5G Communication Framework for Smarter Autonomous Vehicles

By

IOANNIS MAVROMATIS

Department of Electrical and Electronic Engineering
UNIVERSITY OF BRISTOL

Word count: Seventy-two thousand, three hundred and thirty.

ABSTRACT

In recent years, the interest around *"Intelligent Transportation Systems"* (ITSs) has rapidly increased, and they are currently being deployed in commercial, industrial and residential domains. The Next-Generation Cooperative ITSs (C-ITSs) are expected to bring the paradigm of Mobility-as-a-Service (MaaS) to a whole new level. A crucial role to this is attributed to *"Connected and Autonomous Vehicles"* (CAVs), as the focus is steered from being autonomous systems to becoming cooperative entities. The cooperation between them is mainly enabled by sharing sensor data and manoeuvring intentions in a Vehicle-to-Everything (V2X) fashion. Thus, an efficient, reliable and robust communication plane between CAVs and the infrastructure network, able to accommodate the demanding exchange of sensor data, is of paramount importance.

This thesis presents the author's work and contributions in the field of connectivity for CAVs within the scope of an ITS. As connectivity is a broad term, it can be examined from different perspectives. This thesis focuses on the design of a novel heterogeneous C-ITS framework that can handle scalable city-wide ITS application data streams. Later, it identifies the various drawbacks and limitations of the current Radio Access Technologies (RATs), and investigates ways to overcome them and improve the system performance.

The contributions of this thesis commence with the introduction of a conceptual design of a city-scale C-ITS implementation. Briefly, this proposed system operates as a heterogeneous network, maps different application data streams on to different V2X layers, and employs different RATs to deliver each stream in a V2X manner. Based on this fundamental concept, various drawbacks are primarily identified on existing simulation models and tools for IEEE 802.11p. Later, the existing limitations are addressed by designing an experimental testbed and conducting a sizeable experimental campaign. This not only helps with the accurate calibration of the existing models, but also provides a deep comprehension of how a real-world large-scale implementation works.

In the later parts of this thesis, the focus is directed towards Millimetre Waves (mmWaves), as is a competent technology for accommodating significant amounts of exchanged sensor data. At first, the beamforming problems in mobile environments are identified. They are approached through an intelligent MAC-layer solution for V2X beam steering. Furthermore, the resource allocation problem at the highly dynamic network topology of CAVs is investigated, designing a multi-link association scheme for V2V mmWave communications. Finally, the last research effort of this thesis, is related to the city-scale positioning of the mmWave Road-Side Units (RSUs), where an optimal automated procedure within an urban scenario is proposed. The thesis is concluded with the critical review of the research activities mentioned above as well as some ideas for future research.

DEDICATION AND ACKNOWLEDGEMENTS

First of all, I would like to express my thanks to my academic supervisors Prof. Robert J. Piechocki and Prof. Andrew Nix, for their constant guidance, motivation, patience, and substantial contribution to my Ph.D. research work. Special thanks and my sincere gratitude goes to Dr. Andrea Tassi, that acted as my mentor throughout these three years, passed on his scientific wisdom to me and guided me in every step until now.

I am grateful for my fellow doctoral students and researchers of the Communications and the Communication, System and Networks (CSN) and High Performance Networks (HPN) laboratories of the University of Bristol for the scientific conversations we had, their feedback, and of course their friendship. Besides, I would like to thank the members of staff of the University of Bristol for the technical and admin help they provided.

My heart is smiling for all my friends, and especially my buddies Lefteris, Odysseas, Stathis, Nikos, and Kostas, for accepting nothing less than excellence from me and being there in good and bad times. Words are powerless to express my thanks to that amazing person, Dr. Angeliki Katsenou for the endless conversation we had, her encouragement and willingness to support, and for lifting me up in any given moment. Leonidas and Nelly, a special thanks to you too. You rock! Last but not least, I would like to express my appreciation for my parents, for the support they provided and the sacrifices they did, to be able to be here at this moment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

4G . . . . Fourth Generation

5G . . . . Fifth Generation

A-BFT . . . . Association Beamforming Training

ATI . . . . Announcement Transmission Interval

AU . . . . Application Unit

AoA . . . . Angle-of-Arrival

BC . . . . Beam Combining

BHI . . . . Beacon Header Interval

BI . . . . Beacon Interval

BRP . . . . Beam Refinement Phase

BSM . . . . Basic Safety Message

BSS . . . . Basic Service Set

BTI . . . . Beacon Transmission Interval

CALM . . . . Communications Access for Land Mobiles

C-ITS . . . . Cooperative Intelligent Transportation System

C-V2X Cellular-V2X

C2C . . . . Car-to-Car

CAM . . . . Cooperative Awareness Message

CAV . . . . Connected and Autonomous Vehicle

CCH . . . . Control Channel

COTS . . . . Commercial Off-The-Self

CSMA/CA . . . . Carrier Sense Multiple Access with Collision Avoidance

xix

D2D .... Device-to-Device

DENM .... Decentralized Environmental Notification Message

DSRC .... Dedicated Short-Range Communications

DTI .... Data Transmission Interval

EDCA .... Enhanced Distributed Channel Access

ETSI .... European Telecommunications Standards Institute

FDD .... Frequency-Division Duplexing

GA .... Genetic Algorithm

GC .... Greedy Construction

GPS .... Global Positioning System

HD .... High Definition

IFS .... Interframe Space

IMU .... Inertial Measurement Unit

ITS .... Intelligent Transportation System

IoT .... Internet of Things

KPI .... Key Performance Indicator

LDPC .... Low Density Parity Check

LGA .... Local Government Areas

LIDAR .... Light Detection and Ranging

LOS .... Line-of-Sight

LTE .... Long Term Evolution

LWA .... LTE WiFi Aggregation

MAC .... Medium Access Control

MANET .... Mobile Ad-Hoc Network

MBIFS .... Medium Beamforming Interframe Space

MCS .... Modulation and Coding Scheme

MIMO .... Multiple-Input-Multiple-Output

MaaS .... Mobility-as-a-Service

MmWaves ...... Millimetre Waves

NR	New Radio
NLOS	Non-Line-of-Sight
OBU	On-Board Unit
OCB	Outside the Context of a BSS
OFDMA	Orthogonal Frequency-Division Multiple Access
OFDM	Orthogonal Frequency-Division Multiplexing
OSI	Open Systems Interconnection
P2P	Point-to-Point
PBSS	Personal Basic Service Set
PCP/AP	PBSS Control Point/Access Point
PDES	Parallel Discrete Event Simulations
PDR	Packet Delivery Rate
PER	Packet Error Rate
QoS	Quality-of-Service
RAT	Radio Access Technology
RSSI	Received Signal Strength Indicator
RSS	Received Signal Strength
RSU	Road-Side Unit
SAMBA	Smart Motion Prediction Beam Alignment
SBIFS	Short Beamforming Interframe Space
SC-FDMA	Single Carrier Frequency Division Multiple Access
SCH	Service Channel
SDLC	System Development Life Cycle
SDN	Software-Defined Networking
SF	Stable Fixture
SIFS	Short Interframe Space
SINR	Signal-to-Noise-and-Interference Ratio
SLS	Sector Level Sweep
SNR	Signal-to-Noise Ratio

SSW-ACK .... Sector Sweep Acknowledge

SSW-FB .... Sector Sweep Feedback

STBC .... Space Time Block Coding

SUMO .... Simulation of Urban Mobility

TDD .... Time-Division Duplexing

V2G .... Vehicle-to-Grid

V2I .... Vehicle-to-Infrastructure

V2P .... Vehicle-to-Pedestrian

V2V .... Vehicle-to-Vehicle

V2X .... Vehicle-to-Everything

VANET .... Vehicular Ad-Hoc Network

VHT .... Very High Throughput

WAVE .... Wireless Access in Vehicular Environments

e-CAV .... Emergency CAV

r-CAVs .... Regular CAV

1

INTRODUCTION

Since the beginning of this research activity, Intelligent Transportation Systems (ITSs) have moved from being a vision of the future, to become one of the most recognised acronyms in the research community. Dozens of different cities around the world work on, or have already deployed, frameworks that will alter the experience, the expectations and the mobilities of passengers and vehicles in an urban environment. Some significant advancements can be found in cities such as London, Singapore, and Barcelona[1]. In these cities, the governments and the National Research Foundations have already introduced state-of-the-art solutions, creating networks of systems able to manage road traffic, increase road safety and provide various intelligent transportation services to the citizens.

For this to happen, advances have taken place in a variety of fields, including automotive engineering, sensor design, automation and machine intelligence, and finally, wireless communications. The progress in the above fields addressed challenges that ITSs faced in their infancy. The design, development and deployment of a fully functional ITS is receiving close review. Nonetheless, it is also a huge market, and companies and corporations invest billions and rush the development of products with state-of-the-art features having as a goal to dominate this market.

There is a basic idea behind an ITS. An ITS will improve the safety, the efficiency and service, and the traffic situation through the exchange of real-time information. For example, the exploration of real-time traffic variables on accident patterning could prevent future accidents by providing the necessary insight to an ITS. Another example could be the urban traffic management predicting the performance of local roads using real-time public transport information systems. To achieve that, we require the harmonic combination and cooperation of different

---

[1]https://www.intelligenttransport.com/transport-articles/28483/top-five-smart-cities/

technologies of electronics, communications, computers, control, detection and sensing. One of the fundamental entities of an ITS will be that of the intelligent driverless vehicle. It will be equipped with numerous sensors and the information acquired from them will be its eyes and ears on the road. According to estimations, an autonomous vehicle will generate up to 4 terabytes of data daily [5, 6]. The vast majority of these data will be exchanged with the surrounding environment to achieve a reliable service. All the above define the concept of a Connected and Autonomous Vehicle (CAV), and its connectivity will be the centre of our research activities for this thesis.

The exchange of real-time data was always one of the main challenges for wireless connectivity. Taking into account the amount of data that needs to be exchanged, we realise that an ultra-reliable and robust communication framework able to achieve multi-gigabit-per-second data rates and very low delays is of paramount importance. Our current wireless technologies, spanning from WiFi to traditional fourth generation (4G) networks, do not have the capacity, speed, and most importantly, the latency and reliability to enable a truly connected experience. This is the motivation behind this thesis, i.e. the realisation that the existing communication frameworks are not adequate to achieve the strict Quality-of-Service (QoS) constraints required for an ITS. More sophisticated solutions and steps towards the fifth generation (5G) of wireless systems are required, as well as a cooperation between different wireless technologies to achieve the desired result.

Before we discuss in more depth about the potential solutions for the existing communication challenges, it is necessary to define more thoroughly what an ITS and a CAV are, their impacts on modern society, and explain the role of connectivity for a successful implementation.

1.1 What is an Intelligent Transportation System (ITS)?

There are tens of different definitions for the idea of an Intelligent Transportation System (ITS). That alone can illuminate its current state. Different aspects are not explicitly designed yet, there are many open challenges to be addressed, and there is an intense competition from different industries and universities to create the optimum solution and define the next standard. An eloquent definition can be found in the Official Journal of the European Union announced in 2010 [7]:

"ITS is the use of technology, communications and information to deliver informed and efficient mobility and transport. It includes smart motorways, autonomous/driverless and communicating vehicles, urban and inter-urban traffic management, enforcement of speed limits, transport safety and security, and improved mobility. ITSs optimise existing infrastructure to make transport more efficient, rather than to provide additional physical infrastructure with its environmental dis-benefits and financial costs."

ITSs, being an evolution of traditional transportation systems, introduce the concept that both the creator and the recipient of the information generated will be machines. All decisions

and actions will be a result of either programmable protocols or means of machine learning and artificial intelligence techniques. Of course, everything is subject to the nature of a specific application. It should be noted that the definition introduced before does not necessarily imply exclusive wireless connectivity. Certain aspects of it can be wired, such as the connectivity of the infrastructure with a Cloud Service. In this thesis, however, we are going to focus on the wireless aspects of an ITS, as they are the only means of interconnecting a large fleet of self-driving vehicles and providing fully autonomous capabilities.

The above definition also encapsulates the computing, actuating and sensing capabilities of the machines within an ITS. They not only have to implement the necessary protocols but should also interact with the surrounding environment depending on the requirements of each application. These can include sensing a pedestrian crossing a road, forcing a vehicle to reroute to avoid a traffic jam, decelerating hard to avoid a crash, sharing patients information with a nearby hospital, remotely control the traffic lights within a city, and many more. In general, providing computing, actuating and sensing capabilities to a large number of machines in our environment, we can build upon the idea of Mark Weiser for ubiquitous computing [8]. By that, we can utilise smart objects and entities within our ITS that will eventually be able to acquire, share and process information to serve the user.

As it is evident from the above, ITS is not a new technology or standard. It is instead a result of complementary technologies in a big number of fields, i.e. sensing, machine learning, cloud computing, wireless communications, etc. These technologies allow the smart devices to:

- *Be uniquely identified*: Each vehicle, traffic light, basestation, etc. within an ITS can be uniquely located, addressed and configured.

- *Communicate and Cooperate*: The different entities can create interconnected clusters, better utilise resources and enhance their services.

- *Sense and Act*: Information is collected from the surrounding environment and is either transmitted to another device or used.

- *Process data*: Certain computing capabilities are required for a smart device.

Depending on the application, the entities in an ITS might have a subset or a superset of these characteristics. Usually, the capabilities are defined subject to cost reduction, without losing functionality given that the ulterior motive is the enhancement of the road safety.

1.2 Applications and Benefits of Traditional ITSs

ITSs will have social, economic and environmental impacts. Daily, more and more parts of our transportation network will become networked and part of an ITS. This will change the way drivers, businesses and governments deal with road transport. The advancements introduced

can improve transportation in several ways. Frameworks are already under development or will be designed in the near future to exploit the full potential of an ITS.

Some example applications can be seen in Fig. 1.1. Major organisations around the globe focus on the different aspects of ITSs. For example, ERTICO - ITS EUROPE [9] is a European public-private partnership of 120 companies and organisations, promoting research activities and industrial standards for ITS applications. Different projects can be found in this consortium in the area of connected, automated and cooperative driving, e.g., the project Autopilot [10] that brings the Internet of Things (IoT) into the automotive world to transform connected vehicles into a moving IoT ecosystem. Similarly, the platform ADASIS [11] provides a map-enhanced driver assistance system. In the area of automated freight transport the NewRail team [12] focuses on freight transport using the national railway system, and the European Truck Platooning project [13] that uses trucks for the same reason. The project FABRIC [14] focuses on reducing the emissions of vehicles with sophisticated on-road charging solutions for electric vehicles. The research activities for the next-generation of ITSs are not limited in Europe. Similar organisations can be found in America (e.g., ITS-America [15]) and Asia-Japan (e.g., ITS-Japan [16]) with many different projects focusing on all the aspects of an ITS. Finally, it is worth mentioning the two projects that the University of Bristol actively contributes to. The first one is VENTURER [17], which focuses on the users aspects, the required technologies to enable the connectivity of the autonomous vehicles and how we can achieve a wide-scale adoption of them. As part of this project, a vehicle was designed and developed introducing a significant number of autonomous features as well as a communication link with the infrastructure network. The second project is Flourish [18] that again focuses on the connectivity aspect of CAVs as well as the trust of the wireless link and data privacy. Both VENTURER and Flourish projects were funded by Innovate UK, a non-departmental public body operating very close to the Government, having thepurpose to accelerate UK economic growth by stimulating and supporting business-led innovation.

As we can understand from the above, there are numerous applications within the context of an ITS. CAVs, being probably the fundamental entities of an ITS get the most attention and many of the applications designed are related to them. One of the first works in the literature [19] classified these CAV-related applications into three main categories: safety critical, traffic efficiency management, and information/entertainment. Even though we can somehow say that there is an overlap between these categories, generally they focus on the following:

- **Active road safety-critical applications**: Applications regarding primarily the decrease of the probability for traffic accidents and collisions due to unsafe speeds and driving, dangerous weather conditions and heavy traffic. Some example applications can be: cooperative forward collision warning, overtaking vehicle warning, lane change assistance, etc. In fully networked systems they can also help in preventing the loss of life under emergency situations. This is the case for emergency vehicles, that can quickly respond to accidents being informed by real-time traffic monitoring alerts and by updating warning signs and
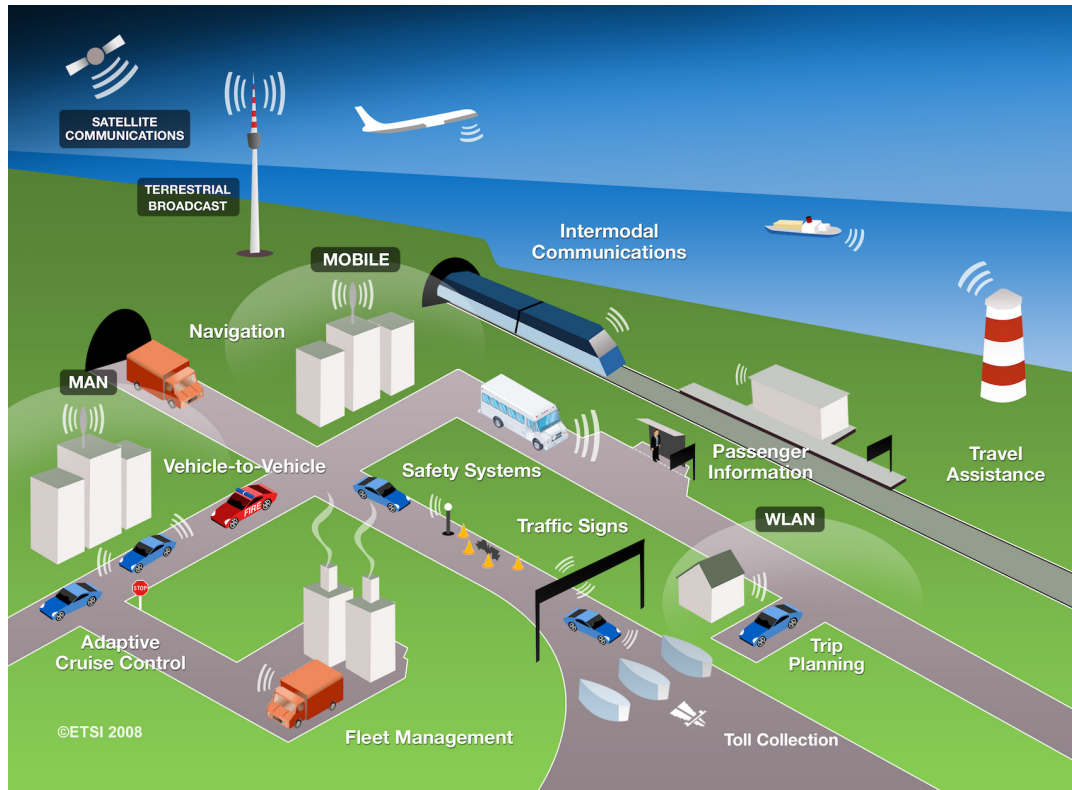
Figure 1.1: Potential applications for Intelligent Transportation Systems (ITSs) (source: ETSI).

speed limits - thus quickly and safely navigated to the incident.

- **Traffic efficiency management applications**: Applications regarding the improvement of vehicle traffic flow, traffic coordination and assistance, and finally provide access to map and local information data. For example, we can have adaptive traffic light systems that control traffic in response to patterns or traffic data collected. Also, a large number of intersections can work together and change lights in response to traffic conditions rather than based on a fixed schedule. In [1], this category was further subdivided to reflect better the different applications that are represented. The classification is: (i) navigation, (ii) road systems, (iii) speed management, (iv) vehicle control, and (v) driver-related services. A summary of classes and some potential applications can be found in Table 1.1.

- **Infotainment applications**: Applications regarding the information and entertainment that can be obtained from locally based services, such as media downloading and point of interest notifications. They also provide access to global internet services, many times related to the traffic efficiency management applications that can improve the commuting experience (e.g., parking zone management, commuting services, etc.).

More details about the above categories can be found in [1, 20, 21]. We can see a comprehensive

Table 1.1: Traffic efficiency and management of ITS applications [1].

Application Type	Application
Navigation	Traffic Information and Recommended Itinerary
	Enhanced Route Guidance and Navigation
Road Systems	Intersection Management
	Traffic Management
	Electronic Toll Collection
	Road Pricing
Speed Management	Regulatory/Contextual Speed Limits
	Traffic Light Optimal Speed Advisory
Vehicle Control	Cooperative Adaptive Cruise Control
	Cooperative Vehicle-Highway Automation (Platooning)
Driver-related Services	Eco-Driving
	Ride Sharing

view of them in Fig. 1.2. From the above categories, we see that there are applications related to all the aspects of an ITS and CAVs. Some of them will operate in the background of an ITS, such as parking zone management. This application requires a centralised city-scale implementation having more relaxed requirements and increased tolerance for errors. However, other applications, e.g., overtaking vehicle warning, cooperative forward collision warning, require a more haptic-like approach and they should meet some minimum standards to operate adequately. In this thesis, we are more interested in that second category of applications.

Overall, with all these potential applications for future ITSs, it is no wonder research activities in this field are so heated. The smart transportation market will surpass \$130bn by 2024, according to the report published from Global Market Insight, Inc. in 2018 [22]. Regardless of the application, it is evident that all of them will require to gather and exchange data with the surrounding environment and with all the different entities of an ITS. The challenges are many though, and to achieve this hyper-connected ecosystem of devices within an ITS requires many years of research.

1.3 Autonomous Vehicles, Connectivity and the Next-Gen ITSs

In the previous section, we described some of the most critical applications that can be found within an ITS. As said, the most critical entity of an ITS is that of a CAV. This definition consists of two parts. A fully autonomous vehicle is described as *"one in which a driver is not necessary, and everybody is a passenger in it"*.

The introduction of autonomous vehicles represents the most significant revolution on our roads since the advent of the internal combustion engine. The benefits include traffic reduction and increased traffic predictability, better road safety, new mobility options and social inclusion.
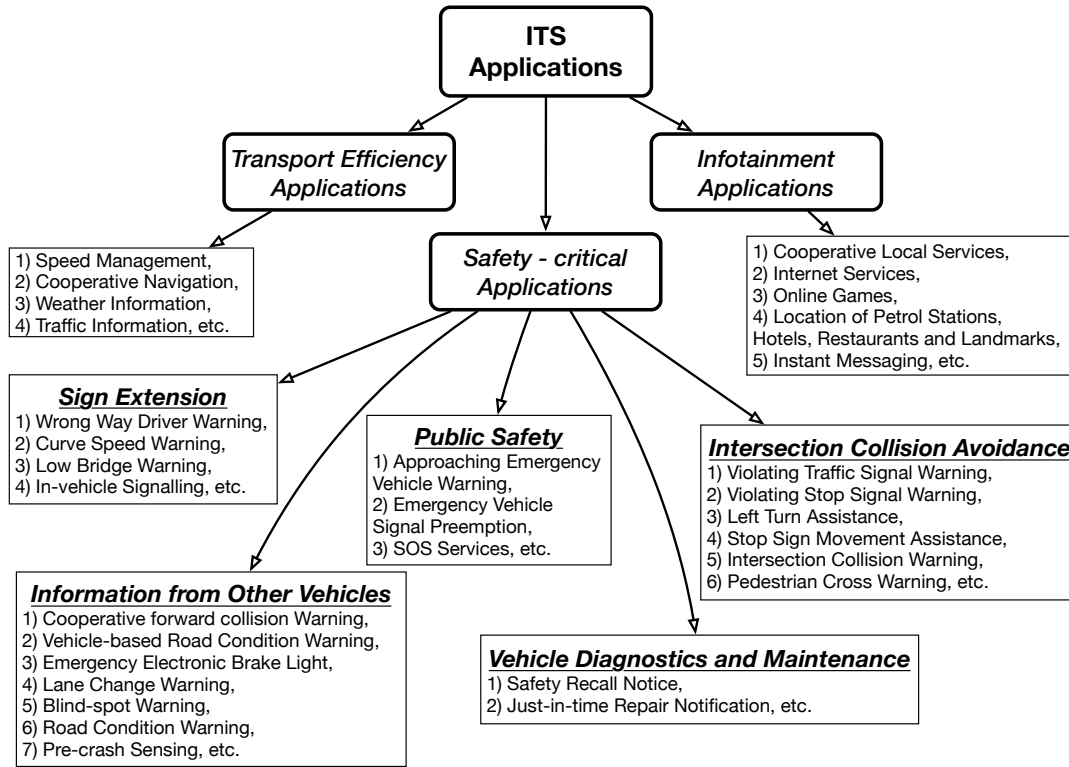
Figure 1.2: Categories of applications related to the Cooperative and Autonomous Vehicles (CAVs).

