freely in the 3D space; on the contrary, VANETs are restricted to 2D movements along streets. [ZK16]. The use of multi-UAV systems can also be beneficial for improving the attainable transmission range and efficiency, as packets can be relayed and forwarded between UAVs to minimize the drawbacks of link interruptions [WJHRMH17].

2.3.3 UAV Assisted Vehicular Networks

By combining UAVs and terrestrial vehicles in the same network, FANET and VANET can be integrated to create a seamless network for C-ITS applications. Figure 2.2 depicts how the UAVs can support VANETs for ITS applications. The research work presented in [MGAUKT17], suggested that UAVs can be used for capturing data about accidents and relay information to the vehicles in the network. In this particular case, a UAV can act as a flying RSU that is mobile contrarily to conventional fixed RSUs on the street. The communication between UAVs and cars (air-to-ground communications) is discussed in [ZCLS15]. For instance, UAVs can assist vehicular networks by acting as intermediate relays. UAVs can also be deployed in an area where a disaster occurs, therefore acting as bundle carriers and relays under the Delay-Tolerant Network (DTN) paradigm. In another work [JZ17], the authors analyzed the possibility of deploying UAVs as flying base stations for communications to rescue vehicles in disaster events; in that scenario, vehicles should cooperate and maintain connections between them when terrestrial communication infrastructures are not available, a task that is assumed by UAVs. In such case, the authors in [SKPK16] analyzed the inter-connectivity of disconnected groups of cars that rely on UAVs. They studied the impact of increasing the number of cars in the group on the overall transmission quality. This case can also be applied in a scenario where a UAV is deployed as a Store-Carry-Forward node for vehicular networks [FAAK17]. More efforts in this area include the deployment of VD-NET (Vehicle-Drone hybrid vehicular ad hoc Network) [WFZGW16], which offers better message transmission by equipping vehicle nodes with UAVs under its instruction so that it can communicate with other vehicle nodes over a greater transmission range. In another related work [OLZGLY17], networks of UAVs can be deployed throughout a city to achieve optimal information distribution. The presence of tall buildings and landmarks can cause disruption to the radio signal, resulting in frequent communication failures between vehicles. Hence, UAVs can assist on routing the information to the vehicles. In [SK17], authors considered the fusion of various ad-hoc networks, whether it is on air (FANET) or on the ground (VANET), to create a search and tracking-based guidance system.

One of the challenges of FANETs is obtaining an accurate radio propagation model, as this problem differs from the typical scenarios addressed in the literature. Most works focus on the link between UAVs and a static ground base station [YD16], which typically has line-of-sight conditions with the UAV. In [AHKJ14], the authors modeled air-to-ground path losses with UAVs; again, the ground receiver was not a moving node. Instead, in this thesis, we focus on UAV-to-car communication, which is currently a very important topic [ZCLS15; JZ17; OLZGLY17], and that differs from the ground base case due to the vehicular mobility. Thus, it becomes necessary to characterize the communications between UAVs and moving vehicles, and to derive a model that can be used in simulations combining FANET and VANET scenarios.

2.4 Implementations of C-ITS

Intelligent Transportation Systems (ITS) are able to provide efficient solutions for traffic-related issues, such as safety and efficiency [SABB17]. When attempting to make roads safer, ITS can provide systems that reduce the number of accidents taking place [MTCCM10], along with safety-related applications [HPCCM18]. Various examples of ITS applications take advantage of the communications capabilities of vehicles. For instance, it is possible to provide services and applications that can help at improving safety, efficiency, and comfort in transportation by gathering and sharing information.

2.4.1 Classification of C-ITS Applications

Various challenging applications can take advantage of C-ITS technologies. In fact, we can classify applications according to their purpose. European Telecommunications Standards Institute (ETSI) has defined a Basic Set of Applications that is composed of three main application categories [ETS09]:

- Traffic Safety
- Traffic Efficiency
- Other Value-Added Applications

Traffic Safety Applications

A remarkable advantage of ITS solutions is the capability of providing safety applications. In fact, ITS solutions can conveniently provide warning notifications in emergency situations. For example, notifications about dangerous traffic conditions or about emergency breaking can be beneficial in providing a safer traffic flow for the drivers in a city [PDLFEBC09]. With the goal of reducing the probability of traffic accidents, recent works include the development of safety-related applications [EZL14]. One example is the application that delivers emergency alerts. This application can be deployed when an accident occurs, providing an emergency call [Qi08].

Among all safety-related applications, the technology implemented relies on the exchange of information inside two types of safety messages that have been standardized by ETSI [ETS10b; ETS10a]:

- Decentralized Environmental Notification Messages (DENMs)
- Cooperative Awareness Messages (CAMs)

CAM messages are sent periodically with the information containing the position, time, and mobility status in order to create situational awareness among vehicles. On the contrary, DENM messages are triggered by specific events, and aim at alerting users of problems on the road. Both CAM and DENM messages are broadcasted to vehicles to disseminate safety information. In general, ETSI has categorized the type of applications that are safety-related as [ETS09]:

- Vehicle Status Warnings
- Vehicle Type Warnings
- Traffic Hazard Warnings
- Dynamic Vehicle Warnings
- Collision Risk Warnings

We should bear in mind that, for the distribution of real-time safety information, required by critical applications in the context of ITS, the diffusion of messages should be reliable and time-bounded [SG12]. Another consideration to consider in the critical safety application domain is the delay of the message delivery process itself. The performance of distributing messages through V2V communications is expected to have a low transmission delay, although with a limited reliability. So, vehicles that belong to a certain V2V network and facing a particular emergency situation should be alarmed as soon as possible. Otherwise, if the delay is too high, the relevance of the message would be reduced, and it would probably expire [MCR09]. According to [GMU14], improving traffic safety is currently the second highest strategy priority in ITS, being the first priority for future ITS solutions. Thus, we find that safety issues in the context of ITS are indeed essential.

Traffic Efficiency Applications

Another purpose of providing C-ITS is to improve the traffic fluidity. This is done by reducing the travel time and traffic congestion. Different applications have been proposed to improve traffic efficiency, including applications providing navigation in order to improve traffic efficiency. One example is the vehicular social networks that promote information sharing [VL15].

By sharing information, the user can take navigation decisions, for example, to avoid highly dense traffic, or areas with road constructions. This way, road users can also obtain indirect economic and environmental benefits. Other features in providing traffic efficiency includes the dissemination of speed limits information, recommended itineraries, or route guidance and navigation, detour notifications, electronic toll collection, and platooning applications [MK18].

Value-Added Applications

Besides driving efficiently and safely, road users can benefit from C-ITS technologies to improve comfort and convenience. Applications providing infotainment, travel information, or journey planning, are available for this purpose. As examples, there are already applications related to the notification of Points-of-Interest, parking access, media downloading, instant messaging, and remote diagnosis for car manufacturers to provide after-sales services [HL08].

2.4.2 Accelerating the Deployment of C-ITS

Although C-ITS technologies are ready for deployment, car manufacturers are expected to include them gradually, starting with high-end models. Considering that the renewal rate of the vehicle fleet is low, this will slow down the deployment of VNs. Moreover, dashboard-integrated OBUs can become obsolete as time goes by. In general, the usefulness of these OBUs remains below the vehicle lifetime.

In the meantime, the widespread adoption of smartphones and tablets enables the deployment of technologies with high connectivity requirements. Since smartphones can be easily integrated in vehicles, and since they have multiple network interfaces, it makes them a suitable platform to be integrated within C-ITS environments.

Various efforts in integrating smartphones in VNs have been proposed. The Car Connectivity Consortium (CCC) released Mirrorlink, that provides technology to use smartphones to run tasks instead of the OBU, and enables the OBU to display information computed from the smartphone. Other renown efforts, such as Android Auto and Carplay, use the same methods [Gre15]. However, these products rely on the Internet infrastructure to provide vehicular communications, meaning that they do not benefit from VANET communications. In addition, these proposals are dependent on specific content providers and on telecommunication operators [TPCCM16]. Hence, there is a need to provide an interface that integrates off-the-shelf smartphones into VANETs. This will allow users to crate their own VN, and to deploy VANET applications using readily-available technology.

Using Smartphones for C-ITS Applications

Several research works have been proposed to exploit the benefits of smartphone technology in C-ITS applications. An example of such efforts includes developing an Android platform that integrates the Open Gateway Service Initiative Vehicle Expert Group (OSGi/VEG). This allows us to generate a vehicular Android/OSGi platform that provides an open environment, rich class-sharing, proprietary vehicular applications, and remote management of the Application Programming Interfaces (APIs). With the ceaseless research efforts involving smartphones with vehicle telematics, the driving experience is nowadays enriched with multiple features related to sensors, efficiency, and interfaces for the sake of safety and convenience [WSH17].

Specifically, when integrating safety-related applications with Android-based smartphones, authors like Whipple et al. [WAB09] proposed using an Android application in a school area to notify passing-by drivers about the speed limit. The Global Positioning System (GPS) feature is used by this Android Public Safety application to discover the location of the vehicle, and it then uses the Google Maps API to define the location of nearby schools. When the driver surpasses the speed limit in a school zone, this application will send an alert. Other safety-related application development efforts include analyzing the behavior of drivers and notifying it. Drivingstyles [MCCM13] is one an example of such an application, warning the driver based on the behavior analyzed.

Other contributions combine visual information with sensor data to provide safe driving. One example is Carsafe [YLC+13], where the researchers developed an application that can detect the condition of the driver. Using computer vision technology, by capturing the visual information from the front camera and also the sensors that are embedded, the application analyses the driver's state, determining whether he/she is drowsy or distracted. The use of visual information for safety purposes is also proposed in EYES [PACCM15]. The authors used the smartphone cameras to capture video that is streamed between drivers. This video sharing application is developed by pairing two driver smartphones, enabling safe overtaking.

In addition, safety applications consider navigation data as well. There have been a lot of applications involving navigation services on smartphones for vehicles. In particular, these applications aim at improving user convenience. One example is a navigation system for smartphones that is not infrastructure-dependent [TD13], using maps that can be rendered offline. Another example involves the use of a projector for cyclists [DFF14], to provide navigation services and minimize distractions while riding. In a car context, a few works even proposed a user interface for displaying audio as an in-vehicle device that provides car navigation [YIIKK14; MYTK14], where the interface is based on a smartphone. The authors of [KKMP17] built a Java API for these kinds of in-vehicle devices. In terms of providing efficient driving, a smartphone can also be used for navigation assistance considering an efficient route to avoid traffic congestion [LLKXBNI14; LLKXBNI16]. Navigation services can also be combined with safety or emergency alerts. The application called iOnRoad[2] uses a smartphone to assist users when driving. This application provides information like an advanced collision warning system, off road alerts, headway distance alerts, and speeding alerts deployed in the smartphone. Another application

[2] iOnRoad. http://www.ionroad.com/, Jan. 2018

[VZB17] involves OBD devices as sensors to trace events and broadcast alerts using the Wi-Fi AP/client mode.

Navigation applications can make use of social networks to circulate information between users. In [SKNI13] authors used voice tweets as part of vehicle social network groups to share driving experiences between drivers. These tweets are based on location and destination, and can be used to calculate an optimum route based on collected tweets. This proposed application, called NaviTweets, is later expanded to benefit from a vehicular cloud, allowing to report and share traffic information [KLKNI16]. Waze[3] is the most popular driving assistant application. Not only does the application provide navigation features, but it is also based on social networks. Hence, the information is shared among users, allowing to offer traffic reports, traffic jam alerts, under construction road alerts, or notify the presence of the police. However, these social network driving applications are heavily dependent on infrastructure support or the Internet. Thus, they do not benefit from VANET-assisted features.

GRCBox: an On Board Unit for C-ITS Applications Support

A device that is able to provide ad hoc network connectivity in the 5.9 GHz band is needed to enable V2V communications. If we use smartphone devices, like Android phones, to launch applications, an option would be to enable ad hoc network connectivity in this device. However, in order to do that, a rooted Android phone is required, thus not being very practical for end users. In addition, the communications range achieved would be quite limited. An alternative approach is GRCBox [TPCCM15], a solution capable of providing ad hoc communications without having to root smartphones. GRCBox is a multi-interface low-cost connectivity device based on Raspberry Pi devices. GRCBox emerges as a solution to bypass the difficulties of configuring an Android device in the Ad-hoc mode. In fact, the GRCBox can act as a router for the automatic deployment of ad-hoc mode communications. With this capability, V2X communications are fully supported, and a seamless integration with smartphones is achieved.

A Raspberry Pi 2 device model B1 is the main hardware of our GRCBox, a single board computer that has the size of a credit-card, and that costs only 35 USD. This device has enough CPU power to perform low-scale network routing. A Raspbian distribution based on Debian is installed in this device.

[3] Waze. https://www.waze.com/, Jan. 2018

This Raspbian distribution supports the current networking hardware while avoiding common problems of other embedded operating systems.

Each GRCBox is equipped with several network interfaces: one inner interface acting as an Access Point (AP) for the users, allowing them to connect to the GRCBox using smartphones supporting WiFi communications in the 2.4 GHz band. The outer interface offers vehicular communications, where it connects to a vehicular network in the 5.8 GHz band. In addition, one can add other network interfaces that connect to the Internet. For instance, one network interface can connect to a WiFi access point, and yet another one can be used to connect to a 4G cellular base station. Figure 2.3 shows a descriptive diagram of our GRCBox connectivity features.

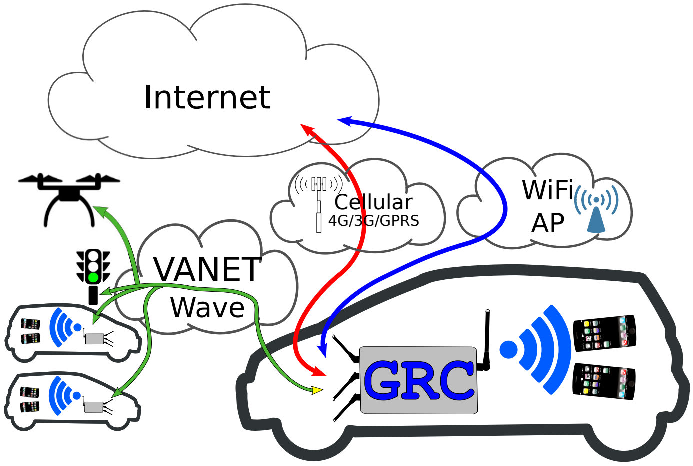


Figure 2.3: GRCBox hardware module connected to a VANET with three different nodes [TPCCM15].

Several services are provided by the GRCBox. GRCBox's inner interface acts as a soft-AP (Access Point) for smartphones. Once these smartphones are connected to the GRCBox, they can access the services that run on the external networks. Since every connection is forwarded by the GRCBox, any application that requires using an available interface that differs from the default one (offering Internet connectivity) must notify the GRCBox. These steps require rules that are defined by rule type, interface name, protocol, source port, source address, destination port, and destination address. By tuning an Android application to be able to communicate with the GRCBox, a full VANET integration can be achieved, enabling C-ITS support.

2.5 Summary

In this chapter, we analyzed the most important concepts related to C-ITS fundamentals. Additionally, we have provided an overview of vehicular network theories. Moreover, we analyze previous works focusing on C-ITS applications. Several efforts include accelerating the deployment of C-ITS. With the broad usage of smartphones, these can be integrated with OBU to support the vehicular communications.

One of the highlights of the use of C-ITS is safety. However, since safety applications are usually time critical, an adequate wireless channel should support the transmission of time-bounded messages. The next chapter will be dedicated to vehicular channel fundamentals and the modeling of vehicular communications.

Chapter 3

An Overview of Modeling Vehicular Networks Communications

3.1 Introduction

In this chapter, we present an overview of the vehicular communications channel as one of the components that support C-ITS applications. First, we will explain the characteristics of wireless channels. Then, we will review existing radio propagation models. finally, we will analyze in more detail the modeling of the vehicular communications channel, followed by a description of the simulation tools where channel models will be integrated into.

3.2 Wireless Channel Characteristics

In wireless communications, the signal quality received from the transmitter is defined by the propagation channel. Problems such as dropped or lost packets, interference, or coverage, are caused by several phenomena. Signal strength, which defines the quality of the signal, will attenuate as a function of distance due to path loss, or be blocked by an object due to shadowing effects [Rap+96].

Moreover, signals in form of electromagnetic waves usually do not reach the receiver directly due to obstacles that block the LOS path. The waves are affected by reflection, diffraction, and scattering caused by obstacles like trees or buildings, etc. The effect is known as multipath propagation, which consists of a sum of attenuated, delayed, and phase-shifted replicas of the signal that is transmitted, which are mixed at the receiver causing signal quality degradation [Pat01]. Fig. 3.1 illustrates the scenario where signals from a base station are affected by multipath propagation effects.

In more detail, the main phenomena that affect radio waves are:

- Reflection: a condition when waves bounce back toward the source.
- Refraction: a condition when the waves change the angle from their general direction, as they are deflected when traversing a specific medium.
- Diffraction: a condition when a wave has a flared out geometric shadow of the slit when passing through a small hole.
- Scattering: a condition when the waves are forced to deviate from a straight trajectory by one or more paths due to localized non-uniformities in the medium through which they go through.

In addition, the signal quality can also be influenced by the Doppler effect. Since wireless nodes can be mobile, mobility can cause a small frequency shift of electromagnetic waves.

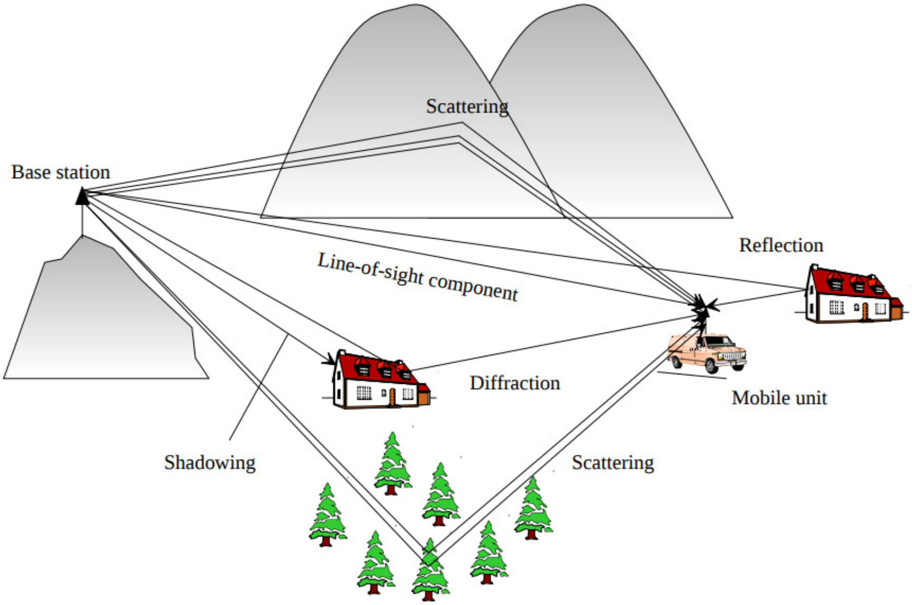


Figure 3.1: Multipath Propagation Scenario [Pat01].

3.3 Radio Propagation Models

Various radio propagation models exist that define the characteristics of wireless channels. In this section we will categorize them according to propagation scale and modeling approaches.

3.3.1 Models by Propagation Scale

We can classify models into three groups according to their propagation scale. The first one is path loss, in which the model estimates the average signal loss due to the propagation distance. Then, we have signal attenuation due to the propagation environment, called shadowing or large scale fading, which fades slowly. The last one is called multipath propagation, which fades very fast, thus being classified as small-scale fading [Gol05]. An illustration to describe the difference between these three models is represented in Fig. 3.2. This figure illustrates the ratio between the received and the transmitted power on the log distance, which is affected by the combination of path loss, shadowing, and multipath effects.

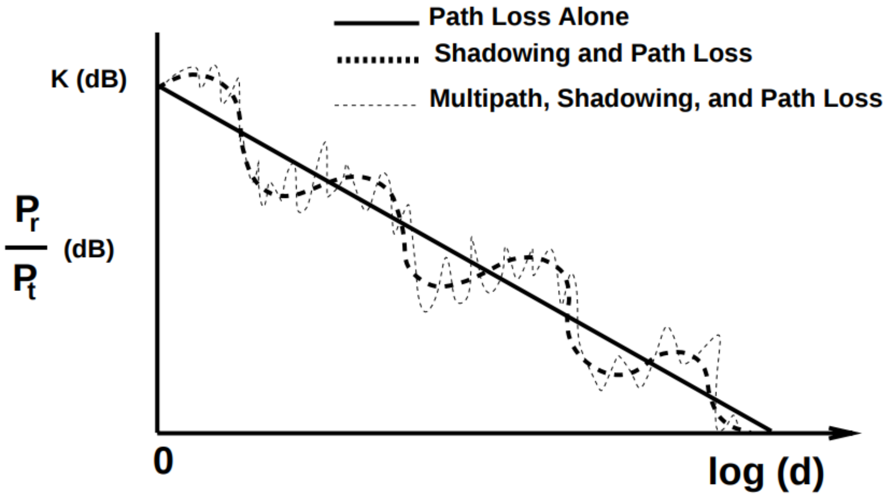


Figure 3.2: Path Loss, Shadowing, and Multipath effects versus Distance [Gol05].

Path Loss

Path loss is the attenuation of an electromagnetic wave through space. The received power decays exponentially with distance.

Path loss is usually expressed as:

$$PL(dB) = 10log\frac{P_t}{P_r} \tag{3.1}$$

which represents the ratio of the transmitted (P_t) and received signal power (P_r). Path loss usually attenuates the signal, and so the path gain is usually defined as the negative of the path loss measured in decibels.

Several path loss models have been proposed in the literature, for instance [Rap+96]:

- Free space propagation.
- Two ray model.
- Ray tracing.

Shadowing or Slow Fading

Predicting the received signal strength should take into account shadowing, diffraction and scattering created by the environment. The condition where the average received power is different at different location, despite having the same separation distance, is called shadowing. This condition usually occurs due to the presence of large objects within the signal path. Hence, blocking objects may result in a faster variation of the received signal strength.

Log normal shadowing is the most used model to predict signal attenuation due to the shadowing effect [HM77]. However, in most cases involving mobile networks, several efforts have been made to empirically model shadowing effects.

Fast Fading or Multipath

The case where the signals received, which are actually copies of the same transmitted radio wave, may have different amplitude and phases and arrive at different times, is usually caused by multipath fading. This effect is known as small-scale fading or fast-fading effect due to the fast fluctuation of the received signal at the receiver for short distances or short periods of time. Usually, this multipath effect is caused by small transient objects reflecting the radio signal.

As for fast fading or multipath effects, the models commonly used are:

- Rayleigh distribution model [SS93].
- Nakagami-m [Cha79].
- Durgin [DR00].

3.3.2 Modeling Approaches

Since a channel model defined by certain radio propagation characteristics is usually represented by a set of algorithms and mathematical expressions, there are several approaches that can be adopted to model a wireless channel. We will classify these models as Deterministic, Semi-Deterministic, and Empirical ones.

Deterministic Models

Deterministic models are usually defined by environment models. The environment model describes the objects that have an impact on propagation according to their position, geometry, type of material used and surface properties.

The most common used deterministic channel model is the ray tracing [ANM95] one, which is an approximation model based on Maxwell's equations. One of the characteristics of this model is that it is site specific. In addition, calculating the signal power using deterministic channel models is usually computationally demanding.

Empirical Models

Empirical models are derived from extensive measurement campaigns, where the measured data is analyzed statistically so as to retrieve information on the propagation parameters and the path loss decay exponent. The empirical mathematical formula is dependant on the environment, the antenna height, and the signal frequency. The advantage of using empirical models is that they offer a fast approach to compute signal attenuation. However, the model is not always accurate as the scenarios and the environment limits their applicability. Examples of empirical models are Okumura-Hata [Oku68; Hat80] and Longley-Rice [LR68].

Semi-Deterministic Models

Deterministic models are able to achieve the most accurate estimations when calculating signal attenuation. However, the computational time is usually high. On the other hand, empirical models can provide a quick estimation of the signal attenuation. However, they can be less accurate as they are site-specific. Thus, semi-deterministic models emerge as a compromise between these two approaches. In particular, they combine the use of the two methods. A semi-deterministic model usually includes the path loss, as well as scattering and diffracting mechanisms. One of the semi-deterministic models is the COST 231 Walfisch-Ikegami Model [CK99], which takes into account free space losses, along with losses associated to diffraction and scattering from rooftops to streets, and the loss due to multi-screen diffraction.

3.4 Channel Models for Vehicular Environments

For scenario-specific communications, in our case involving vehicles as communication nodes, several measurement campaigns already exist. In addition, empirical models were also proposed as a result of measurement campaigns. In this section we will describe the existing related works in the field of vehicular communications channel modeling.

3.4.1 Vehicular Networks Channel Measurement Campaigns

Several efforts have been made to measure the quality of the vehicular communications channel. In particular, some of these works have studied its impact on the packet delivery ratio in VANET environments. The work by Bohm et al. [BLJL10] investigated the impact of the loss of LOS in terms of V2V communications performance, finding that limited LOS between vehicles can still enable communications. Other interesting related works include the deployment of RSUs in vehicular environments [LLLC12; GSB12], studying how urban scenarios with buildings and vegetation affect V2I (Vehicle to Infrastructure) communications. These measurement campaigns usually lead to results that can be used statistically to produce an empirical model.

3.4.2 Car-to-Car Communications Modeling

The most common channel model used for vehicular communications usually involves the radio link between cars (i.e., V2V or Car-to-Car). The models can have specific purposes, and be affected by environment conditions. Below we have classified such models based on the environment or the road topology, and the shadowing objects.

Building Shadowing Models

Several works studied specifically the impact of the presence of building obstructions on the signal path between vehicles. The work of Sommer et al. [SEGD11] modeled the IEEE 802.11p/DSRC radio shadowing in urban environments. Their model can estimate the signal attenuation of the wireless radio transmission when having buildings as obstacles. Other works include assuming the presence of the building defining it as Non Line-of-Sight (NLOS) merely by analyzing the road segment itself, without requiring maps that include the accurate positions of buildings [GFPG10]. In addition, the work by Oishi et al. [OAW06] examined the influence of the building density and the inter-vehicular angle on the LOS probability. Another attempt includes the work by Martinez et al. [MTCCM09], which proposed a radio propagation model called Building and Distance Attenuation Model (BDAM). The proposed model takes into consideration the signal attenuation due to the distance between vehicles, and the presence of buildings blocking LOS that absorb radio waves. Derived from measurement campaigns, the work by Tchouankem et al. [TZS15] investigated the impact of building at urban intersection, and proposed a realistic geometry-based path loss model.

Vegetation Shadowing Models

Other than buildings, the presence of trees can also obstruct the signal path from the transmitter to the receiver. Several works investigated the impact of foliage or trees on the communications performance. One of such works includes a proposed empirical model by Tchouankem et al. [TZSW13]. These researchers proposed a model that can calculate path loss caused by vegetation obtained empirically from experiments in rural areas conducted in different seasons to see the impact of leaved and leafless vegetation. Another research work was performed by Chen et al. [CC14]. A model was proposed in their research work derived from an analysis of foliage scenarios where high profile trees, along both roadsides, affected LOS communication links between testing vehicles. The model is based on the multi-ray model to estimate the attenuation.

Car Shadowing Models

Vehicles blocking the LOS can also significantly attenuate radio signals. The work by [MBSTB10] investigates the impact of a van acting as an obstacle between communicating cars. A similar comprehensive study on the obstruction of large vehicles was made in [VANWOT14].

Other works also include a model derived from measurement campaigns. An example of such a work was presented by Sommer et al. [SJSTCD15], where they proposed a model for signal shadowing by vehicles based on dynamic beaconing. Another contribution is the work by He et al. [HMTZAZ14], where the researchers proposed a signal attenuation model due to the shadowing of a large school bus. The work by Wang et al. [WJHRMH17] specifically studied the impact of a truck as an obstruction for truck-to-truck communication. The work proposed a double-directional characterization of the truck-to-truck channel.

The issue of having cars shadowing communications is also investigated by Al et al. [AAAAKL18] under different environmental conditions by varying both the vehicle speed and the vehicle density. Their work presented an obstacle-based radio propagation model that considers the effect caused by the presence of obstructing vehicles in the line of sight.

Highway Models

Since cars can obstruct communications, there is a need to study communications in specific environments where many cars are present. An example of such an environment is the highway. A highway having more lanes, hence having more vehicles present, is the environment chosen by different researchers. One of such works was presented by Renaudin et al. [RKVO10], where they measured and modelled the V2V radio propagation channel in highways. The empirical model is a combination of the log-normal and the Weibull distribution.

Other measurement campaigns that led to similar models include the work by Abbas et al. [ASKT15], where they proposed an empirical model applicable to highway environments. Moreover, the work of Akhtar et al. [AEO15] characterized communications in a highway scenario; in their work they compared the performance with other common obstacle-based channel models.

Intersection Models

Analyzing the importance of intersections in vehicular communications is an interesting issue. At intersections, various obstructions can be present, whether it is a building, a plant, or a car.

Focusing on street intersections, Karedal et al. [KTAKPBM10] and Mangel et al. [MKH11] designed specific propagation models from measurements performed at intersections. Other works proposing an empirical model, able to predict packet delivery ratio results from measurements made at three different street intersections, was performed by Mahler et al. [MPWKK14]. The different street intersections are classified according to the presence of obstructing buildings, and the availability of large reflection surfaces between transmitter and receiver. In addition, the work by Jaktheerangkoon et al. [JNNR18] investigates the impact of blind corners with obstacles. According to their experimental results, they observed that the minimum distance between the vehicle and the corner can effectively be represented as the key parameter in the model.

The work by Schumacher et al. [STNKZLW12] finds that, in an urban scenario where there are buildings blocking the line of sight, communications are possible for distances ranging from 85 m to 115 m. By using V2V communications under NLOS in the 5.9 GHz band, their results showed that, not only do buildings affect the communication, but they also the width of the street representing the largest street intersection has a significant impact on the delivery probability.

Another research work [TZSW13] investigated the effects of vegetation on the performance of V2V communications at intersections. Tests were based on the 5.9 GHz communications band, under NLOS conditions in a rural environment having different types of vegetation for the different seasons; results showed that the packet delivery ratio clearly depended on the type of vegetation and season. When transmitting a message between vehicles, a third vehicle located at the intersection would also affect communications, and the effect of co-channel interference will have an impact under both LOS (open space) and NLOS (with buildings as obstructions) conditions. The work in [TL15] addressed this issue through measurements in the 5.9 GHz band. Experiments showed that a single vehicle would interfere and decrease the delivery ratio, no matter whether the vehicle is placed near the receiver, near the sender, or between the sender and the receiver.

3.4.3 Car-to-Infrastructure Communications Modeling

Chanel modeling studies also include the analysis of the communication between car and infrastructure, or V2I. A work by Aygun et al. [ABVW16] investigated V2I propagation issues due to the presence of vehicle obstacles, foliage obstacles, and building obstacles. Other works [LKLCS16; GAVR+18] also analyzed the V2I environment, and proposed a model for this type of communication. In more detail, the work by Granda et al. [GAVR+18] focused on the statistical analysis, concluding that the Lognormal, Gamma and Weibull distributions can explain the fading behavior of this multipath environment, being that the amplitude of the fading is highly affected by data dispersion.

Li et al. [LHJ18] made a channel model analysis for V2I communications in rural areas. These researchers proposed a model based on geometric information to calculate how the variable vegetation depth affected V2I communications. On the other hand, the work by Zhang et al. [ZCZZZ18] was performed in a typical urban street environment; they proposed a best-fit statistical model for spatial channel parameters based on measured data.

3.4.4 UAV-to-Infrastructure Communications Modeling

With the emergence of UAV applications in the scope of Smart Cities, UAV can be deployed in larger networks and communicate with the static fixed infrastructures available in our cities. Several research works analyzed the performance of the link between ground stations or base stations with flying UAVs. One of the works proposed recently [GKD18] used stochastic geometry to estimate the connection probability between UAVs and their base stations.