We will start our discussion from the second part of the above definition, i.e. the autonomy.

Autonomous features introduce a safer driving experience. The majority of accidents occur because of human errors [23]. On the other hand, computers, given the technological advancements can easier handle low visibility situations (e.g., fog, night) [24], they cannot be distracted, and they never get tired. Of course, this is an ongoing research topic, that rapidly progresses the last years [25, 26]. Vehicle autonomy can be described in six different levels, ranging from a person manually driving a vehicle without any artificial assistance, to no options being given to the driver - no steering wheel or control [2]. The description of each level and the involvement of the driver can be seen in Table 1.2. Currently, many car manufacturers are developing commercial solutions for Level 2 of autonomous driving. A self-driving vehicle will be equipped with numerous sensors, each one with its unique attributes and limitations. Based on the technology of the existing sensors, applications already exist that can improve road safety (adaptive cruise control, lane keeping technologies, etc.). To meet the requirements for next-generation ITSs, new sensors, such as LiDARs or camcorders, will be equipped by the vehicles [27]. However, sensors are not always accurate, they might fail, and all the computer systems are as good as they are programmed.

Recent forecasts estimate that globally the number of people living in urban areas is due to increase to over 66% by 2050 [28]. In particular, road congestion determines substantial productivity losses. Consider the simple act of searching for a parking space; this represents

Table 1.2: The six levels of autonomous driving [2].

Level	Autonomy	Driver Duties	Vehicle Duties
0	No Automation	Performs all tasks	No artificial help
1	Driver Assistance	Handles all the accelerating, braking, and monitoring of the surrounding environment	Assists with some functions, e.g., brake at the highway when close to another vehicle
2	Partial Automation	Responsible for most safety-critical functions and monitoring of the environment	Assist with steering or acceleration functions and allow the driver to disengage from some of their tasks
3	Conditional Automation	Disengage from "safety critical" functions like braking and leave it to the machine	Vehicle itself controls all monitoring of the environment and acts accordingly
4	High Automation	Driver enganged for driving only when conditions are "unsafe"	Capable of steering, braking, accelerating, monitoring the vehicle and the roadway responding to events
5	Complete Automation	Chooses the destination and becomes a passenger in the vehicle later	Driving requires absolutely no human attention and intervention

around 30% of all road traffic in mid-to-large cities, mainly due to two factors:

- a *lack of knowledge* - If the drivers were aware of the presence of a traffic jam, they would try and avoid congested roads by selecting a different route (vehicle rerouting), or they would choose alternative transportation means (road off-loading);

- a *lack of confidence* - Drivers are not aware of the location of the next available parking slot and cannot quickly reach their destinations, wasting time in painstakingly looking for available spaces and even driving erratically.

Traditional ITSs should allow a user: (i) to plan the journey ahead, and (ii) to react to traffic jams by rerouting vehicles. It becomes evident that the existing ITS solutions are not adequate. Given the expected population growth, road congestion and most importantly the road safety need to be addressed more proactively.

The above leads us to the idea of a Cooperative Intelligent Transportation System (C-ITS). C-ITSs are expected to bring the paradigm of Mobility-as-a-Service (MaaS) to a whole new level. A critical factor that will be introduced in a CAV-based MaaS paradigm is represented by autonomous vehicles that cease to be *autonomous systems* and become *cooperative entities*. This is where the first part of the definition comes in place, i.e. the connectivity. Specifically, cooperation among autonomous vehicles is enabled by the sharing of sensor data and manoeuvring intentions in a Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) fashion. This will have two

benefits. The first one is that MaaS city models based on CAVs have the potential to overcome the users' lack of knowledge and confidence employing intelligent route planning systems and efficient resource allocation strategies for road systems. Most importantly, vehicle cooperation can significantly increase the road safety protecting us from malfunctioning equipment and providing a *"bird's-eye view"* of the surrounding environment [29].

1.4 Towards fully Autonomous and Connected Vehicles

While the reality of fully autonomous vehicles on our roads is some way off, many connected vehicles already exist on our roads. Connected vehicle technologies allow vehicles to talk to each other and the infrastructure around them. Many people will already be familiar with connected vehicle technologies, such as satellite navigation, telematics, etc. One of the main differences between these links and the ones that will be required in the future is the demand for data transmission. For example, a Global Positioning System (GPS) dongle requires the exchange of a few kilobits of data. However, a Light Detection and Ranging (LiDAR) system can generate more than ~250 Mbps of raw data and a Full-High Definition (HD) camcorder usually creates more than ~15 Mbps, even after the compression. These sensors will drastically increase the amount of raw data generated per vehicle. As a figure of merit, this is expected to be ~1 Gbps [6]. To achieve fully autonomous navigation and meet the QoS requirements of next-generation C-ITS applications, will require the utilisation of these generated data, as discussed in Sec. 1.3.

Being inspired by the above, we will now define the different questions that motivated us for the research activities introduced in this thesis. Of course, the questions that arise are innumerable. Some of them can be: *"How can we secure our vehicular links?"*, *"How do we process the data generated?"*, *"How can we use these data to navigate an autonomous vehicle?"*, etc. For this thesis, we will focus on the following ones.

The first question that arises is: **[Q1]** – *"Do we always need all these data to be exchanged?"*. To answer that we should more carefully consider the potential applications that we introduced before. Applications such as the weather information exchange, generate meagre data rates that are tolerant to delays introduced from a wireless link. However, considering a smart emergency vehicle rerouting application, our vehicle driving in high speed should exchange at least two different types of data with the surrounding environment, i.e. (i) LiDAR data with the vehicles in close proximity for a more comprehensive visualisation of the obstacles around it, (ii) traffic light synchronisation information with various coordination points mounted on buildings or traffic lights to minimise the overall journey time. Taking these into consideration, we realise that we require different types of data to be transmitted over various distances and data rates depending on the application. This leads us to two important conclusions. The first one is the importance of data scalability. Based on the requirements of each application, we should either decide if we should transmit raw sensor data or some features extracted from them. Secondly, we

require a combination of different communication links to exchange the above information. It is essential to combine different wireless technologies in a system to achieve the requirements of each application. Therefore, the idea of heterogeneity will be considered for our proposed solutions.

In addition to the first question, we can ask: **[Q2]** – *"How can we transmit these data?"*. At the moment, the prevailing technology for vehicular communications is the IEEE 802.11p amendment that is based on the Dedicated Short-Range Communications (DSRC) set of protocols and standards [30]. The maximum data rate that can be achieved by IEEE 802.11p/DSRC is 27 Mbps. This value is nowhere near the 1 Gbps of sensor data that should be exchanged. To that extent, new wireless technologies, such as the Millimetre Waves (mmWaves) and 3GPP Long Term Evolution (LTE) will be considered to achieve these requirements.

Based on that observation we should also ask: **[Q3]** – *"Are the existing technologies mature enough to meet the strict QoS requirements?"*. The current wireless standards, as they stand right now, lack features that will overcome the highly dynamic environment of a network of moving CAVs. They require more sophisticated algorithms and protocols to counteract that. As numerous problems might be addressed, within the context of this thesis, we will focus on the problems of mmWaves when used for vehicular communications.

Another question that we address is: **[Q4]** – *"Is it feasible to have a city-scale communication plane for CAVs?"*. Having a large scale real-world implementation is a challenging task, that requires a lot of effort, time and resources. Therefore, large-scale city-wide deployments are usually managed by legislative bodies of cities or Local Government Areas (LGA), big organisations of industrial companies. However, the knowledge of how a real-world implementation works is of paramount importance to accurately represent an actual system in theory or simulation models. Therefore in the context of this thesis, we design our own IEEE 802.11p testbed and conduct large-scale experimentation. We will use the acquired knowledge to strengthen our provided solutions for the next research question.

This question is about: **[Q5]** – *"How can we accurately, efficiently and cost-effectively test our hypothesis and our proposed algorithms?"*. To do so, and given the lack of existing autonomous vehicles as well as the immense cost of using a big fleet of real vehicles for experimentation, we consider the problems of existing simulation frameworks. We try to overcome them by considering smart ways of improving their accuracy as well as their simulation time. By that, we can overcome their existing problems and have a better representation of reality.

1.5   Thesis Outline and Original Contributions

Based on the above research questions, and as the connectivity of autonomous vehicles is a broad term, our contribution lies mainly in two of the lower layers of the Open Systems Interconnection model (OSI model) [31], as shown in Fig. 1.3. With red, the reader observes the layers we chose to

Application (Layer 7)

Presentation (Layer 6)

Session (Layer 5)

Transport (Layer 4)

Network (Layer 3)

Data Link (Layer 2)

Physical (Layer 1)
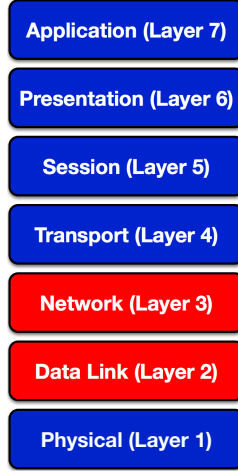
Figure 1.3: The OSI model is a conceptual model that characterises and standardises the placing of the communication functions of a telecommunication or computing system without regard to their underlying internal structure and technology. The model partitions a communication system into abstraction layers. With red, we specify the layers that we primarily focused our research activities for this thesis.

focus our scope of this work. Our research methodology starts initially with an overview of the main requirements for Vehicular Ad-Hoc Networks (VANETs) and the main system components that can be found in a real-world implementation. Before focusing on our specific directions, we describe all the related technologies that have been used in the past for vehicular communications, and why they were not able to achieve the necessary requirements. We later present the ones we consider more suitable for the future. Building on existing knowledge makes our designs more robust and applicable, as widely used as possible. However, providing new solutions will be able to enhance the existing systems and make them suitable for the next-generation of C-ITSs. The outline of this thesis is as follows:

Chapter 2: This chapter presents the necessary information for the reader to understand our motivation behind this thesis as well as the contributions and the results of it.

Chapter 3: The thesis contribution begins with this chapter. We present our novel C-ITS framework based on a heterogeneous communication infrastructure and a scalable data design, that can bring the MaaS paradigm to a whole new level. This architecture enables efficient Vehicle-to-Everything (V2X), handling scalable data streams and mapping them into different layers and wireless technologies. We describe why this architecture is important for a C-ITS and the benefits that we will introduce.

Chapter 4: We proceed by focusing on the base layer of the above architecture, i.e. the IEEE 802.11p technology. In this chapter, we present our real-world implementation of an IEEE 802.11p testbed and the knowledge we acquired from the large-scale experimentation conducted. We apply this knowledge to existing simulation frameworks, introducing at first an agile calibration

method for the INET Framework [32]. We later describe our parallelised design implementation of the above framework. The above can help us the leverage large-scale city-wide experimentation efficiently and cost-effectively without the need for real equipment.

Chapter 5: For the rest of the thesis, we focus on the problems related to mmWave technology. We start by discussing the limitations of the existing beamforming algorithms for mmWaves. We later introduce a novel Medium Access Control (MAC)-layer beamforming strategy that can establish gigabit-per-second and ultra-low latency V2I links and adapt to highly dynamic vehicular network topologies and CAVs mobility patterns.

Chapter 6: This chapter presents a multi-link association scheme for mmWave V2V communications. Our design manages to utilise the wireless channel better, and prioritises the exchange of demanding sensor data between CAVs without dominating the rest of the exchanged data. By that, we can improve the channel utilisation, by taking into account the types of vehicles on the road, their movements, positions and their propagation characteristics any given time.

Chapter 7: Our final contribution chapter introduces a novel mmWave-RSU placement scheme. Our strategy is a scalable implementation, able to find the best positions to deploy a mmWave-RSU in large-scale city-wide scenarios. We do that by taking into account different QoS-constraints of vehicular applications. Our algorithm is compared to other well-known optimisation algorithms from the literature, and not only manages to fulfil these constraints but also manages to reduce the required number of RSUs throughout the city.

Chapter 8: In this chapter, we offer some conclusions and present a critical review of the work presented in this thesis. Following our research contribution in the field of CAVs, we also discuss possible future directions for improving the V2X communication links, being either extensions on our solutions described above, or being independent and new research activities.

A subset of the above has been previously published, or is under submission in conference proceedings and journal transactions. In Table 1.3, we present an overview of how the research questions introduced in Sec 1.4, are connected with the remaining chapters of this thesis, and the related publications. We order the list of publications with respect to the order they appear in this thesis.

Table 1.3: List of published/submitted papers, the research questions they address and the chapter number they appear in this thesis.

Publication	Research Question	Chapter Related	Published / Submitted
"Multi-Radio 5G Architecture for Connected and Autonomous Vehicles: Application and Design Insights", 2018 EAI Transactions on Industrial Networks and Intelligent Systems [33]	Q1,Q2	Chapter 3	Published
"Agile calibration process of full-stack simulation frameworks for V2X communications", 2017 IEEE Vehicular Networking Conference (VNC) [34]	Q5	Chapter 4	Published
"A City-Scale ITS-G5 Network for Next-Generation Intelligent Transportation Systems: Design Insights and Challenges", 2018 Ad-Hoc Now [35]	Q4,Q5	Chapter 4	Published
"Parallel Implementation of the OMNeT++ INET Framework for V2X Communications", 2018 IEEE Vehicular Networking Conference (VNC)	Q4,Q5	Chapter 4	Submitted
"Beam alignment for millimetre wave links with motion prediction of Autonomous Vehicles", 2017 Antennas, Propagation & RF Technology for Transport and Autonomous Platforms 2017 [36]	Q3	Chapter 5	Published
"MmWave System for Future ITS: A MAC-Layer Approach for V2X Beam Steering", 2017 IEEE Vehicular Technology Conference (VTC-Fall) [37]	Q3	Chapter 5	Published
"Efficient V2V Communication Scheme for 5G MmWave Hyper-Connected CAVs", 2018 IEEE International Conference on Communications Workshops (ICC Workshops) [38]	Q3	Chapter 6	Published
"Efficient Millimeter-Wave Infrastructure Placement for City-Scale ITS", 2018 IEEE Vehicular Networking Conference (VNC)	Q3,Q5	Chapter 7	Submitted

BACKGROUND MATERIAL

In this Chapter, we present background information that a reader should be aware of to fully understand the motivation behind our work, the methods that are employed, and the results that we produced. Sec. 2.1 discusses the key characteristics of Vehicular Ad-Hoc Networks, such as the mobility models and the types of vehicular traffic used, the hardware constraints, the QoS requirements, etc. Later, Sec. 2.2, elaborates on the VANET system components, such as the different devices used mounted within a vehicle or at the infrastructure network, and the different communication domains that can be found in a VANET. Sec. 2.3, discusses the various Radio Access Technologies (RATs) used for the traditional ITSs, comparing them with the prevailing technology of IEEE 802.11p/DSRC and describing the advantages and disadvantages of each one of them. This is followed by Sec. 2.4 that describes the three different RATs that are expected to be the main solutions for vehicular communications presented their unique capabilities and drawbacks. Finally, in Sec 2.5, we briefly introduce our motivation for the remaining chapters of this thesis, describing the necessity for each research task that we conducted and how this can fulfil our ultimate goal for a next-generation C-ITS framework, able to accommodate different streams of data and state-of-the-art services, that will bring the CAV-related MaaS to a whole new level.

2.1   Key Characteristics of VANETs

As we discussed in Sec. 1.1, a C-ITS is a group of advanced vehicular applications incorporating intelligent functions. These applications, working jointly and efficiently, can achieve innovative services related to different modes of transport and traffic management. Their outcome can be a reliable, safe, fast and "smart" use of a transport network.

Before developing a communication framework for a *Next-Generation C-ITS service*, it is essential to understand the main characteristics of VANETs. Although some characteristics of VANET resemble with the ones of Mobile Ad-Hoc Networks (MANETs), many different design goals need to be considered to achieve a continuous, faultless operation. These characteristics may differ, depending on the application or the layer of the OSI model they operate within (security, MAC, routing, etc.), but can be classified as in the following subsections:

2.1.1 Mobility Models and their Effects on the Wireless Channel

A mobility model represents the movement of a mobile vehicle/pedestrian and how the direction, speed, acceleration and location change over time. Such models are frequently used for simulation purposes when new communication or navigation techniques are investigated. To that extent, these models should reflect the real mobility of the vehicles as accurately as possible.

An appropriate model is chosen based on the different factors and attributes of a vehicular network and its surrounding environment. For example, key factors are the number of lanes in each direction, the number and position of junctions and traffic lights, obstacles that may cause a detour for a vehicle or signal attenuation (e.g. buildings, forests, mountains, etc.), the road design (curvatures, slopes, etc.) and many more. An overview of these key factors and attributes can be found in [39]. Generally, based on their characteristics, the different mobility models can be classified into three different categories [40]:

- *Highway Mobility Models:* High-speed vehicles, limited presence of traffic lights and intersections, with a small number of obstacles, and variable density depending on the different time and day.

- *City/Urban Mobility Models:* Low speed, pedestrians interfering with traffic, increased density, especially in the peak hours, and frequent acceleration/deceleration because of traffic lights, junctions and obstacles.

- *Countryside/Rural Mobility Models:* Average speed with low vehicle density and without much surrounding building

Appropriate models should be chosen based on the type of roads and traffic that should be investigated for a particular scenario. The unique characteristics of each model affect the wireless communication channel. For example, a very dense urban scenario will have to deal with the increased interference and the saturation of the wireless channel. On the other hand, the increased mobility of a highway scenario can cause significant Doppler spread effects and thus, reduce the signal quality.

Overall, nodes in VANET scenarios are moving faster compared to traditional ad-hoc networks. Typical speeds, depending on the environment, are between $30\,\mathrm{km\,h^{-1}}$ to $50\,\mathrm{km\,h^{-1}}$ for a city/urban environment, $50\,\mathrm{km\,h^{-1}}$ to $80\,\mathrm{km\,h^{-1}}$ for a countryside/rural scenario and between

$90\,\mathrm{km\,h^{-1}}$ to $150\,\mathrm{km\,h^{-1}}$ for vehicles moving on the highway. When discussing about the traditional VANETs, we face the spontaneous creation of a wireless network for V2V data exchange. Therefore, the highly dynamic topology of a VANET, the multipath effect and the unpredictable wireless channel conditions, lead to intermittent communications between the nodes, especially due to the wide range of relative speeds between the vehicles. This is one of the main challenges that should be considered by an ITS application

It is evident that the accuracy of a mobility model is of paramount importance. A mobility model should not only accurately model the position of the vehicles on the road, but it should also realistically model their movement and the traffic flow. In [39], an overview of the existing mobility models is presented and how they can interact with a network simulator. What is more, a framework is introduced proposing the guidelines and the necessary tools for generating a new vehicular mobility model. Authors in [41] summarise the most significant simulation models for wireless signal propagation, DSRC technologies, and vehicular mobility and present the steps that must be undertaken to fine-tune a mobility model. In the following chapters, we will carefully choose our mobility models to reflect the reality of interest that we would like to investigate.

2.1.2 Hardware Resource Constraints

VANET nodes and the autonomous vehicles of the future, unlike the MANET or the IoT nodes, will not have any major resource constraints. CAVs will be most probably electric. Today's electric vehicle can drive for more than $360\,\mathrm{km}$ without recharging [42] and most certainly can be parked for months without its battery being completely drained.

Also, the technological advancements in semiconductors and GPUs has drastically increased available processing power while reducing the size of the chipsets. An example can be given by NVIDIA [43]. They presented a autonomous car development platform, called Pegasus, that is a data-centre solution integrating a big number of powerful GPUs. Their system provides an advanced visualisation software simulating camera, radar, and LiDAR data as inputs, processes these data, and provides a virtual driving simulation environment. In their website, it is mentioned that generates processing power equivalent to *"a 100-server data centre in the form-factor size of a license plate"*.

Nevertheless, as we move towards full autonomy, the requirements for battery, storage and processing power should be further investigated. Especially for the later ones, it is already discussed that the data either generated or collected from the surrounding environment of an autonomous vehicle will require a more active approach of processing and storing. Cloud and Fog computing capabilities are viable solutions for that [44]. For example, fog nodes can be deployed at the edge of a vehicular network to effectively and efficiently collect, process, organise and store the data in real-time, reducing the latency in the system.

2.1.3 Geographical Position, Time Reference and Sensor Data

Many of the applications introduced in Sec 1.2 require the exact position of a CAV as an input. This position is usually extracted via the Global Positioning System (GPS) coordinates acquired from a GPS dongle. GPS coordinates though are not always accurate. GPS introduces a position error with a mean value of ~3 m and a standard deviation of ~1 m [45]. However, the performance is degraded in urban environments due to the effect of urban street canyons, introduced from the height of the buildings. This error can increase up to ~20 m [46], and in some extreme cases up to ~100 m, as described in [47], a value that is unacceptable for an autonomous vehicle performing delicate manoeuvring.

CAVs, except from a GPS dongle, will be equipped with sensors such as magnetometers, accelerometers and gyroscopes, that can be used to enhance the positioning in conjunction with the GPS coordinates. This can be done with the help of data fusion algorithms (e.g. Kalman Filter, Madgwick algorithm, etc.). Sensor data can be processed, enhanced and combined with the GPS position achieving results with centimetre-accuracy even in urban environments, as shown in [48, 49].

In [50], an overview of the most well-known data fusion techniques is presented. Authors also explain the methods that the position information is combined with various sensors. Even though the combination of the above-mentioned data, can significantly increase the accuracy, it is evident that these data might not always be available (e.g. when a vehicle is driving inside a tunnel). Therefore CAVs should be equipped with more sensors such as LiDAR devices or Camcorders to provide redundancy, especially under Non-Line-of-Sight (NLOS) scenarios with the satellites. These sensors can be used to create the cartography of the surrounding vehicles with a precise representation of the road design, the buildings, the surrounding vehicles and obstacles and the pedestrians, increasing the resolution and the accuracy. The only drawback of that is the increased generation of data that should be exchanged with the surrounding environment, as discussed in Sec. 1.4.

In general, sensors equipped by a CAV can be a combination of proprioceptive (related to the behaviour of the CAV itself, e.g., inertial measurements and engine status) or exteroceptive (related to the environment outside the CAV, e.g., cameras or proximity sensors) sensors. Fig. 2.1 shows a suite of possible exteroceptive sensors equipped with an autonomous car and their potential applications and ranges. We will discuss a bit more about the sensors and the data that they generate in a later chapter.

2.1.4 QoS Requirements of Traditional ITS Applications

QoS constraints can vary significantly between different applications. Differences in factors such as end-to-end delay, throughput, tolerance in errors or dropped packets, jitter etc. can affect service and how it is perceived by the end user. All the applications set a minimum value for the above factors with stringent requirements to achieve acceptable performance. For an efficient
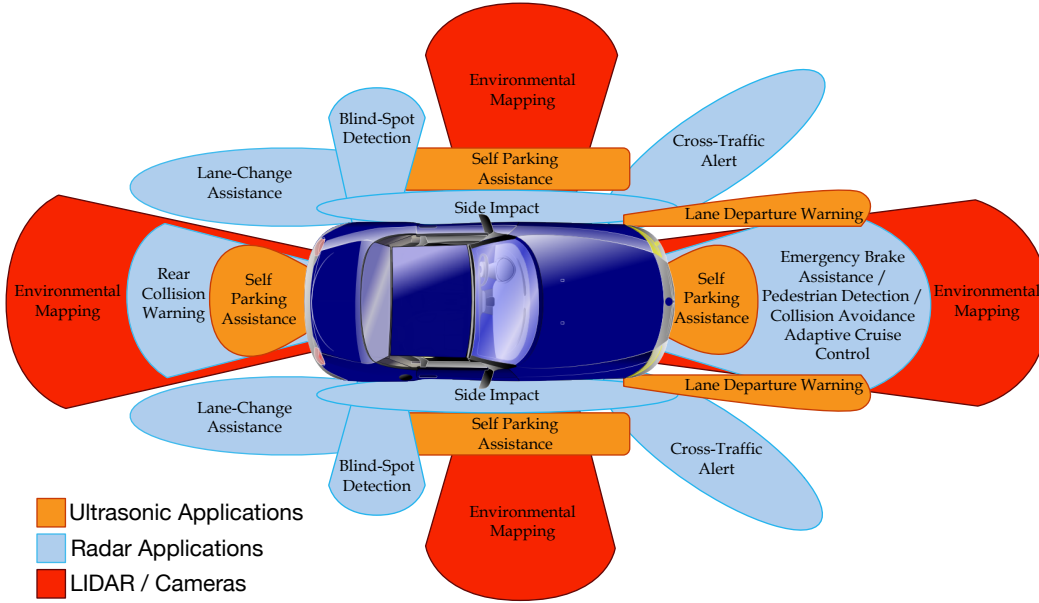
Figure 2.1: Illustration of possible exteroceptive sensors and their potential applications.

and reliable system, a wide range of QoS requirements needs to be met. We introduced the three different categories of ITS applications in Sec. 1.2. Briefly, the QoS constraints of each category are the following [51, 52]:

- *Real-Time applications and services for road safety:* Requires guaranteed access to the channel, low end-to-end delay with very low jitter, and low packet error rate.

- *Traffic management applications:* Guaranteed low latency in the presence of high bandwidth traffic and control the maximum bandwidth per connection.

- *User-oriented/infotainment applications:* Require high throughput and increased data rates, especially for the downlink.

- *Data offloading for forensics applications:* Require low latency and high throughput for the uplink to offload data to a Cloud service.

Of course, even though we can group all the services provided above, each application has some unique requirements that may differ from others. For example, taking some example applications from the first group (Real-Time applications and services for road safety), we see that some of them may trigger an event and periodically transmit data for a short period of time mode, e.g. emergency vehicle warning, slow vehicle warning, etc. Others are designed to periodically transmit data, e.g. motorcycle warning, vulnerable road user warning, etc. and finally, others will broadcast a packet at a specific time, e.g. pre-crash warning. Each application will have its unique QoS constraints, e.g. pre-crash sensing warning requires a latency of $< 50\,\text{ms}$,

while an emergency vehicle warning application or an overtaking vehicle warning is more tolerant to the latency (< 100 ms). A comprehensive table of the available safety and non-safety related applications with their minimum QoS requirements can be found in [53].

2.1.5 QoS Requirements as we Move Towards Full Autonomy

The above QoS constraints are proven to be inadequate as we move towards more sophisticated, yet more demanding, next-generation C-ITS applications. All the aforementioned applications are services that inform or warn a vehicle about an event. A C-ITS is an extreme case that the users' involvement is merely to indicate a destination. Later, the vehicle is solely responsible for all the controls. To that extent, we now have to answer the question of what are the requirements to control and not just inform the vehicles within a C-ITS.

Let us take for example a platooning application of a fleet of autonomous vehicles. Non-automated platooning vehicles travel with an intervehicle distance of 8 m to 10 m. Current research activities show that these vehicles will drive with a distance of a few centimetres between them [54]. An application like that requires < 20 ms of end-to-end latency [55]. Another example can be that of an automated vehicle overtaking. The maximum end-to-end latency, in that case, should not exceed the 10 ms [56]. Comparing that with the overtaking vehicle warning requirements that we discussed in Sec 2.1.4 (100 ms), we can see that an automated manoeuvre requires ten times less delay compared to just the warning. As we move towards more sophisticated solutions and more automated C-ITSs, we will require more *tactile-like* delays that may reach up to 1 ms. Some more example C-ITS applications with their QoS requirements can be found in [57]. What is more, as we discussed before, the various sensors equipped with a CAV, generate different amounts of data that should be processed or transmitted. Later in this thesis, we will present a table with the requirements concerning the data rate for each sensor.

2.2 Vehicular Communication Systems

In the previous section, we described the key characteristics and the minimum QoS requirements for a vehicular communication link. From the above, it is obvious that a VANET is more complex compared to a MANET. As we mentioned before (Sec. 1.1), traditional ITS should allow a user to plan the journey ahead and to react to traffic jams by rerouting vehicles. Before discussing further about that, we should introduce the main components of a VANET system.

2.2.1 Vehicular Communication System Components

Vehicles, regardless of the road characteristics, drive with variable speed and density, leading to a highly dynamic topology. ITS services are designed to operate in a distributed manner creating a non-centralised unbounded system. Even though VANET do not operate within a fixed infrastructure, they consist of three main system components (Fig. 2.2):

1. *On-Board Units (OBUs)*: These devices are mounted inside a roaming car and are responsible for reliably exchanging safety and non-safety-critical information with other Road Side Units (RSUs) or OBUs. They are mainly used for accessing the wireless medium, IP mobility, geographical routing, etc.

2. *Application Units (AUs)*: They are responsible for using the capabilities of an OBU, such as the mobility and the network functions. The AU can be a dedicated device for safety applications or a normal device, such as Rasberry Pi or a laptop, to provide access to the Internet. The AU is usually connected to the OBU through a wired or wireless connection, and the difference between them is in most cases logical and can be considered the brain of the OBU.

3. *Road Side Units (RSUs)*: Fixed device along the side of the road or at central locations in the road network, such as the intersections. They represent the infrastructural network of a VANET. Main functions are the: extension of the network by redistributing or forwarding the information to other RSU or OBU, running safety-critical applications or providing internet access to the network devices. An RSU may be equipped with various wired or wireless radio interfaces. These interfaces will be used to provide access to cloud services, other networks, and connections to other RSUs to extend the communication range of the ad-hoc network.

More information about these devices can be found in the IEEE 1609.0-2013 - Wireless Access in Vehicular Environments (WAVE) - Architecture standard [58] and in the Car-2-Car (C2C) Communication Consortium [59]. Also [21, 60] present a comprehensive description of these devices and their functions. The above terminology was initially used within the scope of Dedicated Short Range Communications (DSRC). However, due to the necessity for new RATs for vehicular communications and for simplification, we will use this terminology for all the communication technologies that will be referred for the remaining of this thesis.

2.2.2 Vehicular Communication Domains

ITSs require a V2X communication plane. A CAV, which is the main component of a next-generation ITS should be able to communicate with other self-driving vehicles, the infrastructure network and the surrounding environment.

Generally, it can be said that there are three different communication scenarios in a VANET. The first one, where all vehicles communicate via an RSU, creating a centralised network similar to a WLAN. Secondly, CAVs communicate directly, forming an Ad-Hoc network. Finally, a hybrid approach, combining both direct and indirect communication between the vehicles. C2C Communication Consortium categorises the system architecture in three different domains and [21] summarises them (Fig. 2.2):
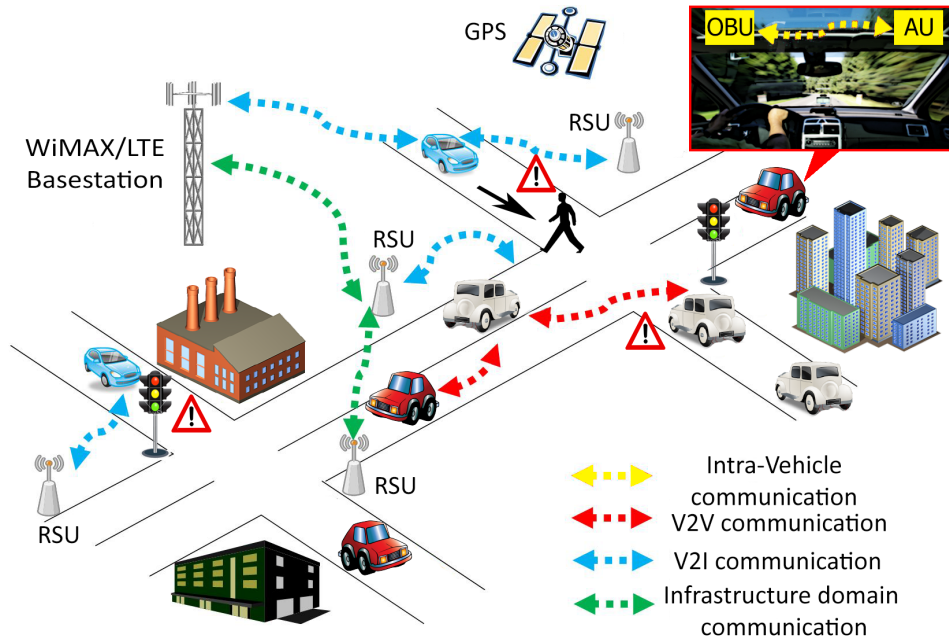
Figure 2.2: VANET architecture and main system components.

1. *Intra-Vehicle domain*: Communication between OBU and AU. It can be either wired or wireless as said before and most times the difference is logical.

2. *Ad-Hoc domain*: For these types of communications, CAVs should be equipped with an OBU and RSUs should be deployed in the network. In this domain, two different communication links can be found:

   - Vehicle-to-Vehicle (V2V) links: Vehicles using *single-hop* or *multi-hop* communications with other surrounding vehicles. If no direct link can be formed between two CAVs, a relay node is responsible for forwarding the information towards the destination vehicle.

   - Vehicle-to-Infrastructure (V2I) links: Vehicles communicate with other nodes via an RSU, to increase their coverage. They can exchange data, share road information, utilising the infrastructure network as a backbone link and exploiting the ability of the RSU for intense processing.

3. *Infrastructure Domain*: RSUs can be connected with the rest of the infrastructure network or the Internet, allowing OBUs to exchange information with other networks or have access to cloud/fog computing capabilities. The infrastructure network could be any host such as the Internet or a server for safety and non-safety vehicular applications.

Except for the vehicles and the RSUs, many more connected entities are added to the ecosystem of a next-generation C-ITS. These entities can be bicycles, pedestrians, or even the

power grid, forming new types of communications such as Vehicle-to-Pedestrian (V2P), Vehicle-to-Grid (V2G), etc. These types of communication have gained significant momentum in recent years, as more complex next-generation services are being developed daily. All the above communication links can be expressed within the scope of V2X communications.

Of course, there are many similarities between these communications, but the requirements are not always the same. For example, a V2P communication link does not require gigabits-of-data-rate as in a V2V case. This is because only a few short safety-critical messages will be required to be exchanged in conjunction with data coming from non-safety-critical services. LiDARs and camcorders mounted on vehicles will most probably be responsible for sensing and avoiding a walking person. For this thesis, we will focus on the V2V and V2I communication links as these are the most demanding ones. For the remainder of this thesis, if we use V2X for simplicity, we will be referring the V2V and V2I links within a C-ITS.

2.3 Dominant RAT for V2X Links within Traditional ITSs

As we discussed in Sec 1.4, the above mentioned demanding QoS requirement cannot be fulfilled with the use of only one RAT. At the moment, the prevailing technology for vehicular communications is IEEE 802.11p/DSRC. For years, many researchers tried to use different RATs within an ITS and enhance the performance of the communication link. Some of these efforts were very successful, and some of them ended up degrading the performance. For the remainder of this section, we will at first compare IEEE 802.11p/DSRC with some older RATs that were not finally able to qualify for vehicular communications. This will be following by a brief introduction of the three different technologies that we believe are the most suitable ones, the reasons behind that, and a performance comparison between them.

2.3.1 IEEE 802.11p/DSRC vs Other Technologies

2.3.1.1 IEEE 802.11 amendments vs IEEE 802.11p

IEEE 802.11b was at first one of the optional technologies for vehicular networks. It is based on Direct Sequence Spread Spectrum (DSSS), introducing a Direct-Sequence Code Division Multiple Access (DS-CDMA) scheme, reduced multipath fading, etc. In [40], a detailed comparison between IEEE 802.11p and b can be found comparing both technologies under highway, rural and urban environments, concluding that IEEE 802.11p, introduces a great improvement in network throughput and end-to-end delay. IEEE 802.11a was designed and optimised for networks with low or no mobility. To that extent, is inferior to IEEE 802.11p, where high-mobility effects should be addressed. Finally, an example comparison between IEEE 802.11p and IEEE 802.11n can be found in [61]. Authors in this work show that IEEE 802.11p/DSRC manages to achieve less delay and better packet delivery rate.

In general 802.11 was designed for local networks. When they are working in ad-hoc mode, all nodes are assumed to be in a specific range and able to interconnect with each other. Adding a multi-hop characteristic will introduce significant overhead, and therefore, even though it was tried to improve these technologies, they were never standardised. One of the main reasons that IEEE 802.11p/DSRC manages to outperform the rest of the 802.11 amendments is the Outside the Context of a BSS (OCB) mode. The introduction of OCB mode meant that no authentication/association is necessary to communicate. The two main parameters related to the channel that should be configured are the central frequency and the bandwidth, and they has to be a priori known.

2.3.2   WiMAX vs IEEE 802.11p

IEEE 802.16 standard series introduced an amendment able to work in mobile environments. This was the IEEE 802.16e amendment, or as is more widely known Mobile WiMAX. WiMAX supports point-to-multipoint connections as well as mesh topologies. Using Orthogonal Frequency-Division Multiplexing (OFDM) in the physical layer manages to support very high data rates. Also, QoS is supported, and longer distances can be achieved compared to the DSRC of IEEE 802.11p. Many researchers compared the performance of both technologies under a vehicular scenario.

In [62] authors introduced a VANET application and compared the performance to the mean Throughput and Packet Loss. Their simulator calculated the total time needed for a packet to be delivered. In [63], simulations were used to evaluate the coverage, average throughput, and end-to-end delay for different vehicle speeds, traffic data rates, and network deployments. Both works concluded that WiMAX shows better performance with more extensive networks concerning the packet delivery rate and reasonable delays. On the other hand, IEEE 802.11p is better suited for scenarios with a small number of vehicles introducing very low latency, even in increased speeds. As we mentioned before, the safety-critical applications require very low delays ($< 100\,\text{ms}$). These QoS requirements cannot be achieved with WiMAX technology.

2.3.3   Bluetooth vs IEEE 802.11p

Finally, Bluetooth is a wireless standard used for short-range communications. It is easy to be used and is a very cheap technology. It supports low data rates (1MB/s) that are sufficient for some vehicular applications. The research community tried to evaluate Bluetooth performance, and it was concluded that it has several drawbacks due to the piconet structure that is used and the difficulty in maintaining it. From the specifications of Bluetooth, it can be seen that the range allows connections up to 10m which is considered very short when compared to the DSRC ranges. Also, increased latency is introduced in the formation of a new piconet (7 s) or a new scatternet (45 s) [64] and or when new nodes join an existing piconet [65].

Table 2.1: Candidate RAT solutions for V2X communications.

Feature \ RAT	IEEE 802.11p / DSRC [69]	IEEE 802.11px [70]	C-V2X (LTE-A Pro) [71], [72]	MmWaves (IEEE 802.11ad) [73]	MmWaves (IEEE 802.11ay) [73]
Frequency Band	5.85 GHz - 5.925 GHz	5.85 GHz - 5.925 GHz	450 MHz - 4.99 GHz, 5.725 GHz - 5.765 GHz	57.05 GHz - 64 GHz	57.05 GHz - 64 GHz
Channel Bandwidth	10 MHz	10 MHz	Up to 640 MHz	2.16 GHz	Up to 8.64 GHz
Range	≤ 1 km	≤ 1 km	≤ 30 km	≤ 50 m	≤ 500 m
Bit Rate	3 Mbps-27 Mbps	Up to 60 Mbps	Up to 3 Gbps	Up to 7 Gbps	Up to 44 Gbps
End-to-End Latency	≤ 10 ms	≤ 10 ms	30 ms to 50 ms (UL/DL) 20 ms to 80 ms (V2V)	≤ 10 ms	≤ 10 ms
Link Establishment Latency	~0 ms	~0 ms	40 ms to 110 ms	10 ms to 20 ms	10 ms to 20 ms
Coverage	Intermittent	Intermittent	Ubiquitous	Intermittent	Intermittent
Mobility Support	≤ 130 km h^{-1}	Under Investigation	≤ 350 km h^{-1}	≤ 100 km h^{-1}	≤ 100 km h^{-1}
QoS Support	Yes	Yes	Yes	Yes	Yes
Broadcast Support	Yes	Yes	Yes	No	No
V2I Support	Yes	Yes	Yes	Yes	Yes
V2V Support	Yes	Yes	Over PC5 Interface	Yes	Yes
Relay Mode	Yes	Yes	Yes	Yes	Yes
MIMO	No	Yes	Yes	No	Yes

From all the above it was evident that IEEE 802.11p is a suitable candidate for vehicular communications and therefore, it became the prevalent technology until now. However, as we discussed before, IEEE 802.11p cannot always fulfil the demanding requirements of the next-generation C-ITS applications. A more sophisticated solution is required for that, and the idea of heterogeneity needs to be introduced in a network. An interesting work [53], shows the requirements for heterogeneous VANET architectures based on DSRC and cellular networks, in addition to the challenges and the solutions for these approaches. There are already many works in the literature that build on top of the concept of heterogeneity to enhance VANETs performance. For example, authors in [66, 67], designed two beamforming algorithms for connectivity with high-speed trains based on a heterogeneous DSRC/mmWave network. Another example using IEEE 802.11p/LTE can be found in [68] where the authors present a novel data traffic steering algorithm using both technologies.

Generally, the research community started steering its interest towards mmWaves and 3GPP LTE technologies as alternatives to IEEE 802.11p. As mentioned before, many research activities build on the idea of heterogeneity combining two of these three technologies. In the next section, we will present the unique characteristics of each one of them and a comparison between their performance presenting the pros and cons and their impact on vehicular communications.

2.4 Candidate RATs for V2X Communications within C-ITSs

Currently, the IEEE 802.11p/DSRC [74] and its extension IEEE 802.11px [70], 3GPP's LTE-Advanced (LTE-A) Pro [75] with its Cellular-V2X (C-V2X) capabilities [71] and, mmWave frequency systems as described by IEEE 802.11ad [76] and IEEE 802.11ay [77] have emerged as potential RATs for V2X communications. Table 2.1 summarises the main features and the additional capabilities provided by all the considered communication solutions.

2.4.1 IEEE 802.11p/DSRC and IEEE 802.11px

IEEE 802.11p/DSRC represents the suite of IEEE 802.11s and IEEE P1609.x standards, describing a system operating in the frequency range 5.850 GHz to 5.925 GHz, with a decentralised architecture that supports V2X communications. The IEEE 802.11p standard defines the PHY and MAC layers. More specifically, on top of the MAC layer, a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access to the medium, supports different QoS profiles by the Enhanced Distributed Channel Access (EDCA) protocol. The IEEE 802.11p/DSRC implementation of the EDCA is inherited from IEEE 802.11e with little modification. With regards to the PHY layer, the OFDM mechanism is adopted, which allows users to achieve a maximum PHY transmission rate of 27 Mbps at a speed of 130 $\mathrm{km\,h}^{-1}$. However, due to the overhead of the communication protocols, the actual network throughput achieved is limited to about 15 Mbps [34]. The reserved 75 MHz of the spectrum is divided into seven channels each of 10 MHz bandwidth, where channel
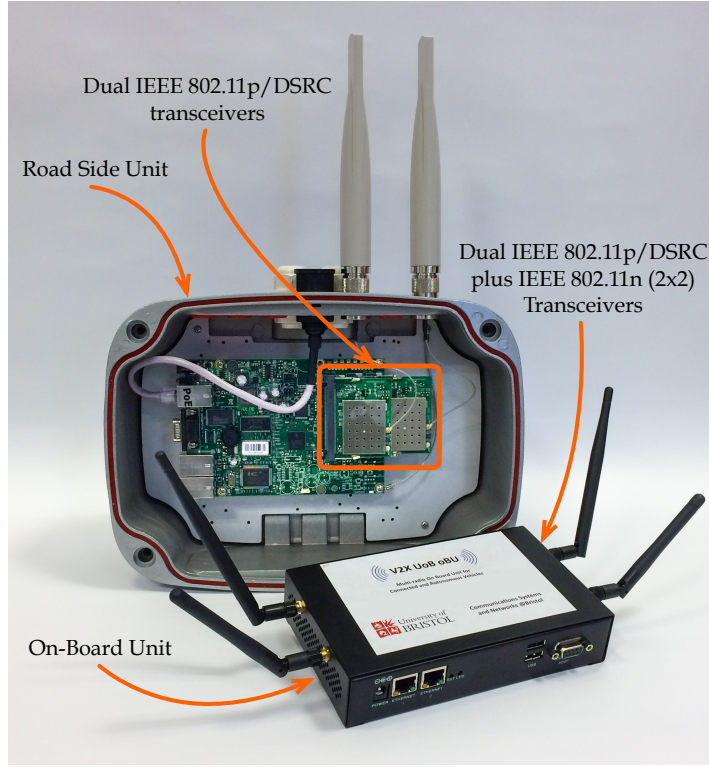
Figure 2.3: Example of low-latency Linux Kernel implementation of IEEE 802.11p/DSRC units prototyped by the University of Bristol. The figure shows units designed for a road side and an on-board vehicle deployment.

178 (the Control Channel, CCH) is solely intended for broadcasting safety and mission-critical messages, and six channels (the Service Channels, SCHs) being used for all other applications.

Due to the adopted MAC contention mechanism, the density of vehicles per area may have a disruptive impact on the overall end-to-end delay. This is caused by the lack of coordination among the devices, i.e., the number of packets dropped increases more than linearly with the vehicle density. As a result, IEEE 802.11p/DSRC tends to be more suitable for transferring low bitrate data streams in vehicular environments characterised by low-to-medium road density and vehicle speed. Examples of an IEEE 802.11p/DSRC Road Side Unit (RSU) and On-Board Unit (OBU) are shown in Fig. 2.3.