Other work includes air-to-ground path loss between an UAV and a static ground receiver [AHKJ14]. Authors found that the prediction is based on the near urban environment properties, and that it is dependent on the elevation angle between the terminal and the platform. Another work that studies the link between UAVs and a ground node was performed by Matolak et al. [MS15], where these researchers characterize the air-to-ground communications channel.

With the emergence of larger-scale UAV deployments, the field of channel modeling involving UAVs still faces significant research challenges. For instance, most UAV measurement campaigns were performed in environments with good LOS conditions. More extensive measurement campaigns are required for environments where obstacles are present, like dense urban scenarios.

3.4.5 UAV-to-Car Communications Modeling

We can only find few research works that specifically focus on the communications channel between UAVs and cars. As seen above, most research works focus on the link between UAVs and a ground base station [YD16], typically facing LOS conditions. These works involved a ground receiver that is not a moving node.

Through simulation experiments, the work by Jia et al. [JZ17] includes an analysis of both air-to-ground and ground-to-air propagation channels. However, their work relies on the model by Al et al. [AHKJ14], which actually assumes a static ground receiver, and not a mobile ground receiver.

3.5 Simulation Tools for Vehicular Communications

In order to obtain results from estimating the signal attenuation in particular scenarios, simulation experiments must be performed. For scenarios involving vehicular networks, there are several network simulators already available. Below we will discuss the characteristics of these simulators, and attempt to determine which one can be more adequate for simulating vehicular scenarios.

3.5.1 Network Simulation Tools

Network simulators usually model the behavior of a network. In particular, the simulator calculates the interaction between different network entities.

OMNeT++

Objective Modular Network Testbed in C++ (OMNeT++) [VH08] is a discrete event network simulator based on C++. OMNeT++ is open source and supported by a complete Graphical User Interface (GUI) with many features. However, the basic OMNeT++ framework is only used for networking with fixed links. It does not provide models for wireless network simulation. Nevertheless, there are frameworks developed for OMNeT++ to support mobile ad hoc networks. One of such frameworks is INET, that includes propagation models designed for vehicular communication.

With OMNeT++, there is a possibility of designing custom modular simulation models. In addition, the models can be composed with any granular hierarchy. One of the advantages of using OMNeT++ is the availability of extensive simulation libraries. Moreover, since OMNeT++ is a C++ based simulation tool, it can be combined with larger communications.

Network Simulator (NS-2 and NS-3)

NS-2 is a discrete event network simulator developed by UC Berkeley with the support of DARPA. For wired networks, due to the large number of models developed, it is used widely, and it was extended by the research group in Carnegie Mellon University with the feature of wireless networking. However, NS-2 is not widely used for vehicular communications simulation due to its high complexity and the difficulty of importing the vehicular mobility models to the framework. In addition, for large scale scenarios, the simulator is computationally expensive.

An optimized version of NS-2, known as NS-3 [HLRDK08], as been proposed. Compared to NS-2, the advantage of NS-3 is the ability to handle large-scale scenarios as it removes the C++/TCL interactions used by NS-2. The drawback would be the incompatibility of the models developed in NS-2 to be imported into NS-3.

SWANS

The Scalable Wireless Ad hoc Network Simulator (SWANS) [Bar04] is a MANET simulator developed at Cornell University. The simulator is based on Java, and it can simulate large scenarios. Real applications written in Java can be directly tested within this simulator. However, the simulation does not include any propagation or any mobility model for VANET simulations.

QUALNET

Qualnet [Sim11] is a commercial scalable simulator for wireless and wired environments that builds upon the GloMoSim platform. The QualNET framework is also able to handle large-scale simulations. This simulator has different layers that are connected through APIs. However, its usage is limited in the community since the simulator has only a commercial license.

3.5.2 Traffic Simulation Tools

Traffic simulators are able to generate the mobility of vehicles, being quite useful for traffic engineering purposes. Usually, for VANET simulation, the simulator models from the flow that is assumed as basic entity. Despite several traffic simulators have been proposed, we will cover those that are widely used by the research community.

SUMO

Simulation of Urban Mobility (SUMO) [BBEK11] is a traffic simulator that is open source, and that aim at simulating large scale vehicular mobility. Three different types of elements are supported: routes, trips, and vehicle types. A vehicle type refers to the physical properties of a vehicle in the simulator. A trip contains the information of the departure time and the destination edge. A route expands trip by defining all the edges through which a vehicle will pass. To define the simulation map in SUMO, different files should be imported so as define the routes used and the presence of obstacles. We can create our own simulation map, or we can extract a real map from the openstreetmap.org website.

VANETMObisim

VANETMObiSim [HFBF06] is designed for modeling vehicular mobility. Using VANETMobiSim, we can import maps and produce mobility traces with different formats. The pattern for the mobility of the vehicle can be set as a random trip, or be customized origin-destination pair. This simulator supports intersection management, lane changing, and overtaking models.

3.5.3 Interlinking Tools

To enable the full integration of a vehicular network in a simulation, different modules are proposed to link the network and traffic simulator. The mobility of a node in VANET simulations is affected by its surroundings, and is dependent from other nodes. Hence, the traffic simulator must react by changing the initial trace file during the simulation run, which is based on the feedback received from the network simulator.

In addition, other than involving ground vehicles into simulation, several efforts has been presented in developing simulation tools to model the characteristics of UAV-to-UAV communications. Recent contributions in interlacing UAVs and networking simulation tools will be detailed in this subsection.

Veins

Vehicles in Network Simulation (Veins) [SGD11] is a framework proposed by Sommer et al. It integrates a full simulation capabilities to replicate real scenarios involving vehicular communications. Veins can connect SUMO and OMNeT++, allowing the latter to influence the vehicles' movements produced by the traffic simulator in simulation time.

The Veins framework supports different channel models. Most notably, it supports the Break point Path Loss Model, Jakes Fading, Log Normal Shadowing, Nakagami Fading, Simple Obstacle Shadowing, Simple Path loss, and Two Ray Interference. Of all the channel model available by default in the simulator, only the simple obstacle shadowing model takes into account the real physical location of the obstacle that characterizes the communication. This simulator does not support yet the fast fading events associated to the irregular terrains that hinder the communication.

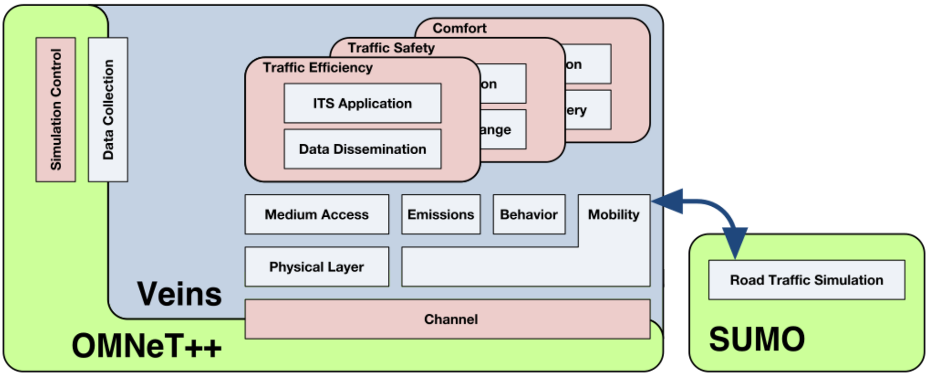


Figure 3.3: The architecture of Veins [SGD11].

iTETRIS

The Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTetris) [RMK+13] can link NS-3 and SUMO through a central control block named iTetris Control System (iCS). It is an improvement upon one of the first linking frameworks, called TraNS.

VsimRTI

V2X Simulation Runtime Infrastructure (VsimRTI) [Sch11] is a more generic framework. This simulation tool can be used by different simulators. In addition, this framework can work as an emulator for the direct testing of real applications in V2X environments.

AVENS

The Aerial Vehicle Network Simulator (AVENS) [MRPPLFPB17] was introduced as a specific FANET simulator. This simulation framework merges OMNeT++ and a flight simulator which can control the aerial vehicles. AVENS can model multiple UAV mobility structures. Hence, AVENS can not only simulate network aspects, but also aircraft information.

CUSCUS

Communications-Control Distributed Simulator (CUSCUS) [ZTSNDFB17] are proposed to simulate the networked control systems that involves UAVs. This simulator is capable of integrating the ns-3 and their own framework, called FL-AIR, that can create applications for UAVs. CUSCUS can run networking and control simulation at the same time.

FlyNetSim

FlyNetSim [BSL18] is an open source simulator that can simulate and evaluate UAV swarms operating within the Internet of Things. FlyNetSim interfaces two tools, ArduPilot and ns-3. FlyNetSim allows the simulation of UAV operations, such as sensing and navigation, and inner state dynamics. This simulator has been studied through case study-scenarios such as; UAV control over Wi-Fi, multi-network communications, Device-to-Device (D2D) communications for UAV swarms, Internet of Things (IoT) and data streaming, and emulation mode, where the simulator runs on a real-world UAV.

3.6 Summary

In this chapter, we presented the fundamental theories for vehicular communication channel modeling. We have reviewed several generic radio propagation models, and also the ones that specifically address vehicular communications. We have found that there are some research opportunities in which we can propose new models that extend or improve upon the existing ones. In addition, with the presence of simulation tools, we can test our model to validate its results, that performs a more realistic analysis. Throughout our literature study, we realize how to the best of our knowledge, there is no research work that specifically studies the channel model involving the connection between UAVs and cars (mobile ground vehicles) as well as simulating in simulation. In addition, there is no specific interlinking tool that involves UAVs in 3D space and cars (as mobile ground nodes) in the same framework. Hence, this will be the research opportunity, where we will fill the gap with this thesis. In the following section we will present our research work, which takes into account the theories and related works presented in this chapter and the previous chapter.

Chapter 4

Car-to-Car Communications Application and Modeling

4.1 Introduction

In this chapter we introduce Messiah, an ITS application that builds upon the theories studied in the previous chapter. In addition, since the performance of C-ITS applications might be affected by the environment, we also perform an empirical study of vehicular communication effectiveness at intersections in the 5.8 GHz band. This way, we are able to predict Messiah's performance in these kinds of environments. Moreover, detailed explanations of the application and channel modeling are also presented.

4.2 Smartphone-Based Driving Safety ITS Application for Vehicular Networks

We aim to design, implement, and deploy an application based on Android [Goz08] devices that displays maps with location information of relevant nearby vehicles as an implementation of C-ITS. This way, the application is able to warn the user about the presence of important vehicles (e.g. ambulances) that are approaching, allowing the driver to take navigation decisions based on that information. To accelerate the development phase, our application works on top of the OSM Automated Navigation Directions [OSM] application. OsmAnd is a popular open source navigation solution that can be run online or offline, and that was modified to meet our purposes. It relies on OpenStreetMap (OSM) [HW08] for displaying and rendering maps. The application has various useful features, and it can be used by cars, bicycles or pedestrians. Our application runs as a plugin for OsmAnd that allows creating a vehicular network; this way, participating vehicles can communicate and exchange information for notifying the presence of relevant nearby vehicles.

4.2.1 Application Features

By using our proposed application, called Messiah, users can participate in a "safe driving" network which allows users to choose among the different vehicle modes provided by the application. The application distinguishes users by classifying them into three kinds of categories. The first one is the administrative entity, acting as a police car or ambulance. The second one is the normal civilian car. The third one is a car that is in need of help. Based on the input from the user, three modes of operation are provided by the application:

- Administrative Mode.
- Civil Mode.
- SOS Mode.

By default, the application starts in the "Civil Mode". This means that, whenever the user launches the application in the default mode (without choosing the "Administrative" or the "SOS" mode), the application will assume that the user is in a standard vehicle. In "Civil Mode", the application will only receive and forward data using V2V communications to and from neighboring nodes. Any received information is displayed on-screen, as shown in Figure 4.1. In this Figure, we can see that the on-screen display shows not only the location of the user (in this case, the civil mode), but also the current location of an administrative vehicle through a siren light icon. The future trajectory of the administrative vehicle is also drawn by a line with an arrow showing its direction on the street. As the screen shows two sirens and two routes, this icon indicates the presence of two administrative vehicles. Another icon displayed in the screen is the SOS icon, which indicates that, within that area, a vehicle in need of help is present.

Whenever the user is in an emergency vehicle, whether it is a police car or an ambulance, the Administrative mode is selected. By working in this mode, participating nodes will forward packets received from neighboring nodes, and, in addition, they will announce their presence in the network by using broadcast messages that contain their current geographical location. Additional information is included in the message, such as the current route and destination. When in the administrative mode, the displayed screen shows the route and the destination, as well as the nodes other than civilian nodes. In this case we can see one administrative node (including its route), and one SOS node (see Figure 4.2). The interface of the application display is similar in the "Civil Mode", with the difference that, in the Administrative mode, the display will not show other vehicles. The exception is the case when there is a vehicle in SOS mode. In the background, the vehicle in administrative mode will also receive and forward this information.

Another feature of this application is the SOS mode. This mode enables users to ask for rescue. Thus, in the SOS mode, the user will generate SOS beacons to notify neighboring vehicles about its problems. These beacons will cause a special alert icon to be displayed on the receiving vehicle's screen. This method is used so that nearby vehicles can be warned that a problem is present, or that someone in need of help is close by, so that they can act accordingly.

A closer look at both Figure 4.1 and Figure 4.2 shows some basic features of the default OsmAnd application that are available. When the application is launched, the user can start the Messiah plugin by pressing the "eye" button, causing two other buttons to appear. This way, the user can choose whether to switch to the administrative or SOS modes, if applicable. If the user does not press these additional buttons, the application will assume to be running in the civil mode.

The dissemination of messages from the Administrative and SOS vehicles is limited to a 1 km range. This means that the information is broadcasted to neighboring vehicles within a 1 km radius, but not beyond that. So, if the source of information is more than 1 kilometer away from the receiving node, the forwarding process stops. This design choice accounts for the relevance of the information. In fact, by the time the information reaches some distant node, the source of the information may have moved to another remote location. This way, we also detect and avoid unwanted packets from circulating in the network for long periods. We discarded the use of time-based deadlines as a limitation since time is generated by each individual system, and so we cannot have a uniform and synchronized time base for all devices. Hence, the distance parameter was considered as the most adequate option.

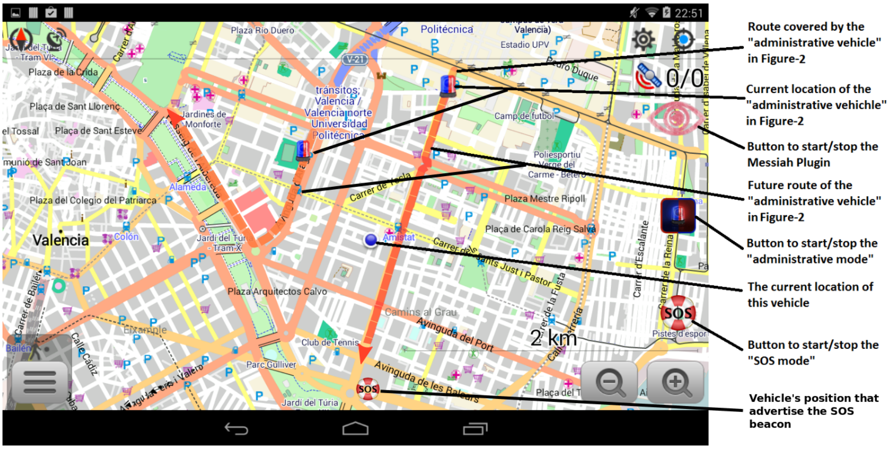

Route covered by the "administrative vehicle" in Figure-2
Current location of the "administrative vehichle" in Figure-2
Button to start/stop the Messiah Plugin
Future route of the "administrative vehicle" in Figure-2
Button to start/stop the "administrative mode"
The current location of this vehicle
Button to start/stop the "SOS mode"
Vehicle's position that advertise the SOS beacon
0/0
2 km
SOS
22:51
Valencia

Figure 4.1: Device working in "Civil Mode".

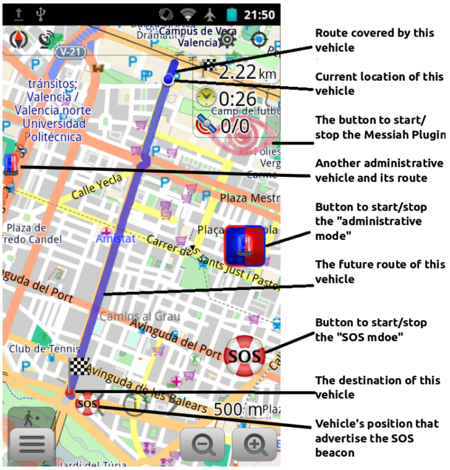

Route covered by this vehicle
Current location of this vehicle
The button to start/ stop the Messiah Plugin
Another administrative vehicle and its route
Button to start/stop the "administrative mode"
The future route of this vehicle
Button to start/stop the "SOS mdoe"
The destination of this vehicle
Vehicle's position that advertise the SOS beacon
21:50
2.22 km
0:26
0/0
500 m
SOS

Figure 4.2: Device working in the "Administrative Mode".

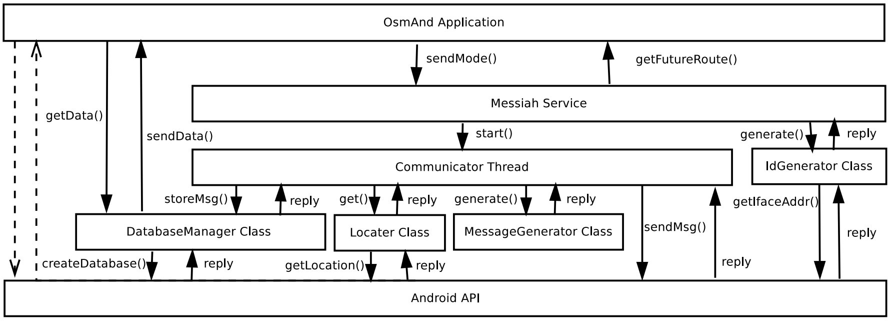


Figure 4.3: Architecture of the Messiah Application.

In the case of broadcasting future routes, the information is also limited according to distance criteria. In addition, the whole route information is not sent entirely to the receiver, but only a part of it. Hence, as seen in Figure 4.1, this node's display only shows a part of the sender route, with the other part being symbolized as an arrow's head indicating the possible route.

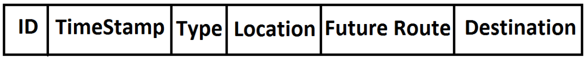
On the other hand, at the source's side (see Figure 4.2), it initially shows the destination location and a straight line without an arrowhead. The line indicating the source's route will be updated when the source's node approaches the receiver's node, or if it is still receiving notifications.

With the eye button present in the display, the user can join and leave the network at any time by pushing that button. In particular, this button will enable the application to activate or deactivate the Messiah plugin. Additionally, the application can also be run in the background. The user can still participate on the network of vehicles by running the Messiah application, even if the application is minimized, or if the user is running another application.

4.2.2 Implementation Details

Figure 4.3 shows different working blocks of our proposed application. The "Messiah Service" controls the functioning of the Messiah application. This service also receives the mode of operation based on the user's input. For sending and receiving data, the responsible thread is the "Communicator Thread".

In order to generate a unique ID for a node, the working block responsible for that task is the "IdGenerator Class". To track the current location of the device, the location service is provided by the "Locator Class". The messages that are received from the network are stored by the "DatabaseManager Class".



ID	TimeStamp	Type	Location	Future Route	Destination

Figure 4.4: Information contained in the message generated by the Messiah Application.

This working block also works as a cache for messages, and it is used to retrieve information later on, when needed.

The messages circulating in the network follow a structure composed of the following fields: node-ID, time stamp, type of message, current location of the sender, intended future route, and destination. As shown in Figure 4.4, the fields are:

- ID: Contains a unique ID generated by the IdGenerator Class.
- TimeStamp: Indicates the time when the message is sent on the sender side, and the time when the message is received on the receiver side.
- Type: Contains the type of message based on the sender, whether it is an administrative mode, SOS, or bye-bye message.
- Location: Contains the geographical information of the sender (in latitude and latitude coordinates).
- Future Route: Contains the GPS points that define the sender's path. By default, we have defined 50 GPS points that will draw the path or trajectory of the source sender.
- Destination: Contains the geographical location of the destination point.

Messages coming from administrative or SOS nodes are basically identical. The administrative and SOS message types have the same function, requiring a distinction between them in order to be represented by different icons when displayed on the screen. The other type of message, known as "Bye-Bye" message, is used to signal when a node that had been previously working in the administrative or SOS mode is now leaving that mode. This notifies nearby nodes that they are either going to switch to the civil mode or to leave the network.

The sending procedure sequence is presented in Figure 4.5 in more detail. The sending process will have to go through packet generation, and it will check the user input mode, the future route, and the destination. Initially, the appli-

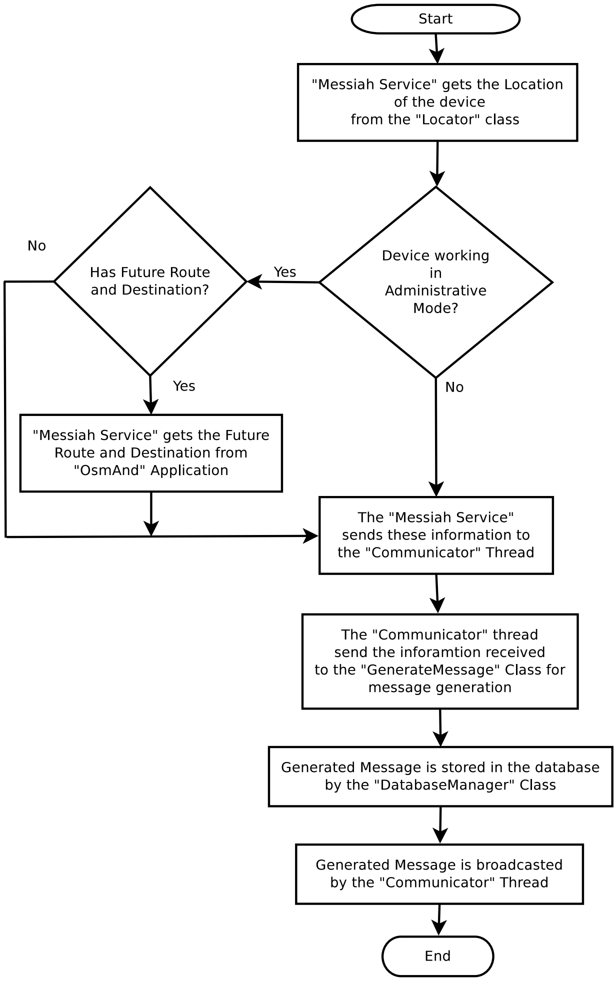

Start
"Messiah Service" gets the Location of the device from the "Locator" class
Device working in Administrative Mode?
Yes
Has Future Route and Destination?
No
Yes
No
"Messiah Service" gets the Future Route and Destination from "OsmAnd" Application
The "Messiah Service" sends these information to the "Communicator" Thread
The "Communicator" thread send the inforamtion received to the "GenerateMessage" Class for message generation
Generated Message is stored in the database by the "DatabaseManager" Class
Generated Message is broadcasted by the "Communicator" Thread
End

Figure 4.5: Flow Chart for Sending Packets.

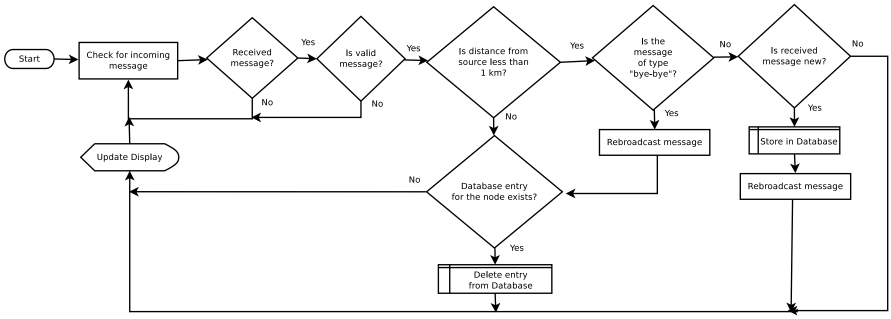


Figure 4.6: Flow Chart for Receiving Packets.

cation will get the current location of the device with the help of the Locator Class. The application will then prompt the user to indicate whether the application will use the administrative mode. If it remains in civil mode, the application will directly generate messages. However, if the device switches to the administrative or SOS modes, the application will then check if a user has a future route and a destination. If it has no future route and destination, the process continues to the Communicator thread. On the other hand, if the user has a future route and destination, the application will extract them, and then continue to send the information to the Communicator thread. The Communicator thread, which has the role of handling the communication functions, will send the information to the GenerateMessage class for message generation. The message generated is stored in the DatabaseManager class. Finally, these messages that have gone through the whole process, are broadcasted onto the network with the aid of the Communicator thread.

All three modes work similarly when receiving messages, which means that the vehicle under the administrative or SOS profiles will receive and broadcast received packets in the same manner as vehicles working in the civil mode. Each node will generate and broadcast messages containing their ID, local timestamp, type of message (indicating the mode they are currently in), current location, future route, and destination.

Figure 4.6 provides a more general overview of the process for all modes. In this process, nodes only receive information and forward it to their neighbors. Initially, the communicator will check for incoming messages. When a message is received, it will first check its validity. The validity of the message is then checked based on the distance towards the source of the information. If the distance to the source is more than 1 kilometer, the application will check the database to determine if the node already exists. If this is the case, the information entry is deleted from the database, and the message is dropped. Otherwise, if the distance to the source is less than 1 kilometer, the application will then check the message type. If it is a Bye-bye message, the message is rebroadcasted, and then dropped. At the same time, the entry associated with the node whose message is received is deleted from the database. Otherwise, if the message is of the administrative or SOS types, it is checked to determine whether it is new. Notice that a packet is defined as new if the ID of that incoming packet does not match any of the identifiers of existing packets in the database. In case the message is new, the local timestamp is added to the message and it is stored in the database. This way, when a node connectivity is lost or network partition event occurs, the message is still stored with the adequate time stamp. The message without the local timestamp will then be rebroadcasted in the network. As a side note, nodes working in the administrative or SOS modes may leave the network at any time, without notifying its neighbors using Bye-bye messages. In order to prevent having a message that is stale in the database, the application will check whether database entries have been there for 10 seconds without any updates. If the mentioned entries are present, they are eliminated from the database.

4.2.3 Deployment

Ad-hoc connectivity is required for our application since it is deployed for V2V communications. Android devices do not include options to enable ad hoc network connectivity by default. Thus, a support device that provides functionality to create an ad-hoc network is required. We have selected the GRCBox [TPCCM15] as the device that delivers V2V functionality, as it can create ad hoc communications between smartphones without having to root them. GRCBox is a low-cost device that has multiple interfaces based on the Raspberry Pi platform. This connectivity device integrates Android smartphones, and it supports V2X communications.

The GRCBox works as a router for our Messiah application. In particular, the GRCBoxes are located on vehicles to create a vehicular ad hoc network between them. The GRCBox also allows creating a connection with an Android device that is used by a user inside the vehicle by offering a local network. With the Messiah application running at the user's side using an Android device, packets are generated and sent locally to the GRCBox. These packets will then be forwarded from the local network to the external ad-hoc network used for VANET communications. In order to realize this type of communication, rules have to be defined. We have extended our Messiah application with an additional class so that the required rules are set automatically. This allows the GRCBox to distinguish between regular Internet traffic and VANET traffic in a transparent manner.

In terms of hardware, a configuration of the interfaces is required so that it discriminates between the internal and the external one. A minimum of two interfaces is required to provide VANET communications. One inner interface would be an Access Point (AP) for the Android devices. On the other hand, the outer interface would provide a VANET of GRCBoxes that are located in nearby vehicles. The network devices used as outer interfaces have the ability to transmit in the 5.8 GHz band. V2V communications can be realized as all the configuration is correctly set up. The integration of our Messiah application with the GRCBox platform is explained in detail in the next section, where we perform the validation of our prototype.

4.3 Empirical Evaluation of V2V Communications at Intersections

Since we know that the performance of a C-ITS application that involves V2V communication is affected by the network performance, it is interesting to study the impact of the environment. The environment, in this case, the street layouts, might affect the quality of the communication. In parituclar, an interesting issue to be analyzed is the intersection problem. On various types of intersections, the obstacles that are present can severely hinder the LOS. For this reason, a study on the impact of intersections on network performance should be performed.

To this purpose, we selected different types of intersections available in the city of Valencia (Spain), and then performed actual field tests using vehicles to determine the communication restrictions imposed by the different types of intersections. In addition, based on the empirical results obtained, we modeled the packet delivery probabilities at different distances to the center of each particular intersection to determine the expected success ratio when delivering event-based messages, and to allow integrating the results here presented in different simulation platforms. In this section, we describe the methodology of the experiments performed. The goal is to measure the packet delivery ratio depending on the distance to the center of the intersection for different types of intersections and antenna locations. The obtained results will then be used to create models. Later, in Section 4.5.3, we will also discuss the applicability of the obtained models, and compare them with other path loss models.

4.3.1 General Overview

Our experimental work requires the utilization of appropriate hardware/software to measure the packet delivery ratio, and also of a proper data analysis methodology to process the gathered data after experiments are completed.

Two devices are used in the experiments. The first one is the *GRCBox* [TPCCM15], which is our on board unit providing fully functional V2V communications. This GRCBox is equipped with an antenna that supports VANET communications in the 5.8 GHz band. The transmitting antennas have a 5 dBi gain, and a 200 mW transmission power. Packet transmission tests using this device will consider two alternative positions for the antenna. In one case we will put the antenna inside the vehicle, specifically on the dashboard. The other alternative considered will be installing the antenna on the rooftop of the vehicle. These variations will allow us to achieve new findings regarding the impact of antenna locations, expecting that having the antenna on the rooftop will provide a better transmission quality than having it on the dashboard.

Another device used in this experiment is an Android smartphone. Taking into account the trends of using smartphones for vehicular communications and ITS-related research works, deploying an Android smartphone can be an alternative solution to exploiting the cost of high-end ITS equipment for research purposes [VMD11; TCCMFM13; FBCE14]. The Android phone runs a custom application (the *Android tool*). This Android tool will allow performing controlled experiments in real environments by generating messages resembling those associated to the European (ETSI) standard, particularly the DENM [ETS10a].

For the experiment itself, which includes real moving vehicles, at least two vehicles are needed: one acting as data sender, and the other one acting as a data receiver. Regarding the location of the vehicles, the one acting as a data receiver will be static, and remains static a few meters away from the center of the intersection, representing a vehicle stopped at a semaphore or at a *stop* sign. The other vehicle, acting as a data sender, will be moving along a different street, crossing a common intersection. The experiment parameters in general are summarized in Table 4.1.

Once the sender is moving, and the receiver starts to receive packets, the developed test tool will record the location of both sender and receiver vehicles, which will then be saved in a log file stored on the Android device. After we record the location of the sender in the log file, our data analysis consists of calculating the packet delivery ratio with regard to the distance between

Table 4.1: Experiment Parameters for Car-to-Car Communications.

Parameters	***Values***
Frequency Band	5 GHz
Transmitter Antenna Gain	5 dBi
Packet Size	30 Bytes
Packet Rate	30 Hz
Transmission Rate	6 Mbps
Obstructions Level	Minimum, Moderate, Maximum

sender and receiver. Specifically, by obtaining the packet delivery ratio for each intersection at different distances, we can draw conclusions on how intersection characteristics will impact message dissemination in vehicular scenarios. In addition, the different intersection types can be modeled, and further analysis can be done based on the obtained models.

4.3.2 Testing Tool and Data Analysis

We have built a small specific tool for measuring the data delivery ratio in the target scenarios. The tool is actually an Android-based application that is GRCBox-aware. The Android application contains libraries and plugins able to connect to the GRCBox module so that, at the user side, one does not need to configure the connection to the GRCBox's outer interface. Hence, once the Android smartphone connects to the access point (AP) of the GRCBox (in this case GRCBox's inner interface), it would instantly be connected to the whole GRCBox environment and the VANET without further settings. Also, based on its functionality, we have different instances running at the sender and at the receiver ends.