IEEE 802.11p/DSRC drawbacks are currently being addressed by an enhanced vehicular communication standard, denoted as IEEE 802.11px [70]. IEEE 802.11px PHY is built on the IEEE 802.11ac PHY layer. As such, the Low Density Parity Check (LDPC) codes for channel coding and the Space Time Block Coding (STBC) increase the performance under noisy channel transmissions exploiting the different versions of the received signal and combining them in an optimal way. Packet delivery rate (PDR) is expected to improve by 40% at a transmitter-receiver distance of 300 m compared to the legacy IEEE 802.11p. Also, Multiple-Input Multiple-Output

(MIMO) antenna capabilities, the improved OFDM pilot layouts and the introduction of Very High Throughput (VHT) frame, are expected to enhance the channel capacity of a factor of 10 compared to the IEEE 802.11p PHY layer. The new IEEE 802.11px/DSRC will adopt the OCB operation of IEEE 802.11p/DSRC introducing backward compatibility with its predecessor. Despite the IEEE 802.11px being in its early development stages, it is expected to replace the legacy IEEE 802.11p in the next years. It is worth noting that even though IEEE 802.11px is expected to improve the system performance compared to its predecessor, still it cannot accommodate the huge amount of data required by next-generation C-ITS services.

2.4.2   3GPP LTE-Advanced Pro and C-V2X

LTE and its major enhancement, LTE-A Pro, provide broadband capabilities to mobile users for several years [71]. Furthermore, 5G New Radio (NR) represent the 5G of cellular communications standards defined by 3GPP to provide high data rate, ubiquitous coverage and global connectivity for mobile cellular users [78]. Some of the key features provided are the ability to accurately position using single reference station, operation on frequency bands $\leq 6\,\mathrm{GHz}$ and $\geq 24\,\mathrm{GHz}$, macro/small cell coverage, etc. Currently, this essential procedures are only accepted in principle, for this thesis we will focus on LTE-A Pro standard amendments. The air interface can support Time-Division Duplexing (TDD), Frequency-Division Duplexing (FDD), and Half-Duplex FDD, as well as scalable channel bandwidths (1.4 MHz - 20 MHz). Furthermore, a maximum of 32 component carriers was aggregated in LTE-A Pro, leading to a maximum aggregated bandwidth of 640 MHz. Support for Licensed Assist Access (LAA), enhanced LAA and LTE WiFi Aggregation (LWA) is also added, which means that additional bandwidth can be made available by aggregating data together from regular LTE bands, the 5GHz LTE-unlicensed spectrum, and common WiFi networks.

Downlink and uplink access technologies are based on Orthogonal Frequency Division Multiple Access (OFDMA) and Single Carrier Frequency Division Multiple Access (SC-FDMA), respectively, thus guaranteeing high flexibility and efficiency in frequency-time resource scheduling. Due to the advanced MIMO capabilities, significant spectral efficiency can be obtained, even at high mobility speeds and under dynamic propagation environments. Besides, high data rates are potentially supported, ranging from 300 Mbps and 75 Mbps in the LTE downlink and uplink respectively, to theoretically 3 Gbps in the case of the LTE-A Pro downlink [79]. Multicast/broadcast services are also fully supported employing the evolved Multimedia Broadcast and Multicast Service (eMBMS) – thus enabling broadcasting of vehicular service messages in cross-traffic assistance applications [80].

Along with the eMBMS framework, V2V capabilities are built upon an enhanced LTE Direct mode and allow any network entity to engage to device-to-device (D2D) communications. The C-V2X operation of LTE-A Pro introduces two new radio interfaces (C-V2X Mode 3 and C-V2X Mode 4). The cellular interface (namely as *Uu*) that supports V2I links and the PC5 interface, which is

responsible for the V2V communications. The configuration of the sidelinks (V2V links) depends on the availability of cellular coverage. When it is present (C-V2X Mode 3), control information is transmitted to CAVs using the Uu interface and V2V communications is established, i.e. the cellular network is responsible for allocating the resources. On the other hand (C-V2X Mode 4), vehicles do not require cellular coverage, and autonomously select their radio resources using a distributed scheduling scheme supported by congestion control mechanisms. Mode 4 is considered the baseline mode and represents an alternative to 802.11p or dedicated short range communications. One of the drawbacks we observe, is the need of a basestation for allocating efficiently the radio resources to each CAV (for C-V2X Mode 3). Secondly, the formation of links significantly increase the end-to-end latency [81] even for C-V2X Mode 4, that a basestation is not required. Currently, these procedures are part of the 3GPP Releases 14 and 15.

Due to the flat architecture, applications demanding low-to-medium latency requirements can be supported without affecting the network scalability. In particular, up to 10 ms end-to-end delay can be reached over an LTE-A Pro network under low cellular traffic conditions. However, *tactile-like* latency requirements (smaller than 10 ms) cannot be fulfilled, especially when the cellular traffic is increases and latencies of 50 ms to 80 ms are observed. Furthermore, terminals in idle mode need to re-establish a connection with the base station (additional link establishment latency of up to 110 ms), thus spending additional time to reach to the connected state. This leads to severe performance degradation in safety-critical applications [82].

2.4.3 Millimeter Wave Systems - IEEE 802.11ad and IEEE 802.11ay

Systems based on mmWave are expected to play a pivotal role in 5G cellular systems. A mmWave system operates in the spectrum between 30 GHz and 300 GHz. For what concerns the application domain of local area networking, the IEEE 802.11ad standard is gaining increased momentum while being tested for vehicular communications [66, 67, 83, 84]. In this standard, the carrier frequencies are spread around the 60 GHz, with a channelisation of 2.16 GHz. Four channels are available for usage, as described in IEEE 802.11ad standard. Also, any mmWave system imposes the adoption of large antenna arrays to achieve high array gains through beamforming techniques. The broad array gains, along with large channel bandwidths, allow the system to achieve high data rates (typically several gigabits-per-second). IEEE 802.11ad ensures data rates higher than 7 Gbps and an end-to-end latency smaller than 10 ms. IEEE 802.11ay [77] that is expected to replace IEEE 802.11ad in the next years introduces a channel bonding. 2,3 or 4 of the available channels can be used together increasing the channel utilisation up to 8.64 GHz. Because of that and because of the introduced MIMO capabilities, throughput performance is significantly increased, reaching up to ~100 Gbps.

From the signal propagation perspective, Line-of-Sight (LOS) communications are characterised by path loss exponents smaller than 2.8, while Non-Line-of-Sight (NLOS) communications may present much higher path loss exponents. In fact, due to their reduced wavelength, mmWave

systems are susceptible to blockages. For these reasons, typical NLOS path loss exponents span between 3.8 and 5.6 [79].

As a prospective 5G wireless solution, mmWave systems proved to be a viable alternative to traditional cellular networks and wireless backhauling systems, while the possibility of using this technology to support the communications in next-generation ITS systems are being extensively investigated. In particular, the European Commission is currently considering the possibility of supporting standardisation activities of ITSs based on mmWave systems to be operated across a dedicated band spanning between 63 GHz and 64 GHz [83].

With ideal propagation conditions, mmWave systems significantly outperform vehicular communication systems based on the IEEE 802.11/DSRC and LTE/LTE-A standards. Large values of penetration loss and errors in the alignment of the antenna beams will have a disruptive impact on the stability of mmWave links. IEEE 802.11ad is shown to be able to achieve adequate link quality at distances up to 50 m, while IEEE 802.11ay is expected to achieve distances of up to 300 m to 500 m. One of the main problems of both standards is the latency introduced by the legacy beamforming training. It can add latency of up to 20 ms in a system [37], a value that is not compliant with the strict QoS constraints of an ITS. The introduced latency will be further investigated in a later chapter.

2.5   A Step Closer to the C-ITS of the Future

In Chapter 1, we described what is an ITS (Sec. 1.1) and its impact on our world (Sec. 1.3). Later, we presented some of the existing services, gave an insight about the applications of the future (Sec. 1.2) and introduced the CAV, the key entity of a C-ITS. This was followed by a brief introduction on the necessity for a reliable and robust communication framework between the CAVs and the infrastructure network, as well as some questions that will try and answer in this thesis (Sec. 1.4). In Chapter 2, we discussed the key characteristics of a vehicular network and a C-ITS (Sec. 2.1) as well as the QoS requirements of the wireless link for the current and the next-generation systems. Building on top of that, we presented the main architectural components of a VANET (Sec. 2.2) and talked about the technologies that did not qualify for a C-ITS (Sec. 2.3) and the potential enabler ones (Sec. 2.4) describing their main drawbacks. In this section, we will break down the different problems introduced and present some of the existing research activities to address them and why a further investigation was still required.

2.5.1   Multi-Radio 5G C-ITS Future Architecture Design

The increased demand in data processing and storage requirements of the next-generation C-ITS services should be addressed by more powerful machines than the ones equipped by an autonomous vehicle. Also, many services will require access to data from different city regions, cities, or even from other countries [85]. This implies the necessity for a more centralised network

architecture design and a cloud computing system architecture. Some example implementation of Vehicular Cloud architectures can be found in [86, 87]. Both works describe the design of a Cloud computing service that is suitable for ITS services. However, C-ITS services require also ultra-low latency to operate. Cloud service data centres can be physically located anywhere in the world far away from the physical location of a vehicle. The increased number of hops to reach the destination node can quickly lead to increased latency and thus, service performance degradation.

The solution to this problem is the usage of Fog computing capabilities. Fog services, like the cloud-based ones, provide computation, storage, and networking resources. Their main difference though is that they are being hosted in the vicinity of the end users (edge of the network). As a result, reliable access is provisioned to delay-sensitive mobile applications. Deploying a fog architecture provides a more reliable service to the end user and is characterised by a low latency, geographical distribution of devices and mobility for a large number of vehicles [44]. Example Fog architecture designs for vehicular communications can be found in [44, 88]. Taking into account the different services provided within a C-ITS, we realise that the use of either a cloud or a fog architecture is service-centric, making them both finally equally important. From that, we realise the importance of abstraction in the system and an access control plain that will be responsible for the allocation of the resources in real-time.

Another critical factor is that the data generated are service-centric as well. We can recall the examples of the transmission of LiDAR data that requires very high data rates while the information of the next available parking space has more relax QoS requirements. To account for the unique features of each application, we require a *scalable data architecture*. The idea of scalability in data originates from the video transmissions where a scalable video stream consists of a *base layer* and multiple *enhancement layers*. The base layer allows users to achieve a primary reconstruction quality, which is gradually improved as soon as the enhancement layers are successfully received. The same principle trespassed the natural boundaries of multimedia communications and is being applied to design systems capable of an analogue-like service degradation [89].

To meet the QoS demands of each unique data stream, we should map them with the different RATs. Having the access control plain introduced before, we can have the necessary abstraction and central coordination required to achieve a Software-Defined Networking (SDN)-like approach. A similar heterogeneous approach can be found in [90] where the authors propose the multi-layer Cloud-RAN architecture, where the multi-domain resources can be exploited as needed from vehicle users. Similarly, authors in [91] designed a context-centric heterogeneous system and evaluated different joint multi-RAT user association and resource allocation strategies.

From the above, we can realise the complexity required to design a reliable and robust C-ITS network architecture. This motivated us and in Chapter 3 we kick off our research activities with our proposed multi-RAT 5G architecture for CAVs. Some of the main aspects of a 5G network are the: (i) use of heterogeneous RATs, operating at different frequencies at the both frequency

bands of $< 6\,\mathrm{GHz}$ and $> 24\,\mathrm{GHz}$, (ii) the utilisation of cloud/fog computing capabilities to enhance the system performance providing a powerful computing infrastructure network very close the RSU, (iii) small/large cell architectures to extend the service coverage and network capacity, (iv) beamforming capabilities for directional communications (as in the mmWave standards), etc. Throughout this thesis, we will target the concept of 5G network more from a system level and an architecture perspective rather than focusing on targeted specifications (for example 5G NR). Our solutions build on the ideas introduced in the 5G networks and the fundamental principles behind them, and could potentially be integrated in an actual 5G system in the future.

For our proposed system, we will be taking into account all the above design paradigms and propose a system that is not only flexible enough to accommodate different types of next-generation C-ITS applications but will be able also to meet its demanding QoS requirements. Similar activities have been presented by ETSI (i.e. ETSI ITS Communication architecture – ETSI - EN 302 665 standard [92]) and ISO 21217, that describes the Communications Access for Land Mobiles (CALM) for C-ITS [93]. CALM is a set of standards for C-ITS as described in ISO 21217 [93]. The idea behind both is that of a "hybrid communications framework". More specifically, this implies the utilisation of different wireless technologies in a technology-agnostic way, in order to achieve the necessary level of abstraction required for C-ITS applications. Similarly, in our approach, we build on the idea of heterogeneity. Both ETSI and CALM architectures focus more on how to design a universal router able to provide all the required interfaces for the different technologies. Our novel approach tackles the problem from a system level perspective. Providing the notion of scalable data dissemination as well as Fog computing capabilities, we can open new avenues of providing an abstract way of handling the data that will benefit with the advancements in areas such as Decision Making and AI.

2.5.2 Enabling Large-Scale V2X Communications using the IEEE 802.11p/DSRC Protocol Stack

Having a well-defined system architecture is the first step for a successful deployment. However, delivering such a complex system, like the one introduced before, is not an easy task. We can think of that as the well-known model of the System Development Life Cycle (SDLC), that there are various steps from the preliminary idea and the system analysis until the actual deployment of the system. Throughout the different steps of this model, we will face different bottlenecks and drawbacks that we should address. This will be the idea for the remaining chapters of this thesis, i.e. we will identify different problems within the above system and individually address.

We start with the base layer of our proposed architecture paradigm. CAVs will require an agile interconnecting framework [94] that provides a constant service and optimal system behaviour. The optimisation and the enhancement of this framework require experimental evaluation with real-world trials and computer simulations. In this chapter, we will present our research activities related to the IEEE 802.11p/DSRC protocol stack. We utilised both simulation models and real-

world field trials for that. As we will discuss, there is an established connection between them, that makes both tools equally essential and complementary to each other. Both tools have their advantages and disadvantages, and when interconnected, we can achieve the best possible results from both.

More specifically, the simulation tools are approximations of the physical world. They can easily and inexpensively obtain near-perfect results. For example, different scenarios can be easily validated, as in [95], where the impact of the different beacon intervals is investigated. They can also be easily scaled up, as in [96], to simulate larger scenarios. On the other hand, real-world trials can show the performance in the actual world. This can be done by Commercial Off-The-Self (COTS) devices [97], or by custom testbeds as in our case. However, many physical parameters cannot be easily isolated or ignored, and the real-world testbeds are costly and demanding to be built and managed. In this chapter, we will start our discussions by establishing a hierarchical framework for the field trials and the simulations. Later, we will describe a way of calibrating a simulation framework given specific hardware and setup. Our chosen simulation framework will be OMNeT++ Veins-INET [98] that supports the IEEE 802.11p/DSRC protocol stack and is well-established for simulating vehicular networking scenarios (for e.g. [95, 96]).

One of the main problems with the simulation models is that very complex real-world systems are hard to be modelled and require increased resources to be simulated. The experimentation of very large-scale real-world testbeds is almost impossible to happen, especially in the academic world that the resources are limited. Therefore, the research community of Parallel Discrete Event Simulations (PDES) has tried to leverage from the new high-performance computing platforms in order to decrease the required simulation time (for e.g. [99]) for large-scale experimentation. However, the current implementations are quite laborious and require the reconfiguration of the existing scenarios, pushing the researchers to prefer single-threaded old-fashioned evaluation. We will address that problem, by designing our solution for OMNeT++ Veins-INET that will be able to operate in a multi-threaded fashion, being though entirely transparent to the end-user.

Nonetheless, CAVs will become a reality in the next years, and they will require an actual communication framework to exchange their sensor information. Inspired by that, we will close this chapter by introducing our IEEE 802.11p/DSRC prototyped testbed and evaluating its performance under an extensive experimental campaign that was carried out at the City of Bristol, UK. We will present our preliminary results, the drawbacks of our system and ways to overcome them.

2.5.3   A MmWave MAC-layer Approach for Device-to-Device (D2D) V2X Beam Steering

Previously, we introduced the idea of integrating mmWave links within a C-ITS to achieve ultra-low delays and gigabit-per-second throughput performance. However, the propagation characteristics of mmWaves can lead to performance degradation due to severe shadowing,

intermittent connectivity and increased Doppler spread [100, 101]). Given these limitations, it can be considered sceptical that mmWave bands will be viable for cellular systems and vehicular communications especially across a more extended range and NLOS paths.

Referring to the existing mmWave standards (IEEE 802.11ad and IEEE 802.11ay), the beamforming antenna training requires a bidirectional frame exchange while operating with quasi-omnidirectional antenna patterns in a beam-sweeping manner. It is a known problem that the increased mobility of a vehicle, or a device in general, introduces increased Doppler spread, thus making the traditional beamforming process to fail [66]. For lower speeds and LOS links, the Doppler shift can be corrected with frequency offset correction techniques [102]. For higher speed though, adequate performance still cannot be achieved. Also, the beam-sweeping increases the delay dramatically. According to [103], the response time required from the chip to change the phase and the gain of a phased-array antenna is roughly $\simeq 50$ ns, proving that the beamforming delay is entirely related with the number of frames exchanged.

Adopting mmWaves with C-ITS implies that the increased mobility of CAV will require frequent beam steering. This will lead to significant in-band overhead. Various approaches tried to tackle this problem. For example, authors in [104] introduced a new codebook design scheme able to accelerate the beam training process. Their approach managed to reduce the overhead and improve the system performance; however, the beam training is still performed in quasi-omnidirectional mode. Similar examples can be found in [105, 106] where authors proposed novel algorithms that discover the optimal beam pair and reduce the protocol overhead. Again though, their training is quasi-omnidirectional. For all the above examples, a system with increased mobility will still suffer from the Doppler Spread.

Mobile systems require frequent beam steering. This leads to significant in-band overhead. Leveraging the idea of heterogeneity, zero in-band overhead can be achieved. Authors in [107], train the antenna beams by passively overhearing frames in the legacy band of 2.4/5 GHz and estimating the Angle-of-Arrival (AoA). Though, in dense urban environments, AoA is not accurately estimated due to the multipath effects. Feedback information from a vehicle, sent over IEEE 802.11p/DSRC links in the form of CAMs, can be facilitated to overcome that. In [67], a vehicle transmitting its initial position and speed, provides feedback for the infrastructure-side beamforming. However, position errors were not taken into account, vehicle speed was constant, and no complex manoeuvres were considered limiting the utilisation of the algorithm on a straight-road scenario.

Following the same idea, we designed a MAC-layer algorithm that enhances the V2X beamforming for mmWaves. Compared to the previously introduced algorithms, our approach utilises the scalable data mapped on the different RATs and, fusing the information exchanged via the CAMs, i.e. the position, the motion, and the velocity of a vehicle, makes it able to enhance the performance and operate on a broader scale under more complex road networks. Our novel algorithm will be presented in Chapter 5. Two different algorithms will be presented in this

chapter. One is related to the position error that will maximise the performance based on the given position error. The second one will be based on the distance between an RSU and a CAV. To do so, we derive equations for analysing the position error. Also, and to show the benefit of our algorithm concerning the end-to-end latency, we derive an analytical model that can calculate the beamforming latency given the number of antenna beam sectors.

2.5.4 Efficient Multipoint V2V Communication Scheme for 5G MmWave Hyper-Connected CAVs

MmWave links, even nowadays, due to their increased throughput performance and their propagation characteristics, are usually used for Point-to-Point (P2P) communications and especially as backhaul solutions when the usage of a wired or optical cable is not an option. Two very recent publications discussing this potential scenario are the [108, 109]. In both works, authors discuss the potential of cellular heterogeneous basestation deployment, incorporating mmWaves and another cellular technology. The idea is that the cellular links will be used to exchange information with the user while the mmWaves will be used to connect the different basestations as a mesh network.

MmWave links, given their high directivity and their increased signal attenuation, can be a potential solution for a crowded mesh network of vehicles as they will provide increased spectrum reusability, limiting the amount of interference between adjacent cells. Having a more efficient beamforming algorithm can enhance the V2X link performance as said before and can overcome the beamwidth drawbacks due to the Doppler shift. We mentioned before that CAVs are estimated to generate up to 1 Gbps of sensor data, required to achieve their *birds-eye* view of the road. MmWaves can achieve multi-gigabit-per-second data rates. For example, IEEE 802.11ad can achieve up to 6.75 Gbps of data rate under Line-of-Sight (LOS) conditions [84]. Comparing this value with the previous estimation regarding the sensor data, it is evident that forming P2P links will lead to under-utilisation of the channel.

Also, *emergency CAVs* (e-CAVs), compared to *regular CAVs* (r-CAVs), due to their increased size and speed introduce more driving risks while performing obstacle avoidance manoeuvres and advanced trajectory planning [110]. To that extent, they require prioritised connectivity, without dominating the wireless channel as this may lead to limited access to sensor data for the remaining r-CAVs.

Taking the above into account, we realise that except for an efficient beamforming algorithm, it is necessary to have proper resource allocation for the mmWaves links based on various system parameters. Therefore, in Chapter 6, we will describe a multi-link association scheme for mmWave V2V communications that will be able to utilise the channel better and prioritise the exchange of sensor data between the e-CAVs, without though limiting the acquired information from the r-CAVs. The adoption of mmWaves for V2V vehicular communications is receiving significant attention in the literature. Some examples are the [67, 111, 112], where the authors

analyse different parameters of IEEE 802.11ad and mmWave antennas under V2V scenarios. Researchers also focused on resource allocation schemes for V2V mmWave communications, e.g. [113–115]. Topics vary from the adoption of different scheduling intervals to achieve low-latency and high reliability, to the introduction of sophisticated queue models that will store the generated packets before transmission. However, all the above activities did not take into account the different types of vehicles within a vehicular network and the different types of data that are exchanged. This will be our approach to this problem, and we will base our algorithm on a Matching Theoretic framework [116]. A similar framework can be found in [113], where authors use a many-to-one Gale-Shapley's Deferred Acceptance matching algorithm. Trying to balance the channel state and the queuing state information with the context information generated, they separate the CAVs into groups of transmitters and receivers. Similarly, [114] presents another usage of Gale-Shapley's Deferred Acceptance matching algorithm. The authors take into account the channel state information and transmission queue state information, but again their problem is formulated as a many-to-one association problem. In [117], authors designed an algorithm to share the acquired sensor data in a V2X manner over LTE links [117]. In our work, we will utilise the Stable Fixtures (SF) matching game [118] and will generalise the problem to a many-to-many link association scheme. Our goal will be to increase the channel utilisation taking into account the upper bounds of mmWave channels, the different types of CAVs and their unique requirements.

2.5.5 Efficient Millimetre-Wave Infrastructure Placement for City-Scale ITS

In the two previous sections, we introduced strategies to enhance the performance of mmWaves under V2I and V2V scenarios. Especially for the V2I case, the positioning of a RSU is critical. Correct RSU deployment that meets some QoS standards can significantly enhance the road safety and efficiency in urban vehicular environments [119]. Especially under dense urban environments, mmWaves will significantly improve the performance of the small-cell access networks [120]. As discussed before, and as shown in the literature, mmWave-V2I links have the potential of enabling gigabit-per-second data rates and ultra-low latency [84].

The communication capability between a RSU and a vehicle is highly dependent on the number of deployed RSUs and their coverage range. What is more, RSUs are costly to deploy and maintain. Therefore, compromises between the coverage provided and the deployment costs have to be made. MmWave RSUs especially, are bounded by their LOS requirements and their strict propagation characteristics [121]. However, they are a perfect candidate for small-cell vehicular deployments as they can meet rigid bitrate and latency Quality-of-Service (QoS) constraints needed by next-generation vehicular applications.

Therefore, it is crucial to deploy a number of RSUs in the most suitable locations, to improve the overall network performance. Given the variety of urban environments, it is necessary to find an agile method to obtain the best locations for each street layout and to deploy thousands of

RSUs throughout a city. In Chapter 7, we will propose a strategy that can automate the RSU placement process for different urban scenarios, taking into account the unique road and building layout of an urban environment as well as the strict QoS constraints of vehicular applications. Within a city, traffic lights are usually placed at road intersections. Also, lamp posts are typically equally located along the sides of a road. In this chapter, we assume that our RSUs are deployed on top of lamp posts and traffic lights. Thus, by positioning the RSUs only on the road, we avoid wall and rooftop blockages and hence, improve the network efficiency. Lately, extensive map data providing information about the building and road shapes became readily available for the general public [122]. This allows studies over large-scale scenarios without the need for synthetic data. To observe the differences between the different street and building layouts, we will present two scenarios, one from the area of Manhattan (NY, USA), and the second from Paris (FR).

In our approach, we consider two main Key Performance Indicators (KPIs). The first one is the LOS network coverage achieved after all the chosen RSUs are deployed. The second one is the Received Signal Strength (RSS) averaged throughout the considered deployment area. A similar work can be found in [123], where the authors proposed a strategy for placing IEEE 802.11p RSUs. They considered the delay tolerance of the warning notifications as their optimisation variable and solved their optimisation problem utilising a Genetic Algorithm (GA). GAs are commonly used for basestation deployment (e.g. [123–125]) and more information can be found in [126]. GAs produce a high-quality solution. However, they are computationally expensive, and they may be stuck in local maxima. On the other hand, Greedy Construction (GC) algorithms, commonly used again for similar problems, are scalable but never find a good-quality solution.

In this contribution, we will address our problem through a novel approach to find the best RSU locations by taking into account the propagation characteristics in mmWaves environments. Then, we will compare our outcomes with solutions obtained by running GA and GC algorithms. Our approach manages to fulfil the KPIs we considered before and is not only scalable but can also achieve a reduced number of deployed RSUs for the same area. Similarly to our work, authors in [127] presented an automated base station placement algorithm for mmWaves. However, they did not consider any QoS constraints for their optimisation algorithm apart from the LOS coverage rate. Finally, several works only consider the distance or the propagation characteristics [128, 129], and not the shape of the buildings and the roads. In our case, we will utilise tools from Computational Geometry (as in [130]), and consider all the above to find the desirable RSU locations.

MULTI-RADIO 5G C-ITS ARCHITECTURE FOR CONNECTED AND AUTONOMOUS VEHICLES

3.1 Introduction and Contributions

3.1.1 Introduction

In this chapter, we will introduce a novel C-ITS framework based on a heterogeneous communication infrastructure, able to bring the MaaS paradigm to a whole new level. Our framework enables efficient V2X communications among CAVs by i) handling scalable ITS applications consisting of independent data streams, and mapping onto three different V2X layers, and ii) employing multiple Radio Access Technologies (RATs) to deliver each data stream, according to the specific QoS constraints.

Within our system, our ITS interacts with a Decision-Making agent that is responsible for collecting and fusing different sensor data focusing on the automated driving issues, requiring immediate decision-making. Later, our ITS Agent is responsible for handling the generated data streams and allocate the resources with respect to their geographical relevance and the QoS requirements. We will also incorporate the idea of a Fog Orchestrator and a Fog Computing Infrastructure in our system, giving us the leverage to use a more powerful machine one-hop away from an RSU in order to reduce the introduced latency from the processing of the data and improve the flexibility, manageability and configurability of our system.

We will start our system description with the CAV-based MaaS paradigm introducing three different C-ITS services that are expected to be of paramount importance (Sec. 3.2). We will also briefly present the overall picture of our system. Later, in Sec. 3.3, we will describe our Cooperative Decision-Making Agent very briefly for CAVs and what will be the various interactions with our ITS Agent, introduced in Sec.3.4. More specifically, in Sec. 3.4.1, we will describe the Service

Plane, responsible for the service introduced in the Sec. 3.2, and how the different messages exchanged are associated with these services and the estimated data rate generated. In Sec. 3.4.2, the Scalable Data Plane are described, responsible for handling and grouping the data streams. The Access and the Control Plane of our System will be described in Sec. 3.4.3 commenting about their distinct system components and functionality. Finally, our system description will be concluded in Sec. 3.4.4, that we will introduce the importance of Fog Computing Infrastructure and how this can be incorporated with the different RATs.

3.1.2 Contributions

For the work presented in Chapter 3 the author:

- Illustrated the key components of CAVs, identifying the various types of information data exchanged between vehicles and the infrastructure.

- Decomposed the C-ITS architecture in two different main agents (Cooperative Decision Making Agent / ITS Agent).

- Described the basic features of the Cooperative Decision Making Agent and analysed the message types generated from the various equipped sensors, their data rate requirements and their sharing potential, i.e. what will be the benefit of sharing these information.

- Introduces three example use cases that can be found within a C-ITS paradigm.

- Linked the aforementioned sensor data with the potential RAT technologies to be used (IEEE 802.11p, LTE, mmWaves) and the C-ITS services introduced before.

- Described the proposed ITS Agent decomposing it into its different components: i) Service Plane, ii) Scalable Data Plane, iii) Access Plane, iv) Control Plane.

- Introduces the potential for cooperation between the different RATs, linking this work with the remainder of the thesis.

The contents of this chapter have been accepted for publication in the following Journal:

- EAI Transactions on Industrial Networks and Intelligent Systems 2018 [33].

3.2 Mobility-as-a-Service Paradigm and C-ITS Services

In Chapters 1 and 2, we described that the end-user would access a range of next-generation transportation services aided by accurate navigation and journey information that is expected to vastly improve the road safety and overall the users' travel experience. As discussed, a reliable communication system is an enabler for cooperative MaaS frameworks and a key component in next-generation C-ITSs.

CAVs within this MaaS framework will manoeuvre and plan their trajectory using their equipped sensors. The trajectory planning can be described either as short-term or long-term. An example short-term planning movement is the avoidance of an obstacle, e.g., a pedestrian, that requires a swift reaction time [131]. On the long-term, we find more sophisticated scenarios (e.g., a CAV intending to park at an empty space beside the next roadblock), and the trajectory planning requires the exchange of sensor data with the surrounding environment [132]. Both manoeuvres though will need a *Decision-Making agent* [133]. This agent, using the acquired data, is responsible for the decision-making (e.g., changing lane for better positioning on the road [134], rerouting CAVs due to traffic congestion, etc.).

Each C-ITS service has specific requirements regarding the data that should be acquired and exchanged. For example, a "collision avoidance by cooperative manoeuvring" application benefits from the almost instantaneous exchange of high-resolution LiDAR or camcorder data [135]. Considering that a 3D-LiDAR can generate a data stream of several *hundreds of megabit-per-second*, both the data rate and latency of communication links are crucial fundamental performance indicators (KPIs) in delivering a high precision and low-risk maneuver [136]. On the other hand, information such as the position of the next traffic light or the next parking space is of less significance, requiring a lower data rate and is more tolerant to communication latency.

The diversity of services within a C-ITS requires meticulous architecture design. Before introducing our novel system architecture, we will define three example key next-generation C-ITS services that combine all the above characteristics. We consider these services, as they are expected to play a pivotal role for a novel C-ITS framework:

- *Intelligent Traffic Planning*: Future C-ITS traffic planning services are needed to reroute autonomous vehicles in the event of traffic jams, to coordinate traffic lights for offloading a congested road, or to provide drivers with essential, up-to-date information. Pedestrians can also be benefited and enhance their safety. For example, replacing the normal push-button for signalling and provide a better detection for pedestrians approaching a cross-road, it can overcome the inattention of a driver.

- *Smart Emergency Vehicle Routing*: This service is a specialised case paradigm of the previous one, focusing on the emergency vehicles on the road. It assists them by providing the best route to reach a destination. Typical critical situations correspond to immediate medical assistance or disaster emergency management, such as fire alarms, where police vehicles need to escort the fire trucks promptly. Under these circumstances, the surrounding vehicles must be informed of the approaching emergency vehicles and rerouted to reduce the traffic congestion drastically. Furthermore, a traffic light synchronisation system can minimise the overall journey time. A more complex use case scenario corresponds to a fully autonomous E-Ambulance system, where vehicles are equipped with health monitoring devices, such as wearable sensors, able to transmit the collected patient's data to the hospital or a control centre before reaching their destinations.

- *Multimodal Commuting - MaaS*: This service aims to dynamically and adaptively plan the route of a road user, combining different transportation systems, depending on start and end-points. For instance, an intelligent commuting system can notify drivers of available parking areas, where a shuttle bus service is offered to carry the employees to their work efficiently.

Based on the unique application requirements and taking into account the diverse features of each potential RAT, we have designed our system based on a *scalable data architecture*. The idea of scalability in data originates from the video transmissions where a scalable video stream consists of a *base layer* and multiple *enhancement layers*. The base layer allows the users to achieve a primary reconstruction quality, which is gradually improved as soon as the enhancement layers are successfully received. The same principle trespassed the natural boundaries of multimedia communications and is being applied to design systems capable of an analogue-like service degradation [89].

The different data streams in our system are mapped onto different RATs, and the central coordination takes place in a Software-Defined Networking (SDN)-like approach (Fig. 3.1). The *Decision-Making agent* is responsible for the automated driving applications, which require nearly instantaneous decision-making and reliable communication links. The design of this agent is outside the scope of this work. A brief introduction to the functionality of a decision-making agent will be given later in this chapter. The reader can refer for more details to [133].

In Fig. 3.1 we can see the separation of the data and the control plane in our system. This enables two prominent features. At first, it enables the management capabilities and the dynamic network resource allocation required by a 5G vehicular communication framework [137]. Decisions are taken with respect to various network KPIs, such as the availability of a RAT or the load of a communication link. Utilising QoS information for appropriate network resource reservation is an effective cross-layer approach to achieve guaranteed QoS throughout the entire network. This is very important for our system due to its unpredictability about the load, the number of vehicles and the requirements. Later, centralised control over the heterogeneous network provides a global view and a unified configuration interface. The centralised and automated management and consistent policy enforcement can help to reduce configuration errors. This will also help with the design of comprehensive diagnostic and prognosis tools for automated network maintenance.

In our system, we assume that our network is clustered in different management areas called as *Fog Areas*. Each Fog Area is centrally managed by a *Fog Orchestrator* (FOs) and consists of a number of multi-RAT RSUs and controls multiple CAVs. The FOs represents the logical entities encapsulating the core components of our system. These entities are responsible for taking the automated driving decisions, and mapping the data streams onto the different RATs. The number of constraints in an FO and its building blocks (e.g., microservices) can be many. The FO orchestrates them in either a centrally or distributed manner, meaning that the FO can
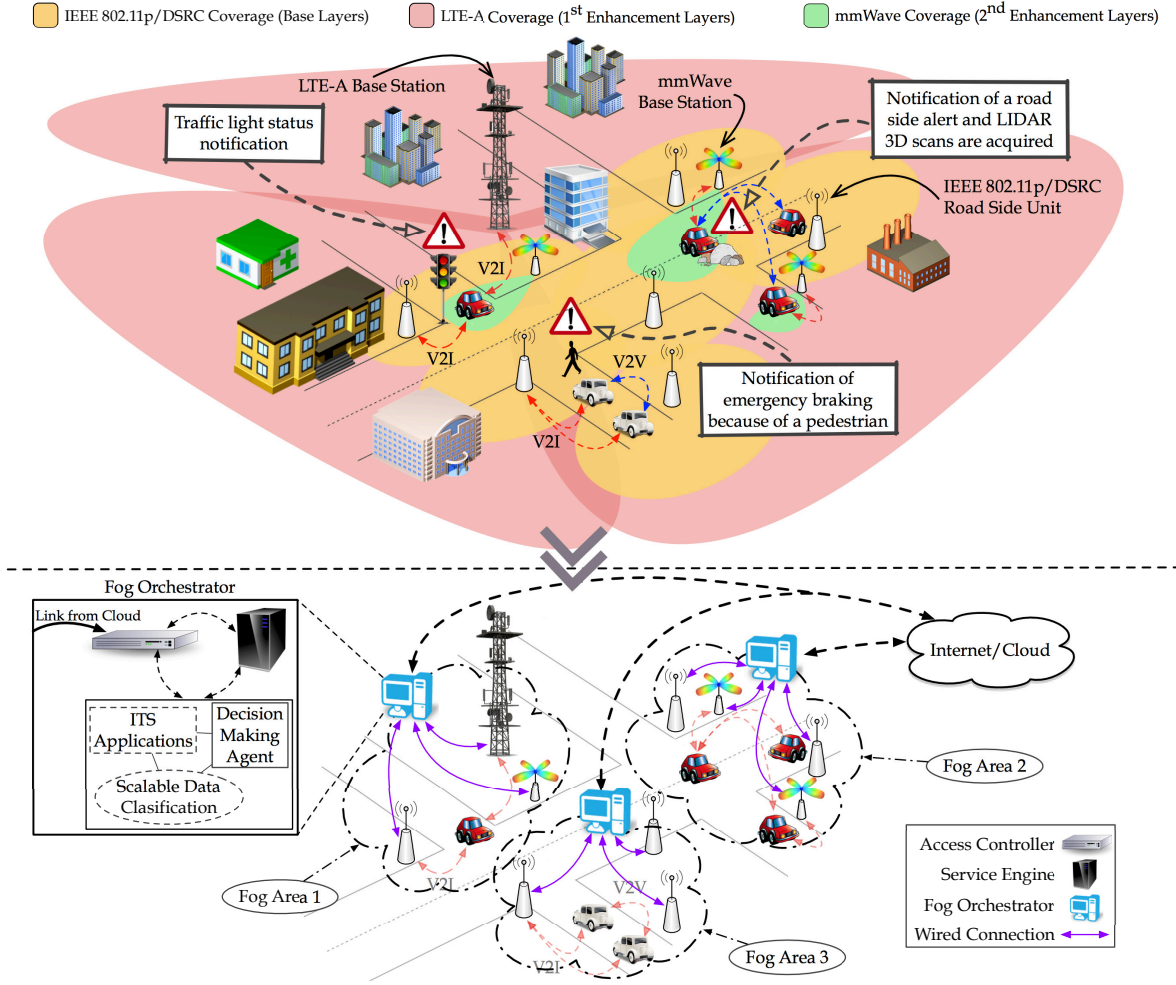
Figure 3.1: General overview of the considered system model. The proposed ITS design framework ensures analogue-like performance degradation communications among CAVs by means of multi-layer V2X communications over an heterogeneous network infrastructure.

start orchestrating services if necessary, maintaining though its core functionality. This Fog node ensures the programmability of the system simplifying the deployment of new functionalities, which rely on a variety of capabilities, locally attached devices, sensors, and actuators. All the above will ensure the QoS required for a specific service.

Utilising this Fog Computing implementation, our centralised controller will be responsible for the generation and processing of all the messages. The different radios will be solely responsible for relaying the message to and from the CAVs. Our envisaged solution still interacts with a cloud-based city-wide connection, which it interfaces with the *Access Controller* in the FO (Fig. 3.1). In particular, the cloud-based service will only be in charge of recording city-scale data, interconnecting the different Fog areas and enforcing city-scale policies to be put into practice.

The above design paradigm can provide the necessary abstraction for a next-generation C-ITS
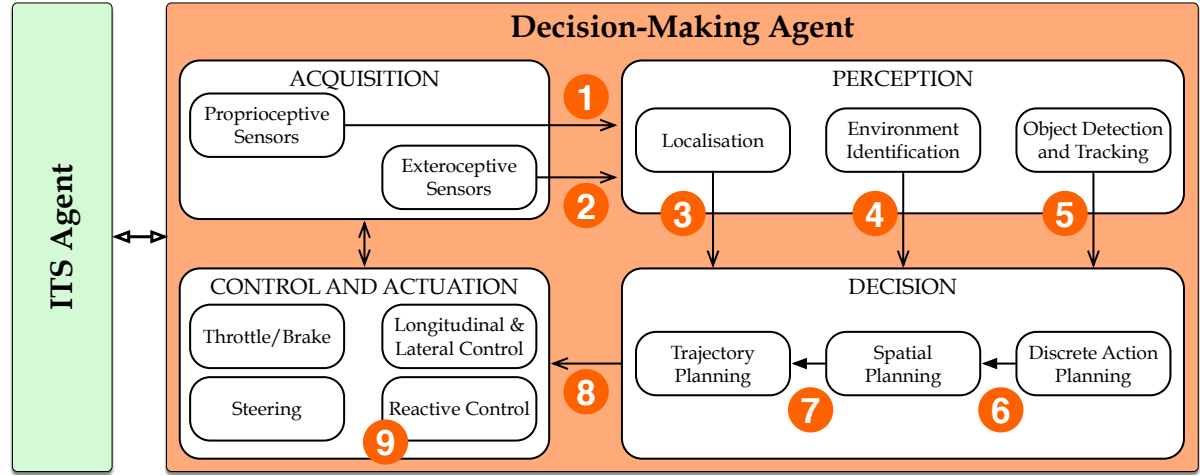
Figure 3.2: Decision-Making Agent architecture.

system. This will give us the leverage to hide the different heterogeneous characteristics of the various RATs and CAVs and simplify the deployment of a new C-ITS service. This system architecture can accommodate next-generation applications with strict QoS constraints bringing us closer to the paradigm of MaaS. In the following sections, we will describe the core network architecture and system components starting with some insights about the Decision-Making agent that will be accommodated by our system.

3.3 Cooperative Decision-Making Agent for CAVs

In an attempt to transform CAVs in C-ITS actors, we refer to the architecture shown in Fig. 3.2. In particular, the proposed CAV design comprises an *ITS agent* and a *Decision-Making agent* [133]. As will be presented in Section 3.4, the ITS agent allows each CAV to exchange sensory data and driving intentions with other CAVs and city-level emergency rooms.

The ITS agent interacts with the Decision-Making agent, which comprises a combination of *proprioceptive* (related to the behaviour of the CAV itself, for e.g., inertial measurements and engine status) and *exteroceptive* (related to the environment outside the CAV, for e.g., cameras or proximity sensors) sensing that begins the control loop structure. Fig. 2.1 shows a suite of possible exteroceptive sensors for a CAV. The *Perception* system processes raw sensory outputs into features describing: (i) the position and movement of the vehicle [138], (ii) the road geometry [139], and (iii) the kind (for e.g., cars, cyclists or pedestrians) and motion of moving objects. The Perception system interacts with the *Decision* system, which takes basic manoeuvring choices (*Discrete action planning*) and determines the path to be followed on a chosen road (*Spatial planning* and *Trajectory planning*) [140]. Finally, *Control and Actuation* system implements the manoeuvring operations needed to follow the chosen trajectory.

The considered Decision-Making agent focuses on automated driving issues, which require

Table 3.1: Characterisation of messages exchanged among the entities within the Decision-Making agent.

No.	Message Type	Example(s)	Data Rate Estimate	Sharing Potentials
1	Proprioceptive sensing	Global Positioning System (GPS)	10 Kbps to 60 Kbps	Simple V2V beaconing will improve reliability of tracking. Could also be augmented with extra information to report accidents or prioritise progress for emergency vehicles.
2	Exteroceptive sensing	Light Detection And Ranging (LiDAR) scans	30 Mbps to 100 Mbps	Sharing raw-sensory data increases field of view, enabling cars to see around corners or remove blind-spot [136]. Also improves the reliability of object detection and tracking.
3	Localization	Fused GPS, Inertial Navigation System (INS) and Simultaneous Localisation and Mapping (SLAM) position	10 Kbps to 800 Kbps	Most simply, this could be implemented by means of the same beaconing as in message type 1. Obtained localisation features could also be shared to improve navigation quality across many CAVs [138].
4	Environment	Driveability grid or road shape	100 Kbps to 10 Mbps	Sharing processed road data (for e.g., classified map grids or highly parametrized road shapes) offers benefits similar to those associated to the adoption of message type 2.
5	Objects	Positions and velocities of other road users	80 Kbps to 800 Kbps	In addition to an enhanced field of view (similar to the adoption of message type 2), improved reliability and accuracy of tracking.
6	Discrete plans	Intended route choice and turns	1 Kbps to 10 Kbps	Roughly equivalent to flashing indicators; short-term route sharing would enable greater cooperation on the road and smoother rides thanks to improved prediction. Knowing the intentions of other road users can improve global performance. May help predict traffic levels and alleviate congestion.
7	Spatial plans	Single path or mesh	80 Kbps to 800 Kbps	Sharing the path enables a simple 'follow the car in front' behaviour, aiding robust decision making, at least as a starting point for a path planner. Can be viewed as an extreme version of what discussed for message type 4, with the road environment distilled to a single driveable path. Could also share spatial planning meshes used to determine routes. Meshes can be very costly to produce but likely to be similar over time and cars [140].
8	Trajectories	Single path with time profile	80 Kbps to 800 Kbps	An enhanced form of what discussed for message type 6; this information improves predictability of other road users, reducing the corresponding conservatism in planning and enabling smoother rides during interactions. Can be considered as feed-forward information, known to damp out instabilities in vehicle platoons [141].
9	Reactions	Emergency brake signal	1.5 Kbps	Possibly as an augmentation of beaconing as in message type 1, rapid sharing of sudden braking or other reactive responses will improve response times in traffic, equivalent to brake lights and hazard light signalling.

45

immediate decision-making, thus reliable communication systems [136, 141]. For each of the
listed message-based interactions identified in Fig. 3.2, Table 3.1 covers the implications of
sharing the information content of these messages via a communication system.

3.4   Proposed ITS Agent Design for Next-Generation CAVs

This section describes the proposed ITS agent design for next-generation C-ITSs featuring CAVs
through heterogeneous RATs, as shown in Fig. 3.3. As said in Sec. 3.3, our ITS agent interacts
with the Decision-Making agent to achieve the ITS service goals, e.g., implementing long-term
driving manoeuvres (ITS Agent to Decision Making agent signalling), or adapts the service goals
in accordance, e.g., detecting a car accident (Decision Making agent to ITS agent signalling).
The responsible FO for a specific fog area, and most importantly the embedded Decision-Making
agent, can directly access all the generated data within this area. Application data from other
city regions or Fog areas can be accessed via the cloud-services communication links.

In Sec. 2.4, we already described the three different wireless technologies that will be in-
corporated in our system. Since IEEE 802.11p/DSRC access network does not rely on any core
networks, this ensures low latency for V2X communications at the cost of reduced coverage. To
this end, it appears natural to relay safety/mission-critical messages via an IEEE 802.11p/DSRC
network when: (i) they are relevant to surrounding vehicles, (ii) are characterised by low data
rate, and (iii) tolerate a *tactile*-like end-to-end latency. For instance, this is the case of messages
signalling that a vehicle triggered its emergency braking system. This information is likely to be
relevant only to the immediate surrounding vehicles. On the other hand, if vehicles get involved
in an accident in the middle of an intersection, this can have a disruptive impact on all the traffic
flow across a large area of the city. In that case, it is worth taking advantage of an LTE-A Pro
network to notify the disruption across geographically larger areas. Whereas, the authorities via
the mmWave communication infrastructure can gain access to the vehicle camcorders or LiDARs
to assess the severity of the accident (see Fig. 3.1).