At first, the application will check if it is connected to the GRCBox device on the sender's side. Also, the user can input the log file name, the packet transmission rate, and the size of the packet. In this case, we have chosen the sending parameters similar to those typical of CAM/DENM messages [ETS10a; ETS10b], having a size of about 300 bytes, and selecting a packet rate of 30 packets per second to allow quickly gathering large amounts of data. Also notice that, since packets are being broadcasted, the transmission rate is limited to 6 Mbps. At the receiver's side, GRCBox connectivity is also tested, and the user can introduce the log file name and select when to start gathering data. The transmission of packets is started when, at the sender's end, the user presses the start button, triggering the transmission of a packet train at the defined rate and packet size, and using the broadcast mode. Similarly, the transmission stops as the user stops the application. This will cause the application to automatically store the whole log file in a local file at both sender and receiver ends. In a nutshell, our Android-based application is used to define parameters such as data rate, packet size, and log file name. It starts the sending process upon receiving the corresponding user command; then, when the user stops the application, the log file is automatically stored.

The data collected is saved in a log file located in the Android device's storage. This log file contains all the data required to analyze the packet delivery ratio at different distances. For this purpose, we are interested in comparing the geographic information of both endpoints, and so the log file contains the coordinates of the sender and the receiver in terms of latitude and longitude (flat terrain is assumed). Based on this geopositioning information, we then calculated the distance in meters between the vehicle localization and the center of the intersection, where the sending vehicle passes through as part of its designated trajectory.

The distance is calculated with the help of Geographiclib [Kar11], a tool offering a straightforward calculation of distance based on latitudes and longitudes. Using that tool, we then analyzed the log file that is stored on both sender and receiver sides.

As for the delivery ratio, we need to compare the log files containing packets sent at the sender side and the packets received at the receiver side. This can also be calculated by referring to the geoposition information provided by both sides.

4.3.3 Selection of Target Intersections

For our experiments, we selected three different types of intersections, each with different characteristics. In particular, the types of intersections were chosen to obtain different degrees of obstruction. The geographical location of each intersection is shown in figures 4.7, 4.9, and 4.11. The yellow line with an arrow indicates the trajectory and the direction of the sending vehicle, while the yellow point indicates the location of the receiving vehicle (static). We now proceed to provide more details about each of them.

Intersection 1 (Minimum Obstruction)

The first intersection selected is an open space. It was taken in a sparsely populated area in the outskirts of the city of Valencia (Latitude 39.483920, Longitude -0.333793). In this intersection, no relevant signal blockages are present. In fact, figure 4.7 shows that the only blocking structures are two buildings, one south of the roundabout, and the other one north. Thus, the line of sight is not blocked along the trajectories of the vehicle, meaning that the degree of obstruction is minimal. As shown in figure 4.8a, the receiving vehicle is located near to the intersection, being surrounded by grass fields and facing no significant signal blockage.

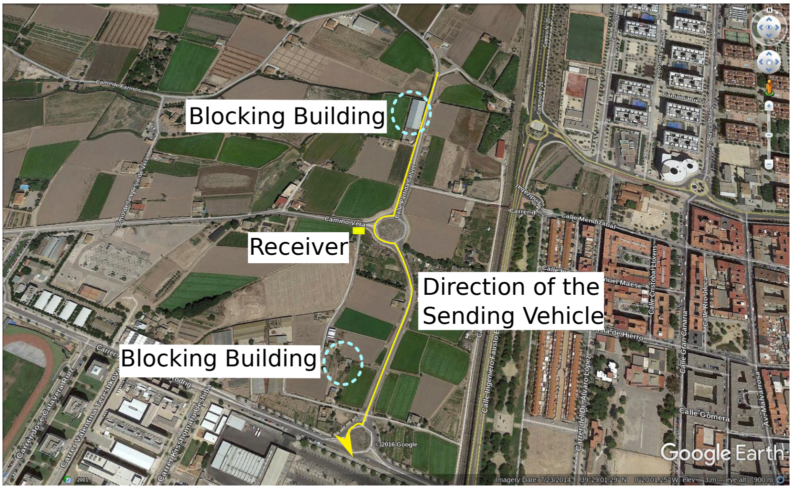
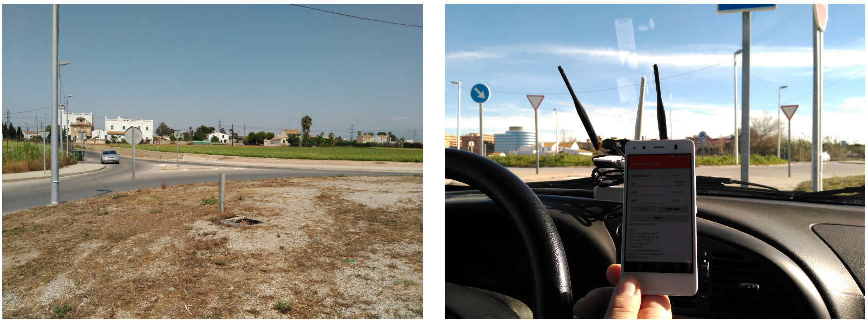


Figure 4.7: Location and the trajectory for Scenario 1 (open)

Intersection 2 (Maximum Obstruction)

The second intersection selected is in a residential area, which is located in a crowded and dense neighborhood of the city (latitude 39.473695, and longitude -0.332307). In this intersection, buildings represent obstructions to the line of sight, and so the degree of obstruction is nearly maximum. Based on the aerial view shown in figure 4.9, we can see that the environment indeed consists of a very dense neighborhood, without additional urban elements separating them except streets themselves, meaning that an urban canyon is formed. In figure 4.10a we can see that the chosen intersection is surrounded by at least two-floor buildings. So, from the perspective of the receiving vehicle, the line of



(a) Outside View.

(b) Inside View.

Figure 4.8: View of the vehicle parked at Intersection 1.

sight is quite limited as the furthest view that one can glimpse from inside the mentioned vehicle is of, at most, 20 meters. By defining this type of intersection, we expect that the transmission of packets in this location would face a worst-case scenario.

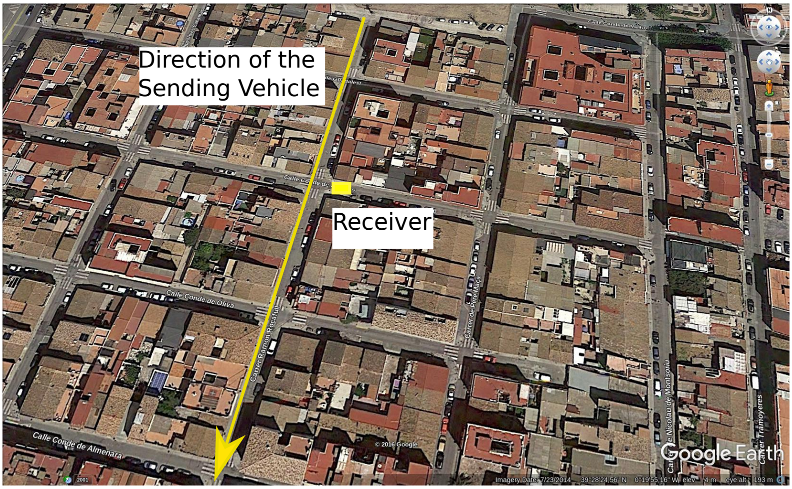
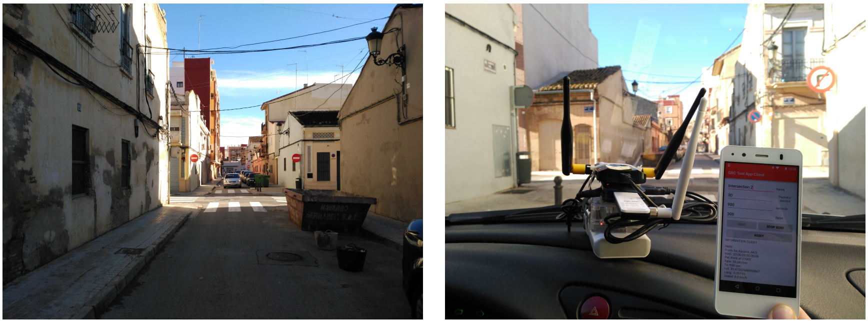


Figure 4.9: Location and trajectory for Scenario 2 (buildings).



(a) Outside View. **(b)** Inside View.

Figure 4.10: View of the vehicle parked at Intersection 2.

Intersection 3 (Moderate Obstruction)

The third selected intersection is a mix of the previous two scenarios, as the line of sight is blocked by either buildings or trees. The intersection lies in a residential area near a university campus (latitude 39.473848, and longitude -0.341330), and the degree of obstruction can be considered moderate. Figure 4.11 shows that this kind of intersection has some buildings in the surroundings, but has also open spaces and vegetation along the trajectory. A mix between these characteristics causes communication in this kind of environment to produce interesting results, as the line of sight characteristics are variable, being that sometimes it is blocked by a building or tree, while at other times no obstacle blocks sight. Figure 4.12a presents the real view of the intersection. As shown, the receiving vehicle is surrounded by trees and, within meters, there is an open field. However, the street itself is located in a residential area full of tall buildings.

4.4 Experimental Results

To assess the impact of intersections on V2V communications, we have performed two sets of experiments. First, we analyzed the packet delivery ratio for vehicles that are located at intersections. Secondly, we analyzed the probability of successful delivery of notifications associated to critical events from one vehicle to another at intersections.

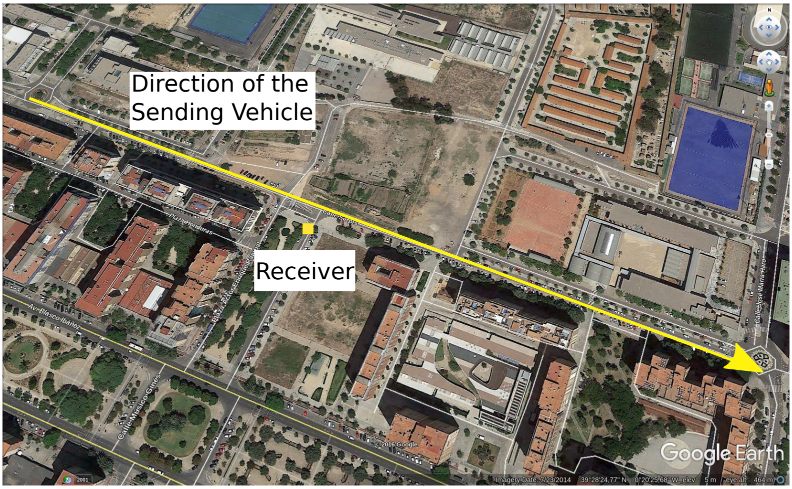


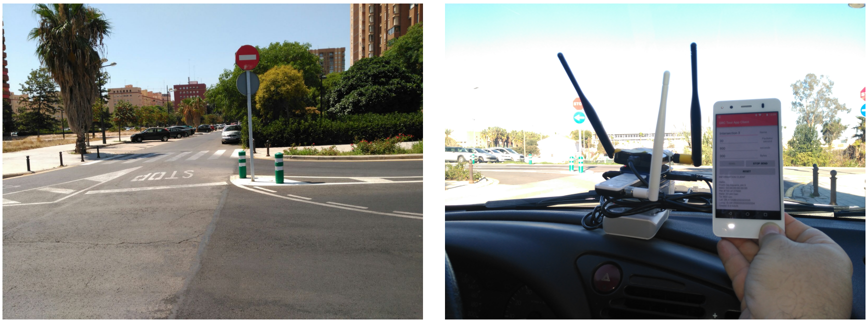
Figure 4.11: Location and trajectory for Scenario 3 (trees).

4.4.1 Packet Delivery Ratio at Intersections

We have performed real experiments with vehicles to gather the location (coordinates) of these vehicles when packets are successfully delivered at each intersection. We also gathered data for scenarios where the antennas are located either inside the vehicle, on its dashboard, or on the rooftop of the vehicle. The data were gathered from a set of five vehicle runs at each intersection, and the measured coordinates are then validated using real maps.

Results for Intersection 1

Figure 4.13a shows the percentage of messages received as a heat map for the first intersection, which has the lowest degree of obstruction, based on the locations of the sender associated with successful packet delivery events. In this experiment, the two antennas involved were located in the vehicles' dashboard. As expected, the packets can be delivered successfully having as source nearly any position along the vehicle trajectory. The only gap detected occurs due to the blocking caused by the two available buildings. However, the delivery ratio is much higher when it is at the center of the intersection, and gets lower as the vehicle moves away up to the maximum tested distance (about 300 meters).



(a) Outside View.

(b) Inside View.

Figure 4.12: View of the vehicle parked at Intersection 3.

Figure 4.13b presents the results when the antenna is located on the rooftop instead. We can now observe that the point density is higher, and that there are no gaps. In fact, the impact of physical obstructions is still relevant, but it is merely limited to a slight reduction of the packet delivery ratio.

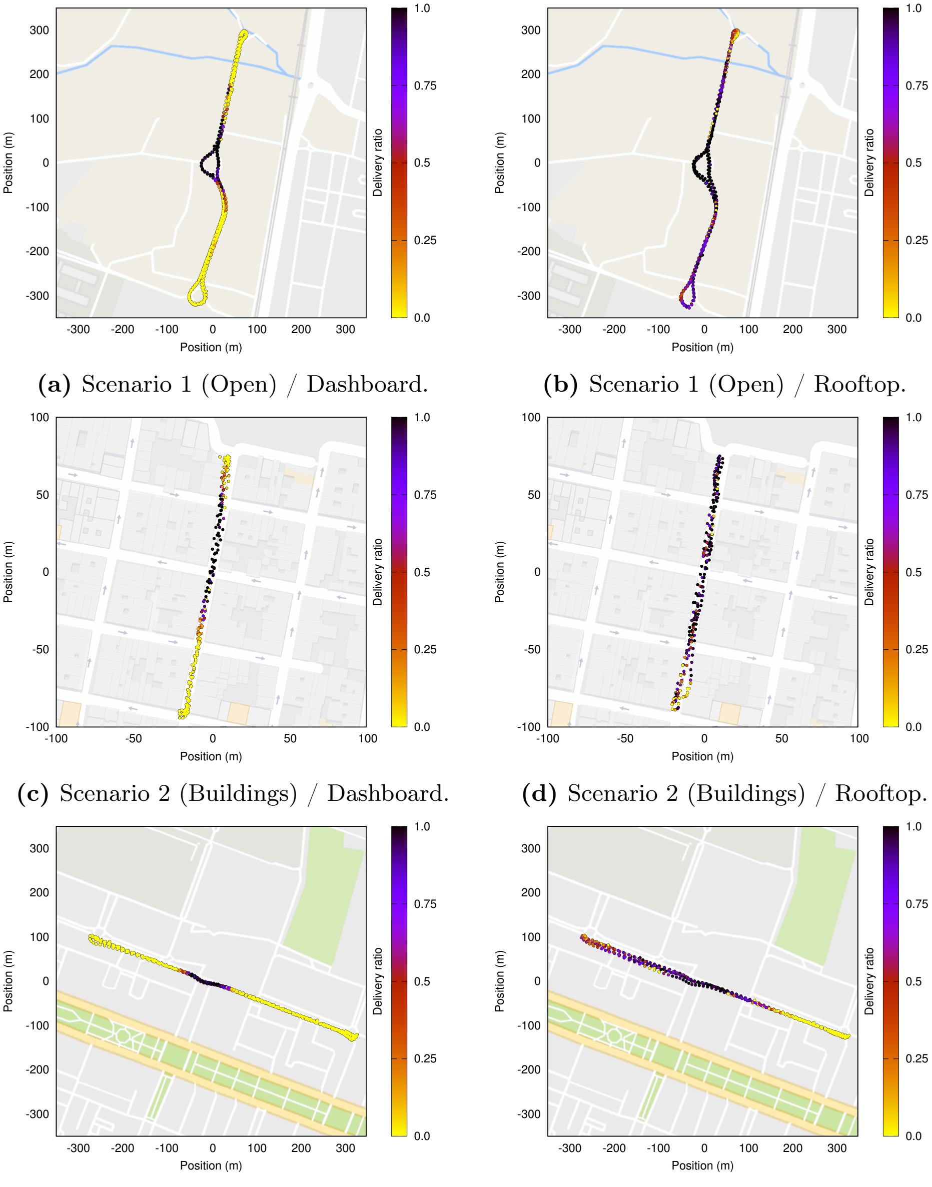
Results for Intersection 2

This second intersection is quite narrow, having tall buildings on all sides that create an urban canyon. Figure 4.13c shows the locations associated with successful reception of the packets when the antenna is located in the dashboard. We can see that successful packet receptions are basically restricted to a range of just one block (about 50 meters), again experiencing a descending delivery ratio when moving away from the center of the intersection.

If, instead, the antenna is located on the rooftop, figure 4.13d shows results similar to those in the previous figure, although now the overall delivery ratio increases, reaching up to two blocks away from the intersection (about 100 meters), while mostly maintaining an acceptable delivery ratio.

Results for Intersection 3

Since this third type of intersection presents a moderate degree of obstruction, the results in terms of radio range should be in-between the results for the first and second types of intersections. Figure 4.13e shows that the locations associated with a successful transmission indeed reach greater distances compared to the second type of intersection, having values resembling more



(a) Scenario 1 (Open) / Dashboard.

(b) Scenario 1 (Open) / Rooftop.

(c) Scenario 2 (Buildings) / Dashboard.

(d) Scenario 2 (Buildings) / Rooftop.

(e) Scenario 3 (Trees) / Dashboard.

(f) Scenario 3 (Trees) / Rooftop.

Figure 4.13: Heat Maps for the different Scenarios. Each plot shows the packet delivery ratio depending on the sender position, scenario (open, building and trees) and antenna location (dashboard, rooftop).

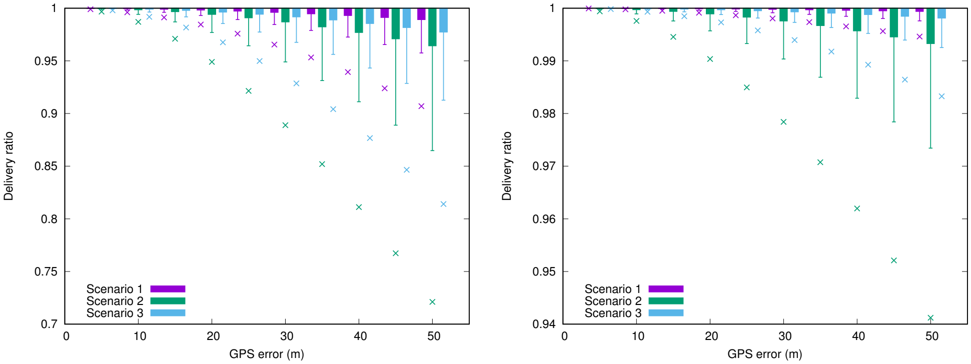
the first intersection. This occurs since the obstacles closer to the vehicle in this situation are mainly vegetation-related, which does not negatively impact communications the same way as buildings; in addition, the streets are wider than in the second scenario.

When locating the antenna on the rooftop, results are found to be much better. Now, despite we cannot associate the entire vehicle path with good transmission conditions, the results of figure 4.13f clearly show that high delivery ratios can be maintained for up to 100/150 meters. Again, the delivery ratio near the intersection is significantly higher, experiencing a constant drop as we move away.

4.4.2 Event Notification Effectiveness on Intersections

To further validate our research work, we have also analyzed the probability of successful delivery of notifications associated to critical events. As explained earlier, such event notification dissemination typically relies on multi-hop broadcasting to make sure that the information arrives to all vehicles in a certain target area. However, since such dissemination procedure is prone to cause broadcast storm problems, and since urban obstacles will typically hinder dissemination towards vehicles in nearby streets, different proposals consider optimal to perform timely broadcasts when vehicles are located at intersections to maximize reachability. Such timely broadcasts for moving vehicles, though, rely on mapping GPS coordinates to map details, and the overall effectiveness will highly depend on the GPS error introduced at the time of broadcasting.

Taking the aforementioned issues into consideration, in this section, we will use the models derived in section 4.5 (see next section) for the different intersection types and antenna locations, to study the probability of successfully delivering an event-related message at an intersection when considering different GPS error values. We are assuming that the vehicle intends to send a packet when located at the center of the intersection to maximize the packet delivery ratio. However, if we take the GPS error into account, we could expect that the error it introduces could impact the packet delivery ratio, especially in urban canyon scenarios. To this purpose, we define different maximum values for the GPS error (which typically ranges between 5 and 50 meters), and create normal distributions where 99.7% of the values are inside this maximum distance (3σ rule). Then, considering this probability distribution for the vehicle location when transmitting a packet, we combined it with the models derived in the previous section to gain awareness about the expected success ratio for the event message delivery.



(a) Dashboard.

(b) Rooftop.

Figure 4.14: Delivery ratio at the three intersections for different GPS error ranges.

In figure 4.14, we evaluate the impact that the GPS error ranges will have on the delivery ratio for each scenario, either with the antenna located on the dashboard or on the rooftop. The three intersections having different levels of obstruction are compared. In these plots, the delivery ratio from the fitted model is shown for three significant points in the error distribution: the bar corresponds to the interval from 0 to σ (68% of values), the line corresponds to $[0, 2\sigma]$ (95%), and the cross is $[0, 3\sigma]$ (99.7%).

If we focus on the case where the antenna is located in the dashboard (see figure 4.14a), a significant difference is detected when we have a GPS error of 50 meters. In the case of intersection 1, a GPS error of up to 50 meters still shows acceptable packet delivery levels; on the contrary, for intersection 2 (urban canyon), the delivery ratio is much worst than for the other two cases.

Figure 4.14b shows that, when installing the antenna on the rooftop, the impact of GPS error is now reduced as the delivery ratio in these cases, when compared to the previous ones, is much better. This means that, in general, effective propagation of messages at intersections is possible, even in urban canyons, and despite GPS errors, as long as rooftop antennas are used so that their extended radio range compensates for the poor radio visibility and positioning error.

In a nutshell, again we find that the different antenna positions and the characteristics of intersections clearly affect the probability of successful packet delivery, even in the presence of GPS errors. That being said, the most reliable sending process takes place when we put the antenna on the rooftop of the vehicle, and the transmission occurs at an open space intersection (with minimum obstructions). The worst case occurs at the urban canyon intersection (maximum obstruction), when the antenna is located within the vehicle (in the dashboard), thereby matching our initial hypothesis.

4.5 Intersection Modeling

Based on the results obtained in our experiments, we now proceed to detail how the different intersections were modeled. Our purpose is to obtain a generic model that allows integrating the different behaviors observed in simulation tools, as well as analytically studying the effectiveness of event-related message delivery at intersections.

4.5.1 Modeling procedure

In order to model the different intersections, our procedure was the following: first, we obtained the number of packet transmissions and receptions for each position registered; second, we determined the packet delivery ratio value associated with each distance range; finally, we performed a curve fitting process to derive optimal parameters.

In detail, the results of the first step of the experiment consisted of a list of coordinates (i.e., latitudes and longitudes) stored at the sender, being another similar list stored at the receiver's side. The sender coordinates are its actual location when a packet is sent. Logically, the list at the sender side has more entries than the one at the receiver side, as several packets get lost. So, we must compare the difference between these two lists of coordinates. A packet is successfully received if an entry (coordinate) at the sender's list is also present in the receiver's list. The coordinate is then translated into distance by considering the coordinates relative to the center of the intersection using the *haversine* formula implemented in the GeographicLib library.

The outcome of the first step is then grouped into small intervals, being the interval width equal to 5 meters. The delivery ratio derived for the different intervals is then plotted using a bar chart, thereby resembling a histogram (although it is not so in a strict sense). By following the same procedure for the different antenna locations (rooftop vs. dashboard), we can then obtain a comparative chart for the target intersection that allows checking packet delivery ratio in both cases.

Once these distributions are calculated for the three types of intersections, and for both antenna locations, we proceeded to find the best fit for our data. The curve fitting was done using the nonlinear least-squares Marquardt-Levenberg algorithm (implemented in the GNUPLOT software) to derive a general model for the curve, determining which function is more adequate for our purposes, as well as the best parameter values for each distribution.

4.5.2 Fitting results

For each curve, we tried to find a common model that would be suitable for both antenna positions (dashboard or rooftop), and for the different types of intersections (open, building, trees). Our intention was that only one parameter would vary from one scenario to another, allowing to seamlessly model different types of intersections having variable degrees of radio visibility. After evaluating several fitting functions (polynomial, power) for the different types of intersection and antenna locations, the best fitting was obtained using a Gaussian function:

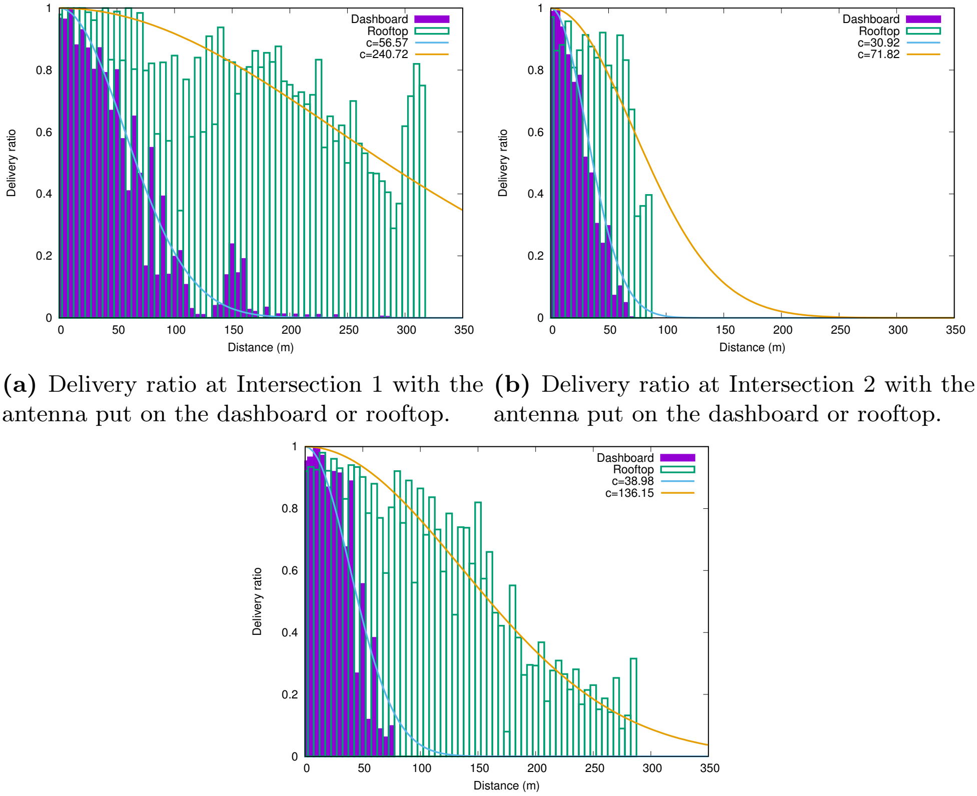
$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \tag{4.1}$$

Notice that variables x, a, b, and c are generic variables, meaning that equation 4.1 will be adapted to our specific requirements. In particular, we set x as the distance from the intersection, and fix the probability of reception to one exactly at the intersection (distance zero), so that a is one and b is zero. Thus, the only parameter to fit is c, that is, the *standard deviation*. Finally, we have the following expression:

$$f(x) = e^{-\frac{x^2}{2c^2}} \tag{4.2}$$

This exponential function computes the delivery ratio for a particular distance x. As the distance grows, this probability asymptotically becomes 0. The value of the constant c (or standard deviation σ) will depend on the scenario and the antenna position, and reflects the variation or dispersion of the data values.

The resulting bar charts and fitting results are shown in figures 4.15a, 4.15b, and 4.15c. If we take a look at the experimental results for intersection 1 (see Figure 4.15a), we can quickly notice that there is a significant difference between the delivery ratio for the *dashboard* and *rooftop* antenna locations.



(a) Delivery ratio at Intersection 1 with the antenna put on the dashboard or rooftop.

(b) Delivery ratio at Intersection 2 with the antenna put on the dashboard or rooftop.

(c) Delivery ratio at Intersection 3 with the antenna put on the dashboard or rooftop.

Figure 4.15: Curve Fittings for Delivery Ratio vs. Distance.

Table 4.2: c parameter and χ^2 error values for each scenario and antenna position.

	Antenna on dashboard		**Antenna on rooftop**	
	c	χ^2	c	χ^2
Intersection 1	56.57	15.35	240.72	30.15
Intersection 2	30.92	12.84	71.82	21.67
Intersection 3	38.98	6.18	136.15	20.14

The curve fit for the *dashboard* antenna location shows that, for low distances, the delivery ratio is still comparable to the one from the *rooftop* fit. After a distance of about 20 meters from the intersection, the bars show a quick attenuation. Also, we observe that it loses contact after about 200 meters. We can observe how, when the distance is of about 120 meters, the delivery ratio suddenly drops, being followed by a moderate increase. This is an effect of the buildings present in the environment, as we can see in an aerial view of the street shown in figure 4.7. Concerning the delivery ratio for the rooftop scenario, high delivery values are sustained for a distance up to 70 meters, after which they experience a 20% drop. In this same scenario, for the *dashboard* case, the drop ratio is significantly higher (50%). Thus, we can conclude that antenna location has a very significant impact on packet delivery success.

Concerning the second intersection, the bar chart shown in figure 4.15b clearly shows a significant difference compared to the previous one. In fact, the distance range is now quite reduced, being that the packet delivery ratio drops to only 35% in about 40 meters (for the "dashboard" antenna position). Beyond 50 meters, the delivery ratio becomes near zero. If we now focus on the curve fit for the rooftop scenario, we find that differences towards the dashboard case are quite clear, similarly to what occurred for Intersection 1. However, respect to that first intersection, we now see that, at a distance of 50 meters, the packet delivery ratio for the rooftop is nearly 0.8, while for the dashboard it is only 0.3, both values being much lower than those measured in intersection 1. This is why we can categorize this second intersection as an urban canyon, which is a worst-case scenario associated to the maximum degree of obstruction.

Regarding the results for Intersection 3, figure 4.15c shows that, when the antenna is located in the dashboard, there is a loss of radio connectivity after about 70 meters. Instead, when the antenna is on the rooftop, contact is maintained beyond 250 meters, a much greater distance. Compared to the two previous intersections, the fittings in this scenario are indeed a situation in between intersections 1 and 2.

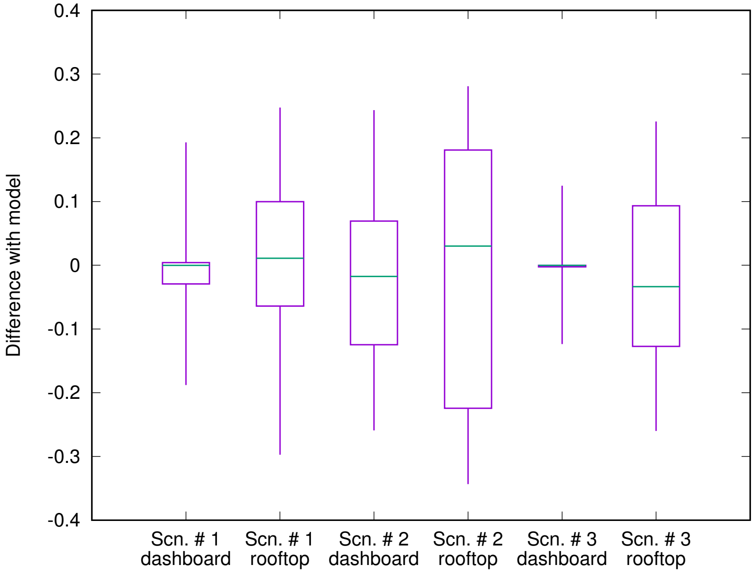


Figure 4.16: Differences among the fitted models and data.

Overall, we consider that the obtained results are quite reasonable by considering that our experiments were made in scenarios where no interference is hindering our communications band, meaning that the channel only experiences the effect of additive white Gaussian noise. In such a situation, the fitting corresponds to a standard AWGN channel model.

We now focus in detail on the outcome of the fitting results and the corresponding fitting errors. Notice that equation 4.2 introduced parameter c, the standard deviation of the Gaussian function, which allows adapting the fitting curve to each type of intersection and antenna location. In table 4.2 we detail the values of this parameter for each case. It is interesting to observe that this parameter decreases for lower radio ranges at intersections, being directly related to the packet delivery ratio. In fact, the higher the parameter, the higher will be the packet delivery ratio for a certain distance towards the intersection.