As discussed in Sec. 3.2, the proposed system paradigm is based on the idea of *scalable data
dissemination*. The data streams admission control and the RAT resource allocation are part
of the ITS agent. For the proposed system architecture one base layer and two enhancement
layers will be considered. For the remainder of this chapter, we will describe the key ITS agent
components and their fundamental operations.

3.4.1   Service Plane

The Service Plane is the place where all the next-generation ITS services and applications are
developed. We can use as examples the C-ITS applications described in Sec. 3.2. The Service Plane
operates regardless of how the data streams will be mapped onto multiple layers, and regardless
of which RAT will be used to transmit each stream. With regard to the above mentioned ITS

(a) Structure of an ITS agent.

(b) Detail of the Access Plane.
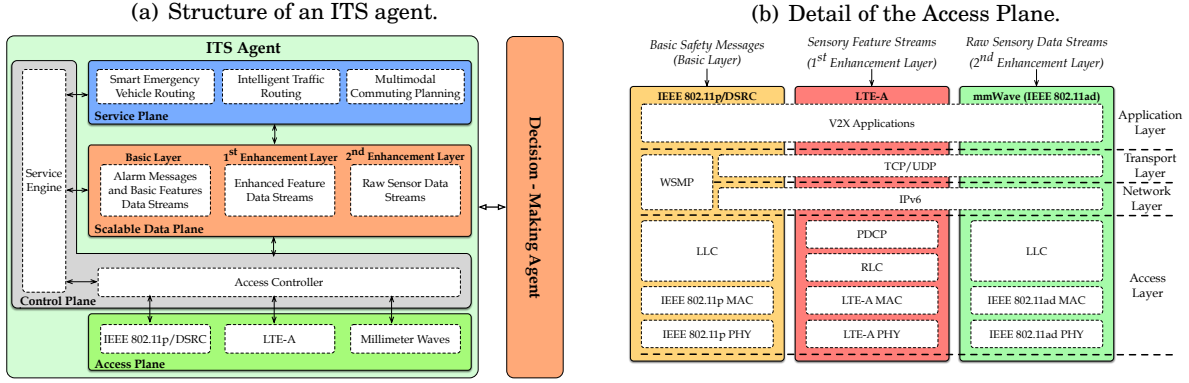
Figure 3.3: Proposed ITS agent architecture and detail of the proposed Access Plane incorporating the considered RATs.

applications, Table 3.2 considers some examples of data streams being exchanged, their impact on the ITS service and the estimated amount of sensor data to be transmitted/received. Of course, each application can generate more message types than the one presented in Table 3.2, but from that, we can get an indication about the different requirements of each service and each message type specifically. As described in Section 3.3, each C-ITS service has its unique requirements with regard to the sensor data to be exchanged. The above figures of merit prove the necessity for a scalable data plane design and a flexible system architecture to accommodate these needs.

3.4.2 Scalable Data Plane

In the proposed system design, the Service Plane builds upon a Scalable Data Plane, which is in charge of:

- Handling data streams from the lower system planes and dispatching them towards the Service Plane;

- Grouping data streams by their geographical relevance and QoS requirements.

The Scalable Data Plane assumes that each next-generation C-ITS service exchanges scalable data streams that comprise up to three independent data layers. The data streams mapped onto the base layer convey messages for Road Safety Applications in a V2X fashion. These messages are described in SAE J2735 and are called Basic Safety Messages (BSMs), as defined by the Society of Automotive Engineers (SAE), or CAMs, as defined by the European Telecommunications Standards Institute (ETSI). The above-mentioned BSMs or CAMs, are being broadcast every 0.1 s and contain core vehicle information, such as vehicle size, GPS location, braking system status, etc. Along with the other messages, (e.g., Intersection Collision Avoidance, Road Side Alert, etc. described in SAE J2735), they can allow the proposed C-ITS design to support basic safety/mission-critical C-ITS functionalities, such as support to navigation, obstacle avoidance,

47

Table 3.2: Relevant messages for next-generation ITS Services.

ITS Services	Example Message Types	Impact on ITS Services	Estimated Data Rate
Intelligent Traffic Planning	City-wide map grids and road shape reports.	Enabling the origin-to-destination long-term journey planning	10 Kbps to 10 Mbps
	City-wide knowledge of CAV positions	Congestion prevention	10 Kbps to 800 Kbps
	Routes and destination in low resolution	Congestion prediction and high level rerouting	80 Kbps to 800 Kbps
Smart Emergency Vehicle Routing	LiDAR sensor raw data streams exchanged and processed in real-time	Precise high-mobility manoeuvres	50 Mbps to 250 Mbps
	Accurate representation of the nearby moving obstacles	Decision making through accurate object tracking	80 Kbps to 800 Kbps
	Trajectory paths with time profiles	Enhanced cooperation between CAVs improving long-term manoeuvre smoothness	80 Kbps to 800 Kbps
Multimodal Commuting	Available parking spaces in close proximity	Reducing the overall commuting time	10 Kbps to 10 Mbps
	City-wide knowledge of CAV positions	Refinement of the expected arrival time	10 Kbps to 800 Kbps
	Information on road disruptions (for e.g., accidents, adverse weather conditions)	Smart CAVs rerouting	30 Kbps to 100 Kbps

traffic light status notification, etc. Given their low data rate, nearly *tactile*-like latency, and local relevance, the V2X base layers are transmitted over IEEE 802.11p/DSRC communication links.

The considered next-generation C-ITS applications described in Sec. 3.2, build upon those above basic ITS functionalities. They are expected to impact on potentially large areas of a city. The vehicles involved in these services will generate *feature streams* from their onboard sensors, which correspond to the V2X first enhancement layer in the proposed ITS design. An example of a feature stream is given by the 3D bounding box representation of objects surrounding each vehicle [142]. These data are being processed using sensor data fusion techniques and are independently generated by each CAV [142]. These highly refined information streams need to be shared among a large number of vehicles to allow them to take long-term decisions. In particular, the first enhancement layer requires a communication system providing extensive coverage and links capable of megabits-per-second, without though the need for *tactile*-like latencies. Hence, we refer to LTE-A Pro as the proposed RAT for this layer.

As discussed, the first enhancement layer is the result of sensory data processing carried out by each vehicle independently. Considering the Smart Emergency Vehicle Routing application, in the case of large-scale accidents, city-level emergency rooms may find it convenient to gather raw sensor data from multiple vehicles, combine them and then extract the required features. In fact,

in the case of LiDAR data, combining raw data acquired from different locations can eliminate multiple blind spots and lead to a more accurate 3D bounding box object representation [143]. Streams of raw sensory data define the V2I second enhancement layer. Transmitting raw sensory data requires communication links capable of a gigabit-per-second, which implies the adoption of the mmWave infrastructure. However, due to the intermittent connectivity level associated with this technology, no reliability constraints should be associated with the second enhancement layer.

3.4.3 Access and Control Planes

To guarantee high system adaptability, controllability and flexibility, it is necessary to separate the Control Plane, responsible for the data stream admission control and the RAT selection, with the Access Plane, that incorporates all the available RATs. As discussed in Section 3.3, this approach is mostly adopted in the SDN domain, where decoupling the network functionalities from the RAT in use enables high system programmability, and abstracts the network infrastructure from the different services [137].

In our system particularly, the Access Plane encapsulates the three proposed RATs, building a network operating in a heterogeneous manner, capable of operating under various conditions and fulfilling various QoS profiles. As shown in Fig. 3.3(b), each of the adopted standards is characterised by a different protocol stack. The PHY and MAC layer of these standards have been described in Sec. 2.4. IEEE 802.11p/DSRC and IEEE 802.11ad mmWaves include the typical Logical Link Control (LLC) layer in charge of flow and error control, whereas LTE-A adopts the Radio Link Control (RLC), responsible for error detection and recovery, and the Packet Data Convergence Protocol (PDCP) for packet integrity protection and header compression. At the Transport and Network layer, IEEE 802.11p/DSRC supports apart from the IP protocol the Wireless Access in Vehicular Environments family of protocols as well. This is very important as Short Message Protocol (WSMP), supporting the one-hop broadcast transmission of high priority and time sensitive data messages.

To achieve the necessary level of abstraction, we consider a Control Plane consisting of two distinct system components: the *Service Engine* and the *Access Controller*. The former is responsible for classifying the incoming data streams based on the Scalable Data Plane configuration, while a decision algorithm determines which data stream must be processed. The *Access Controller* then chooses the appropriate RAT based on the decisions made by the *Service Engine* in accordance with the Scalable Data Plane, as discussed in Section 3.4.2, as well as taking into account the availability of the chosen technology. Both system components are integrated within the logical entity of the FO.

3.4.4 Cooperation between Different RATs

In the previous sections, we described the key components of the proposed multi-radio 5G architecture paradigm, and we established a connection between the different scalable data layers and the individual RATs. The abstraction and virtualisation introduced by this design, as discussed, can pave the way for more reliable ITS applications and services. Building upon the flexibility provided by this system, we can enhance the performance of each RAT even further and overall, provide a more robust system.

More specifically, the cooperation between different RATs can be enabled within a fog area. The FO can utilise the already exchanged ITS application data to fine-tune each radio. For example, a significant fraction of the Beacon Intervals (BIs) introduced in IEEE 802.11ad is occupied by the necessary beamforming operations between the devices. In particular, as observed in [36], about 1/3 of the BI length is to be allocated for beamforming operations. However, by taking advantage of the CAMs (exchanged via IEEE 801.11p/DSRC), which encapsulate position and heading information of a CAV, our FO unit can track a vehicle on the road, inform the mmWave RSU about the estimated position of the vehicle and provide the necessary and pre-calculated beamforming information. This has the potential of improving the network throughput [36, 37].

By following a similar approach, FOs can contribute to delivering a system with advance user-authentication functionalities [144]. In fact, users' credentials can be generated and processed by each FOs rather than in a single centralised node. Similarly, FOs can also be used to implement advanced PHY layer security features, for instance, based on random network coding strategies [145, 146]. Overall, the Fog computing implementation within our system gives us the leverage that using small, powerful units within our FO, is possible to virtualise services one-hop away from an RSU. This will reduce the processing latency in our system, improving the configurability and the flexibility required for the high demanding next-generation C-ITS applications.

3.5 Conclusions

In this chapter, we introduced a novel C-ITS multi-RAT architecture, able to enhance the CAV-related MaaS paradigm. More specifically, our system can accommodate different streams of data, intelligently allocated the resources of each RAT, and mapping them with respect to the geographical and QoS constraints. Our framework is designed in an SDN-like fashion incorporating Fog Computing capabilities as well, to enhance the performance and minimise the end-to-end latency. The distinct system components, being the Decision-Making Agent and the ITS Agent, as well as their sub-components, can give us a more unified view of the system and improve the flexibility, the manageability and the configurability, aspects essential for a C-ITS framework.

This work is envisioned to play a pivotal role in the design and deployment of the next-

generation C-ITS frameworks. Of course, one of the main challenges is the difficulty of such a system to be designed and implemented. This is because the different system components should be precisely designed, tested and debugged before being integrated, being an effort of many years of research. Having said that, and keeping all the above in mind we will focus our research activities on some more specific subsections of this system. These activities will be introduced in the next four chapters of this thesis, having as a short-term goal to enhance the performance of some distinct components and an ultimate goal for the future, a complete system implementation.

Enabling Large-Scale V2X Communications using the IEEE 802.11p/DSRC Protocol Stack

4.1 Introduction and Contributions

4.1.1 Introduction

Chapter 4 will present our efforts to accurately and cost-effectively evaluate a large-scale V2X communication framework using the IEEE 802.11p/DSRC protocol stack. There are two ways of evaluating an experimental framework: 1) with simulation studies, 2) with real-world trials. In this chapter, we will describe the interconnection between these two methods and why both of them are necessary for accurate performance evaluation. Later, we will present a calibration procedure, designed for OMNeT++ Veins-INET simulation framework, that can enhance our performance investigation and bring us closer to a realistic representation of the reality. However, the more representative of the reality a simulation framework is, the more computationally expensive it becomes. To that extent, we will describe our solutions for reducing the required simulation time. To do so, we modified our simulation framework and operated it in a multi-thread fashion. Even though, parallelised versions of the software already exist, they are quite laborious and require the reconfiguration of the existing scenarios. Our amendments will ensure the seamless integration with the existing scenarios and easy reconfiguration to speed up the execution time.

Except our efforts on accurately and cost-effectively simulate a vehicular networking scenario, we will also investigate the requirements for an actual real-world deployment. We will start by describing the features required for an experimental testbed in order to fulfil the demanding requirements of vehicular communications. Building on that, we will present our efforts on building an experimental IEEE 802.11p/DSRC testbed. We based our prototyped system on

open-source firmware and low-cost hardware components, having in mind a cost-efficient large-scale deployment around the City of Bristol. Later, we will present the evaluation of our testbed throughout three days of field trials. Our testbed logs all the messages generated and exchanged in a V2I and V2V fashion. Accessing the logged information, we will discuss our findings and our experiences from these trials.

More specifically, in Sec. 4.2, we describe the benefits and the limitations of both evaluation methods and raise the questions answered in the rest of the chapter. Later (Sec. 4.3), we establish the hierarchical framework between the trials, the simulations and the reality-of-interest, discussing what the interrelationship between these entities is. We continue with presenting our designed and developed experimental testbed (Sec. 4.4). In Sec. 4.5 we describe the agile calibration procedure we followed, to microscopically calibrate our full-stack network simulation. Furthermore, in Sec. 4.6, we discuss ways of parallelising the above framework, in order to speed up its execution time and make it a viable option for large-scale experimentation. In Sec. 4.7, we start our performance evaluation, describing at first the results acquired from our fine-tuned simulation. Secondly, we present the improvement in the simulation time, achieved by operating the simulation framework in a parallel fashion. Finally, we introduce the initial performance investigation of our experimental testbed under a large-scale experimental campaign around the City of Bristol. Based on the knowledge acquired from the aforementioned field trials, we later identify the drawbacks that should be addressed in the future. We conclude this chapter in Sec. 4.8, with a summary of the above some critical thinking about our findings.

4.1.2 Contributions

The contribution of the author for the work presented in Chapter 4 is:

- Established the connection and the hierarchical framework between the trials, the simulations and the reality of interest.

- Actively contributed with the debugging, and the design of the experimental devices introduced (as part of the VENTURER Project).

- Performed the preliminary evaluation of the experimental testbed using the channel emulator, designing and conducting both experiments related to that (throughput and video PSNR evaluations).

- Defined a calibration procedure for Veins-INET simulation framework, taking into account the different parameters from the real world (as part of the VENTURER Project).

- Designed and implemented the multi-threaded version of Veins-INET.

- Was involved in the experiments related to the agile calibration of Veins-INET.

- Designed and simulated all the scenarios described in this chapter, for both the calibration procedure and the parallel implementation (as part of the VENTURER Project).

- Was actively involved in the design and deployment of the testbed presented.

- Was actively involved in the design of the large-scale experimental campaign described (as part of the Flourish Project).

- Was part of the team that conducted the large-scale experimental campaign around the City of Bristol (as part of the Flourish Project).

- Designed the website that hosts the dataset of the field trials [147].

A subset of the contributions and the results of this chapter was presented at the conferences:

- IEEE Vehicular Networking Conference 2017 (IEEE VNC 2017) [34].

- International Conference on Ad Hoc Networks and Wireless – AdHoc-Now 2018 [35].

- IEEE Vehicular Networking Conference 2018 (IEEE VNC 2018).

4.2 Interconnection between Trials and Simulations

As we discussed in the previous chapters, the communication framework for CAVs is essential. The exchanged information can increase the vehicle safety, provide new services, reduce traffic jams improving the fleet routing, etc. This is one of the main reasons that 75 MHz of the spectrum have been allocated for DSRC in the frequency band of 5.9 GHz. Performance enhancement in this frequency band is a hot research topic as CAVs require an agile interconnecting framework [84, 94], able to provide a constant service and optimal system behaviour.

Concerning the performance evaluation of novel algorithms introduced by the research community, we find that the common tools used are either a combination of theoretical models and simulations, or real-world field trials. As discussed in Sec. 2.5.2 the simulations are based on approximations of the physical world, being able though to obtain near-perfect results, quickly and inexpensively. They provide a high degree of flexibility, making it possible to test different configurations and isolate specific parameters, in order to examine a system under specific conditions. One of the main problems of the existing simulation frameworks is their scalability. A very complex real-world system design is hard to model and requires increased resources to be simulated, resources that even high-performance computing platforms are not always able to provide.

On the other hand, real-world trials are based on a "perfect" model. Conducting a real-world experiment, we can observe the behaviour of a system, under a particular scenario taking into account all the disruptive or constructive parameters of an environment. These parameters

are not easily considered within a simulation framework. For the same reasons, conducting large-scale experiments, we know that the results acquired are an actual representation of the reality-of-interest. However, the wireless environments, and more specifically the vehicular networks are very dynamic. Some of the main disadvantages of real-world trials are the inability to reproduce the results, as the physical parameters are not easily isolated and ignored, the cost, and the required time.

All the above is the main reason that researchers usually prefer one of the two evaluation methods. The question that arises though is: *"Is it necessary to have an interconnected framework between the real-world trials and the simulations?"*. The short answer to that is: *"Yes, it is essential in order to accurately, time- and cost-effectively evaluate the performance of a novel strategy."*. To achieve this interrelationship between these two distinct entities, we need to establish the hierarchical control system between them and find the links between the different subsystems. Having a hierarchical system representation, we can identify the different interactions between the various entities, introducing any abstractions when required. By the above, we can achieve the "accuracy" mentioned in the answer above. We will further discuss this approach in the next sections, starting from Sec. 4.3.

4.3 Hierarchical Framework: Trials and Simulations

Consider a C-ITS consisting of a number of RSUs and CAVs on a road network. The different services running on the C-ITS, as well as the safety-critical and the infotainment-related applications, exchange various kinds of data. For example, safety-critical messages denoted as CAMs in the ETSI Standards [148] or as WAVE Short Messages (WSMs) in the IETF standards [149], are broadcast with a frequency of 10 Hz. These messages encapsulate core information about the vehicles, such as the position, the velocity, the size, etc. Similar safety-related messages, regarding road hazard warnings (e.g. the DENM within the ETSI Standards [150]) can be triggered any time within the network. The above messages are relatively short (~300-800 B) and are transmitted using QPSK ½ MCS. The above messages require low one-hop end-to-end delay and high-delivery data rates.

Apart from the above messages, safety-related applications in the future C-ITSs may generate different kinds of data. Example applications can be the video-assisted overtaking or the traffic monitoring using road-mounted camcorders. These applications are already tested on CAVs [151] and require the transmission of video frames via UDP streams. Their main QoS constraints are the increased data rate and the low jitter. These applications are more forgiving with the BER performance, due to the new generations higher efficiency video encoders and the adoption of Forward Error Correction (FEC) techniques. Even the non-safety related applications may have some strict QoS constraints. For example, a map database or a geolocation information application downloading via FTP servers requires zero packet-loss but with lenient latency requirements.
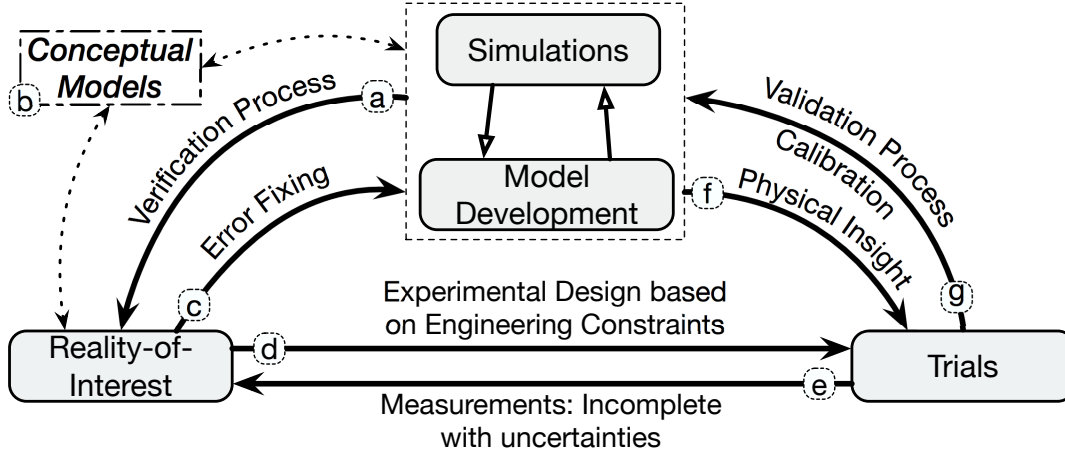
Figure 4.1: The interrelationship between the experiments, the simulations and the real world as well as the processes that connect them.

Each application, except its unique content-related QoS requirements, behaves differently under various physical environments. In Sec. 2.1 we described the three different environments we find in vehicular networks, i.e. urban, suburban, rural, their unique characteristics and the way they affect a particular network. For instance, an experiment within a city will be affected by blockages from the buildings, that significantly attenuate the signal. Though urban canyon behaviours can be introduced, under specific circumstances, wave-guiding the signal. On the other hand, foliage is the primary form of blockage in rural areas while vehicles tend to move faster introducing a more noticeable Doppler Shift. Given the above, each environment should be addressed differently under a simulation scenario. For example, an accurate representation of the buildings/obstacles within our simulation framework is necessary in order to find the intersections of the signal with the buildings. Also, a proper channel model configuration will give us the correct signal attenuation for the above intersections. From all the above, it is evident that a vehicular network is a complex entity within a C-ITS. To correctly evaluate the network performance, careful consideration of the different parameters is required as well as a thorough investigation of the different network characteristics.

4.3.1 Cooperation and Coexistence of Trials and Simulations

Cooperation between simulations and trials is mandatory to enhance the accuracy of a system performance validation. The exchange of information between them can significantly improve the results while minimising the required cost and time. Establishing a framework between the utilised *simulation models*, the *trials* conducted and the *reality-of-interest*, i.e. the part of the real world (e.g. a city, a neighbourhood or a road) that we are investigating, will help us to understand the requirements and the limitations of each one better. The interrelationship between these three entities can be seen in Fig. 4.1.

The assessment of the simulation accuracy is divided in two phases, i.e. the *verification* and
the *validation* [152]. The verification (Fig. 4.1 – a) is the confirmation that a model is correctly
implemented and reflects the real world. This is confirmed by conceptual models – abstractions
of a system that characterise and standardise a network function, e.g. the OSI model (Fig. 4.1 –
b). Verifying a model, the existing errors can be determined and fixed to assure that it matches
specifications and assumptions with respect to the initial concept (Fig. 4.1 – c). Throughout
this chapter, we will not address the model verification for Veins-INET any further. The models
implemented in Veins and INET Frameworks have been thoroughly verified in several works by
the research community, e.g. [153–156], etc.

Apart from the simulation-based evaluation, we can evaluate the performance of a system
using real-world trials. A trial-based evaluation is limited by the engineering constraints (Fig. 4.1
– d). For example, a NIC might support decreased transmission power compared to the standard
specifications, limiting the operational range of a device. What is more, trials suffer from un-
certainties (e.g. the attenuation of the signal with respect to the weather is unpredictable, not
easily measured and environment-dependent) affecting the reliability and the validity of the
experiment. The replicability of the experiment is also a big concern. All roads are not the same,
and the devices have different specifications, so replicating an experiment is difficult. This leads
to incomplete results (Fig. 4.1 – e) as it is impossible to validate all possible combinations.

The validation of a theoretical model assesses the precision of a model when compared with
the behaviour of the real world (Fig. 4.1 – f). The verification of a model usually precedes its
validation. The simulation models are validated using simple experiments and isolating the
external factors that affect the performance. A meaningful representation of the real world can be
achieved by fine-tuning a simulation framework at first (Fig. 4.1 – g). This is done by using inputs
from measured results (e.g. path loss exponent) or applying weights at the output to minimise the
divergence error. A direct comparison between the absolute values of the experimental results is
misleading and should be avoided. The validation process should identify and compare the trends
of the performance for the different experiments (e.g. both the simulated and trial-based results
have a relative degradation when one parameter is changed).

4.3.2 Hierarchical Validation of a Simulation Model

Inconsistencies between the simulated and real-world results can arise for several reasons.
Various examples can be the *measurement errors* (e.g. noise from the data acquisition methods,
calibration errors, etc.), *formulation errors* (e.g. incorrect channel models) or even *numerical
errors* due to the floating point units of a computer (e.g. subtraction of floating points, overflow of
integers, etc.). Overall, we can categorise the errors in two different groups. The *random* errors
that affect the relative precision of the simulation models, and the *systematic* errors, that affect
the absolute value of a result. The second group of errors is easily repeatable and therefore
predictable. To that extent, it can quickly be taken into account introducing a counteract that

will cancel its effect in a system.

The necessary level of accuracy can only be achieved by applying the validation procedure throughout the entire development process following a hierarchical approach [157]. To do so, we fragment the bigger problems into smaller entities and solve them individually. By doing that, we can achieve the required level of accuracy with low complexity. Generally, the trials should test the essential features of the simulation models, such as the impact of the considered assumptions or the simplifications. On the other hand, simulations allow incremental validation towards a "real-world-like" system.

The hierarchical validation, however, is not a one-way process. Problems identified at more complex systems may be more appropriate to be tackled in smaller scenarios, or the other way around. It is required to well-define the boundaries and the initial conditions in order to make it easier to identify the uncertainties later. When the individual errors and the inconsistencies have been addressed, and the required level of accuracy is achieved, we can proceed to complete system validation. For that, the unpredictability of various parameters creates the necessity for more extended and increased number of experiments and simulations to achieve a clear understanding of the real-world behaviour.

4.4 Experimental Testbed Architecture

The design of real-world ITS solution is a complicated procedure. All the above should be taken into account throughout the entire design and implementation process. Also, considering the initial conditions, the boundaries and the trends of the performance is of paramount importance. When fragmenting the system into smaller entities and isolating the characteristics that disruptively affect the performance, it can be easier to identify the random and the systematic errors. Identifying them, we can approach an "ideal-like" system. Before using a network simulator to validate different scenarios, it is necessary to configure it correctly. As said, there is a direct connection between the real-world trials and the simulations. So, in this section, and before proceeding with the validation of our simulation framework, we will present a real-world testbed that we designed, implemented, and deployed around the City of Bristol. This testbed is capable of providing both V2I and V2V communication links and can be deployed on a large scale. When designing this testbed, we were interested in the following features:

- Continuous availability for delay-critical applications.

- Full-stack system implementation to support various vehicular applications and dissemination of scalable data types.

- Centralised coordination via a SDN-like framework.

- Open-source operating system to be easily customisable and compatible with Fog and Cloud Computing architectures [158].
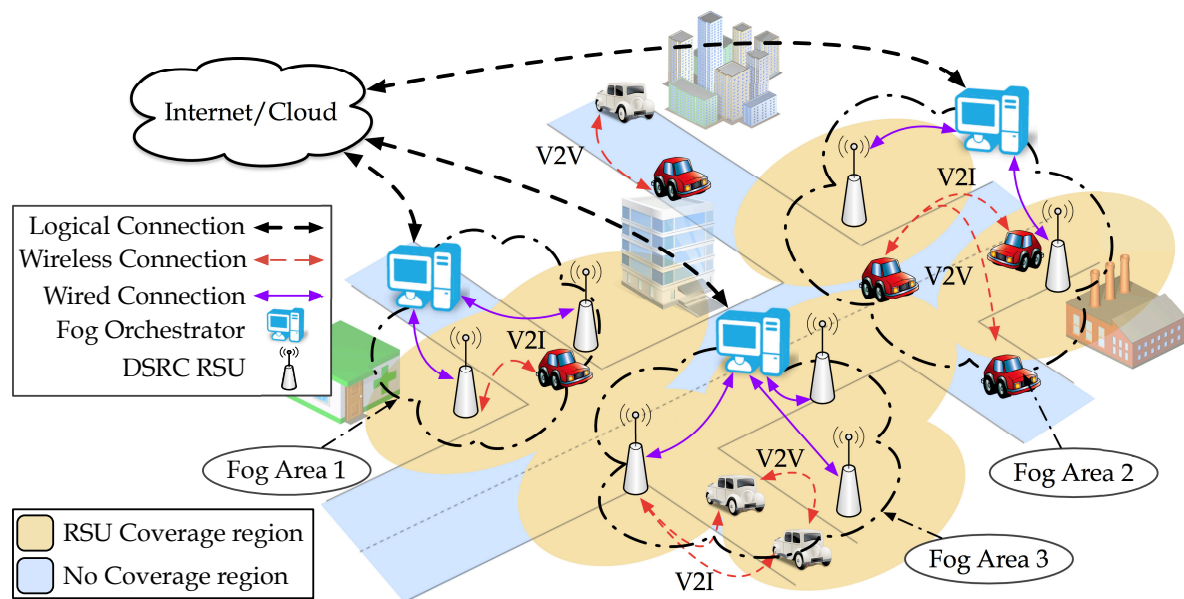
Figure 4.2: A general overview of the considered system model. The C-ITS design framework
ensures V2X connectivity and an NFV architecture in the infrastructure domain.

- Dual-operation as RSU and OBU to support both V2I and V2V links.

- Low-cost for large-scale deployments.

- Weatherproof.

In the next section, we will describe our ideal experimental VANET testbed, describing the
different characteristics and features. Finally, in Sec. 4.4.2, we describe the system that we
designed, deployed and implemented in the City of Bristol. This system was utilised for the
experiments that will be described in the remaining of this chapter.

4.4.1   Description of an Ideal System Architecture

Our ideal experimental VANET testbed consists of various devices and entities. Each one will play
a significant role in the operation of our system. An ideal design paradigm is shown in Fig. 4.2,
where the different devices and entities are presented. Our system paradigm, as discussed before
will be consisted of three main entities: the RSUs, the OBUs, and the *Fog Orchestrators (FOs)*.
The RSUs will be responsible for providing V2I connectivity and will be mounted on building
walls and connected to a centralised control plane. The OBUs, installed in the vehicles, will
exchange vehicular application data with surrounding RSUs and OBUs. Finally, the FOs are
devices that can centrally manage different clustered areas, named as *Fog Areas*. These areas
are ideally within one-hop distance from an RSU to reduce the end-to-end delays.

The RSUs and the OBUs will form IEEE 802.11p/DSRC links between them, as shown in Fig. 4.2. Throughout the city, we have coverage and non-coverage regions, where a vehicle can respectively connect to both the RSUs and the OBUs or just with the surrounding vehicles. We also assume that our infrastructure network is clustered in different management areas called *Fog Areas*. A Fog Area is managed by an FO. All the FOs share a direct wired link with the different RSUs within the same area. Being one-hop away from all the RSUs, they can be used to process all the time-critical information received or generated at the infrastructure side and reduce the end-to-end delay. In our system paradigm, each FO interacts with a cloud-based service. The cloud-based service will be responsible for recording city-scale data, interconnecting the different FOs and Fog Areas and pushing city-scale policies in the entire network.

We regard the above system as capable of exchanging any data that may be required for safety and non-safety critical vehicular applications. We are interested more particularly in the exchange of CAMs on both V2I and V2V links and UDP data streams for our performance investigation. More specifically, later in this chapter, we will describe two different experiments conducted. The first is large-scale experimentation using CAMs, while the second is a smaller-scale scenario utilising both CAMs and UDP packets. In the next section, we will describe in greater detail the testbed components that we have already designed and deployed around the City of Bristol. As our work stands at the moment, the FOs and the Fog areas were not implemented and are being only part of our ideal system solution. We consider them as a task for our future research activities. In the next section, we describe in more detail the experimental testbed that we designed.

4.4.2 Description of our Experimental Setup

For our experimental validation, we prototyped an open-source IEEE 802.11p/DSRC testbed (Fig. 4.3). Our devices consist of a single-board computer and two wireless cards connected to their accompanied antennas. More specifically, we used a Mikrotik RB433 single-board computer (CPU 300 MHz, 64 MB RAM, 64 MB storage space, x3 Ethernet slots, x3 MiniPCI slots) [159], acting as the main processing unit for our devices. As for the NICs, we utilised two IEEE 802.11a wireless cards for redundancy. The first one was a Mikrotik R52H [160], regarded as a low-power (LP) NIC and operating at 25 dBm. The second wireless card, being the high-power (HP) transceiver, is a Mikrotik R5SHPn [161] with 29 dBm maximum transmission power. The wireless interfaces of the RSUs and the OBUs in our system are accompanied by different antennas as shown in Figs. 4.3(a) and 4.3(c), one bolted on the RSUs and the second magnetically attached to the roof of our vehicles. The RSU NICs were connected to a dipole antenna with a gain of 7 dBi and 9 dBi for the LP and the HP transceivers respectively, while the OBU antennas have a gain of 5 dBi. Our RSU devices were powered up via Power-over-Ethernet (PoE), while a battery pack was used for the OBUs to avoid the voltage spikes experienced when using a lighter inverter within the vehicle. All the device and the key driver characteristics can be found in Table 4.1.

61
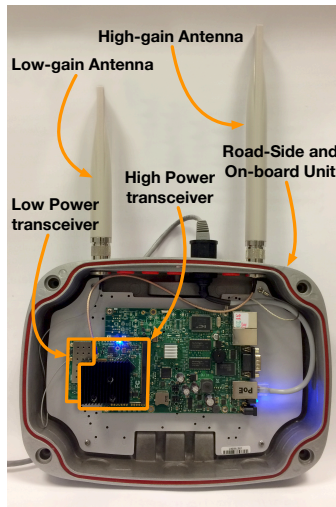
(a) IEEE 802.11p / DSRC RSU units.
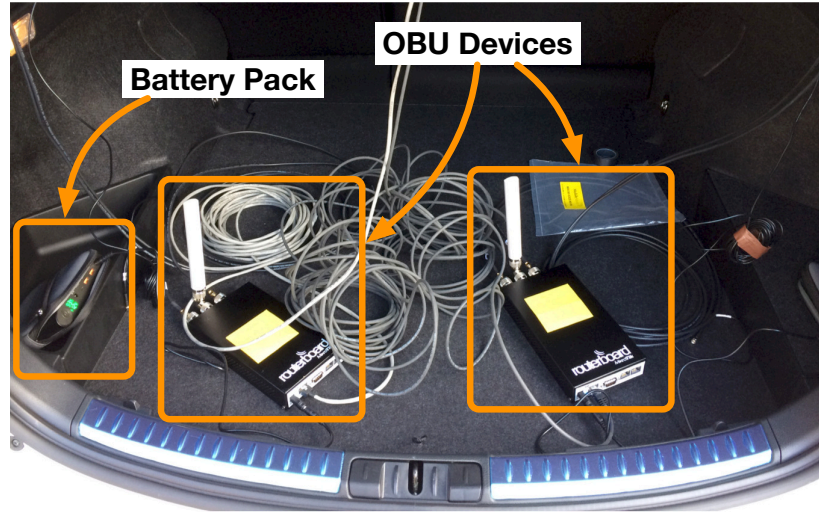
(b) IEEE 802.11p / DSRC OBU units.
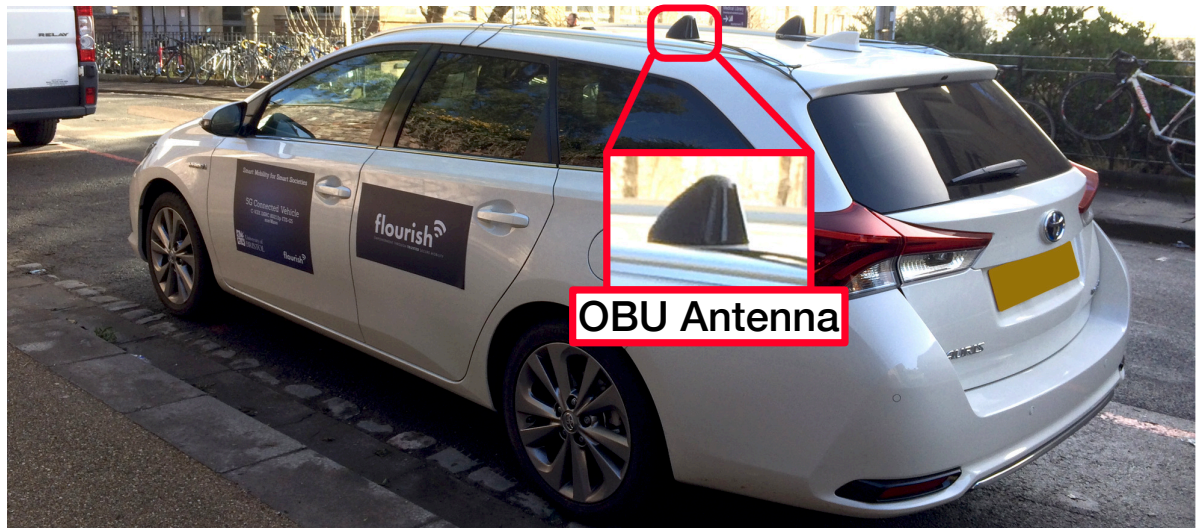
(c) OBU antenna mounted on the roof of the car.

Figure 4.3: Our experimental testbed. We prototyped both RSUs and OBUs units, equipped them with different antennas and conducted our experimental trials around the City of Bristol.

A low-latency OpenWRT Linux distribution was used[1] as the operating system. The different Atheros chipsets of each transceiver (AR5414 for the LP and AR9220 for the HP) required the use of two different Atheros drivers (ath5k for the LP and ath9k for the HP). Both were modified accordingly to enable IEEE 802.11p compatibility adding the 5.9 GHz band to the regulatory domain. The Linux kernel modules that we modified have been summarised in Fig. 4.4. The software modules *cfg80211* and *nl80211* act as interfaces between the user and kernel space,

---

[1]OpenWRT Barrier Breaker Release no. 14.07 - https://openwrt.org/

Table 4.1: Wireless network interface controller characteristics.

	LP-RSU	LP-OBU	HP-RSU	HP-OBU
Model	Mikrotik R52H [160]		Mikrotik R5SHPn [161]	
TX Power	25 dBm		29 dBm	
Antenna Gain	7 dBi	5 dBi	9 dBi	5 dBi
Linux Driver	ath5k		ath9k	
Bandwidth	10 MHz			
Frequency	5.89 GHz		5.9 GHz	
$CW_{\min}, CW_{\max}$	[15, 1023]			
MCS	QPSK ½			

mac80211 is the general driver framework, and *iw* is the NIC configuration utility. Furthermore, *cfg80211_ops* and *ieee80211_ops* define the operations and the callbacks between the different blocks. The Outside the Context of a BSS (OCB) mode was enabled in the MAC layer, allowing the NICs to operate without being associated with an access point and the *iw* utility was modified accordingly to include the new commands for using OCB mode. The values for the contention windows and the MCSs were chosen to follow the regulation for the ITS-G5 standard specification. Integration with a GPS dongle via a USB interface was enabled. A beaconing interface was also developed that generates IEEE 802.11p DSRC CAMs and broadcasts them in the network.

The GPS coordinates, the speed, the heading and the timestamp of the GPS are being encapsulated within the transmitted CAMs. A logging interface was designed that logs all the packets generated, transmitted and received. An example of the packets exchanged can be found in Fig. 4.5. At the TX side, the acquired GPS coordinates are represented as *GpsLongitude*, *GpsLatitude*, being respectively the longitude and latitude values. The *InterLongitude* and *InterLatitude* values are the interpolated values based on the acquired GPS coordinates. The *SeqNum* is the sequence number of the packet generated (starting at zero when the device boots up). The *GpsSpeed* and *InterSpeed* are the acquired values from the GPS dongle and the interpolated value respectively. Finally, the *Timestamp* is the time that the packet is generated, given in Unix Epoch format. The rest of the fields are used for debugging purposes only.

At the RX side, the *RxMAC* is logged at first, which is the MAC address of the device transmitted the packet. *RxLongitude* and *RxLatitude* are the GPS coordinates encapsulated in the transmitted packet. Finally, the *InterLongitude* and *InterLatitude* values represent the current longitude and latitude of the receiver, acquired from the GPS dongle and interpolated later. The remaining values are similar to the transmitted packet. The above system is highly customisable, and in the future, more features extracted from different sensors can be encapsulated in the exchanged frames to introduce different vehicular applications and expand the cooperative awareness of a vehicle.
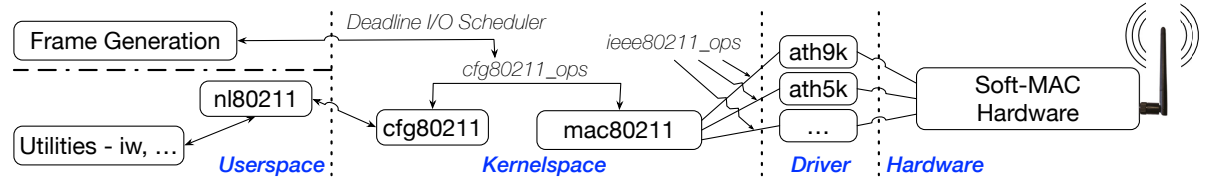
Figure 4.4: Linux Kernel Modules modified to enable the IEEE 802.11p/DSRC capabilities in our system.

```
TX-REQ-CAM ; Protocol;1 ; StationID;3608578767 ; SeqNum;1000 ; GpsLongitude;-2.6025489 ; GpsLatitude;51.4558020 ;
InterLongitude;-2.6025490 ; InterLatitude;51.4558020 ; GpsSpeed;4.3320000 ; InterSpeed;0.0000000 ;
Timestamp;1519032043343 ;
TX-MSG-LENGTH;107;

RX ; RxMAC;4c:5e:0c:84:35:f6;107 ; RX-CAM ; Validation;1 ; Protocol;1 ; StationID;3608578767; SeqNum;32563 ;
RxLongitude;-2.6031772 ; RxLatitude;51.4563665 ; RxSpeed;28.3320000 ; SpeedConf;3 ; RxHeading;3222 ; InterLongitude;-
2.6025490 ; InterLatitude;51.4558020 ; InterSpeed;12.9980000; Timestamp;1519035271768;
```

Figure 4.5: Example of the log file generated at the transmitter and the receiver side.
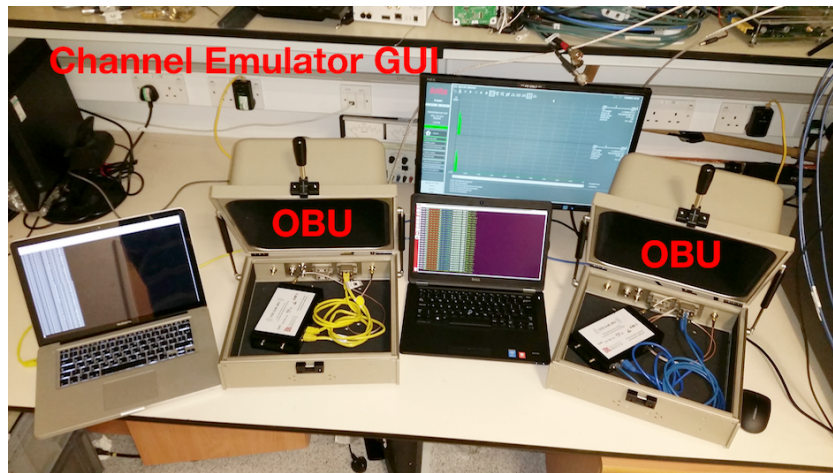
Figure 4.6: Using a Channel Emulator and two Faraday Cages to isolate the devices from the surrounding environment, we conducted a "real-world like" preliminary experiment within our laboratory.

4.4.3   Preliminary Evaluation of the Experimental Testbed

Before deploying the above-described testbed in real-world, we had to thoroughly validate its performance under a controlled laboratory environment. To do so, we utilised the Anite Channel Emulator (Fig. 4.6), provided by the Communication, Systems and Networks (CSN) laboratory of the University of Bristol and performed our test under an emulated V2I scenario. We emulated two devices, one stationary and one mobile, on one of the main roads from Bristol (Park Street – Fig. 4.7(a)) and used the device parameters shown in Table 4.1 (using the HP-RSU transceivers)
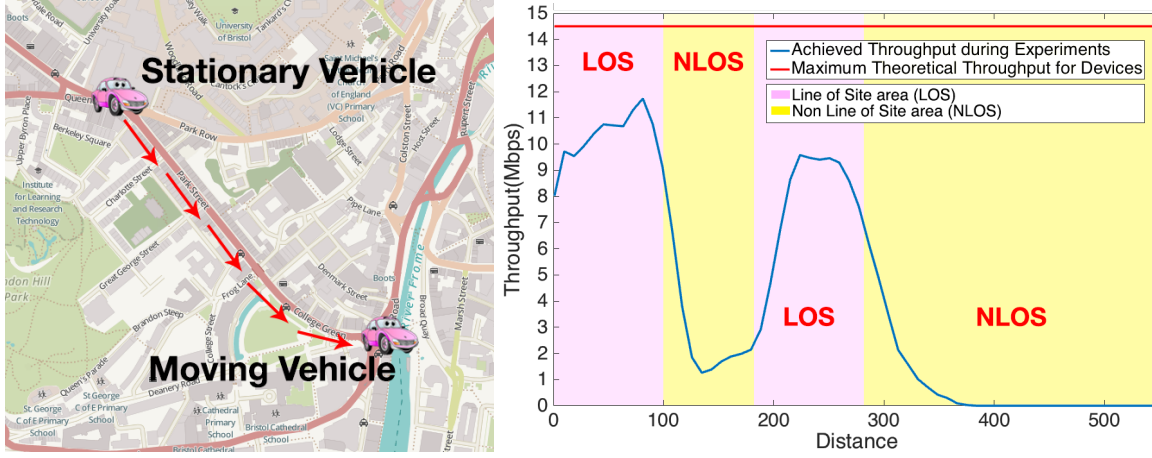
(a) The route used with the preliminary experiment using the Channel Emulator.

(b) Throughput results for the devices shown in Fig. 4.3(a).

Figure 4.7: Preliminary results for our prototyped testbed. The synthetic data were generated using Iperf and the experiment was conducted using a Channel Emulator.

and a different MCS. The MCS used for that particular experiment was 64-QAM with a coding rate of ¾. The channel impulse responses needed for the emulator, where acquired using the Ray Tracing tool of the University of Bristol, and later were processed in a format suitable for the emulator. In Fig. 4.7(b), the initial throughput results can be found. For this experiment, we generated a UDP data stream using iPerf traffic generator[2] and transmitted that from the stationary to the moving vehicle. As it can be seen, the maximum theoretical throughput (measured with the devices in a controlled environment) was roughly about 14.5 Mbps. The maximum achieved throughput was about 12 Mbps and degrades as the distance is increased. As expected, the obstacles found and the curvature of the road significantly drop the performance. The above test proved that the devices could work adequately under urban vehicular scenarios.