In detail, we can see that the c values for the first intersection are the highest ones. On the other hand, for intersection 2, the lowest values are obtained, being intersection 3 characterized by intermediate c values. Also, regarding antenna locations, we find that the c value for the rooftop results is always more than twice those obtained when having the antenna installed in the dashboard.

The largest relative difference is detected for intersection one, where the c value for the rooftop case is more than four times greater than the one for the dashboard case. This occurs because, for this kind of intersection, packet losses are mostly related to signal power drops due to distance, and the rooftop antenna location thereby emerges as the optimal option to mitigate such power losses.

Table 4.2 also shows the fitting error expressed as χ^2, the sum of the squares of the differences between the model function and the actual delivery ratios obtained from the experiments. Additionally, figure 4.16 shows a box and whisker plot of the difference distribution for each scenario and antenna position. The box shows the 1st, 2nd, and 3rd quartiles, and the whisker is the mean value plus/minus the standard deviation. The model fitting is clearly more accurate for the dashboard scenarios than when mounting the antenna on the rooftop. This occurs because, in the latter case, the range is not large enough to reach near-zero values.

4.5.3 Model Applicability to Simulation Environments

In general, a detailed channel characterization between two endpoints requires studying the signal to noise plus interference values at the receiver, which includes modeling in detail the signal propagation conditions in the target environment. In the specific case of vehicular networking environments, this includes the modeling of signal reflections and Doppler spread in the presence of various obstacles, including buildings, trees, and vehicles. However, such a detailed signal propagation analysis is extremely complex, and so it becomes computationally prohibitive to undertake such a detailed analysis when studying traffic communications in a large area, especially for vehicular networking studies, where this area can grow up to the size of an entire city, or even more. To address such a problem, empirical path loss models for urban environments have emerged (e.g. Nakagami [Nak60] and Durgin [DRX98]). However, these models provide a generalization of the propagation behavior, meaning that they fail to provide a detailed characterization of very specific transmission conditions, such as the intersection propagation conditions addressed in this paper. Yet, the problem of how to adapt our model to simulation environments remains, as it requires knowing in advance the actual characteristics of each specific intersection in order to adequately model it.

To achieve the intersection modeling requirements enabling the adoption of our models, we propose automating the intersection classification process by analyzing the street width and the presence of buildings in a preprocessing

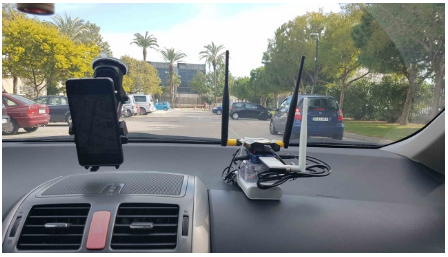


Figure 4.17: Photo taken from within the car.

step before the actual simulation. This way we avoid having to manually tag each intersection manually, and benefit from the models hereby derived with little additional complexity.

It is worth highlighting that widely used map providers, such as OpenStreetMap [HW08], already include such street and building information for many relevant cities, which simplifies and makes feasible the adoption of our solution.

4.6 Applicability on the ITS Application in Relation to Intersection Modeling

In this section, we present the applicability of our C-ITS application, Messiah, in vehicular environments. A functionality test was performed to ensure that application can work with our designed architecture. Another assessment includes the impact of the vehicular communications performance at intersections previously presented on the performance of the Messiah application. The features of Messiah were tested, and we also analyze the applicability of our application in vehicular network environments, based on the previous network performance test. Reliability, timeliness, and packet load are the target metrics, as they are considered critical for this application.

4.6.1 Functionality Test

The functionality of the application was assessed as part of the initial test. The test was completed on the campus of the Universitat Politècnica de València (UPV) by deploying real cars, each with a GRCBox unit inside, along with an Android device.

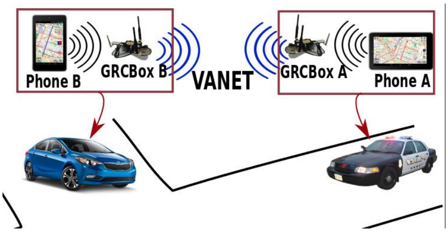


Figure 4.18: Structure of the Ad-Hoc Network.

The application is integrated with the GRCBox module, and it relies solely on the ad-hoc network that is formed by GRCBoxes located inside vehicles, as shown in Fig. 4.17.

These GRCBoxes are used by participating vehicles to forward received data to other GRCBox units within communication range. The forwarding of the data, making the GRCBox acting as a relay node, will occur until the data reaches the destination node.

The ad-hoc networking mechanism is depicted in Fig. 4.18. The Android device connected to the GRCBox will establish a path for the data so that it can be forwarded to the outside ad-hoc network formed by the connected GRCBox devices. Each car has a GRCBox and an Android device inside, hence establishing a VANET. The communication in this network relies on IEEE 802.11-based broadcast traffic.

4.6.2 Network Performance Test

Considering the experiments performed on intersections, we can proceed to determine the expected performance of the Messiah application under realistic settings. Based on the three types of intersections studied, we can now proceed to determine how many packets arrive to the receiver. With the support of additional data, such as timestamps, packet rate, and packet size information, we can analyze the performance of the application based on its packet loss, inter-packet arrival time, and the traffic overhead.

Packet Loss

The first performance assessment targets the packet delivery in these three intersection environments. To determine the maximum distance covered by communications using this application, packet losses are calculated by determining how many packets are lost on the channel when communicating with the sender for a specific distance interval, being that regular spatial intervals are defined.

As a result, in Fig. 4.19 we can see that few packets are lost in all three scenarios for distances up to 10 meters. In the dense residential area, loss levels increase significantly when reaching 40 meters, situation where only 50 percent of the packets are received. In a less-populated urban area, a similar situation occurs when the communication range is of 60 meters, being that the 50th percentile corresponds to a distance of 80 meters.

The application can handle communications for distances up to 240 meters in rural areas, but in a less-populated urban area, nearly 100 percent packet loss is detected when the communication distance grows beyond 100 meters. In fact, under NLOS conditions, and with the highest level of obstruction (urban canyon), the communications are successful only for distances up to 80 meters. This means that, to achieve an effective message dissemination within a wider region, multi-hop forwarding is a desirable feature to include in our application.

Time-Probability Analysis on Packets

The next assessment consists of a time-probability analysis by focusing on consecutive packet arrivals. We have tested a 10 Hz packet sending rate in this experiment. We have chosen this rate so that the inter-packet arrival time is about 100 ms under ideal conditions. The probabilities of having inter-packet arrival times below the indicated time are derived using a cumulative distribution function. We have calculated the probabilities for the three types of scenarios with different street characteristics. Figure 4.20 shows that the inter-packet arrival time is never lower than 100 ms. This is due to the fact that the transmission rate is 10 Hz (10 packets per second). Up to 130 ms, the probability grows in all three scenarios. However, in both rural and regular urban areas, 97% of the inter-packet arrival times are below 200 ms. On the other hand, in an urban canyon (dense residential area) scenario, values of 500 ms are reached for nearly the same probability. The difference we observe between the less-populated urban area and the rural area is that, at 150 ms, the cumulative probability for the rural area reaches 0.8, whereas in the less-

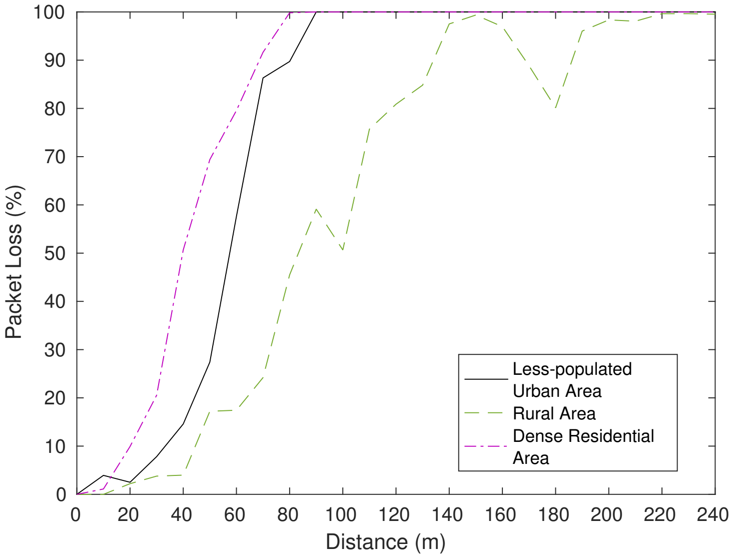


Figure 4.19: Distance vs Packet Loss.

populated urban area it is only 0.75. The cumulative probability reaches its maximum for all scenarios when the inter-packet arrival time is of 900 ms. These inter-packet arrival times results are related to the packet losses taking place in our tests. So, greater waiting times between consecutive packet arrivals mean that packet loss bursts are greater. The ideal condition is that the inter-packet arrival time has the same value as the inter-packet sending time. This condition occurs if there are no packet losses, meaning that no additional delay occurred. The results obtained from our assessments indicate that, when using the application, the time required for notifying nearby vehicles about emergency situations is under 1 second in the worst case, thus being adequate for safety applications.

Traffic Overhead

In our last set of tests, we assessed the traffic overhead based on the route size advertised by the sender. Traffic overhead is calculated based on the GPS points advertised with three different refresh rates. Since we use UDP packets for sending the information, the packets advertised cannot grow beyond 1.5 KB. The GPS points advertised in this context contain the future route information

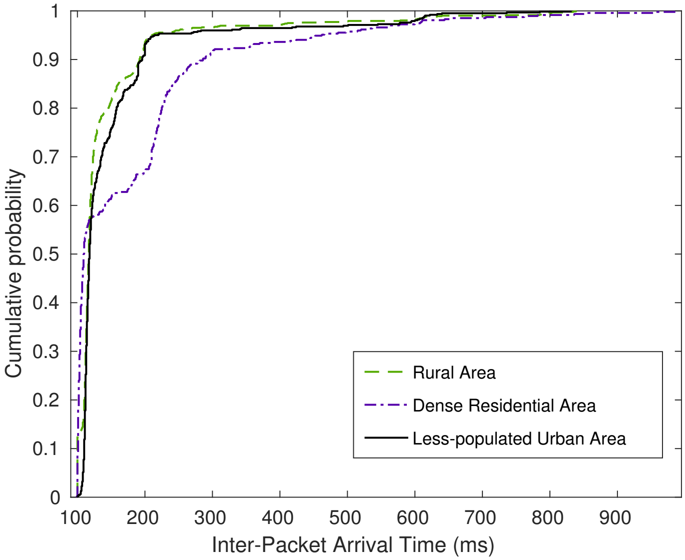


Figure 4.20: Cumulative Distribution Function for the Inter-Packet Arrival Time.

provided by the sender. We have varied the number of GPS points embedded into the packet to detail the future route to a smaller or greater extent. As we can see in Figure 4.21, both the refresh rate and the route size affect the traffic overhead. The more GPS points we have, the higher the value of the traffic overhead; in particular, if packets containing 70 GPS points are advertised every second, this will generate 5 MB of traffic overhead per hour. If the inter-packet period is reduced, the traffic overhead will increase accordingly. These values show the need for adequating the route size that should be included in the packet so that it does not saturate the wireless medium.

4.7 Summary

In this chapter, we introduced Messiah, a novel application providing navigation information and emergency notifications with the goal of creating a safe traffic environment. Messiah's features include an on-screen display that merges maps and the presence of emergency vehicles (ambulances, police cars, fire brigades), along with their routes. This allows civil vehicles to be warned about their presence so that the driver of those vehicles can take alternative

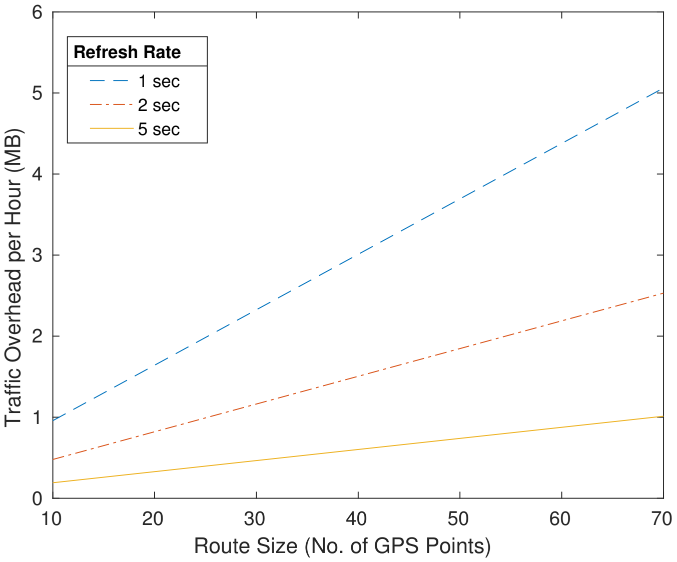


Figure 4.21: Traffic Overhead based on Route Size

navigation decisions to reduce the congestion along the routes followed by emergency vehicles. Also, civil vehicles can benefit from this application by being able to notify nearby vehicles in case help is required, or they may even contact nearby emergency vehicles to request immediate assistance by using the SOS mode. The application is also suitable for VANET environments if combined with GRCBox. In fact, the latter is used to enable V2V communications on vehicles, allowing them to create an ad-hoc network. This way, smartphones running our application can seamlessly distribute messages in the ad-hoc network. Our application can also be immediately deployed and join a VANET without requiring user intervention by automatically detecting and configuring the GRCBox.

To investigate the performance of a C-ITS application, in this case our Messiah application, we must assess the expected network quality in different vehicular environments. In particular, we want to determine how the environment can affect V2V communications quality. Hence, we have studied the packet delivery effectiveness achieved on different types of intersections (no obstacles, urban canyon, partial obstruction), and when locating the antennas on either the dashboard or the rooftop. Extensive experimental results using broadcast traffic have shown that the impact of the intersection type is significant, as differences of up to 150 meters in transmission range were detected. Also, we find that having a rooftop antenna is also a critical factor, allowing to extend the transmission range between 100 and 250 meters, which may represent more than a 100% increase in some cases.

Additionally, we have modeled all the obtained results by finding the best-fitting function, and then applying regression. We find that a Gaussian function offers adequate fits for all cases by just varying one parameter. This way, our model allows to seamlessly represent different types of intersections, and to bring these results to simulation environments in a straightforward manner.

Based on our model, we then made an analytic study to determine the probability of a successful event dissemination process at intersections, for the different types of intersection and antenna locations tested, when varying the maximum GPS error. We find that, in general, dissemination is highly effective, even in urban canyons, and for high GPS error conditions, as long as rooftop antennas are used, being the more restrictive dashboard solutions not recommended. This way, using the previous models, and assisted by real-time geolocation and maps, we can first determine the type of scenario to use and then, knowing the GPS error, determine the expected delivery ratio. In this particular study, we have showed how environmental conditions impact vehicular communications.

Based on our analysis of V2V communications performance at intersections, we then found the expected performance of our Messiah application in terms of network performance under different conditions, showing that it can properly meet the requirements of typical emergency situations. Regarding packet delivery effectiveness, we found that, in the worst-case scenario, where the level of obstruction is high and causes NLOS conditions, packets can still be successfully delivered for distances of up to 80 meters. Additionally, in the same scenario, if we set the message sending rate to 10 Hz, the inter-packet arrival time remained below 900 ms in all cases. This means that, despite channel losses, warnings can still be delivered in less than 1 second, which is an acceptable value for our target application.

In the next chapter, we will perform a feasibility study of including flying nodes (UAVs) to enhance the quality of vehicular communications.

Chapter 5

Experimental Characterization of UAV-to-car Communications

5.1 Introduction

Following the previous chapter, we now move on including UAVs that can support V2V communication as a relay for communication. Based on the work on the previous chapter, it is understood that in rural areas the communication shows greater performance. However, in rural area the infrastructure is limited and there is still a chance of getting the communications obstructed by vegetation or terrains. Hence, we propose a scenario where we include UAVs that communicate with cars to support V2X communications in rural areas.

In this chapter, we describe how we characterize UAV-to-car communications based on real experiments aimed to foster the development of a communications model to be used in simulation studies. In our scenario, UAVs act as mobile RSUs, enabling us to perform a study of vehicular communications between aerial and ground vehicles in the 5 GHz band. The experiments were performed in a rural area of Valencia, with actual field tests using vehicles and drones to determine the communications performance. In addition, based on the results obtained, we have modeled the packet delivery ratio in different

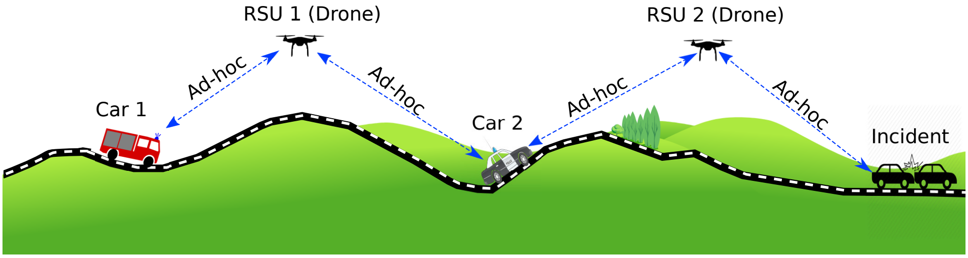


Figure 5.1: UAVs acting as Mobile RSUs.

scenarios (varying drone altitude, antenna rotation, and antenna orientation) using a modified Gaussian function.

5.2 Architecture Overview

In this section we start by providing a general overview of the envisioned scenario, and we then detail the proposed architecture, including the data flow and the different elements involved.

5.2.1 General Overview

Our ultimate goal is to build an environment where UAVs can be deployed to assist communications in an area with limited or no infrastructure support. Specifically, UAVs should be placed strategically to cope with those scenarios where vehicle-to-vehicle communications are hindered by the lack of line of sight. As depicted in Figure 5.1, the UAVs can be located near the top of surrounding hills causing line-of-sight obstructions between cars. Such UAV-assisted communications infrastructure would be useful when detecting a remote accident, especially if ground vehicular communications are not reliable due to large distances and non line-of-sight restrictions.

5.2.2 UAV-to-Car Communications

Our work focuses on measuring the effectiveness of point-to-point ad-hoc communications between UAVs and cars, which can eventually conform the big picture detailed in the previous subsection. In particular, for this experimental work, appropriate hardware/software are required to measure the packet delivery ratio, the range, and to analyze the data obtained. Three different

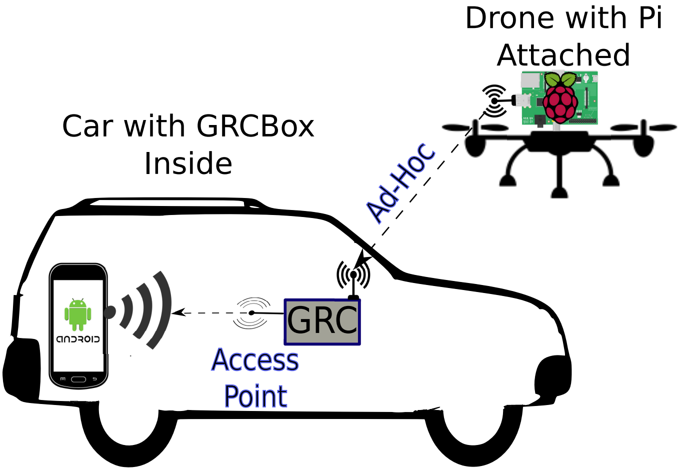


Figure 5.2: Flow of Packets from UAV (Drone) to Car.

devices are used in our communications experiments: a Raspberry Pi mounted on the UAV, an Android smartphone, and, finally, the GRCBox [TPCCM15], which is a car on-board unit that is endowed with multiple communication interfaces, including ad-hoc communication capabilities that is already presented in Chapter 2. The GRCBox is placed on the car, along with the Android device that works as the end receiver.

Taking into account the widespread adoption of smartphones worldwide, this configuration can be a non-expensive and fast solution to deploy ITS equipment. For our experiments, we developed a test application for Android that was installed on the Android smartphone. This application allows the driver to receive data from the drone via the GRCBox. Regarding the drone itself, we embedded a Raspberry Pi on it. The former is used to generate packets periodically, broadcasting them via the ad-hoc network created[1]. In order to join the ad-hoc network, a car on-board unit (GRCBox) is needed, as the Android device cannot operate in this mode by default.

The packet flow generated is depicted in Figure 5.2. Each packet is transmitted wirelessly from the UAV to the car on-board unit (GRCBox) via the ad-hoc network. Since the GRCBox also provides an access point inside the vehicle, it acts as a packet relay, forwarding the packets received to the Android device connected to the GRCBox.

[1] Note that this device differs from the embedded UAV flight controller.

For the experiment described in this chapter, although the communication is single hop (one car to one UAV), the transmission relies on the broadcast mode. Hence, the communication is not a direct communication, but an ad-hoc one. To assume a larger area involving multiple nodes, multi-hop transmission would be required. In that case, the nodes would act as both transmitter and receiver. Every node would relay packets received from another transmitter, that would in turn circulate the information in the ad-hoc network.

5.2.3 GRCBox Use and Drone's Interface for Communications

In our experiment, we will use the GRCBox as a receiver for packets that are broadcasted by the drone at the sender's side. The GRCBox is placed within the vehicle (see Figure 5.2). This GRCBox can act as the entry gate of packets that travel in the ad-hoc network connecting the drone and the car. The packets are then delivered to the passengers equipped with a smartphone running our GRCBox-aware application. Regarding the drone, it is equipped with a Raspberry Pi configured to send packets in ad-hoc mode. This way, the Raspberry Pi on the drone and the GRCBox inside the car can transparently communicate through the same ad-hoc network. The Raspberry Pi's antenna in the drone has the same frequency band as the GRCBox's antenna (5.8 GHz).

5.3 Experimental Settings

In this section, we start by providing an overview of the location where experiments took place. Afterward, we detail the experimental tools used. Finally, we analyze the data gathered in our experiments. The parameters considered in this experiment are summarized in Table 5.1, and will be further explained in the following subsections.

5.3.1 Experimental environment

For the experimental tests, we have selected a rural area located on the outskirts of Casinos, a small town located to the west of Valencia, Spain. The road was located far from the urban area itself, and thus interference was minimal when operating in the 5.8 GHz frequency band. As depicted in the aerial view (see Figure 5.3), we have selected a path for the vehicle that is more than 3 kilometers long.

The experiments took place during the sunny summer season. At the time of the experiments, no precipitations were present. The wind speed was spotted

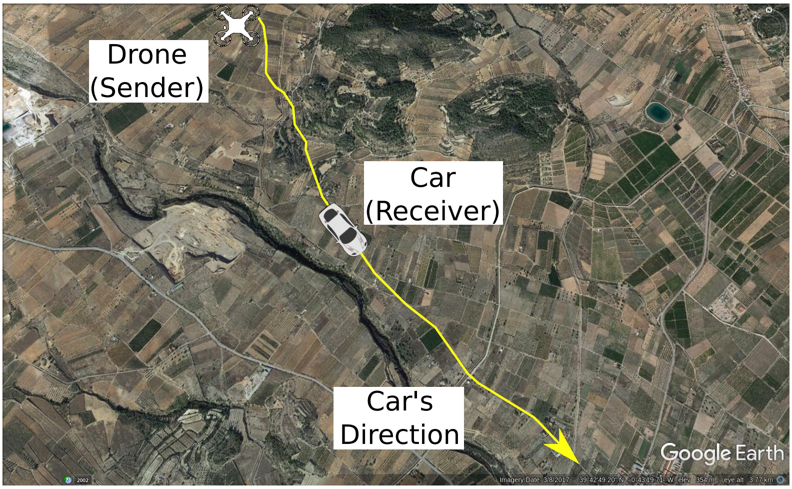

Drone
(Sender)
Car
(Receiver)
Car's
Direction
Google Earth
2002
Imagery Date: 3/8/2017 39°42'49.20" N 0°43'19.71" W elev 354 m eye alt 3.77 km

Figure 5.3: Satellite view of the target road near Casinos.

to be very low, representing a typical weather condition in a valley surrounded by hills. Hence, the weather conditions did not significantly affect the communications performance in our experiments.

In terms of the terrain profile surrounding the experiment location, the trajectory experienced different elevation points (see Figure 5.4). From the starting point to the destination point, the road is found to be running downhill. Specifically, as detailed in Figure 5.5a, the elevation at the starting point is about 400 meters, and at the final point it is about 320 meters above the sea level.

Table 5.1: Experiment Parameters for UAV-to-Car Communications.

Parameters	***Values***
Frequency Band	5 GHz
Transmitter Antenna Gain	5 dBi
Transmission Power	200 mW
Packet Size	1400 Bytes
Packet Rate	10 Hz
Transmission Rate	6 Mbps
Transmitter Altitude	40 m, 100 m
Transmitter Antenna Orientation	Vertical, Horizontal
Receiver Antenna Position	outside the car, inside the car

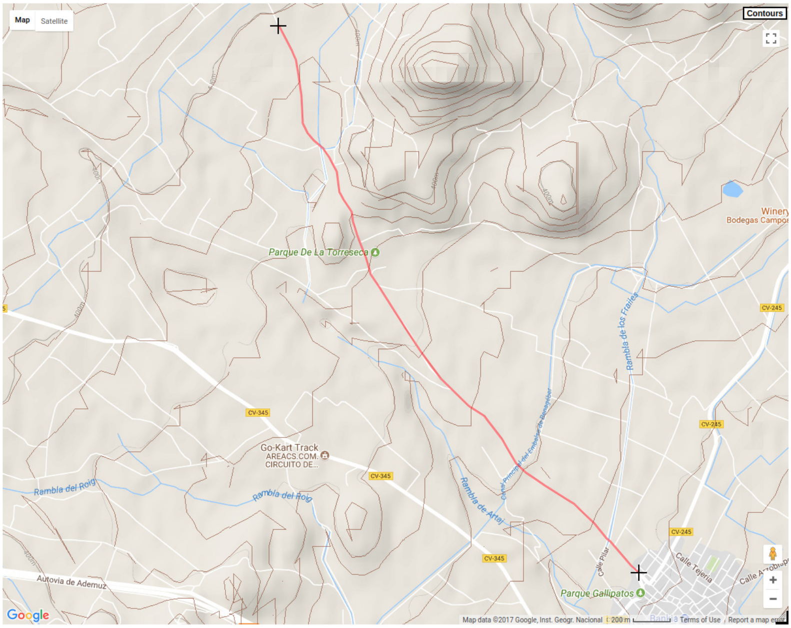
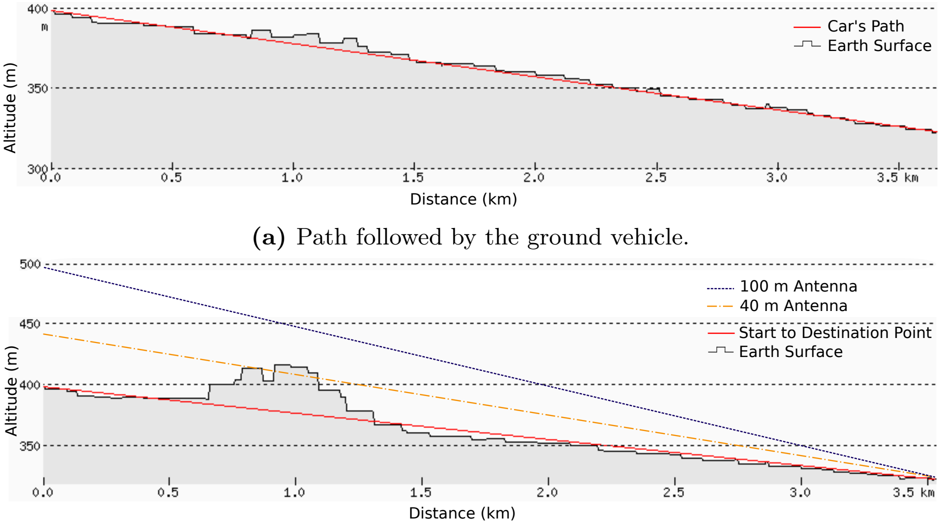


Figure 5.4: Topographic map of Casinos.

If we calculate the LOS between the drone and the vehicle, we can see in Figure 5.5b that, depending on the drone's altitude, the worst-case LOS conditions are expected when the vehicle is located about 900 meters away from the starting point, situation where signal obstructions exist. Such situation causes communications to experience NLOS conditions.

Our driving tests were performed with a static drone and one moving vehicle, as shown in Figure 5.6. Notice that the drone is located at the top of the scenario, near the vehicle's initial position. As the vehicle moves downwards, both its location and the drone's location are stored in a trace file. This is done to determine the positions associated to packet sending and reception events.



(a) Path followed by the ground vehicle.

(b) LOS conditions experienced when the signal source is located at the Initial Starting Point for different drone altitudes.

Figure 5.5: Elevation profiles relevant to our experiments.

5.3.2 Experimental Tools

Figure 5.7 shows our target scenario, illustrating with an example how packets are sent and recorded. The drone will generate UDP packets and broadcast them to an ad-hoc network. While the drone remains static, the car is moving away from the drone. The moving car is equipped with a GRCBox and an Android phone. This way, the car can seamlessly join the ad-hoc network. While moving and receiving packets, the car will record its geographical location. This way, by continuously retrieving the geographic information, the receiver (in this case, the car) can determine with great accuracy the GPS coordinates associated to the reception of each particular packet. The parameters used in this experiment with UAV-to-Car communications using GRCBox are the same as in the experiment described in the previous chapter, where the transmitting antenna has a 5 dBi gain, and a 200 mW transmission power.

At the sender side (Raspberry Pi embedded in the drone), we developed a Java application in order to generate broadcast messages. This test tool was tuned for a packet generation rate of 10 Hz, each with a size of approximately 1.4 Kbytes, being each packet numbered with a sequential ID. In addition, the timestamp and GPS location of the sender are also carried in the packet. These

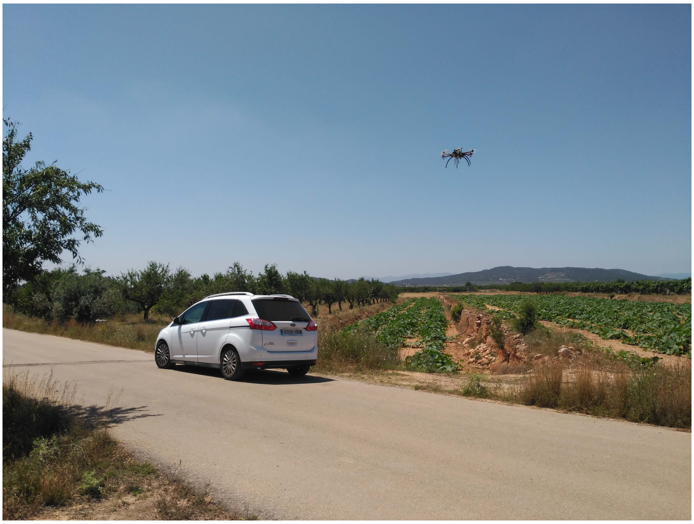


Figure 5.6: Real view of the UAV-to-Car communications scenario in Casinos.

transmission parameters are similar to those of typical DENM [ETS10a] in the context of vehicular networks. Also notice that, since all packet transmissions are broadcasted, the transmission rate is limited to 6 Mbps. The location information was obtained from the drone's GPS device by using the MAVLink protocol, that enables communication between the Raspberry Pi and the UAV's flight controller.

At the receiver's side, the packets were received by the Android smartphone using the GRCbox as a relay. We have developed a specific Android-based application for this purpose. This application provides the rendering of information received from the drone. The tool is also used for measuring the packet delivery ratio in the test scenarios. It is relevant to point out that our Android-based application is fully compatible with GRCBox, meaning that the user does not need to configure the connection to the GRCBox interface. Instead, the application contains libraries and plugins that allow connecting to the GRCBox module in a seamless manner. Once the smartphone is connected to the access point of the GRCBox (inner interface), a bidirectional communications relay is established with the outer interface (ad-hoc network). In this application, packet reception is started when, at the receiver's side, the user presses the receive button. The packets received are then recorded and stored in a log file until the user stops the application. The stored variables

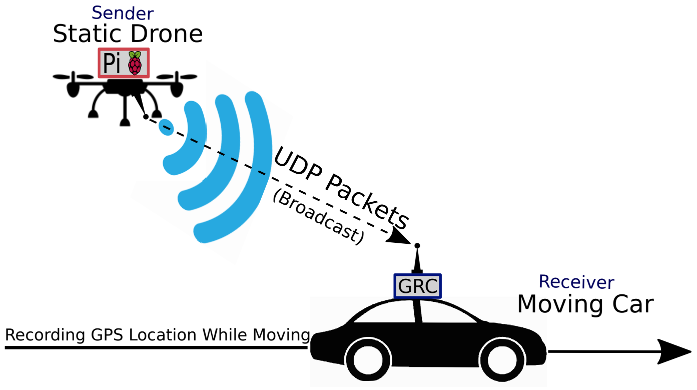


Figure 5.7: UAV-to-Car communications in our experimental scenario.

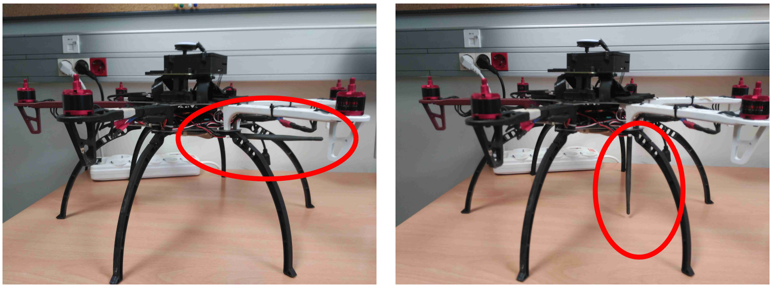
are then measured to determine which parameters have more impact on the communication's quality.

5.3.3 Data Analysis

The log file generated from the experiments contains all data required to measure the packet delivery ratio at different distances, including the geographic information (latitude, longitude, altitude). Since the receiver records the sender's geographical location, the distance between the drone and the moving vehicle can be directly inferred by comparing the sender's location and the receiver's location.

The packet delivery ratio is calculated by comparing the number of packets sent at the sender side, and the number of packets received during a specific time interval. For our study, we have in fact analyzed both endpoints, sender and receiver, as both contain a log file that is stored in each device. The log files are then analyzed to compare and calculate the ratio of packets sent and received by considering the geographical information inside.

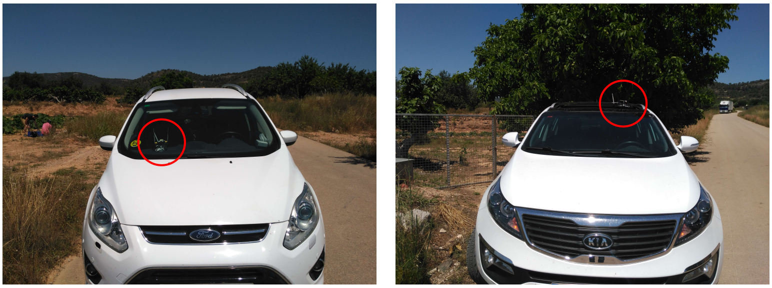
In order to measure the potential factors that could affect the communication's performance, we have selected three variables for the experiment: the altitude of the sender (the drone), the sender's antenna orientation, and finally the receiver's antenna location. First, the drone's altitude will have a clear effect on the LOS probability, as the higher the drone is, the lesser the probability of finding obstacles, and so we considered two altitudes: 40m and 100m, respectively. The orientation of the drone antenna also affects the communications



(a) Horizontal.

(b) Vertical.

Figure 5.8: View of the UAV antenna orientation.



(a) Inside the Vehicle.

(b) Outside the Vehicle.

Figure 5.9: Views of the car antenna location.

Table 5.2: Scenario Categories

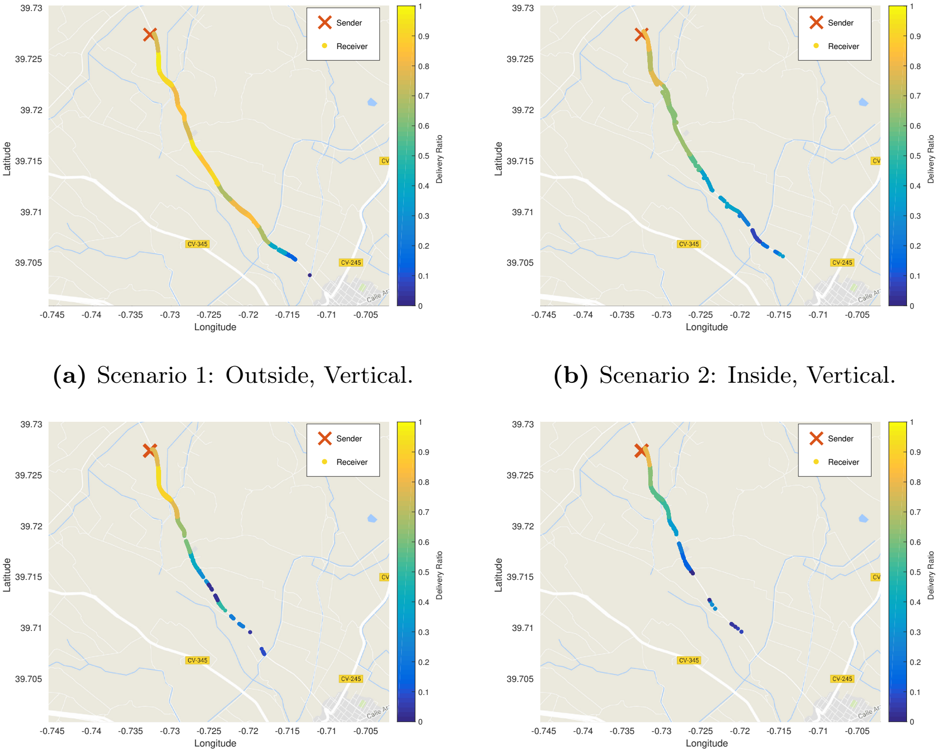
Scenario ID	UAV Altitude	Car's Ant. Position	UAV's Ant. Orientation
Scenario 1	100m	Outside	Vertical
Scenario 2	100m	Inside	Vertical
Scenario 3	100m	Outside	Horizontal
Scenario 4	100m	Inside	Horizontal
Scenario 5	40m	Outside	Vertical
Scenario 6	40m	Inside	Vertical
Scenario 7	40m	Outside	Horizontal
Scenario 8	40m	Inside	Horizontal

range, as the antenna waves propagate according to different radiation patterns. So, we considered horizontal and vertical orientations (see Figure 5.8). Finally, testing with different antenna positions in the vehicle is also interesting to study how communications degrade due to the vehicle's metal elements (see Figure 5.9). So, at the vehicle side, we tested two different positions for the antenna: inside (dashboard) and outside (rooftop).

In order to simplify the definition of our scenarios, we have categorized them (see Table 5.2) depending on the UAV's altitude, antenna position at the car, and antenna orientation at the UAV. The naming defined in Table 5.2 will be used to indicate each scenario in the sections that follow.

5.4 Experimental Results

The experimental results are presented as heatmaps (see Figures 5.10 and 5.11) for different drone altitudes. Each point represents locations where groups of packets were successfully received. The points have different colors according to the associated delivery ratio. In terms of the relationship between communications range and packet delivery ratio, the results obtained are depicted in Figure 5.12. With the results obtained, we propose a model in the following section.



(a) Scenario 1: Outside, Vertical.

(b) Scenario 2: Inside, Vertical.

(c) Scenario 3: Outside, Horizontal.

(d) Scenario 4: Inside, Horizontal.

Figure 5.10: Heat Maps for the different scenarios where the drone's altitude is of 100 meters. Each plot shows the packet delivery ratio depending on the sender's antenna orientation (Vertical, Horizontal), and the receiver's antenna location (Inside, outside).

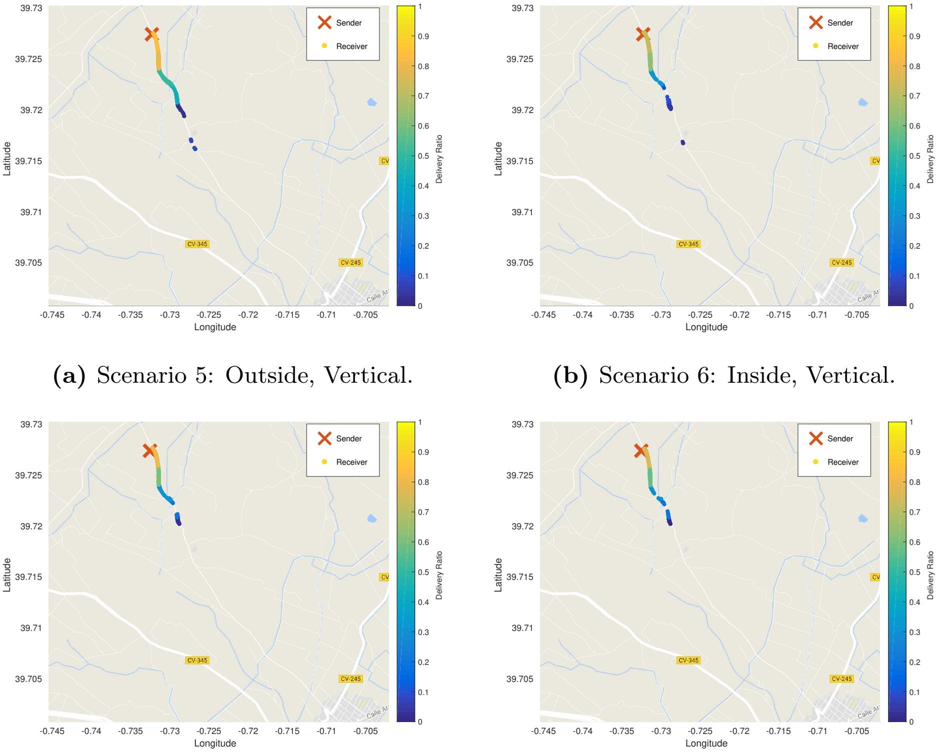
5.4.1 100m altitude

The results of the experiments made with the drone located at the highest altitude are shown in figure 5.10. The best results are achieved when installing the car's antenna outside, and having the drone's antenna pointing down, vertically (Scenario 1). We can see in figure 5.10a that, in this situation, the communication reaches up to 3 kilometers. When the highest distance is reached, the delivery ratio is roughly above 10%. Up until 1.4km, the delivery ratio is above 90%. At 2.8 km range, we can see that this ratio falls from about 0.7 to less than 0.5. As expected, the furthest the car is, the fewer packets are received.

On the other hand, if we put the receiver's antenna inside the car (Scenario 2), the results are slightly worse. As we can see in Figure 5.10b, the reception trend has the same pattern as in the previous scenario, although the delivery ratio is slightly reduced. This is due to the presence of the car's materials, especially metals, which block the signal. At 2.8 km range, the delivery ratio is found to be about 20%. On the other hand, at the furthest point of reception, the delivery rate is less than 10%.

Having the drone's antenna pointing horizontally introduces significant differences in terms of packet delivery. In Scenario 3, if we put the car's antenna outside, the communication range is lower when compared to Scenario 1, where we had the drone's antenna pointing down (vertical). Interestingly, up to a certain (short) distance, the delivery ratio is near 100%.

Finally, scenario 4 presents the worst overall results when compared to the previous three. The range is shorter, and the ratio is lower for all distances compared to the other cases. At the distance of 1.4 km, the packet delivery ratio is already below 50%, and the maximum distance reached in this scenario is only 2.4 km. Compared to the other experiments, where the UAV was flying at a height of 100 meters, this scenario presents the worst results in terms of communications range and packet delivery ratio performance, meeting the theoretical expectations.



(a) Scenario 5: Outside, Vertical.

(b) Scenario 6: Inside, Vertical.

(c) Scenario 7: Outside, Horizontal.

(d) Scenario 8: Inside, Horizontal.

Figure 5.11: Heat Maps for the different scenarios where the drone's altitude is of 40 meters. Each plot shows the packet delivery ratio depending on the sender's antenna orientation (Vertical, Horizontal) and the receiver's antenna location (inside, outside).

5.4.2 40m altitude

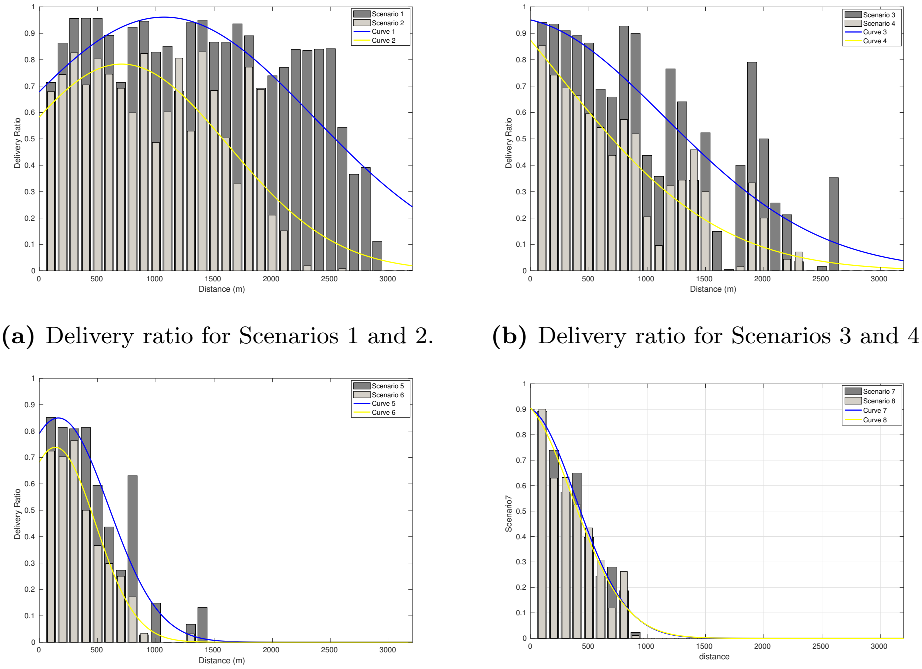
In this scenario we have decreased the drone's flight altitude, maintaining the same variables as in the previous scenario. Figure 5.11 shows the new results achieved. In general, we can see that the range is reduced when compared to the results achieved at 100 meters. We can also observe the impact of a little hill between the road that is located at coordinates with latitude 39.719058, and longitude -0.729005, drastically reducing the delivery rate.

As in the previous experiments, we find that scenario 5 achieves the best results in this group, being that the communications range reaches up to 1.5 kilometers. At 1 km, though, the packet delivery ratio already starts to drop to less than 20%.

When we locate the antenna inside the vehicle (Scenario 6), the result is less satisfactory than the previous one. By having a lower range, the delivery ratio is also lower for all distances when compared to the previous scenario. Another interesting fact is the ratio produced for the first few meters. In the case of scenario 6, this ratio is not decreasing in the range from 0 to 250 meters. In fact, it slightly increases until that point, and it eventually attenuates afterward. This is due to the orientation of the antenna at the drone, as having the antenna pointing directly to the receiver would represent worst-case conditions in terms of energy radiation from that antenna. The maximum distance reached in this scenario is 1.3 km.

On the other hand, when the drone's antenna has a horizontal orientation, the results are more standard, meaning that, as the distance increases, the delivery ratio attenuates in a monotonous manner. However, in terms of range and delivery ratio values, having the vehicle's antenna outside (scenario 7) still achieves lower performance than for the two previous scenarios (scenarios 5 and 6). In fact, compared to the previous scenario where the communications range is of 1.3 km, the maximum distance covered in this scenario is of less than 1 km. At 500 meters, the packet delivery ratio has already dropped to 43%.

Putting the antenna inside the vehicle while having the drone's antenna pointing horizontally is, as expected, the worst possible combination, achieving a low communications range and a low packet delivery ratio. For instance, the performance achieved in scenario 7 is noticeably better than the one for scenario 8. In fact, despite the maximum reachable distance is similar to the previous scenario, the packet delivery ratio obtained as distance increases is significantly less than the previous ones.



(a) Delivery ratio for Scenarios 1 and 2.

(b) Delivery ratio for Scenarios 3 and 4.

(c) Delivery ratio for Scenarios 5 and 6.

(d) Delivery ratio for Scenarios 7 and 8.

Figure 5.12: Curve Fittings of Delivery Ratio vs. Distance, with the antenna installed Inside or Outside the vehicle.

5.5 Experimental Modeling

Using the results of the previous experiments, we now proceed by modeling communications based on the different factors being studied (drone's altitude, transmitter antenna orientation, and receiver antenna location). Notice that our model was obtained based on the number of packets received at each registered position. As the packet delivery ratio is calculated for a small distance interval, we have performed curve fitting to derive the optimal parameters. The models generated can then be integrated into simulation tools to have more realistic performance results.

Specifically, for modeling this system, we used the location of the received packets (latitudes, longitudes, and altitudes), and their corresponding timestamp. The sender also stores information about how many packets were sent, allowing to compare against the packets received to calculate the delivery ratio based on distance ranges. We have sliced the distance range into intervals of 100 meters. The delivery ratio is then calculated for each distance range interval. The delivery ratio information for all scenarios tested is then plotted on a bar chart. Using this information, we continue by performing curve fitting using the nonlinear least-square Marquardt-Levenberg algorithm. This curve fitting procedure allows deriving a general model for the packet delivery ratio vs. distance under different conditions.

The curve fitting results are presented in Figure 5.12. We have tried to find a common model that would suit our variables (antenna orientation, antenna location, and UAV altitude). The goal is having fitting parameters that can vary from one scenario to another while maintaining the same function. Thus, the selected fitting function was the Gaussian function, and the best fitting was evaluated for the different cases studied:

$$f(x) = a \cdot e^{-\frac{(x-b)^2}{2c^2}} \tag{5.1}$$

Based on equation 5.1, we have three parameters for each scenario: a, b, and c. These parameters take the values shown in Table 5.3. We have also added the R^2 value to represent the goodness-of-fit. Based on the table mentioned, for each scenario, the model explains all the variability of the response data around its mean. The parameter is represented as R-squared, which is a statistical measure of how close the data are to the fitted regression line. In our fitting of the delivery ratio and distance values obtained from the experiment, it shows a high value of R-squared, being that it ranges from 0.7162 to 0.9844. Hence, the model represents an adequate fit.

As depicted in Figure 5.12, the curve that represents those scenarios where the antenna is placed outside the vehicle has higher values than the curve representing those scenarios where the antenna is located inside the vehicle.

Comparing the curve that represents the scenarios that involve pointing down the antenna (vertical) to the opposite case (horizontal orientation), we can observe that some curves bend, while others do not. The curves that bend are the curves that represent a scenario having the antenna pointing down (vertical). This is expectable as, when the car is just below the drone's position, the delivery ratio becomes lower, increasing when the car gets a bit farther from the drone. This phenomenon is due to the fact that the antenna installed on the UAV is an omnidirectional antenna, thereby radiating power uniformly in one plane, with the power decreasing with the elevation angle above or below that plane until it theoretically reaches zero on the antenna's axis [McD99].

Table 5.3: Value of parameters a, b and c for each scenario with its R^2 value .

	a	b	c	R^2
Scenario 1 (Curve 1)	0.9613	1071	1815	0.7162
Scenario 2 (Curve 2)	0.7834	705.4	1296	0.8132
Scenario 3 (Curve 3)	0.9593	-180.2	1883	0.7646
Scenario 4 (Curve 4)	1.24	-11.49	1941	0.8627
Scenario 5 (Curve 5)	0.8497	163.4	609.5	0.9279
Scenario 6 (Curve 6)	0.7385	137.7	488.9	0.9844
Scenario 7 (Curve 7)	0.9083	-48.59	616.4	0.9786
Scenario 8 (Curve 8)	0.932	-113.1	654.2	0.971

Hence, the signal that propagates from the antenna has its strongest power when the receiver is perpendicular to it. On the other hand, when the receiver is located parallel to the antenna (in our case, just below), the signal received will be weak.

From another perspective, when comparing the curves for the scenarios where we vary the drone's altitude, it quickly becomes clear that higher altitudes are associated to greater distances. For Scenarios 1 and 2, the curves continue up to distances of more than 3.2 kilometers. For Scenario 3, the curve stops at 3 km, reaching 200 additional meters on Scenario 4. On the other hand, for both Scenarios 5 and 6, the resulting fitting function cannot reach more than 1.5 km. On the last two scenarios, the maximum point reached by the resulting curve is 1.3 km. That is, for the scenarios when the drone's altitude is of 40m, these cut-off values precisely correspond to locations where LOS is blocked by hills, as shown in figure 5.5b.

It is worth pointing out that, since our experiments were located in a quite remote rural area, no significant interferences were present that could hinder the communications band. Thus, the results achieved are quite reasonable as, for this situation, the fitting suits an AWGN (Additive White Gaussian Noise) channel model, meaning that only the effect of additive white Gaussian noise is noticeable. Also, we find that the only obstacles that hindered the communication were little hills that blocked the line of sight in some areas of the selected scenarios.

5.6 Model Applicability and Comparison Against Existing Models

In this section, we discuss the novelty and applicability of our model by comparing it against other existing propagation models, and explaining how it could be used as a part of existing network simulators.

5.6.1 Comparison with Existing Models

The characterization of a data link between two endpoints requires studying the signal propagation conditions in the target environment. The line of sight conditions and the presence of obstacles clearly affect the channel model due to the signal reflections and Doppler spread effects. This is specially true for vehicular communications, where it can be affected by the presence of buildings, terrain profile, or the car metal structure itself.

Various empirical path loss models have been proposed to address the signal propagation phenomenon. However, very specific transmission conditions cannot be addressed by these models, as they only provide a generic propagation behavior. Regarding vehicular communication settings, we have found in previous chapter that different types of intersections have their own characteristics that define the communications performance when the visibility between two nodes is obstructed by either buildings or trees . In our current UAV-to-vehicle communications environment, we have found that terrain conditions (e.g hills, mountains) can be a critical factor affecting communication.

To clearly demonstrate the usefulness and novelty of the models derived from this work, we have also compared our results with those obtained from simulation experiments that mimic the real-life experimental settings. We have chosen the OMNeT++ [VH08] tool for simulation tests. Table 5.4 presents the simulation parameters adopted. Notice that, in order to get simulation results comparable to the real experimental ones, we have set the simulation parameters in accordance to real-life settings, except for the Path Loss Model, which is the parameter under analysis.

For our simulation experiments we considered two existing empirical path loss models. The first one was the Free-space path loss model [Fri46], and the second one was the Rayleigh fading [Skl97] model. Simulation experiments include two nodes, one static and another one mobile, thereby resembling the drone and car used in the real experiment. Then, we have measured the packets sent from the transmitter, and those received correctly at the receiver

Table 5.4: Simulation Parameters for Comparability Test.

Parameters	***Values***
Transmitter Power	200 mW
Antenna Gain	5 dBi
Packet Size	1400 Bytes
Packet Rate	10 Hz
Transmitter Altitude	40m/100m
Path Loss Model	Free-Space, Rayleigh Fading

node. Afterward, we have calculated the packet delivery ratio progressively, depending on the distance between the two nodes. This way, we obtained a graph for packet delivery ratio versus distance, the same way as we did with the results obtained in the real experiment.

Figure 5.13 combines the results for the real tests and the simulation tests for comparison. The curves have distinct colors, where *Scenario 1* represents the real experiment with the drone flying at 100 meters in a green curve, and *Scenario 5* represents the real experiment with the drone flying at 40 meters in a red curve. In both cases, these results correspond to the best case scenario, with the UAV antenna pointing down, and the vehicle's antenna located on top of it. Regarding the *Rayleigh* results represented in the figure in a blue curve, these correspond to the simulation test using the Rayleigh fading model, while the *Free-Space* line (in cyan color) corresponds to simulation results using the Free-Space path loss model. Regarding the altitude parameter, simulation results actually show that the impact of varying the UAV's altitude is negligible, being that very similar results are obtained when having the transmitter at 40 meters or 100 meters. Hence, the curves that are shown in the figure are applicable for the channel model regardless of the transmitter's altitude chosen. The reason why the transmitter's altitude variation did not make difference on the result is that the simulator was not prepared for 3D communications. Hence, even though that we can input the altitude of the transmitter, in fact the communication was two dimensional.

Based on the results presented in Figure 5.13, we can notice that the Rayleigh fading model does not represent the performance of the communications achieved when flying at either 40 or 100 meters. In fact, the Rayleigh fading model only shows a progressive decline as the distance between endpoints increases. In addition, even though in the real experiment the communication was lost after 3.2 kilometers, the curve still shows a successful packet delivery beyond that

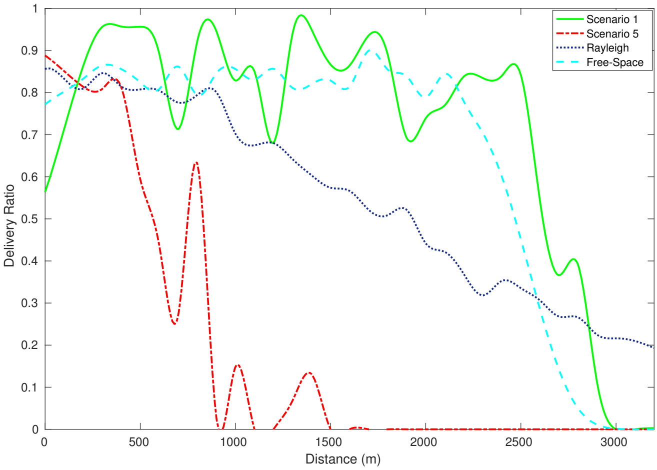


Figure 5.13: Comparison of results obtained from simulations and real experiments.

distance. Overall, we can conclude that it cannot represent the performance for UAV-to-car communication conditions.

Regarding the Free Space path loss model, it is more adequate for the scenario being tested. We can see that, as the distance increases, the power attenuates drastically after a certain distance. This is shown in its curve when it attains 2000 meters, being that the curve plunges from 0.85 to less than 0.5 for a distance variation of only 200 meters. This trend is similar to the curve that represents the real experiment when the transmitter was at an altitude of 100 meters. Notice that, at this particular altitude, the terrain profile did not cause line-of-sight obstructions. However, compared to the curve that represents the experiments having the transmitter at 40 meters of altitude, we find that the Free Space model is no longer applicable. Now, since communication obstructions were present between the drone that acts as the transmitter and the moving vehicle, the packet delivery ratio drops for much shorter distances. This is due to the fact using the Free space path loss model, only shows a condition where the communication is not obstructed or there is no interference along the distance. Whether the fact is that there are hills that interfere the communications, this model did not assume such case which can explain why there is no sudden drop in between. This is due to the fact using the Free space path loss model, only shows a condition where the communication is not obstructed or there is no interference along the distance. Whether the fact is that there are hills that interfere the communications, this model did not assume such case which can explain why there is no sudden drop in between.

Taking into account the results presented above, we conclude that the existing channel models still cannot address the specific characteristics of UAV-to-Car communication scenarios. In fact, we found that the free-space path loss model is applicable, but only when there is line of sight between both endpoints. In case the terrain profile causes obstructions, then traditional models are no longer applicable, and an obstruction model specific to each particular scenario is required.

5.6.2 Applicability of the proposed Model in Simulation

Knowing that our proposed model can be an alternative for defining communications performance in UAV-to-car scenarios, our intended goal is to use this model for larger scale simulations. In order to adapt our model to simulation environments, information regarding the terrain profile in the target area should be known in advance. Hence, a topographic map with elevation information would be needed for the simulation to be as realistic as possible.

By analyzing a 3D map (with elevation information), one can characterize the line-of-sight conditions between UAVs and cars by classifying the elevation and landforms as obstacles to communication. This classification process can take place before the actual simulation. Widely used map providers such as OpenStreetmap [HW08] include elevation information, which makes the adoption of our model feasible. This way, the adoption of the models derived in this chapter can be used in combination with line-of-sight availability based on information of the actual scenario topology, which is used to determine whether signal blocking due to obstacles will take place or not, prior to checking the packet delivery feasibility accounting for distance-related loss. The effort on investigating this issue will be presented in the following chapter.

5.7 Summary

In this chapter, we have studied the packet delivery effectiveness on UAV-to-car communications. For this study, we have varied the drone's altitude, the transmitter antenna orientation, and the receiver antenna location. According to our experiments, the best scenario for UAV to car communications takes place when the drone's antenna is pointing down (vertical), the car's antenna is located outside the vehicle, and the drone's altitude is very high (100 meters, near the maximum allowed limit according to the legislation in different countries), thereby helping to avoid any LOS obstruction. Under this setting, the communications range achieved was of 3.2 kilometers.

In addition, based on the experimental results obtained, we have also found the best-fitting function and applied regression to provide a generic model applicable to the different situations under study. The model is based on the Gaussian function, which offers a suitable result representation, such as the ones obtained in our tests. In fact, by merely adjusting two values of the fitting function, we can adapt it to the different scenarios tested. Simulation results have also shown that the derived models keep some relationship with the standard Free Space model when LOS conditions are met, but not otherwise.

Knowing that we have proposed an empirical model obtained from experimental results, it is desirable to import the scenario into simulation. In the following chapter we will investigate the same communication method in simulation. By getting the experimental results, these can be useful to validate the simulation model proposed in the next chapter. Even though we have tested the use of existing propagation models in the simulation, we can observe that the variation of the drone's altitude did not make any differences. This is due to the fact that the communication was not in 3D, hence the result will be the same at the same distance between the drone and the car. For this reason, in the following chapter we will investigate the impact of the drone's altitude on the communications by considering the terrain profile.

Chapter 6

Simulation Modeling of UAV-to-car communications

6.1 Introduction

Based on the findings of the previous chapter, it becomes evident that the simulation tool used was not prepared for three dimensional communications, which is essential for communications involving UAVs as flying nodes. In fact, the UAVs were assumed to have the location in 2D, which is unrealistic. In this chapter, we propose and implement a novel simulation model for UAV-to-car communications. The model takes into account the results obtained from the real testbed presented in the previous chapter, and it was developed for the OMNeT++ simulation tool [VH08]. The simulation model takes into account three-dimensional communications. A comparison with two-dimensional communications is also presented for the sake of completeness. This model calculates the signal attenuation due to the presence of hilly or mountainous terrains that hinder communications by retrieving elevation information of the Earth provided by a Digital Elevation Model (DEM) [Gal00]. Signal attenuation is obtained from existing propagation models. The use of various DEMs and propagation models (i.e. diffraction models like Bullington and Deygout) are compared to the results obtained from our testbed experiments. The results from the simulation tests show that the model obtains results comparable to testbed experiments, thus validating their adequateness.

6.2 3D Model for UAV-to-Car Communications

The main goal of our research work was to support the use of UAVs as relays for vehicular communications. An UAV can act as a message relay for communications between cars when the area of the network has limited infrastructure support. As seen in Fig. 5.1, UAVs can be useful to detect accidents in rural areas. In this chapter, we develop a simulation framework where the communications between nodes are affected by terrain irregularities that might act as obstacles. Hence, the actual 3D terrain profile is of utmost importance since it can impact the network connectivity.

For UAV-to-car communications, the challenges are not the visibility range nor the interference, especially in rural areas, where infrastructures are limited and the interference was minimal. The terrain profile is the main challenge, especially when the presence of hills or mountains represents obstacles that hinder wireless communications. The detection of obstacles is thus necessary to characterize the communication when obstructed by the terrain features. This is done by extracting the elevation information of the terrain. An obstruction to communications is defined by a higher elevation. This obstruction from the elevation can cause moderate to high attenuation of radio signals.

In the case of a larger scenario, where the communications would involve multiple nodes in the area, interferences would be more common. This is due to the fact that there are transmissions of multiple nodes on a common channel. In the presence of interferences, the signal would suffer noise that would impact the packet delivery. Greater noise would produce failures on the delivery of packets in the network. However, in this thesis work, we did not assume any interference since our focus was the communication link between the UAV and car.

6.2.1 Elevation Model

In order to include terrains as obstacles, such as hills or mountains, information about the elevation of the terrain is required. A DEM provides information about the real-world terrain data, including its elevation. The DEM, which contains data about the Earth's surface, can be used to recognize how high the terrain is above the sea level. Based on that information, the terrain can be defined either as flat, hilly, or mountainous. The DEM also provides information about the elevation of any given points, and it can be obtained from different sources. Specifically, we used two different DEMs: the DEM from National Aeronautics and Space Administration (NASA), which is collected from the Shuttle Radar Topography Mission (SRTM), and the Google's DEM, accessed via the Google Maps Elevation API.

The NASA's DEM is collected from the SRTM [FRC+07], which obtained such information by capturing the Earth's surface with a radar sensor from space. The model has various resolutions that define the accuracy of the elevation information. NASA's DEM files are in "height" format, and contain global elevation data captured during NASA space missions, being named in the format known as *northing* and *easting*, which are based on their coverage in geocoordinates. *Northing* means the northward measured distance when adopting the UTM system, while *easting* means the eastward-measured distance. The file then covers one degree of latitude and one degree of longitude. To include DEM into our simulation environment, a suitable file based on the location of the scenario should be selected. This DEM contains an array of points in the form of a grid and has two resolutions, one is 1-arc second, and the other one is a 3-arc second sampling. The right elevation is obtained by pinpointing the appropriate latitude and longitude into the right cell of the grid.

If we use a DEM corresponding to a 3-arc second sampling in order to obtain elevation data, we will get a grid of 1201 x 1201 cells, as illustrated in Fig. 6.1, in which 1201 cells represent one degree of latitude or longitude. To define the elevation or the height of the terrain, we must index the target coordinate based on its northing and easting information. To pinpoint the exact cell of the grid, we can use the fraction part of the geographical coordinate and multiply it by 1200, in the case of 3-arc second sampling. As an example, 0.5 belongs to the 600th cell.

Google's DEM can be obtained by an API that allows us to query locations around the Earth for elevation data. The Google Maps Elevation API [Dev18] is intended for developing hiking and biking applications, positioning applications, or low-resolution surveying applications. Using this API, we can retrieve the altitude information of basically any location on Google Earth. To retrieve the altitude of the terrain, the API requires the coordinates in terms of longitude and latitude. As a result, it will provide the altitude in meters. The good thing about using this API is that it can access the larger database that Google has, which might have better accuracy as it is daily updated by both developers and users. In addition, it is easier to retrieve the elevation information, as we merely need to input latitude and longitude data to retrieve the altitude

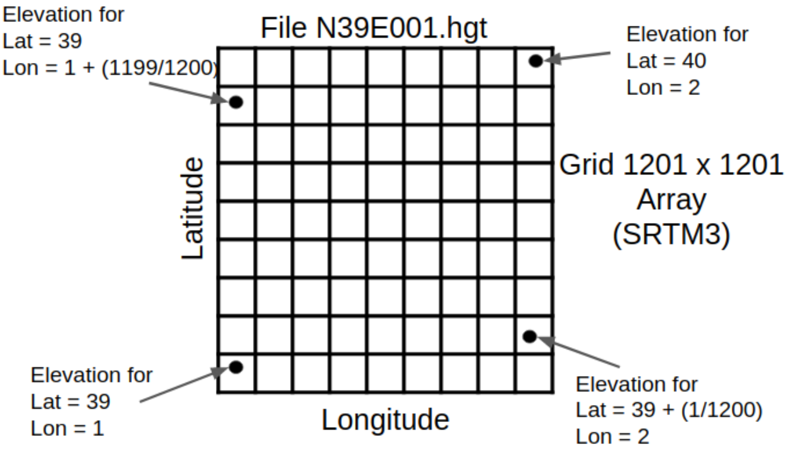


Figure 6.1: SRTM DEM Data Structure.

of the terrain in return. The drawbacks of using this API is the exclusivity of accessing the data. The API has a limited free access, which is only up to 10 requests per second, for a maximum of 2500 requests per day. In addition, there is a maximum of 512 locations that can be retrieved per request. This means that the service is not accessible to the general public who typically has limited resources.

6.2.2 Propagation Model

High terrains can affect the quality of wireless communications since they can represent signal obstructions. The presence of hilly or mountainous terrains can cause non-line-of-sight conditions. This means that the power of the received signals is reduced or even blocked, and so these terrains can act as signal obstructions. In some cases, diffraction effects can also occur due to high terrains. This means waves will bend or deflect when encountering hilly terrains or other types of obstructions. We can analyze the diffraction effect by approximating obstacles as knife edges. In such case, the signal transmitted in the network will be attenuated due to the knife edge diffraction effect. Knife edge diffraction loss is calculated by considering the distance between the transmitter and the receiver, the wavelength of the signal, and the height (h) of the obstacle above the line of sight line formed between the transmitter and the receiver. By retrieving the elevation information, we can obtain this height.

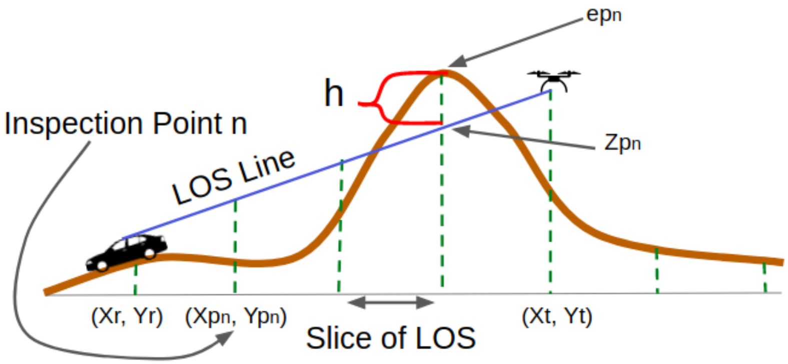


Figure 6.2: Set of slices that divide the LOS into different inspection points.

Calculation with a parametric equation is required in order to get the LOS line between the locations of the transmitter and the receiver. Then, detecting the terrain that acts as an obstacle and blocks the signal is done by moving along the LOS line in discrete steps. We will be able to calculate the height of the obstacle (h) or the knife at each step by getting a positive value. Otherwise, getting a negative value means that the terrain is not an obstacle to the signal.

In more detail, defining a LOS line is done by a parametric equation based on the location of the UAV, or in this case, the sender, which includes its latitude (x_s), longitude (y_s), and altitude (z_s), and the vehicle position (x_r, y_r, z_r). In order to compare the detected LOS line and the terrains present in the network, we have to define an inspection point (x_p, y_p, z_p). We can obtain the inspection point and analyze it depending on how many slices the LOS line is divided into. Notice that the LOS line is divided into slices by inspection points, as shown in Fig. 6.2. The fraction of the LOS can be defined by specifying these slices. This fraction will be the s parameter in the following parametric equation:

$$\begin{aligned} x_p &= x_s + s \cdot (x_r - x_s) \\ y_p &= y_s + s \cdot (y_r - y_s) \\ z_p &= z_s + s \cdot (z_r - z_s) \end{aligned} \tag{6.1}$$

that is defined by the spacing of the resolution of the map. Slicing the LOS or spacing it should not result in portions that are smaller than the resolution of the DEM file, since it can lead to unnecessary computing efforts. However, a higher spacing can lead to losing the information about potential obstacles that are present in the end-to-end link.

Table 6.1: Comparison of diffraction methods.

Features	Single Knife Edge	Deygout	Bullington
Number of Edges	Single	Multiple	Multiple
Accuracy	Low	Improved	Low
Complexity	Simpler	More Complex	Simpler
Main Edge	Edge with Maximum Height	Principal edge with largest v parameter	Equivalent knife edge between transmitter and receiver

The s parameter is needed to calculate the inspection point. Then, we can compare the z_p position for every inspection point. This point acts as the altitude of the LOS point. The elevation point or the terrain height (e_p) is obtained by consulting the DEM file to extract the elevation information by getting the x_p and y_p coordinates of the inspection point.

The knife or the height of the obstacle is obtained by calculating the difference between e_p and z_p, and it is denoted as h_p. When an elevation point at the inspection point is higher than the height of the LOS, or its altitude, an obstacle is detected at that particular location. The outcome of the calculation would be a positive h_p value. On the other hand, when the value of h_p is negative, LOS is not obstructed by an obstacle

Now, we describe the three propagation models used in the experiments: the first one (the *Single Knife Edge*) only considers a single edge, and it is based on the Fresnel-Kirchoff diffraction model; the other two consider several knife edges, assuming that there could be more than one obstacle present in the signal path. Among the most popular and widely used multiple knife edge diffraction models, we will analyze the *Bullington* [Bul47] and the *Deygout* [Dey66] diffraction models. The main reason behind choosing these models is that these two models were suggested in recent ITU-R recommendations when studying signal propagation effects in the presence of diffraction [P.509; P.518]. For a general comparison between these three methods, table 6.1 summarizes their main features.

SINGLE KNIFE EDGE

In the first model, a Fresnel-Kirchoff diffraction parameter (v) is required after getting the height of the obstacle or the knife at a specific inspection point with the purpose of calculating the signal attenuation. It is obtained by taking into account the height of the obstacle (h_p), the distance between the obstacle and the transmitter (d_s), the distance between the obstacle and the receiver (d_s), and the wavelength (λ). Thus, the calculation of the diffraction parameter is done using the following equation:

$$v = h_p \sqrt{\frac{2(d_s + d_r)}{\lambda d_s d_r}} \tag{6.2}$$

We can then define the signal attenuation due to diffraction from the obstacle by obtaining the v parameter and using the Fresnel Integral $F(v)$. According to Lee [Lee82], the Fresnel integral is determined as follows:

$$\begin{aligned} F(v) &= 0 && v \leq -1 \\ F(v) &= 0.5 - 0.62v && -1 \leq v \leq 0 \\ F(v) &= 0.5e^{-0.95v} && 0 \leq v \leq 1 \\ F(v) &= 0.4 - \sqrt{0.1184 - (0.38 - 0.1v)^2} && 1 \leq v \leq 2.4 \\ F(v) &= \frac{0.225}{v} && v > 2.4 \end{aligned} \tag{6.3}$$

Once the Fresnel integral result is obtained, we can define the signal attenuation due to the diffraction using the following equation (in dBm):

$$L(dB) = 20log\left|F(v)\right| \tag{6.4}$$

DEYGOUT MODEL

The Deygout model differs from the single knife edge diffraction model as it not only considers the point with the highest value of the knife along the path, but it also considers "secondary" obstacles that are added to the diffraction loss over the main obstacle. This is a cascaded knife edge method. The diffraction parameter calculation is the same as for the previous method (single knife edge) using equation 6.2. The knife edge with the highest value must be found. However, to get the height of the obstacle, or h_p, the calculation was not done using the previous parametric equation. Instead, it uses the equation shown below:

$$h_p = e_p + [d_s d_r / 2r_e] - [(z_s d_r + z_r d_s)/d] \tag{6.5}$$

where we have to include the Earth curvature in this method; r_e corresponds to the effective Earth radius, which is about 6371 kilometers, and d is the distance between the transmitter and the receiver.

The value of v for every point must be calculated referencing every knife edge. The maximum value of v indicates the principal knife edge, which is the highest obstacle. If v results in a value of more than -0.78, it means that there is no loss in the signal path; otherwise 0 dB is returned. However, if the v value is less than -0.78, then we have to keep calculating the value of v, but this time the points are not between the transmitter and the receiver, but between the transmitter and the principal knife edge, which will return the v_s value. The same process is then repeated for the points between the principal knife edge and the receiver. This time we will get the v_r value.

Once we get the knife height of the principal edge (v), the edge between the principal edge and the transmitter (v_s), and the edge between the principal edge and the receiver (v_r), we can calculate the attenuation using this method. The signal loss, in dB, is denoted as:

$$L(dB) = F(v) + T[F(v_s) + F(v_r) + C] \tag{6.6}$$

Calculating the Fresnel integral for Deygout is not the same as in the single knife edge diffraction model. The Fresnel integral calculation used in this method is as follows:

$$F(v) = 6.9 + 20log(\sqrt{(v-0.1)^2+1} + v - 0.1) \tag{6.7}$$

The C value in equation 6.6 is an empirical correction given by the following equation:

$$C = 10.0 + 0.04D \tag{6.8}$$

where D is the total path length in kilometers, whereas T is a factor denoted as:

$$T = 1.0 - exp\left(- \frac{F(v)}{6.0} \right) \tag{6.9}$$

This way, the $L(dB)$ value obtained is the overall attenuation value for the method when using the Deygout diffraction model.

BULLINGTON MODEL

As recommended by ITU for propagation by diffraction [P.518], the Bullington model is used as a method for a generic terrestrial path. The Bullington diffraction method is used to calculate the attenuation whether the signal path is LOS or is trans-horizon. In this method, the height of the knife is defined by the slope of the line between the transmitter, the inspection point, and the receiver.

In the Bullington method, the first step to do is to get the intermediate profile point with the highest slope of the line from the transmitter to that point. The slope is obtained using equation (S_{tim}):

$$S_{tim} = max\left[\frac{e_p + 500C_e di(d - d_i) - z_s}{d_i} \right] \tag{6.10}$$

where C_e is the effective Earth curvature in km^{-1}, which is given by $1/r_e$. d_i is the distance between the transmitter and the inspection point i (or the point which divides the LOS into slices). After getting the slope of the line between the transmitter to the inspection point, we need to calculate the slope of the line from the transmitter to the receiver assuming a LOS path:

$$S_{tr} = \frac{z_r - z_s}{d} \tag{6.11}$$

After getting the two slopes, there are two cases to be considered. The first case considers that the path has LOS, and so the slope of the line from the transmitter to the point is less than the slope of the line from the transmitter to the receiver ($S_{tim} < S_{tr}$). In that situation, we need to find the inspection point that has the highest diffraction parameter v, obtained by:

$$v_{max} = max\left\{ \left[e_p + 500C_e di(d - d_i) - \frac{z_s(d - d_i) + z_r d_i}{d} \right] \sqrt{\frac{0.002d}{\lambda di(d - d_i)}} \right\} \tag{6.12}$$

In this first case, the knife edge loss for the Bullington method is obtained by:

$$L(dB) = F(v_{max}) \tag{6.13}$$

using equation 6.7, if the v_{max} is greater than -0.78. Otherwise, if it is lower, the value returns 0 dB.

For the second case, the path is trans-horizon, which means that the slope of the line between the transmitter and the inspection point is equal or greater than the slope of the line from the transmitter and the receiver ($S_{tim} \geq S_{tr}$). Hence, for this case, we need to find the slope of the line from the receiver to the inspection point using the following equation:

$$S_{rim} = max\left[\frac{e_p + 500C_e di(d - d_i) - z_r}{d_i} \right] \tag{6.14}$$

Once we have obtained the slope of the line, we have to calculate the distance of the Bullington point from the transmitter using the following equation:

$$d_b = \frac{z_r - z_r + S_{rim} d}{S_{tim} + S_{rim}} \tag{6.15}$$

The Bullington point is required to get the Bullington diffraction parameter, denoted as v_b, which is obtained through the following equation:

$$v_b = \left\{ \left[z_s + S_{tim} d_b - \frac{z_s(d - d_b) + z_r d_b}{d} \right] \sqrt{\frac{0.002d}{\lambda d_b(d - d_b)}} \right\} \tag{6.16}$$

Then, the Bullington diffraction parameter v_b is used to calculate the knife edge loss for the Bullington point as:

$$L(dB) = F(v_b) \tag{6.17}$$

using equation 6.7 to calculate the Fresnel integral. In both cases, whether the signal path is LOS or trans-horizon, we should calculate the final diffraction loss for the Bullington method (L_b) using the following equation:

$$L_b = L + [1 - exp(-L/6)](10 + 0.02d) \tag{6.18}$$

where L_b is the overall attenuation value obtained when applying the method using the Bullington diffraction model. The comparison of using these various diffraction models will be presented and evaluated in Section 6.6.

6.2.3 Implementation of the proposed model in simulation

This subsection is devoted to describing the implementation of the proposed 3D communications model in a standard simulation tool. In particular, for our experiments, we have used OMNeT++ [VH08] for network simulation, SUMO [BBEK11] for vehicular traffic simulation, and the Veins [SGD11] simulation framework to connect SUMO and OMNeT++ for specific vehicular communications simulation. In particular, the effect of 3D wireless transmission has been implemented by modifying its physical layer. For our simulation experiments, we use Wireless Access in Vehicular Environment (WAVE) as the MAC protocol, and the control channel is used for communications.

Our proposed model works as an extension to this simulation framework (combining OMNeT++, SUMO, and Veins) by adding a module that considers 3D space features. Traditional modules for signal propagation only consider 2D space. By modifying the physical layer in the simulation, we can simulate the effect of 3D wireless transmission. When configuring the propagation model in the simulation, we have to take into account the location of the nodes (latitude, longitude, and altitude), and the terrain spacing. In addition, other parameters that characterize the behavior of the signals, like the wavelength and the received power threshold, are configured by default in Veins. All these features are implemented in Veins, as shown in Fig. 6.3.

Inside the simulation tool, we have to extract the elevation information first. By considering the location of the nodes with respect to its northing and easting, the DEM data are imported. By consulting the DEM file, we can retrieve the elevation information that is accessed inside OMNeT++. The DEM file is locally stored in order to reduce computation, and thus simulation time. Thanks to the assistance of TraCI in SUMO, we can obtain the exact geocoordinates (location) by connecting it to Veins, according to the mobility of the nodes.

The detection of obstacles is required after obtaining the coordinates of the vehicle from the mobility module. This functionality is executed inside the propagation module at the physical layer. The terrain profile with its elevation must be estimated in order to detect obstacles. The DEM file inside OMNeT++ is accessed in order to estimate the elevation of the terrain. By being able to detect obstacles using the knife-edge diffraction model, the propagation module can better determine the actual signal attenuation. On the receiver side, OMNeT++ can assess the signal strength, and determine the minimum signal strength to receive packets from this communication model.

As an alternative, if we want to use another elevation model for comparison purposes, we will include the Google Maps Elevation API; in this case, some libraries are required to be included in the simulator. Since OMNeT++ simulator is written in C++, libraries such as *libcurl* [Lib18] and *jsoncpp* [JSO18], both implemented in C++, provide access to the Google Maps Elevation API. The *libcurl* library is used to fetch data from remote websites, in our case the Google Maps Elevation API. In addition, since the Google Maps Elevation API would return the information in JSON format, the *jsoncpp* library is needed to decode and analyze the received information. It will parse the value of the elevation information that will be needed as information about the height of the terrain. Notice that terrain height information is the basic parameter required to assess the signal strength using the propagation module adopted.

6.3 Simulation Setup

In this section, we present an overview of the simulation architecture and the settings used to validate our model. Details about how to import the geographical location into the simulation, as well as using the 3D propagation module in the simulation tools for a specific scenario, are provided in this section. In addition, data analysis covers how data are obtained to produce adequate results.

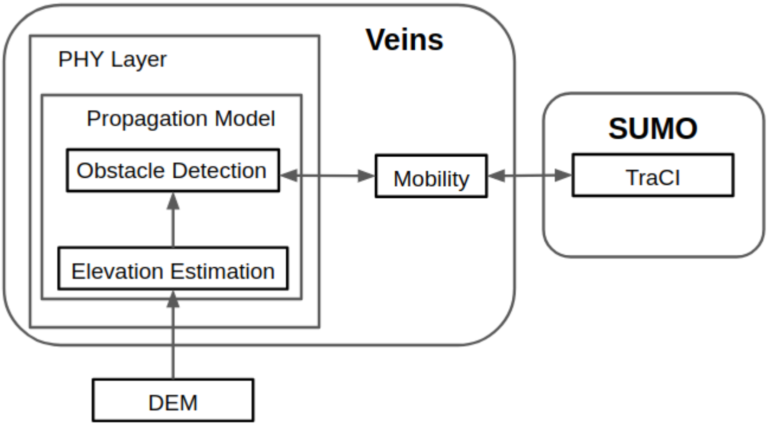


Figure 6.3: Simulation Architecture.

6.3.1 Simulation Architecture

In order to validate our model, the simulation environment is modified to resemble the same environment used in our previous testbed experiments. Since the experiment took place in the outskirts of Casinos, Spain, which is a rural areal, the map corresponding to that particular location was taken from Open Street Map (OSM) and imported into SUMO. The location was far from the urban area, and so the interference in the 5.8 GHz band was minimal. In that particular location, free spaces like valleys or hilly terrains are present, as we can see in the aerial photo shown in Fig. 5.3, which evidences that the experiment's location represents a typical rural area. In particular, the road is running downhill from about 400 meters above the sea level (starting point) to about 320 meters above sea level (destination point). As presented in Fig. 5.5, we can trace a LOS point between the transmitter at the starting point to the receiver at the destination point. Depending on the altitude of the transmitter, the worst-case line-of-sight conditions are expected when the vehicle is located about 900 meters away from the starting point, situation where signal obstructions exist. In particular, communications experienced Non-Line-Of-Sight (NLOS) conditions.

The map of Casinos that was imported from OSM is used to generate the ground vehicle traffic using SUMO. The OSM files imported were modified so that the car used for testing is following the same trajectories as in the real experiments. Overall, the car's path is more than 3 kilometers long. As for

Table 6.2: Simulation Parameters for UAV-to-Car Communications.

Parameters	***Values***
Frequency Band	5 GHz
Transmitter Antenna Gain	5 dBi
Transmission Power	200 mW
Packet Size	1400 Bytes
Packet Rate	10 Hz
Transmission Rate	6 Mbps
Transmitter Altitude	40 m, 100 m
Message Type	BSM

the UAV, it moves freely in the air, meaning that mobility is independent of the map, being defined directly in OMNeT++ without the use of SUMO.

In summary, a static UAV and a moving car are used in the scenario, thereby resembling the testbed experiments used as reference.

6.3.2 Simulation Settings

To simulate our experiments, we have used the map imported from Open Street Map (OSM) [HW08], which was then integrated into SUMO to manage the vehicle's mobility. This way, to mimic the real mobility pattern, the car in the simulation can have the same trajectory and speed as in the real testbed. The Veins simulator is used to mimic real vehicular communications in the simulation.

The communications between UAV and car rely on the ad-hoc mode, and the UAV will act as the data source by generating UDP packets and broadcasting them into the network. The car will be moving away from the UAV that sends the packets. Both UAV and car record their own geographic location periodically.

The packets transmitted into the network, at a rate of 10 Hz, are Basic Safety Messages (BSMs), resembling the same scenario as in the testbed experiments that are similar to those of typical DENM [ETS10a] in the context of vehicular communications. The parameters set in the simulation are specified in Table 6.2. The simulation was performed using a PC with Intel Core i7-4790 @ 3.60GHz x 8 and 8 GB RAM.

In this simulation, we also include our implemented module with the different models for path loss to provide accurate 3D communications simulation. The module implements the diffraction of the terrain and can obtain information about the elevation of the terrain, which can then be used to calculate the signal attenuation. For comparison purposes, we will also perform an experiment considering 2D communications only. In that case, the module that calculates the elevation as an obstruction will not be used. The comparison of these two methods will be presented in the following section.

The simulation is composed of a car moving back and forth, non-stop, following the same path for five consecutive times. During that period, the car will receive packets that are sent by the UAV. The packets that are sent are BSMs, which are transmitted periodically until the simulation stops. The number of BSMs circulating in the network are recorded for statistical purposes.

While recording the flow of the packets, the simulation tool (OMNeT++) also records the location of the nodes in the network. The location that is obtained in the simulation is either in cartesian coordinates or real geocoordinates. Such information will be useful to define the results, and for data analysis. The cartesian coordinates correspond to the location of the node in OMNeT++ and SUMO (in x,y). The real geocoordinates correspond to the location of the node in the real world. Thanks to the TraCI module inside OMNet++, we can have the two types of coordinates seamlessly.

6.3.3 Data Analysis

From the simulation experiments, we can get the data to measure the packet delivery ratio based on distance. The simulation records the geographical location of the transmitter and the receiver whenever the receiver gets a packet. This way, the distance between the two nodes can be obtained directly by comparing the two node locations.

To calculate the packet delivery ratio, we compare the number of packets sent at the transmitter side, and the number of packets received at the receiver side within a predefined distance interval. Both endpoints are analyzed as in discrete simulation, and we can observe events occurring throughout time for every node. By considering the geographical information obtained from simulation, we are able to compare and calculate how many packets are sent and received.

In the simulation that includes the elevation profile, potential factors such as height can affect the communication's performance. Hence, we have chosen to vary the altitude of the transmitter (which in this case is the UAV). So, the LOS probability is affected by the UAV's altitude. Notice that, in general, a low altitude results in a higher probability of finding obstacles. We have considered two altitudes for the UAV: 100m and 40m. In this paper, we will differentiate these two scenarios that involve different UAV flight altitudes. For the scenario where the UAV is flying at 100m, we will name it as Scenario 1. The other scenario, that involves the UAV flying at 40m, is called Scenario 2.

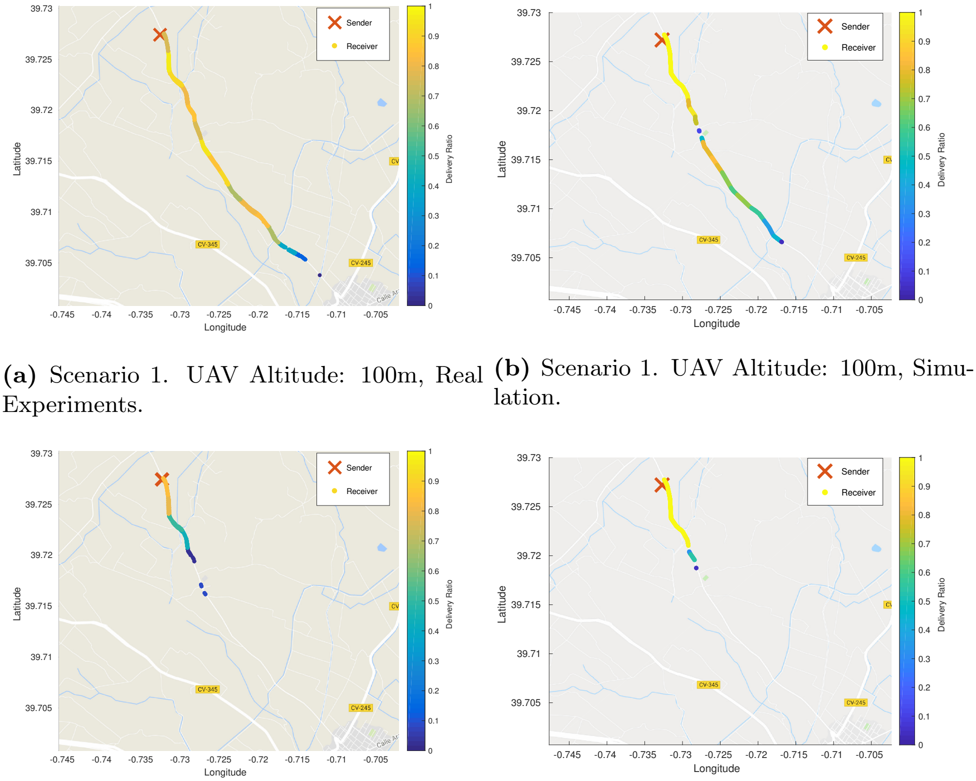
6.4 Simulation Results

The goal of this section is to compare and validate the results obtained using the simulation architecture that includes our 3D communications model, with the results obtained from the real testbed experiments. In all the simulation experiments performed in this section we use the NASA's DEM data and the single-knife edge diffraction model. This diffraction model is used since our idea is to analyze the impact of including the 3D communications, hence the simplest diffraction model is used here. We will study the impact of using other diffraction models in Section 6.6.

Firstly, in Fig. 6.4 we can see the packet delivery ratio using heat maps for better visual description. The heat map is generated from the geographical information retrieved from the experiments. The points on the heat map indicate successful packet reception locations. Based on the packet delivery ratio, the color gradually changes from dark to light, denoting the ratio from low to high.

The location of the received packets in the heat map in Fig. 6.4 represent the exact location of the scenario for both testbed and simulation experiments. When comparing Fig. 6.4a with Fig. 6.4b, where the latter uses our 3D communications model, we can clearly observe that the reception event locations are quite comparable. In more detail, and for this scenario where the UAV flew at 100 meters, the heat map indicates that the points from the starting point to the destination point are associated to decaying packet delivery ratio values. In the real testbed, we can see on the heat map that some points are recorded for higher distances, despite representing very low values. In comparison, in the middle of the car's trajectory in the scenario, there is a drop in the delivery ratio in both simulation and testbed experiments, although this drop is more significant in the case of simulation.

The results when the UAV is flying lower (at an altitude of 40 meters) are also very similar. The points recorded on both simulation and real testbed show



(a) Scenario 1. UAV Altitude: 100m, Real Experiments.

(b) Scenario 1. UAV Altitude: 100m, Simulation.

(c) Scenario 2. UAV Altitude: 40m, Real Experiments.

(d) Scenario 2. UAV Altitude: 40m, Simulation.

Figure 6.4: Comparison of Heatmaps from Real Experiments and Simulation.

only half of its trajectory, as seen in Fig. 6.4c and in Fig. 6.4d. The ratio recorded in the real testbed has a lower value for shorter distances. The color is darker at certain locations, as observed in the heat map. On the other hand, in the simulation the color is lighter, indicating a higher packet delivery ratio. However, both the real testbed and the simulation recorded a very low value at the end of the path, where the car can still receive packets. In that case, the LOS is starting to be obstructed at that particular location.

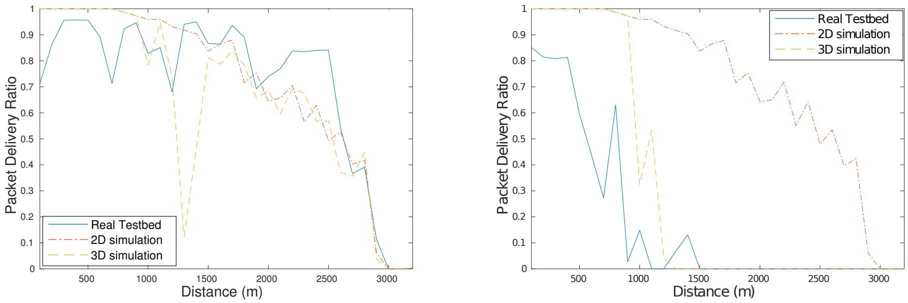
In order to have a more detailed look at the results, we represent the packet delivery ratio depending on the distance between the car and the UAV in Fig. 6.5. The figure shows results for both scenarios (when the drone is flying at

40m and 100m), and using the same parameter settings. To have a better comparison and validate our model, we have also tested using the same parameters and the same scenario when only considering the 2D communications model (it is worth remembering that this model neglects the terrain elevation information). For each figure, the plots include the curves for the simulation taking into account the elevation information (our 3D simulation model), the simulation using only a 2D communications model (no elevation information), and also the results obtained from the real testbed experiments for the validation of the model.

In the scenario where the drone is flying at 100 meters, it attains LOS in most cases, since the drone was high enough above the terrains. Thus, the difference when including the elevation information is not noticeable. A slight difference is shown when the distance is about 1.3 kilometers. A drop of the packet delivery ratio is indicated in the 3D simulation as a result of having the DEM indicating a high elevation in that specific location. This is due to the lack of accuracy of the DEM for this specific scenario. Since the resolution of NASA's DEM is 90 meters, there might be a chance that, at that particular point, there is a change of elevation within 90 meters of the DEM tile. However, for the other scenario, it is found to be adequate.

The case when the UAV flew lower (at 40 meters) leads to NLOS conditions. The real impact of using the elevation information in our proposed 3D simulation is noticeable here. The outcome would greatly differ from the real testbed results had we not included the elevation information in the propagation model. Contrarily, by performing the 3D simulation, the communications are clearly affected by the high elevation of the terrain. If we merely perform a 2D simulation, at a distance of 2 kilometers, the packet delivery ratio remains quite high (more than 60%). On the other hand, no communications were taking place at that particular distance in the real tested experiments. The simulation including 3D communications showed that, at that distance, the communication was obstructed by the terrains since it includes the elevation information. This proves that an acceptable level of realism is reached when using our simulation model as a result of including the proposed terrain model.

Finally, in order to have more detailed simulation results, and to assess how packet reception is affected by the signal quality, we have also recorded the Received Signal Strength Indicator (RSSI) value in our simulation. Figure 6.6 shows the RSSI values based on the distance between the transmitter (the car) and the receiver (the UAV). These figures include both the case when the UAV flew at 100 meters (Fig. 6.6a), and at 40 meters (Fig. 6.6b). In each figure, we have included the results for 3D communications using the red line. The



(a) Scenario 1. UAV Altitude of 100m.

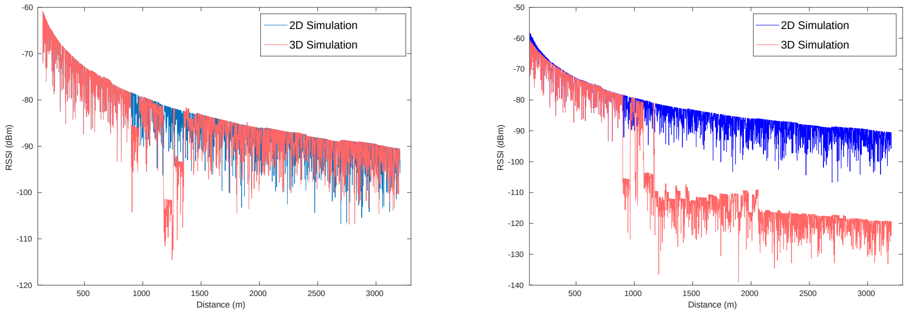
(b) Scenario 2. UAV Altitude of 40m.

Figure 6.5: Packet Delivery Ratio vs. Distance for both scenarios.

other line included, appearing in blue, represents the simulation results where we did not include the elevation information to define the signal quality, or in other words, we did not consider 3D communications.

Fig. 6.6a, which represents the results from the simulation when the UAV flew at 100 meters, shows the reason why packets are lost at certain distances. The simulation spotted that, when the distance is between 1200 and 1300 meters, there is a drop on the signal obtained at the receiver's side. The RSSI indicates the lowest value, at about -115 dBm, at a distance of 1300 meters. At that particular location, the signal path was obstructed by a higher terrain, resulting in a greater loss of signal strength. However, for distances beyond 1400 meters, the signal strength indicator showed a similar value for both when the simulation considers the 3D communications model, and the one with only the 2D communications model.

On the other hand, the results when the UAV flew lower (40 meters), shown in Fig. 6.6b, denote that, for most cases, the signal path experiences NLOS conditions. In more detail, compared to the packet delivery ratio results, we can see that the RSSI is much lower when the 3D communications model is used. When the distance is about one kilometer, the signal strength drops to about -110 dBm compared to the signal strength when using the 2D communications model, where the signal remains higher at -80 dBm. Another significant difference, this time in terms of packet delivery ratio, is detected at 1300 meters, since no packets were received at the receiver's side at that location. This is also confirmed by the results regarding the RSSI. The figure shows that, when we consider 3D communications, the RSSI values range from -110 dBm to -130 dBm. Instead, when the 2D communications model is used, the RSSI ranges



(a) Scenario 1. UAV Altitude of 100m.

(b) Scenario 2. UAV Altitude of 40m.

Figure 6.6: Received Signal Strength indicator vs. Distance for both scenarios.

from -80 dBm to -105 dBm. These data further confirm that elevation information significantly affects the signal path since, at some location, the simulation spotted some terrains that act as obstructions to the signal propagation.

6.5 Impact of Digital Elevation Models on Performance

In this section, we compare the two elevation models included in the simulation framework. The default elevation model is the NASA's DEM. Using this DEM in the simulation we can record the altitude of the car when it follows the route based on the geocoordinates obtained from the simulation.

For the sake of validating the effectiveness of using the appropriate DEM file, we will also compare the results using the NASA's DEM with the ones obtained using Google's DEM (as described in subsection 6.2.1). The goal is to compare its overhead in terms of execution time, and also its performance results in terms of Packet Delivery Ratio and RSSI.

In addition, since in our previous real experiments we have obtained the trace detailing the elevation of the terrain, we will use this real trace as a comparison to the obtained results in the simulation. In this case, the car acting as data receiver records its location, providing the geocoordinates used to retrieve the elevation value; such value will be added to the height of the car in the simulation, thereby obtaining the correct altitude. We will use three methods: first, we use the altitude recorded in the real experiment for simulation, getting the Packet Delivery Ratio and the RSSI. Second, based on the vehicle's geocoordinates, we will retrieve the elevation information using NASA's DEM. In the third method, we will retrieve the elevation information using Google's DEM.

6.5.1 Impact of Digital Elevation Models on Simulation Time

As an effort to select the most efficient DEM for simulation, we have compared the execution time required by the different methods for retrieving the elevation information. In particular, we will compare five methods that run our 3D communications model in simulation.

The results of the execution time are shown in Fig. 6.7. The first bar (the blue one), is the real experiment itself. It represents how much time is needed to run the experiment in the field. The second one, represented by the yellow bar in the figure, is the simulation using the real trace obtained from the previous real experiment to gather information about the car's altitude as it moves along the trajectory. The third one, represented as a green bar, also works by retrieving the car's altitude, and the signal path from the UAV to the car is determined using the NASA's DEM. The fourth one, represented as a purple bar, uses the same method as the latter, but retrieves elevation data using the Google Maps Elevation API (Google's DEM). Concerning the last one, represented as a red bar, refers to the simulation experiment when 3D communications are not considered, meaning that no elevation information is included, and it is included as a reference.

Our assessment of the execution time is made by determining the time required for measuring the communications between the UAV and the car as the latter moves along the trajectory from the starting point (where the UAV is located) to the destination point, right before it makes a U-turn, in both the simulation and testbed experiments.

As shown in Fig. 6.7, the real experiment itself takes about four minutes, for a car speed of about 40 km/h. However, if we execute a simulation using the same parameters using the NASA's DEM for the elevation information, it takes about 19 minutes to execute. This is expectable since retrieving the elevation information for the signal path requires significant computation time. In addition, when calculating the knife edge diffraction, the machine executing the simulation also takes more time. The same thing occurs when we use the real trace as a reference for the altitude of our receiver; in this case, the execution time is slightly lower than the NASA's DEM one, as it avoids retrieving elevation information.

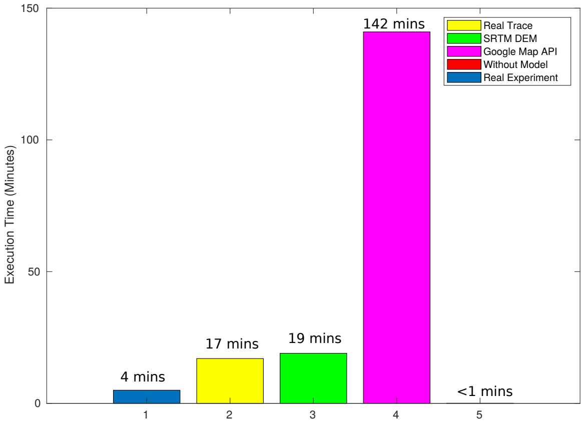


Figure 6.7: Execution Time for the different Elevation Models.

The most time-consuming simulation approach is using the Google Maps Elevation API. Since the elevation information is not stored locally, it has to check online the terrain height values throughout the simulation experiment. Thus, not only do we have a slower execution time associated to connecting to the Internet, but also the Google Maps Elevation API itself consumes a significant time since it represents a very large database of elevation information. Thus, in our experiment, the time consumed was more than 2 hours for one run. This evidences that using Google's DEM is quite inefficient in terms of execution time. Compared to using the NASA's DEM, the gap is huge. Another thing to keep in mind is the exclusivity of retrieving the elevation using Google Maps Elevation API. The request is limited to the public, and it is not adequate for simulation if this approach is adopted by a broader number of users.

Finally, and as a reference of the overhead incurred when using 3D communication, the execution time is lower than one minute when only 2D communications are considered, since this approach does not need more computation time to retrieve the elevation information. This simulation experiment was run with the same parameters as before. This shows that, even though in terms of computation time 2D communications are faster, the degree of realism is far worse, as shown in the previous section.

6.5.2 Altitudes Recorded by the Different Elevation Models

We used information regarding the latitude, longitude, and altitude of the car (receiver), for comparison with the retrieved elevation information using the other DEMs. Thus, based on the latitude and the longitude of the car, we can retrieve the elevation information using both DEMs. The height of the car is added to the elevation obtained so that it has the absolute altitude for that particular location.

In Fig. 6.8, we present a comparison of the altitude of the car from the starting point to the final point, which is calculated based on the distance to the sender (UAV). The blue line represents the trace of the car's altitude obtained in the real testbed. The purple line in the figure represents the car's altitude obtained via NASA's DEM, where the elevation is obtained using the latitude and the longitude of the car's trajectory in the real testbed. The yellow line represents the results using Google's DEM, using the same method as for the previous one.

The results in the figure show that Google's DEM has the closest values to the ones obtained from the real trace, due to its higher resolution. On the contrary, the results for NASA's DEM show a more discretised curve, not being as continuous as the line produced by Google's DEM. This is particularly noticeable for distances between 1300 meters and 1700 meters; in this range, the line produced by the NASA's DEM is a steady line, showing that there was no change in the altitude, meaning that the Earth's surface is assumed to be flat in that location. On the other hand, the real trace showed that there is a slight drop of altitude in that slice of distance. Despite Google's DEM showed a higher value of altitude, it has the same trend as the line from the real trace, unlike the one for NASA's DEM, that has the same altitude value (400 meters). From the figure, we can see that the altitudes produced using Google's DEM are more accurate than NASA's ones, as they have values that better resemble the altitudes recorded in the real trace.

6.5.3 Performance Using Different Elevation Models

Since we have understood the differences between using different elevation models in terms of execution time and accuracy when determining the altitude, we now proceed to compare them in terms of packet delivery ratio performance. We compare the three methods that use different elevation models (real trace, NASA's DEM and Google's DEM). However, the main difference arises depending on the approach taken to retrieve the altitude of the receiver, in this

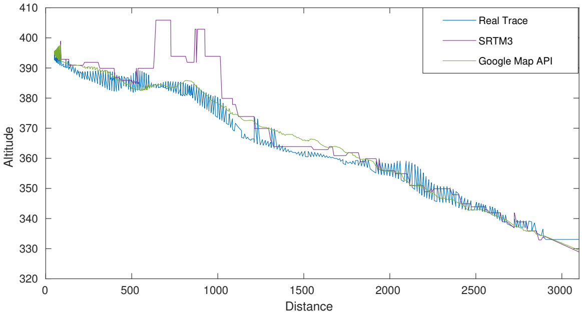


Figure 6.8: Comparison of Altitudes Recorded.

case, the moving car. When the simulation is running, the car has to record its geolocation coordinates in order to define the signal path and the obstructions in between. For the elevation information of the signal path in the simulation, for the three experiments, we will keep using NASA's DEM.

Fig. 6.9 shows the results achieved, where the blue curve represents the result of running the 3D simulation using NASA's DEM. The yellow curve represents the result of running the 3D simulation using Google's DEM as a reference for obtaining the receiver's altitude. Finally, the red line represents the case when the receiver's altitude takes the data directly from the real trace.

The figure shows that the curve produced by Google's DEM has values closer to the curve associated with the real trace. As for the curve produced by NASA's DEM, the trend is similar to the curve from Google, although the actual values differ when compared to both the real trace and Google's DEM. This evidences that Google's DEM, due to its greater resolution, allows obtaining better estimations of the packet delivery rate than NASA's DEM.

For a more detailed comparison, we have also tested the simulation to get the RSSI value to prove the signal strength when receiving packets from the sender. We have tested the three methods using the same parameters to get the RSSI in dBm.

The results are shown in Fig. 6.10. Google's DIM curve is closer to the one obtained using the real trace. We can understand that from the previous figure (see Fig. 6.9), in terms of both RSSI and packet delivery ratio, as the values obtained are quite comparable. However, the trend is found to be more similar to one achieved using NASA's DEM. On the contrary, NASA's DEM presents the greatest difference for a distance between 1000 meters and 1300 meters.

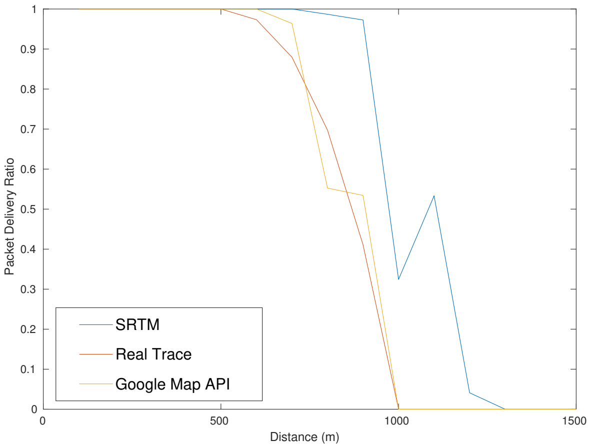


Figure 6.9: Comparison of Elevation Models on the Packet Delivery Ratio.

On the other hand, both the simulation using Google's DEM and the real trace show a drop of the RSSI value, which in turn results in packets losses, presenting a low packet delivery ratio.

Summing up, although the results obtained using Google's DEM are slightly more accurate than the ones obtained using NASA's DEM, its high execution time and limited data access make Google's DEM an unfeasible option when targeting efficient simulations.

6.6 Impact of Diffraction Models on Performance

In this section, we compare the different path loss models that are included in the simulation framework as described in subsection 6.2.2. The first path loss model is a single knife edge model, where we have calculated the highest obstacle (or knife) that is present within the signal path between the transmitter and the receiver. The second path loss model is the Deygout model, and the last one is the Bullington model. The comparison is made between the three models implemented in the scenario, where the drone is flying at 40m, that is, the scenario with obstacles that lead to NLOS conditions.

We will compare the performance of these three models in terms of how much time is needed to execute a particular path loss model on our host. Another

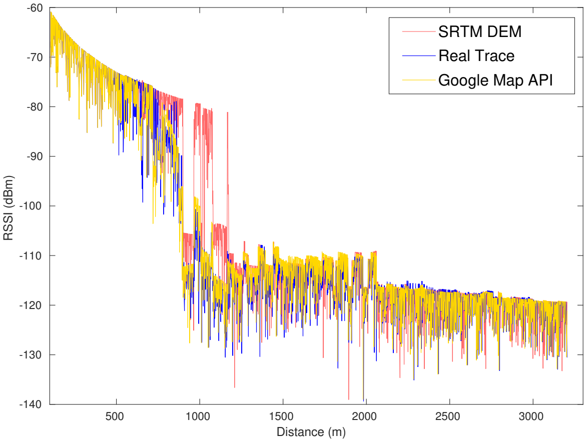


Figure 6.10: Comparison of Elevation Models on RSSI.

comparison is the packet delivery ratio obtained when running the simulation using these models. In addition, for a more detailed comparison, we have also added the RSSI values that are produced using these models.

6.6.1 Performance when using different Path Loss Models

Firstly, we compare the different path loss models (or diffraction models) based on the packet delivery ratio at the receiver's side. That is, the ratio between the packets arriving to the receiver (the car), and the packets sent from the

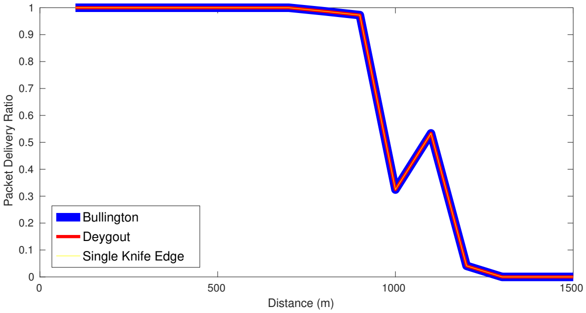


Figure 6.11: Comparison of Diffraction Models on the Packet Delivery Ratio.

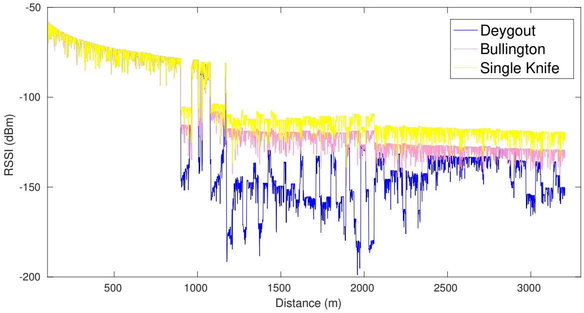


Figure 6.12: Comparison of Diffraction Models on the RSSI.

transmitter (the UAV). As a reference, the single knife edge diffraction model is used, as it has already been compared with the results from the real experiment. The results are shown in Fig. 6.11, where we can see that the outcome is basically the same. This occurs since the three diffraction models experience the same obstacle within the signal path. However, the calculation of the signal strength is expected to have different results for these three models.

Although having the same outcome in terms of packet delivery ratio, we cannot expect the signal strength at the receiver's side to record the same value. To prove this, we have also evaluated the RSSI for each diffraction model.

Fig. 6.12 represents the comparison of the RSSI values from the outcome of the simulation using the three different diffraction models, where we can see that the signal strength for the three diffraction models records a low value after 1300 meters. This is the reason why, at the receiver's side, no packets are successfully delivered. The difference between the three is in fact on how low RSSI values are when detecting an obstacle. In the figure, we can understand that, when using the Deygout model, the RSSI outcome is the lowest compared to the other two. On the other hand, if we only include the single knife edge diffraction model, the RSSI value is the highest. This occurs because the signal loss is affected by a single obstacle, instead of multiple obstacles. In the Bullington and Deygout models, since they are multiple knife edge diffraction models, we obtain lower RSSI values. Thus, the single knife approach seems to be the least adequate one, as it neglects the other obstacles that might be present in the signal path, and that affect the signal quality between the transmitter and the receiver.

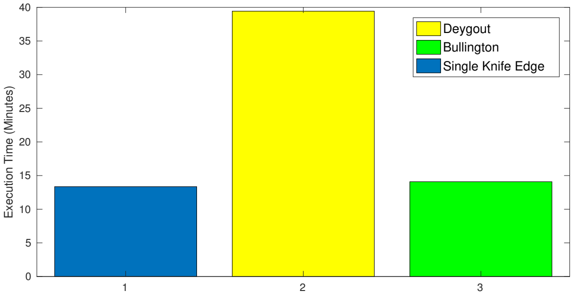


Figure 6.13: Execution time for the different diffraction models.

6.6.2 Impact of Path Loss Models on Simulation Time

To complete our analysis, we now compare the execution time to perform simulations when using different path loss models. In order to have a view of which path loss model is more efficient in terms of execution time, we compare the three models using our 3D simulation framework.

Fig. 6.13 shows that, when using the Deygout diffraction model, the execution time is longer than for the other two (about 40 minutes). On the other hand, the simulation using the single knife edge and Bullington diffraction models show very similar values. In the case of the single knife edge, the simulation takes about 13 minutes, whereas for the Bullington model it takes about 14 minutes. We can conclude from the figure that the Bullington model offers the best trade-off between execution time and accuracy.

Summing up, the best option is to use the Bullington model, given its slightly higher accuracy and low execution time. A comparison of all the results obtained is presented in table 6.3 for the sake of completeness.

6.7 Summary

To make simulation experiments involving UAV-to-Car communications more realistic, we have proposed a simulation framework that includes 3D terrain profiling. In fact, when characterizing the communications involving UAVs and ground vehicles, we found that standard approaches defined by a planar space are inadequate, as communications are affected by the presence of terrains hav-

Table 6.3: Summary of Results Based on Methods Used in the Simulation.

Diffraction Method	Elevation Model	Deployment	PDR Accuracy	RSSI Accuracy	Simulation Time
None (2D)	None (2D)	very easy	not accurate	not accurate	very fast
Single knife	SRTM	easy	less accurate	less accurate	fast
Single knife	Google Elevation API	very complex & limited	more accurate	more accurate	very slow
Deygout	SRTM	very complex	more acccurate	most accurate	slow
Bullington	SRTM	complex	more accurate	more accurate	moderate

ing certain height or elevation; in such cases, 3D space enabled communications should be considered instead.

We can retrieve information about the terrain profile thanks to the DEMs made available by providers like NASA or Google. In fact, terrain information is critical to calculate the attenuation of the signal for this kind of communication. Therefore, in this chapter, we proposed a simulation framework that provides a specific propagation model able to model diffraction effects caused by hilly terrains, and we have implemented our model in the OMNeT++ simulation tool. Our model is then validated by performing several simulation experiments to analyze the packet delivery success of UAV-to-Car communications. The simulation results obtained are then compared with those obtained from real experiments. We find that, if properly tuned, simulation experiments show results that are comparable to the ones from the real testbed experiments for both scenarios tested (varying UAV flight altitude).

Chapter 7

Conclusions, Publications, and Future Works

7.1 Conclusions

In this thesis we studied the deployment of vehicular communications involving both ground and aerial vehicles under the VANET and FANET paradigms. Both these types of communications fall under C-ITS, whose overall goal is to enhance traffic environments. We have investigated the use of this technology in real experiments followed by empirical studies, and we have made an effort to bring testbed results into simulation so as to simulate vehicular scenarios in the most realistic way possible.

In more detail, we have developed a safety-related C-ITS application that makes use of VANETs in order to gain a better perspective on how C-ITS applications can benefit users. We found that the specific environment features can affect the communications quality experienced by the application itself. For this reason, we have then performed an in-depth study of intersections involving vegetation and buildings as obstacles to study in more detail their impact on signal quality. Based on our results we have discovered that other approaches should be sought when the presence of obstacles is inevitable. Hence, based on this observation, we proposed an architecture and a model that involves UAVs as relays to enhance vehicular communications. To ensure the quality of the communications, an empirical study was performed. However, since real experiments involve a lot of resources, simulation studies seem to be more feasible to perform and to conform more generic scenarios. Yet, to have an appropriate degree of realism in simulations, an adequate model must first be proposed. Based on this finding, we have proposed an extension for a simulation tool that considers the impact of real-life restrictions, in this case involving three dimensional communication issues when addressing UAV-to-car links. The model considers the elevation information as a reference, and different diffraction techniques were imported to the simulation framework.

Below we briefly summarize the most relevant contributions of this thesis:

- A survey of C-ITS applications, vehicular communications implementations, and channel modeling. The literature review includes a study of VANET, FANET, and smartphone-based safety applications for vehicular communications. An overview of channel modeling techniques for vehicular communications is also presented.

- A smartphone-based application for safety purposes (Messiah). This application is an extension for OSMAnd that aims to advertise the location of emergency vehicles, like ambulances, police cars and fire brigades, to surrounding civilian cars by relying on the VANET paradigm. Messiah is a navigation application that can display emergency notifications to the user. Hence, the driver of the car receiving these on-screen notifications can take navigation decisions with higher situational awareness. The application was tested in an urban area to validate its functionality.

- An empirical model for vehicular communications at intersections. The model was based on an empirical study addressing the impact of intersections in urban and rural areas in terms of communications performance between cars. By merely changing one parameter, the model can fit up to three types of intersection, with various degree of obstructions.

- Experimental characterization of UAV-to-car communications and an architecture for UAV-to-car communications measurement. Real experiments were performed that involved real cars and UAVs. A set of measurements were performed by varying the UAV altitude, antenna orientation, and antenna location. An architecture for making measurements, and an empirical model were proposed based on the tests made.

- An extended simulation framework for UAV-to-car communications. A more generic model, which is more adequate for a wide range of 3D communication scenarios, was proposed as a module that extends the OMNeT++ simulation framework. The simulation model considers the impact of the terrain profile, as well as the implementation of specific propagation models.

Having accomplished all of our initially defined goals, we consider that the ultimate purpose of this thesis has been achieved successfully. Hence, we conclude this dissertation.

7.2 Publications Related to the Thesis

This section lists the publications that have been produced as a result of this thesis, as well as some other collaborations and non-directly related publications we published during this time. Overall, during the PhD period, we published 4 **ISI/JCR-indexed** journal articles, and 4 **CORE-ranked** papers in international conference proceedings.

7.2.1 Journal Articles

- S. A. Hadiwardoyo, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni. “An Intelligent Transportation System Application for Smartphones Based on Vehicle Position Advertising and Route Sharing in Vehicular Ad-Hoc Networks”. In: *Journal of Computer Science and Technology* 33.2 (2018), pp. 249–262 [JCR Q3, IF: 0.956]

- S. A. Hadiwardoyo, A. Tomás, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, and P. Manzoni. “Empirical Study and Modeling of Vehicular Communications at Intersections in the 5 GHz Band”. In: *Mobile Information Systems* 2017 (2017) [JCR Q2, IF: 1.462]

- S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J. C. Cano, and P. Manzoni. “Experimental characterization of UAV-to-car communications”. In: *Computer Networks* 136 (2018), pp. 105–118 [JCR Q1, IF: 2.516]

- S. A. Hadiwardoyo, C. T. Calafate, J.-C. Cano, Y. Ji, E. Hernández-Orallo, and P. Manzoni. “3D Simulation Modeling of UAV-to-Car Communications”. In: *IEEE Access* (2019) [JCR Q1, IF: 3.557]

7.2.2 Conference Proceedings

- S. A. Hadiwardoyo, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni. “An Android ITS driving safety application based on vehicle-to-vehicle (V2V) communications”. In: *Computer Communication and Networks (ICCCN), 2017 26th International Conference on.* IEEE. 2017, pp. 1–6 [CORE A]

- S. A. Hadiwardoyo, A. Tomás, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, and P. Manzoni. “On the impact of urban intersection characteristics in vehicular to vehicular (V2V) communications”. In: *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International.* IEEE. 2017, pp. 452–457 [CORE B]

- S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, and P. Manzoni. “Evaluating UAV-to-Car Communications Performance: Testbed Experiments”. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA).* IEEE. 2018, pp. 86–92 [CORE B]

- S. A. Hadiwardoyo, C. T. Calafate, J.-C. Cano, Y. Ji, E. Hernández-Orallo, and P. Manzoni. “Evaluating UAV-to-Car Communications Performance: From Testbed to Simulation Experiments”. In: *Consumer Communications & Networking Conference (CCNC), 2019 16th IEEE Annual.* IEEE. Jan. 2019 (Best Paper Runner Up) [CORE B]

7.3 Funding and Projects

The thesis is related to several projects and funding, namely:

- "Smart@CarPhone: Toward Seamless Smartphone and Vehicle Integration to Connect Drivers with Sensors and the Environment in a Holistic Service-Oriented Architecture" (National Project) which was funded by the Kingdom of Spain through the Ministerio de Economía y Competitividad, Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2014, Spain, under Grant TEC2014-52690-R.

- Ayudas para Contratos Predoctorales para la Formación de Doctores 2015 (Predoctoral Fellowship), which was co-funded by the European Union through the European Social Fund (ESF) and the Kingdom of Spain

through the Ministerio de Economía y Competitividad, Programa Estatal de Promoción del Talento y su Empleabilidad del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016, Spain, under Grant BES-2015-075988, which also financed the research stay in St. Petersburg, Russia (at St. Petersburg Electrotechnical University "LETI"), between December 2018 and February 2019.

- National Institute of Informatics (NII) Internship Program for financing the research stay in Tokyo, Japan, between March and September 2018.
- Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers JP16H02817 and JP18KK0279 for financing one of the publications.

7.4 Future work

The results accomplished in this doctoral thesis represent an advance in the state of the art of research in the area of vehicular networking, addressing both ground communications and air-to-ground communications, and its applications for Intelligent Transport Systems. We consider that the contributions made offer a starting point that opens a wide range of possibilities regarding research work. In detail, we believe that this thesis can be extended through the following lines of work:

- To assess the impact of intersections on V2V communications in simulation platforms, in order to model the real testbed experiments in a simulated environment.
- To extend the real testbed experiment for measurements involving more than two nodes, using several UAVs and cars.
- To extend the simulation framework for UAV-to-car communications in order to simulate the communication in urban areas. The simulation model should consider the building heights using the digital surface models as a reference.
- To develop an ITS-based application that relies on UAV-to-car communication to achieve a full C-ITS implementation involving ground vehicles and UAVs.
- To find an efficient 3-D placement of UAVs that can provide a maximum coverage for ground vehicles in environments with irregular terrains.

Acronyms

AP Access Point 19

API Application Programming Interface 16

C-ITS Cooperative Intelligent Transport System 7

CAM Cooperative Awareness Message 14

CCC Car Connectivity Consortium 16

D2D Device-to-Device 36

DEM Digital Elevation Model 95

DENM Decentralized Environmental Notification Message 14

DSRC Dedicated Short Range Communications 9

ETSI European Telecommunications Standards Institute 13

FANET Flying Ad-hoc Network 2

GPS Global Positioning System 17

GUI Graphical User Interface 32

ICT Information and Communication Technology 8

IoT Internet of Things 36

ITS Intelligent Transport System 1

LOS Line-of-Sight 11

MANET Mobile Ad-hoc Network 10

NASA National Aeronautics and Space Administration 97

NLOS Non Line-of-Sight 27

OBU On-Board Unit 2

OMNeT++ Objective Modular Network Testbed in C++ 32

OSM OpenStreetMap 38

RSSI Received Signal Strength Indicator 112

RSU Road Side Unit 9

SRTM Shuttle Radar Topography Mission 97

UAV Unmanned Aerial Vehicle 9

V2I Vehicle-to-Infrastructure 9

V2X Vehicle-to-Everything 1

V2V Vehicle-to-Vehicle 1

VANET Vehicular Ad-hoc Network 2

VN Vehicular Network 9

WAVE Wireless Access in Vehicular Environment 105

Bibliography

[AAAAKL18] M. A. Al-Absi, A. A. Al-Absi, T. Kim, and H. J. Lee. “An environmental channel throughput and radio propagation modeling for vehicle-to-vehicle communication.” In: *International Journal of Distributed Sensor Networks* 14.4 (2018), p. 1550147718772535 (cit. on p. 29).

[ABVW16] B. Aygun, M. Boban, J. P. Vilela, and A. M. Wyglinski. “Geometry-based propagation modeling and simulation of vehicle-to-infrastructure links.” In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. IEEE. 2016, pp. 1–5 (cit. on p. 30).

[AEO15] N. Akhtar, S. C. Ergen, and O. Ozkasap. “Vehicle mobility and communication channel models for realistic and efficient highway VANET simulation.” In: *IEEE Transactions on Vehicular Technology* 64.1 (2015), pp. 248–262 (cit. on p. 29).

[AFF16] M. Alam, J. Ferreira, and J. Fonseca. “Introduction to intelligent transportation systems.” In: *Intelligent Transportation Systems*. Springer. 2016, pp. 1–17 (cit. on p. 9).

[AHKJ14] A. Al-Hourani, S. Kandeepan, and A. Jamalipour. “Modeling air-to-ground path loss for low altitude platforms in urban environments." In: *Global Communications Conference (GLOBECOM), 2014 IEEE*. IEEE. 2014, pp. 2898–2904 (cit. on pp. 13, 31).

[ANM95] G. Athanasiadou, A. Nix, and J. McGeehan. "A ray tracing algorithm for microcellular wideband propagation modelling." In: *Vehicular Technology Conference, 1995 IEEE 45th*. Vol. 1. IEEE. 1995, pp. 261–265 (cit. on p. 26).

[ASKT15] T. Abbas, K. Sjöberg, J. Karedal, and F. Tufvesson. "A measurement based shadow fading model for vehicle-to-vehicle network simulations." In: *International Journal of Antennas and Propagation* 2015 (2015) (cit. on p. 29).

[Bar04] R. Barr. "Swans-scalable wireless ad hoc network simulator." In: *User Guide* (2004) (cit. on p. 33).

[BBEK11] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. "Sumo–simulation of urban mobility." In: *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*. Vol. 42. 2011 (cit. on pp. 33, 105).

[BCCMPR17] A. Bujari, C. T. Calafate, J.-C. Cano, P. Manzoni, C. E. Palazzi, and D. Ronzani. "Flying ad-hoc network application scenarios and mobility models." In: *International Journal of Distributed Sensor Networks* 13.10 (2017) (cit. on p. 11).

[BLJL10] A. Böhm, K. Lidström, M. Jonsson, and T. Larsson. "Evaluating CALM M5-based vehicle-to-vehicle communication in various road settings through field trials." In: *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*. IEEE. 2010, pp. 613–620 (cit. on p. 27).

[BMSMI12] C. T. Barba, M. A. Mateos, P. R. Soto, A. M. Mezher, and M. A. Igartua. "Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights." In: *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE. 2012, pp. 902–907 (cit. on p. 8).

[BSL18] S. Baidya, Z. Shaikh, and M. Levorato. "FlyNetSim: An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot." In: *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems.* ACM. 2018, pp. 37–45 (cit. on p. 36).

[BST13] I. Bekmezci, O. K. Sahingoz, and Ş. Temel. "Flying ad-hoc networks (FANETs): A survey." In: *Ad Hoc Networks* 11.3 (2013), pp. 1254–1270 (cit. on p. 11).

[Bul47] K. Bullington. "Radio propagation at frequencies above 30 megacycles." In: *Proceedings of the IRE* 35.10 (1947), pp. 1122–1136 (cit. on p. 100).

[CC14] Y. Chen and B. Chamadiya. "Propagation model for vehicle to vehicle LOS communication in foliage scenario." In: *Connected Vehicles and Expo (ICCVE), 2014 International Conference on.* IEEE. 2014, pp. 1120–1125 (cit. on p. 28).

[CCGI+05] L Chaimowicz, A Cowley, D Gomez-Ibanez, et al. "Deploying air-ground multi-robot teams in urban environments." In: *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III.* Springer, 2005, pp. 223–234 (cit. on p. 11).

[Cha79] U Charash. "Reception through Nakagami fading multipath channels with random delays." In: *IEEE Transactions on communications* 27.4 (1979), pp. 657–670 (cit. on p. 25).

[CK99] D. J. Cichon and T. Kurner. "Propagation prediction models." In: *COST 231 final report* (1999), p. 134 (cit. on p. 26).

[DDLW09] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld. "AirShield: A system-of-systems MUAV remote sensing architecture for disaster response." In: *Systems conference,* *2009 3rd Annual IEEE*. IEEE. 2009, pp. 196–200 (cit. on p. 11).

[Dev18] G. Developers. *Google Maps Elevation API*. 2018. URL: `https://developers.google.com/maps/documentation/elevation/intro` (visited on 09/24/2018) (cit. on p. 97).

[Dey66] J. Deygout. "Multiple knife-edge diffraction of microwaves." In: *IEEE Transactions on Antennas and Propagation* 14.4 (1966), pp. 480–489 (cit. on p. 100).

[DFF14] A. Dancu, Z. Franjcic, and M. Fjeld. "Smart flashlight: map navigation using a bike-mounted projector." In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM. Apr. 2014, pp. 3627–3630 (cit. on p. 17).

[DR00] G. D. Durgin and T. S. Rappaport. "Theory of multipath shape factors for small-scale fading wireless channels." In: *IEEE Transactions on Antennas and Propagation* 48.5 (2000), pp. 682–693 (cit. on p. 25).

[DRX98] G. Durgin, T. S. Rappaport, and H. Xu. "Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz." In: *IEEE Transactions on Communications* 46.11 (1998), pp. 1484–1496 (cit. on p. 63).

[ETS09] T. ETSI. *Intelligent transport systems (ITS); vehicular communications; basic set of applications; definitions*. Tech. rep. Tech. Rep. ETSI TR 102 638, 2009 (cit. on pp. 13, 14).

[ETS10a] ETSI. "102 637-3, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of decentralized environmental notification basic service." In: *ETSI, Sophia Antipolis Cedex, France* (2010) (cit. on pp. 14, 47, 48, 78, 108).

[ETS10b] T. ETSI. “102 637-2, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Co-operative Awareness Basic Service.” In: *ETSI, Sophia Antipolis Cedex, France* (2010) (cit. on pp. 14, 48).

[EZL14] E. C. Eze, S. Zhang, and E. Liu. “Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward.” In: *Automation and Computing (ICAC), 2014 20th International Conference on.* IEEE. Sept. 2014, pp. 176–181 (cit. on p. 14).

[FAAK17] W. Fawaz, R. Atallah, C. Assi, and M. Khabbaz. “Unmanned Aerial Vehicles as Store-Carry-Forward Nodes for Vehicular Networks.” In: *IEEE Access* 5 (2017), pp. 23710–23718 (cit. on p. 12).

[FBCE14] R. Frank, W. Bronzi, G. Castignani, and T. Engel. “Bluetooth low energy: An alternative technology for VANET applications.” In: *Wireless on-Demand Network Systems and Services (WONS), 2014 11th Annual Conference On.* IEEE. 2014, pp. 104–107 (cit. on p. 47).

[Fes14] A. Festag. “Cooperative intelligent transport systems standards in Europe.” In: *IEEE communications magazine* 52.12 (2014), pp. 166–172 (cit. on p. 9).

[FRC+07] T. G. Farr, P. A. Rosen, E. Caro, et al. “The shuttle radar topography mission.” In: *Reviews of geophysics* 45.2 (2007) (cit. on p. 97).

[Fri46] H. T. Friis. “A note on a simple transmission formula.” In: *Proceedings of the IRE* 34.5 (1946), pp. 254–256 (cit. on p. 89).

[Gal00] J. P.W.J. C. Gallant. *Terrain analysis: principles and applications.* John Wiley & Sons, 2000 (cit. on p. 95).

[GAVR+18] F. Granda, L. Azpilicueta, C. Vargas-Rosales, et al. “Deterministic Propagation Modeling for Intelligent Vehicle

Communication in Smart Cities." In: *Sensors* 18.7 (2018), p. 2133 (cit. on p. 30).

[GFPG10] E. Giordano, R. Frank, G. Pau, and M. Gerla. "CORNER: a realistic urban propagation model for VANET." In: *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on.* IEEE. 2010, pp. 57–60 (cit. on p. 27).

[GK11] M. Gerla and L. Kleinrock. "Vehicular networks and the future of the mobile internet." In: *Computer Networks* 55.2 (2011), pp. 457–469 (cit. on p. 10).

[GKD18] B. Galkin, J. Kibilda, and L. A. DaSilva. "Backhaul for low-altitude UAVs in urban environments." In: *2018 IEEE International Conference on Communications (ICC).* IEEE. 2018, pp. 1–6 (cit. on p. 31).

[GMU14] S. Grant-Muller and M. Usher. "Intelligent Transport Systems: The propensity for environmental and economic benefits." In: *Technological Forecasting and Social Change* 82 (2014), pp. 149–166 (cit. on p. 15).

[Gol05] A. Goldsmith. *Wireless communications.* Cambridge university press, 2005 (cit. on pp. 23, 24).

[Goz08] J. Gozalvez. "First Google's android phone launched [Mobile Radio]." In: *IEEE Vehicular Technology Magazine* 3.4 (2008), pp. 3–69 (cit. on p. 37).

[Gre15] S. Greengard. "Automotive systems get smarter." In: *Communications of the ACM* 58.10 (2015), pp. 18–20 (cit. on p. 16).

[GSB12] J. Gozálvez, M. Sepulcre, and R. Bauza. "IEEE 802.11 p vehicle to infrastructure communications in urban environments." In: *IEEE Communications Magazine* 50.5 (2012), pp. 176–183 (cit. on p. 27).

[Hat80] M. Hata. "Empirical formula for propagation loss in land mobile radio services." In: *IEEE transactions on Vehicular Technology* 29.3 (1980), pp. 317–325 (cit. on p. 26).

[HCCJHM19a] S. A. Hadiwardoyo, C. T. Calafate, J.-C. Cano, Y. Ji, E. Hernández-Orallo, and P. Manzoni. "3D Simulation Modeling of UAV-to-Car Communications." In: *IEEE Access* (2019) (cit. on p. 127).

[HCCJHM19b] S. A. Hadiwardoyo, C. T. Calafate, J.-C. Cano, Y. Ji, E. Hernández-Orallo, and P. Manzoni. "Evaluating UAV-to-Car Communications Performance: From Testbed to Simulation Experiments." In: *Consumer Communications & Networking Conference (CCNC), 2019 16th IEEE Annual*. IEEE. Jan. 2019 (cit. on p. 128).

[HCLMMWY10] E. Hossain, G. Chow, V. C. Leung, R. D. McLeod, J. Mišić, V. W. Wong, and O. Yang. "Vehicular telematics over heterogeneous wireless networks: A survey." In: *Computer Communications* 33.7 (2010), pp. 775–793 (cit. on p. 10).

[HFBF06] J. Härri, F. Filali, C. Bonnet, and M. Fiore. "VanetMobiSim: generating realistic mobility patterns for VANETs." In: *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. ACM. 2006, pp. 96–97 (cit. on p. 34).

[HHOCCM18a] S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J. C. Cano, and P. Manzoni. "Experimental characterization of UAV-to-car communications." In: *Computer Networks* 136 (2018), pp. 105–118 (cit. on p. 127).

[HHOCCM18b] S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, and P. Manzoni. "Evaluating UAV-to-Car Communications Performance: Testbed Experiments." In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE. 2018, pp. 86–92 (cit. on p. 128).

[HL08] H. Hartenstein and L. Laberteaux. “A tutorial survey on vehicular ad hoc networks.” In: *IEEE Communications magazine* 46.6 (2008), pp. 164–171 (cit. on p. 15).

[HLRDK08] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena. “Network simulations with the ns-3 simulator.” In: *SIGCOMM demonstration* 14.14 (2008), p. 527 (cit. on p. 33).

[HM77] F. Hansen and F. I. Meno. “Mobile fading—Rayleigh and lognormal superimposed.” In: *IEEE Transactions on Vehicular Technology* 26.4 (1977), pp. 332–335 (cit. on p. 25).

[HMTZAZ14] R. He, A. F. Molisch, F. Tufvesson, Z. Zhong, B. Ai, and T. Zhang. “Vehicle-to-vehicle propagation models with large vehicle obstructions.” In: *IEEE Transactions on Intelligent Transportation Systems* 15.5 (2014), pp. 2237–2248 (cit. on p. 28).

[HMVMGPGP11] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson. “Smart cities at the forefront of the future internet.” In: *The Future Internet Assembly*. Springer. 2011, pp. 447–462 (cit. on p. 8).

[HPCCM17] S. A. Hadiwardoyo, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni. “An Android ITS driving safety application based on vehicle-to-vehicle (V2V) communications.” In: *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*. IEEE. 2017, pp. 1–6 (cit. on p. 128).

[HPCCM18] S. A. Hadiwardoyo, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni. “An Intelligent Transportation System Application for Smartphones Based on Vehicle Position Advertising and Route Sharing in Vehicular Ad-Hoc Networks.” In: *Journal of Computer Science and Technology* 33.2 (2018), pp. 249–262 (cit. on pp. 13, 127).

[HTHOCCM17a] S. A. Hadiwardoyo, A. Tomás, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, and P. Manzoni. "Empirical Study and Modeling of Vehicular Communications at Intersections in the 5 GHz Band." In: *Mobile Information Systems* 2017 (2017) (cit. on p. 127).

[HTHOCCM17b] S. A. Hadiwardoyo, A. Tomás, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, and P. Manzoni. "On the impact of urban intersection characteristics in vehicular to vehicular (V2V) communications." In: *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International.* IEEE. 2017, pp. 452–457 (cit. on p. 128).

[HW08] M. Haklay and P. Weber. "Openstreetmap: User-generated street maps." In: *IEEE Pervasive Computing* 7.4 (2008), pp. 12–18 (cit. on pp. 38, 64, 92, 108).

[JJUR17] F. Jamil, A. Javaid, T. Umer, and M. H. Rehmani. "A comprehensive survey of network coding in vehicular ad-hoc networks." In: *Wireless Networks* 23.8 (2017), pp. 2395–2414 (cit. on p. 11).

[JMAJAZ17] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal, and S. Zhang. "Communication and networking of UAV-based systems: Classification and associated architectures." In: *Journal of Network and Computer Applications* 84 (2017), pp. 93–108 (cit. on p. 11).

[JNNR18] S. Jaktheerangkoon, K. Na Nakorn, and K. Rojviboonchai. "Blind Corner Propagation Model for IEEE 802.11 p Communication in Network Simulators." In: *Journal of Advanced Transportation* 2018 (2018) (cit. on p. 29).

[JSO18] JSONCpp. *A C++ Library for Interacting with JSON.* 2018. URL: `https://github.com/open-source-parsers/jsoncpp` (visited on 09/24/2018) (cit. on p. 106).

[JZ17] S. Jia and L. Zhang. "Modelling unmanned aerial vehicles base station in ground-to-air cooperative networks." In:

IET Communications 11.8 (2017), pp. 1187–1194 (cit. on pp. 12, 13, 31).

[KAEHJLW11] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions.” In: *IEEE communications surveys & tutorials* 13.4 (2011), pp. 584–616 (cit. on p. 9).

[KAMA13] M. Khabazian, S. Aïssa, and M. Mehmet-Ali. “Performance modeling of safety messages broadcast in vehicular ad hoc networks.” In: *IEEE transactions on intelligent transportation systems* 14.1 (2013), pp. 380–387 (cit. on p. 10).

[Kar11] C. F. Karney. “Transverse Mercator with an accuracy of a few nanometers.” In: *Journal of Geodesy* 85.8 (2011), pp. 475–485 (cit. on p. 49).

[KKMP17] B. Kovacevic, M. Kovacevic, T. Maruna, and I. Papp. “A java application programming interface for in-vehicle infotainment devices.” In: *IEEE Transactions on Consumer Electronics* 63.1 (2017), pp. 68–76 (cit. on p. 17).

[KLKNI16] D. Kwak, R. Liu, D. Kim, B. Nath, and L. Iftode. “Seeing is believing: Sharing real-time visual traffic information via vehicular clouds.” In: *IEEE Access* 4 (2016), pp. 3617–3631 (cit. on p. 18).

[KTAKPBM10] J. Karedal, F. Tufvesson, T. Abbas, O. Klemp, A. Paier, L. Bernadó, and A. F. Molisch. “Radio channel measurements at street intersections for vehicle-to-vehicle safety applications.” In: *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*. IEEE. 2010, pp. 1–5 (cit. on p. 29).

[Lee82] W. C. Lee. *Mobile communications engineering.* McGraw-Hill Professional, 1982 (cit. on p. 101).

[LHJ18] W. Li, X. Y. Hu, and T. Jiang. “Path Loss Models for IEEE 802.15. 4 Vehicle-to-Infrastructure Communications in Rural Areas.” In: *IEEE Internet of Things Journal* (2018) (cit. on p. 30).

[Lib18] Libcurl. *The Multiprotocol File Transfer Library*. 2018. URL: `https://curl.haxx.se/libcurl/` (visited on 09/24/2018) (cit. on p. 106).

[LKLCS16] S.-H. Lee, J.-C. Kim, K.-T. Lim, H.-R. Cho, and D.-H. Seo. “WAVE communication-based V2I channel modeling.” In: *Journal of the Korean Society of Marine Engineering* 40.10 (2016), pp. 899–905 (cit. on p. 30).

[LLKXBNI14] R. Liu, H. Liu, D. Kwak, Y. Xiang, C. Borcea, B. Nath, and L. Iftode. “Themis: A participatory navigation system for balanced traffic routing.” In: *Vehicular Networking Conference (VNC), 2014 IEEE*. IEEE. Dec. 2014, pp. 159–166 (cit. on p. 17).

[LLKXBNI16] R. Liu, H. Liu, D. Kwak, Y. Xiang, C. Borcea, B. Nath, and L. Iftode. “Balanced traffic routing: Design, implementation, and evaluation.” In: *Ad Hoc Networks* 37 (2016), pp. 14–28 (cit. on p. 17).

[LLLC12] J.-C. Lin, C.-S. Lin, C.-N. Liang, and B.-C. Chen. “Wireless communication performance based on IEEE 802.11 p R2V field trials.” In: *IEEE Communications Magazine* 50.5 (2012), pp. 184–191 (cit. on p. 27).

[LR68] A. G. Longley and P. L. Rice. *Prediction of tropospheric radio transmission loss over irregular terrain. A computer method-1968*. Tech. rep. INSTITUTE FOR TELECOMMUNICATION SCIENCES BOULDER CO, 1968 (cit. on p. 26).

[MBSTB10] R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, and J. Barros. “Experimental study on the impact of vehicular obstructions in VANETs.” In: *Vehicular Networking Conference (VNC), 2010 IEEE*. IEEE. 2010, pp. 338–345 (cit. on p. 28).

[MCCM13] J. E. Meseguer, C. T. Calafate, J. C. Cano, and P. Manzoni. “Drivingstyles: A smartphone application to assess driver behavior.” In: *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE. July 2013, pp. 000535–000540 (cit. on p. 17).

[McD99] N. McDonald. “Omnidirectional pattern directivity in the presence of minor lobes: Revisited.” In: *IEEE Transactions on Antennas and Propagation Magazine* 41.2 (1999), pp. 63–68 (cit. on p. 87).

[MCR09] X. Ma, X. Chen, and H. H. Refai. “Performance and reliability of DSRC vehicular safety communication: a formal analysis.” In: *EURASIP Journal on Wireless Communications and Networking* 2009.1 (2009), p. 1 (cit. on p. 15).

[MGAUKT17] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer. “UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges.” In: *IEEE Communications Magazine* 55.3 (2017), pp. 22–28 (cit. on pp. 10–12).

[MK18] M. Mueck and I. Karls. *Networking Vehicles to Everything: Evolving Automotive Solutions*. DEG Press, Jan. 2018, p. 233. ISBN: 9781501507243 (cit. on p. 15).

[MKH11] T. Mangel, O. Klemp, and H. Hartenstein. “A validated 5.9 GHz Non-Line-of-Sight path-loss and fading model for inter-vehicle communication.” In: *ITS Telecommunications (ITST), 2011 11th International Conference on*. IEEE. 2011, pp. 75–80 (cit. on p. 29).

[MPWKK14] K. Mahler, P. Paschalidis, M. Wisotzki, A. Kortke, and W. Keusgen. “Evaluation of vehicular communication performance at street intersections.” In: *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*. IEEE. 2014, pp. 1–5 (cit. on p. 29).

[MRPPLFPB17] E. A. Marconato, M. Rodrigues, R. d. M. Pires, D. F. Pigatto, C. Q. Luiz Filho, A. R. Pinto, and K. R. Branco.

"AVENS-A Novel Flying Ad Hoc Network Simulator with Automatic Code Generation for Unmanned Aircraft System." In: (2017) (cit. on p. 35).

[MS15] D. W. Matolak and R. Sun. "Air-ground channel characterization for unmanned aircraft systems: The near-urban environment." In: *Military Communications Conference, MILCOM 2015-2015 IEEE*. IEEE. 2015, pp. 1656–1660 (cit. on p. 31).

[MTCCM09] F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni. "Realistic radio propagation models (RPMs) for VANET simulations." In: *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*. IEEE. 2009, pp. 1–6 (cit. on p. 28).

[MTCCM10] F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni. "Emergency services in future intelligent transportation systems based on vehicular communication networks." In: *IEEE Intelligent Transportation Systems Magazine* 2.2 (2010), pp. 6–20 (cit. on p. 13).

[MYTK14] S. Matsuyama, T. Yamabe, N. Takahashi, and R. Kiyohara. "Intelligent user interface of smartphones for on-vehicle information devices." In: *Procedia Computer Science* 35 (2014), pp. 1635–1643 (cit. on p. 17).

[Nak60] M. Nakagami. "The m-distribution-A general formula of intensity distribution of rapid fading." In: *Statistical Method of Radio Propagation* (1960) (cit. on p. 63).

[NRSG17] E. Ndashimye, S. K. Ray, N. I. Sarkar, and J. A. Gutiérrez. "Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey." In: *Computer Networks* 112 (2017), pp. 144–166 (cit. on p. 10).

[OAW06] J. Oishi, K. Asakura, and T. Watanabe. "A communication model for inter-vehicle communication simulation systems based on properties of urban areas." In: *International*

Journal of Computer Science and Network Security 6.10 (2006), pp. 213–219 (cit. on p. 27).

[Oku68] Y. Okumura. “Field strength and its variability in VHF and UHF land-mobile radio service.” In: *Rev. Electr. Commun. Lab.* 16 (1968), pp. 825–873 (cit. on p. 26).

[OLZGLY17] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, N. Lagraa, and M. B. Yagoubi. “Intelligent UAV-assisted routing protocol for urban VANETs.” In: *Computer Communications* 107 (2017), pp. 93–111 (cit. on pp. 12, 13).

[OSM] OSMAnd. *OpenStreetMap Automated Navigation Direction.* Accessed: 2017-01-19. URL: `http://osmand.net/` (cit. on p. 38).

[P.509] R. P.526-11. *Propagation By Diffraction.* Tech. rep. International Telecommunication Union, 2009 (cit. on p. 100).

[P.518] R. P.526-14. *Propagation By Diffraction.* Tech. rep. International Telecommunication Union, 2018 (cit. on pp. 100, 103).

[PACCM15] S. Patra, J. H. Arnanz, C. T. Calafate, J.-C. Cano, and P. Manzoni. “EYES: A novel overtaking assistance system for vehicular networks.” In: *International Conference on Ad-Hoc Networks and Wireless.* Springer. June 2015, pp. 375–389 (cit. on p. 17).

[Pat01] M. Patzold. *Mobile fading channels: Modelling, analysis and simulation.* John Wiley & Sons, Inc., 2001 (cit. on pp. 22, 23).

[PDLFEBC09] P. Papadimitratos, A. De La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza. “Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation.” In: *IEEE Communications Magazine* 47.11 (2009), pp. 84–95 (cit. on pp. 11, 14).

[Qi08] L. Qi. “Research on Intelligent Transportation System Technologies and Applications.” In: *2008 Workshop on Power Electronics and Intelligent Transportation System.* Aug. 2008, pp. 529–531. DOI: `10.1109/PEITS.2008.124` (cit. on p. 14).

[Rap+96] T. S. Rappaport et al. *Wireless communications: principles and practice.* Vol. 2. prentice hall PTR New Jersey, 1996 (cit. on pp. 22, 24).

[RKVO10] O. Renaudin, V.-M. Kolmonen, P. Vainikainen, and C. Oestges. “Non-stationary narrowband MIMO inter-vehicle channel characterization in the 5-GHz band.” In: *IEEE Transactions on Vehicular Technology* 59.4 (2010), pp. 2007–2015 (cit. on p. 29).

[RMK+13] M. Rondinone, J. Maneros, D. Krajzewicz, et al. “iTETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications.” In: *Simulation Modelling Practice and Theory* 34 (2013), pp. 99–125 (cit. on p. 35).

[SABB17] K. Sjoberg, P. Andres, T. Buburuzan, and A. Brakemeier. “Cooperative Intelligent Transport Systems in Europe: Current Deployment Status and Outlook.” In: *IEEE Vehicular Technology Magazine* 12.2 (2017), pp. 89–97 (cit. on p. 13).

[Sch11] B. Schünemann. “V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems.” In: *Computer Networks* 55.14 (2011), pp. 3189–3198 (cit. on p. 35).

[SEGD11] C. Sommer, D. Eckhoff, R. German, and F. Dressler. “A computationally inexpensive empirical model of IEEE 802.11 p radio shadowing in urban environments.” In: *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on.* IEEE. 2011, pp. 84–90 (cit. on p. 27).

[SG12] M. Sepulcre and J. Gozalvez. “Experimental evaluation of cooperative active safety applications based on V2V communications.” In: *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM. 2012, pp. 13–20 (cit. on p. 14).

[SGD11] C. Sommer, R. German, and F. Dressler. “Bidirectionally coupled network and road traffic simulation for improved IVC analysis.” In: *IEEE Transactions on Mobile Computing* 10.1 (2011), pp. 3–15 (cit. on pp. 34, 35, 105).

[Sim11] Q. N. Simulator. “Scalable network technologies.” In: *Inc.[Online]. Available: www. qualnet. com* (2011) (cit. on p. 33).

[SJSTCD15] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. L. Cigno, and F. Dressler. “How shadowing hurts vehicular communications and how dynamic beaconing can help.” In: *IEEE Transactions on Mobile Computing* 14.7 (2015), pp. 1411–1421 (cit. on p. 28).

[SK17] V. Sharma and R. Kumar. “G-FANET: an ambient network formation between ground and flying ad hoc networks.” In: *Telecommunication Systems* 65.1 (2017), pp. 31–54 (cit. on p. 12).

[Skl97] B. Sklar. “Rayleigh fading channels in mobile digital communication systems. I. Characterization.” In: *IEEE Communications magazine* 35.9 (1997), pp. 136–146 (cit. on p. 89).

[SKNI13] W. Sha, D. Kwak, B. Nath, and L. Iftode. “Social vehicle navigation: integrating shared driving experience into vehicle navigation.” In: *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. ACM. Feb. 2013, p. 16 (cit. on p. 18).

[SKPK16] P. Shilin, R. Kirichek, A. Paramonov, and A. Koucheryavy. “Connectivity of VANET segments using UAVs.” In: *International Conference on Next Generation Wired/Wireless Networking*. Springer. 2016, pp. 492–500 (cit. on p. 12).

[SS93] S. Sampei and T. Sunaga. "Rayleigh fading compensation for QAM in land mobile radio communications." In: *IEEE Transactions on Vehicular Technology* 42.2 (1993), pp. 137–147 (cit. on p. 25).

[STNKZLW12] H. Schumacher, H. Tchouankem, J. Nuckelt, T. Kürner, T. Zinchenko, A. Leschke, and L. Wolf. "Vehicle-to-Vehicle IEEE 802.11 p performance measurements at urban intersections." In: *2012 IEEE International Conference on Communications (ICC)*. IEEE. 2012, pp. 7131–7135 (cit. on p. 30).

[TCCMFM13] S. M. Tornell, C. T. Calafate, J.-C. Cano, P. Manzoni, M. Fogue, and F. J. Martinez. "Evaluating the feasibility of using smartphones for ITS safety applications." In: *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*. IEEE. June 2013, pp. 1–5 (cit. on p. 47).

[TD13] M. C. Togneri and M. Deriaz. "On-board navigation system for smartphones." In: *International Conference on Indoor Positioning and Indoor Navigation*. Vol. 28. Oct. 2013, 31st (cit. on p. 17).

[TL15] H. Tchouankem and T. Lorenzen. "Measurement-based evaluation of interference in Vehicular Ad-Hoc Networks at urban intersections." In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE. 2015, pp. 2381–2386 (cit. on p. 30).

[TPCCM15] S. M. Tornell, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni. "GRCBox: extending smartphone connectivity in vehicular networks." In: *International Journal of Distributed Sensor Networks* 2015 (2015), p. 5 (cit. on pp. 18, 19, 45, 47, 73).

[TPCCM16] S. M. Tornell, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni. "A novel on-board unit to accelerate the penetration of its services." In: *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE. 2016, pp. 467–472 (cit. on p. 16).

[TZS15] H. Tchouankem, T. Zinchenko, and H. Schumacher. “Impact of buildings on vehicle-to-vehicle communication at urban intersections.” In: *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE.* IEEE. 2015, pp. 206–212 (cit. on p. 28).

[TZSW13] H. Tchouankem, T. Zinchenko, H. Schumacher, and L. Wolf. “Effects of Vegetation on Vehicle-to-Vehicle Communication Performance at Intersections.” In: *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th.* IEEE. 2013, pp. 1–6 (cit. on pp. 28, 30).

[VANWOT14] D. Vlastaras, T. Abbas, M. Nilsson, R. Whiton, M. Olbäck, and F. Tufvesson. “Impact of a truck as an obstacle on vehicle-to-vehicle communications in rural and highway scenarios.” In: *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on.* IEEE. 2014, pp. 1–6 (cit. on p. 28).

[VH08] A. Varga and R. Hornig. “An overview of the OMNeT++ simulation environment.” In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops.* ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2008, p. 60 (cit. on pp. 32, 89, 95, 105).

[VL15] A. M. Vegni and V. Loscri. “A survey on vehicular social networks.” In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2397–2419 (cit. on p. 15).

[VMD11] W. Vandenberghe, I. Moerman, and P. Demeester. “On the feasibility of utilizing smartphones for vehicular ad hoc networking.” In: *ITS Telecommunications (ITST), 2011 11th International Conference on.* IEEE. 2011, pp. 246–251 (cit. on p. 47).

[VZB17] M. Vochin, S. Zoican, and E. Borcoci. “Intelligent System for Vehicle Navigation Assistance.” In: *World Conference* *on Information Systems and Technologies*. Springer. Apr. 2017, pp. 142–148 (cit. on p. 18).

[WAB09] J. Whipple, W. Arensman, and M. S. Boler. “A public safety application of GPS-enabled smartphones and the android operating system.” In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE. Oct. 2009, pp. 2059–2061 (cit. on p. 17).

[WFZGW16] X. Wang, L. Fu, Y. Zhang, X. Gan, and X. Wang. “VDNet: an infrastructure-less UAV-assisted sparse VANET system with vehicle location prediction.” In: *Wireless Communications and Mobile Computing* 16.17 (2016), pp. 2991–3003 (cit. on p. 12).

[WJHRMH17] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo. “Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones.” In: *Ieee vehIcular technology magazIne* 12.3 (2017), pp. 73–82 (cit. on pp. 12, 28).

[WSH17] J. Wahlström, I. Skog, and P. Händel. “Smartphone-Based Vehicle Telematics: A Ten-Year Anniversary.” In: *IEEE Transactions on Intelligent Transportation Systems* (2017) (cit. on p. 16).

[WT10] S. Waharte and N. Trigoni. “Supporting search and rescue operations with UAVs.” In: *Emerging Security Technologies (EST), 2010 International Conference on*. IEEE. 2010, pp. 142–147 (cit. on p. 11).

[XSRC12] Z. Xiong, H. Sheng, W. Rong, and D. E. Cooper. “Intelligent transportation systems for smart cities: a progress review.” In: *Science China Information Sciences* 55.12 (2012), pp. 2908–2914 (cit. on p. 8).

[YD16] M. B. Yassein and N. A. Damer. “Flying ad-hoc networks: Routing protocols, mobility models, issues.” In: *International Journal Of Advanced Computer Science And Applications* 7.6 (2016), pp. 162–168 (cit. on pp. 13, 31).

[YIIKK14] T. Yamabe, S. Ikegami, A. Ishizaki, S. Kitagami, and R. Kiyohara. “Car navigation user interface based on a smartphone.” In: *Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference on.* IEEE. Jan. 2014, pp. 85–86 (cit. on p. 17).

[YLC+13] C.-W. You, N. D. Lane, F. Chen, et al. “CarSafe app: alerting drowsy and distracted drivers using dual cameras on smartphones.” In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services.* ACM. June 2013, pp. 13–26 (cit. on p. 17).

[YQYRHB17] E. Yanmaz, M. Quaritsch, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter. “Communication and coordination for drone networks.” In: *Ad Hoc Networks.* Springer, 2017, pp. 79–91 (cit. on p. 11).

[YZL15] C. Yuan, Y. Zhang, and Z. Liu. “A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques.” In: *Canadian journal of forest research* 45.7 (2015), pp. 783–792 (cit. on p. 11).

[ZCLS15] Y. Zhou, N. Cheng, N. Lu, and X. S. Shen. “Multi-UAV-aided networks: aerial-ground cooperative vehicular networking architecture.” In: *ieee vehicular technology magazine* 10.4 (2015), pp. 36–44 (cit. on pp. 12, 13).

[ZCZZZ18] R. Zhang, L. Cai, Z. Zhong, J. Zhao, and J. Zhou. “Cross-Polarized Three-Dimensional Channel Measurement and Modeling for Small-Cell Street Canyon Scenario.” In: *IEEE Transactions on Vehicular Technology* (2018) (cit. on p. 30).

[ZGFZZ16] W. Zhu, D. Gao, C. H. Foh, W. Zhao, and H. Zhang. “A collision avoidance mechanism for emergency message broadcast in urban vanet.” In: *Vehicular Technology Conference (VTC Spring), 2016 IEEE 83rd.* IEEE. 2016, pp. 1–5 (cit. on p. 10).

[ZK16] W. Zafar and B. M. Khan. “Flying ad-hoc networks: technological and social implications.” In: *IEEE Technology and Society Magazine* 35.2 (2016), pp. 67–74 (cit. on p. 12).

[ZTSNDFB17] N. R. Zema, A. Trotta, G. Sanahuja, E. Natalizio, M. Di Felice, and L. Bononi. “CUSCUS: An integrated simulation architecture for distributed networked control systems.” In: *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual.* IEEE. 2017, pp. 287–292 (cit. on p. 36).