Apart from the experiment with the synthetic UDP packets from Iperf, we conducted another more complex test involving the transmission of a video stream. We did that to ensure the flawless operation of our devices when components like LiDARs or camcorders are connected with our testbed, generating traffic that needs to be retransmitted to another device. This can be very useful for a Netflix-like [162] adaptive video transmission framework (as we described in Chapter 3). For our scenario, we considered H.264/AVC as our video encoder, as it is the most widely used from live video streaming providers. Even though SVC performs better compared to AVC [163], AVC was chosen as the scalable version is not standardised and therefore the functionality of the existing tools is limited and unreliable. The two devices were placed with 50 m separation between them, on the same road as before (Fig. 4.7(a)) and both devices were stationary.

---

[2]iPerf Traffic Generator - https://iperf.fr
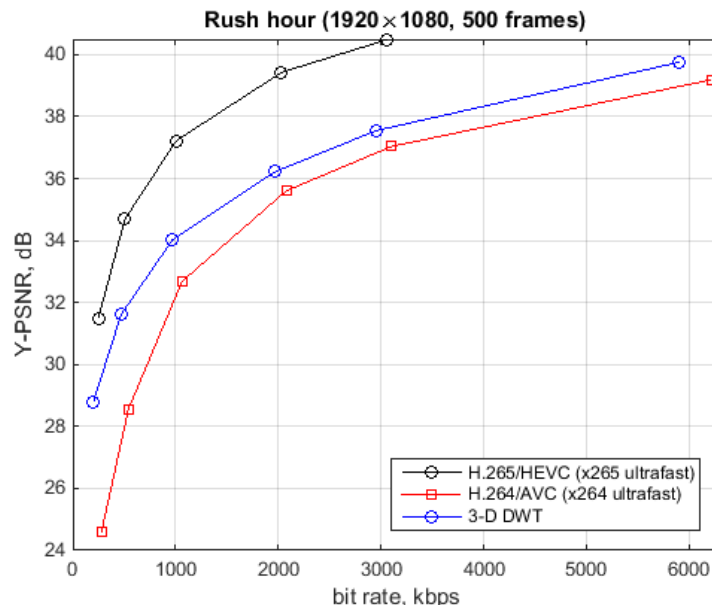
Figure 4.8: PSNR-Rate curves for the video "Rush Hour" [3].

The functionality of our system was evaluated using "Rush-Hour" video sequence [3]. The raw video files were chosen to have similar texture characteristics to the ones encounter in an urban environment (e.g. pedestrians, vehicles, heat distortion, etc.). PSNR was used as a metric for evaluating the performance of the video. The suitable video bitrates were chosen based on the Rate-Distortion (RD) curves of the encoded video [164]. An example can be found in Fig. 4.8, where a video can be perceived as "good" for values of PSNR between 35 dBm to 39 dBm. In Table 4.2, the results from the initial video experiments can be found.

At first, conducting this experiment, we ensured the flawless operation of our devices. However, even from these preliminary results, we can observe that the perceived PSNR values are nowhere near the "good" range for the transmitted video. This strengthens, even more, our idea for scalable and heterogeneous system architecture, as we described in Chapter 3. The above concludes our preliminary evaluation and the validation of the performance of our testbed. However, similarly to the simulation models, the channel emulators are not perfect and need to be adequately calibrated and configured to produce accurate results. For the remaining of this chapter, our research efforts will be focused on the simulation frameworks and the real-world trials.

4.5   Fine-Tuning Veins - INET Network Simulator

In the previous sections, we described the interconnection between the trials and the simulations at first. This was followed by a brief description of an ideal system solution and our actual system implementation that will be used for our performance investigation. In this section, we

Table 4.2: PSNR results from the video transmission experiments.

Video Bitrate	Resolution	Decoded Frames	Mean PSNR	Standard Deviation
506 Kbps	360p	500/500	12.8	2.01
675 Kbps	480p	500/500	12.93	4
1029 Kbps	720p	500/500	12.93	4
1536 Kbps	1080p	493/500	17.74	2.86
1013 Kbps	360p	500/500	15.35	3.09
1351 Kbps	480p	236/500	14.65	0.95
2058 Kbps	720p	485/500	14.99	5.24
3072 Kbps	1080p	479/500	17.32	1.32
1689 Kbps	360p	500/500	19.91	6.87
2252 Kbps	480p	500/500	14.87	2.39
3430 Kbps	720p	500/500	12.36	3.39
5120 Kbps	1080p	493/500	9.25	9.15

will describe the fine-tuning process of Veins-INET, discuss the performance metrics that we used, an insight of how the simulator works and what are the differences of the real world. The calibration of the aforementioned network simulator will take place in two phases. The first step (Sec. 4.5.1) describes the initial calibration scenario, conducted under "ideal-like" conditions. As discussed, to achieve high-fidelity results we need to fragment our system into smaller problems and address each one individually. When all the problems under "ideal-like" conditions have been addressed, we can later move to a more realistic representation of the real world, introducing the various channel characteristics and the obstacle loss (Sec. 4.5.2). This phase is finalised with the integration of a real-device profile within the simulation framework (Sec. 4.5.3).

4.5.1 Initial V2X Calibration Scenario

Consider a scenario with an ideal channel (free-space path loss, no reflections) between two ideal stationary vehicles (isotropic antennas, zero hardware attenuation). Of course, this is an ideal and hypothetical scenario and cannot exist in the real world. To approximate this ideal scenario, we conducted our initial experimental demonstration, for both HP and LP transceivers, inside an anechoic chamber with dimensions 8.5 m×4.5 m×5 m. The distance between the two devices was ~6.5 m. One was denoted as an RSU while the second was considered an OBU for our investigation. We should note, that for this particular experiment, both devices were using the same antenna model to avoid inconsistencies between the hardware. For the experiment, we tested the performance of the devices under different MCSs. More specifically, we generated

a UDP data stream using iPerf traffic generator and a periodic beacon every 100 ms. All the data were transmitted from the OBU towards the RSU device, and for each scenario, enough data were being generated to saturate the channel. As known, the network level performance is affected by the Signal-to-Interference-plus-Noise Ratio (SINR) and the sensitivity levels for each MCS. The SINR degrades based on the disruptive channel characteristics (e.g. distance attenuation, multipath, antenna misalignment, etc.) and the imperfection of the devices (e.g. thermal noise, etc.). Using an anechoic chamber, we conducted the experiments under near-optimal conditions (SINR greater than the sensitivity level), and therefore the near-optimal performance was achieved.

The above scenarios were designed in Veins-INET as well. In order to achieve the necessary level of similarity, we fragmented the design process into smaller steps/problems, that we solved individually. Initially, we considered the IEEE 802.11p Physical Layer (PHY) frame. It consists of three fields [165]: 1) the *Preamble*, 2) the *Signal* field (SIG), and 3) the *Data* field. The preamble marks the beginning of the PHY frame, is responsible for the appropriate antenna selection and corrects the timing and frequency offsets. This is followed by the SIG field that specifies the frame rate and length. Finally, the data field consists of the Physical Layer Service Data Unit (PSDU), that encapsulates the MAC frame, the Physical Layer Convergence Procedure (PLCP) Service, and a Tail field. The Data field can also be padded with extra bits, so its length is a multiple of the coded bits in an OFDM symbol. The above fields are transmitted using BPSK ½ MCS, and their lengths can be found in Tab. 4.3.

Veins-INET describes the transmission duration of the Preamble and SIG fields, with a parameter named as *preambleDuration*. The length (in bits) of the Data field is characterised as the *headerBitLength* (see Fig. 4.9). The PHY bitrates simulated in Veins-INET are defined by the operational mode used, namely *opMode*, that was set to "*p*" in a similar fashion with our devices. The pair of simulation parameters *bitrate* and *modulation* should be manually configured for each MCS in order to appropriately describe the PHY bitrate. Finally, the channel *bandwidth*, the *carrierFrequency*, the *antennaType* (*ConstantAntennaGain* in this case) and the *gain*, all describing different aspects of the channel and the devices, should be configured by the user before any simulation scenarios.

WAVE 1609.4 standard [149] introduced a multi-channel operation, i.e., operating alternately on the control channel and one of several service channel. For our scenarios, we did not consider the above in order to identify the maximum performance under saturation conditions. Furthermore, the RTS threshold – defined as *rtsThreshold* in Veins-INET, was set to a value greater than the frame size. This ensured that the RTS/CTS procedure was disabled. The MAC layer backoff times are drawn from a Contention Window (CW) starting from $CW_{\min}$ (*cwMinData* and *cwMaxData*). All the values used for the above can be seen in Tab. 4.3. The values that were especially chosen for the contention were proven to be optimal for vehicular communications [166]. In the particular case of the ath5k driver, the length of the MAC TX queue size is capped. The

Table 4.3: Simulation and experimental parameters.

Parameter	Value	
Experiment/Simulation Time	10	s
Carrier Frequency	5.9	GHz
Bandwidth	10	MHz
MTU	1500	B
UDP Packet Length	8192	B
Beacon Length	500	B
Beacon Interval	100	ms
Preamble Duration	32	μs
SIG Duration	8	μs
PLCP Service Length	16	bit
Tail Length (Data Field)	6	bit
$CW_{\min}, CW_{\max}$	$[15, 1023]$	
TX MAC queue size	50	
Background Noise	$\mathcal{N}(-110, 3)$	dBm
Connector and Cable Losses	3	dBm

Algorithm 1 Inter-arrival Jitter Algorithm [167]

Require: Inputs: t_{tsmp}, the timestamp of the incoming packet, t_{arr}, the current time.

Ensure: $t^*_{\text{del}} = 0$, the end-to-end delay of the previous packet, $t_{\text{jitter}} = 0$, the variance after the previous packet

  while UDP Packets are arriving at the receiver **do**

    $t_{\text{del}} = t_{\text{arr}} - t_{\text{tsmp}}$ ▷ The end-to-end delay

    $t_{\text{diff}} = t_{\text{del}} - t^*_{\text{del}}$ ▷ The variance of the inter-arrival time

    $t^*_{\text{del}} = t_{\text{del}}$ ▷ Store the end-to-end delay for the next iteration

    $t_{\text{diff}} = |t_{\text{diff}}|$ ▷ The absolute value of the difference

    $t_{\text{jitter}} \mathrel{+}= 1/16 * (t_{\text{diff}} - t_{\text{jitter}})$ ▷ The inter-arrival variance

  end while

---

same value was considered for our simulations as well – namely *maxQueueSize*.

Veins-INET uses a transmission interval describing the generation of two consecutive UDP packets, defined as the *sentInterval*. To match the same level of the channel utilisation as the real-world field trial, without having packets discarded from the MAC TX queue, a very precise value *sentInterval* value had to be chosen for each MCS. Suitable values were found with a trial and error method. A summary of all the simulation and experimental parameters can be found in Tab. 4.3 and Sec. 4.4.2.

The results for the above calibration scenario are shown in Figs. 4.10 and 4.11. The transport layer throughput and the interarrival jitter [167] were chosen as the most relevant metrics for this study. The first can describe the amount of information that can be transmitted via a channel, whereas the second is very important for video-related applications. The rectangle is the interquartile region (IQR) between the first and third quartile, while the line within represents the median. The whiskers are the maximum and minimum values and the asterisks show the
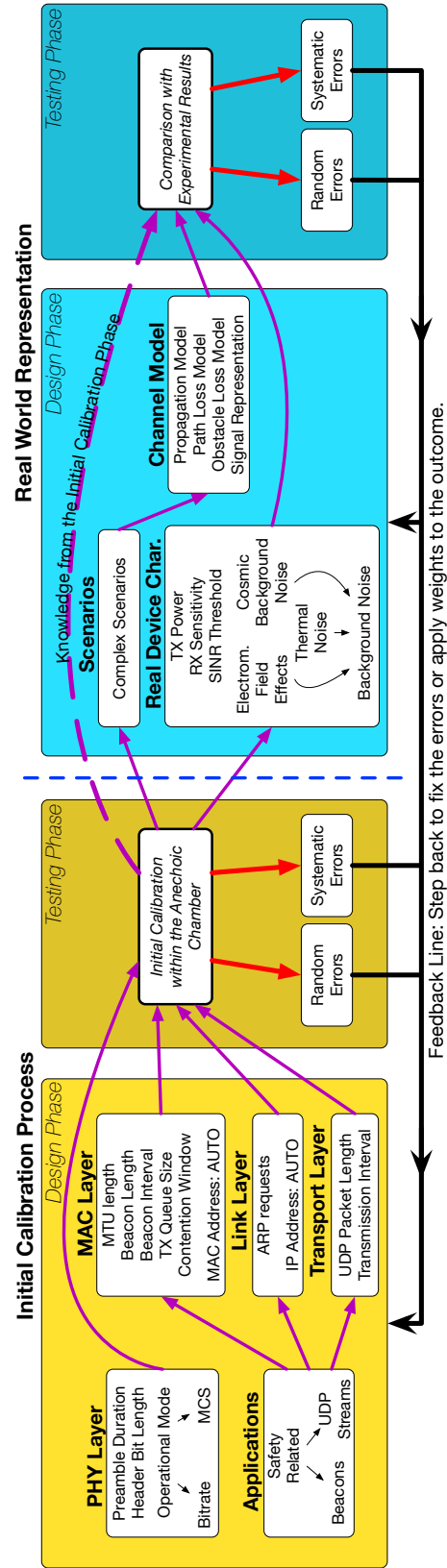
Figure 4.9: Hierarchical Validation Process. Each entity is individually fine-tuned to achieve high-fidelity results after the calibration of the entire system. The fine-tuning is a two-phase process starting from an initial calibration and moving to a more complex real-world representation.
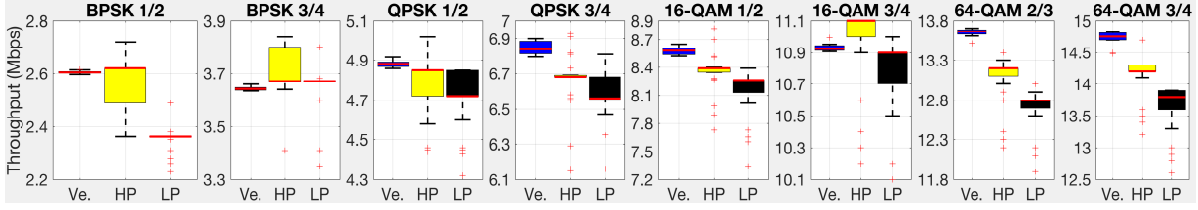
Figure 4.10: Values of throughput obtained from Veins (initial calibration), the HP and LP transceivers, for different MCSs.

outliers. A value is regarded as an outlier if it outside $\pm 2.7\sigma$ (99.3% coverage of the normally distributed samples).

Fig. 4.10 shows the network throughput measured at the transport layer. As mentioned in Sec. 4.3.2, a meaningful comparison should focus on the trends. Therefore, the first trend observed is that some MCSs achieve slightly better simulated performance compared to the trial ones (e.g. QPSK ¾), while the rest are almost identical. The median value deviates up to ~0.5 Mbps for the HP transceiver, and up to ~1 Mbps for the LP device. Overall, we observe that the throughput performance of the LP transceiver is ~5% lower that the HP one; comparing the median values and following the same trend for all MCSs. This observable difference is related with the operation of the two different drivers used.

Fig. 4.11 compares the inter-arrival jitter performance. The jitter, as defined in RFC 1889 [167], is the statistical variance of the inter-arrival time between packets. An algorithmic representation of the jitter calculation procedure can be found in Alg. 1. Comparing the absolute values of the results, we see a huge difference between the trials and the simulations. However, focusing again on the trends in the performance, i.e. the relative variation in the jitter for the different MCSs, it is shown that all the results follow a similar trend. For lower MCSs, the jitter is increased, but when the bitrate is increased, both the simulation results and the actual devices perform better. The jitter values presented from Veins-INET have been multiplied by 56000. We found this value by trial and error, and we observed that multiplying all the results with the same value, we get the same orders of magnitude with the ones obtained from the LP and HP transceivers. The huge difference in the absolute values was somehow expected as our devices are built upon a single-core CPU, which executes tasks with the same priority according to the Linux *Deadline I/O Scheduler* – thus the CPU cannot fetch/push data streams towards the transceivers at a constant I/O rate. Since in Veins-INET this issue is not present, the simulated and measured jitter performance may vary significantly.

This concludes our initial Veins-INET calibration effort. Based on the framework introduced in Sec. 4.3, we fragmented our system into smaller problems, fine-tuned each one individually and managed to replicate our ideal subsystem in Veins-INET. Regarding the performance, we achieved similar results with minor deviations between the trials and the simulations. In the next sections, we will introduce more complex device features and the channel behaviour in order

71
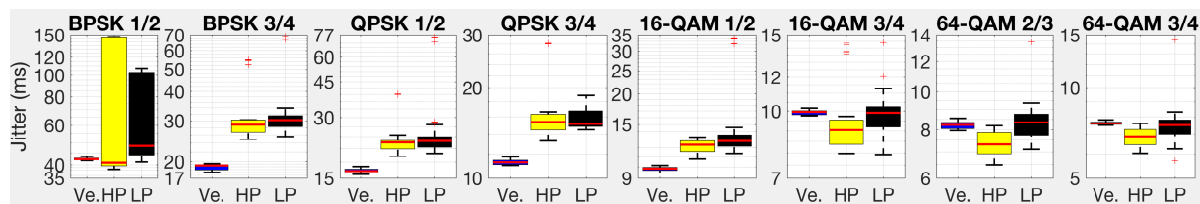
Figure 4.11: Values of jitter associated to the UDP stream. Obtained from Veins (initial calibration), the HP and LP transceivers for different MCSs.

to approach a better representation of the reality.

4.5.2 Moving towards a Realistic Representation of the World

The surrounding environment plays a vital role in the performance of vehicular networks. Each environment has its own characteristics. Therefore a theoretical analysis is required for the signal degradation due to the different channels. Firstly, we have the long-term characteristics where the signal degrades depending on the distance, the atmospheric conditions, the RSU and OBU positioning, the carrier frequency, etc. Secondly, we have the short-term variations that are based on the multipath effect from the surrounding objects and the Doppler shift introduced from the velocity of the vehicle. The channel behaviour within Veins-INET lies beneath the *physical medium model*. This model is further split into different submodules as shown in Fig. 4.9.

At first, the propagation model (*propagationType*) describes the propagation time within the channel. It can be configured either as *constantSpeed* where the propagation time is proportional to the distance travelled or as *constantTime* where the propagation time is a constant parameter. For our scenarios, we considered *constantSpeed* as our chosen model to incorporate the distance between the devices in our simulations. The analogue signal and its fluctuation is a complex physical phenomenon which can be modelled in many different ways in Veins-INET. Choosing the right analogue domain signal representation is the most crucial factor in the trade-off between accuracy and performance. For our performance investigation, we utilised the *DimensionalAnalogModel* that represents the power deviation over both time and frequency.

The long-term signal degradation under a real-world experiment depends on the carrier frequency, the distance, the positions of the vehicles, etc. In Veins, the simulation parameter *pathLossType* describes the path loss model that is responsible for computing the power reduction based on the travelled distance, the velocity factor, the carrier frequency and the path loss exponent for each environment. The multipath distortion caused by the surrounding obstacles and leading to short-term signal degradation is described using the various well-known small-scale fading models (e.g. Nakagami, Rician, etc.). Within Veins-INET their individual parameters (K-factor for Rician, shape-factor for Nakagami, etc.) can be accordingly fine-tuned to represent the real world. An XML file lists all the physical obstacles in the simulated world. This file can be generated by parsing the building characteristics from a map. For each building, a material type

exists that is used to calculate the material absorption when a ray is travelling through an object. The parameter *ObstacleLossType* specifies the rate of signal degradation for all the objects. In Sec. 4.6.1, we will further describe the way Veins-INET interprets a building and how it affects the signal propagation.

4.5.3 Integration of Real Device Profiles in Veins

In the previous section, we configured the parameters related to the signal propagation. The next step is to integrate the characteristics of each NIC in our simulated scenario. Unfortunately, COTS devices should frequently be treated as a "black box" as many of their physical level performance characteristics are unknown and not easily measurable. Therefore, we will base our simulation parameters on some speculations and assumptions derived from the datasheet of each considered transceiver.

We start with the attenuation that is introduced from the cables and the connectors in the system. These are described as the *systemLoss* in Veins-INET, and their values were measured within our laboratory. Later, the SINR of each device can fluctuate from random effects such as the thermal noise, electromagnetic field effects, the cosmic background noise, etc. These effects, compared to the system loss discussed before, are not predictable and do not come from a specific source. Therefore, they cannot be isolated. Veins-INET represents this signal variation with a background noise model (*backgroundNoiseType*), that we configured for our system using the *backgroundNoise* parameter and follows a Normal distribution.

According to the manufacturer datasheet, we have access to the transmission power given at 20 MHz of channel bandwidth, for each MCS. However, the IEEE 802.11p bandwidth is equal to 10 MHz. From the energy-per-symbol-to-noise power spectral density equation it follows that:

$$(4.1) \qquad \frac{E_s}{N_0} = \frac{C}{N} \frac{B}{f_s}$$

where E_s is the energy per symbol, N_0 is the noise power, C/N is the carrier-to-noise ratio, B is the channel bandwidth and finally f_s is the symbol rate. In our case, the only non-constant variable is B. Therefore, for a 10 MHz channel, the E_s/N_0 ratio is expected to be twice as much as that measured using a 20 MHz channel. As N_0 is measured per unit of bandwidth (per MHz), it follows that E_s is doubled. Finally, knowing the number of bits per symbol of each MCS, we can infer the maximum transmission power for a 10 MHz channel. These values are summarised in Tab. 4.4.

As regards the sensitivity of the receiver, from the Minimum Operational Sensitivity (MOS) relation, we know that:

$$(4.2) \qquad \text{MOS} = \frac{SINR_{\text{thr}}\, k\, T_\alpha\, B\, (NF)}{G_{\text{rx}}}$$

where $SINR_{\text{thr}}$ is the minimum SINR needed to process (not just detect) a signal, NF is the noise figure, k is Boltzmann's constant, T_α is the effective noise temperature referred at the input

Table 4.4: Simulation parameters based on the manufacturer datasheet.

Modulation	TX power		RX sensitivity		$SINR_{\text{thr}}$ [168]		Units
	LP	HP	LP	HP	½ MCS	¾ MCS	
BPSK	27	28	−93	−93	10	15	dBm
QPSK	26	27	−85	−88	10	15	dBm
16-QAM	25	26	−80	−84	17	17	dBm
64-QAM	24	24	−73	−80	20	25	dBm

of the receiver, and G_{rx} is the isotropic antenna gain. Obviously, $SINR_{\text{thr}}$ depends not just on the NIC but also on the MCS in use. Authors in [168] measured the $SINR_{\text{thr}}$ under a V2I scenario for two different antenna heights. Their lower antenna configuration was very similar to ours, so their $SINR_{\text{thr}}$ results will be utilised for our scenarios. Knowing the $SINR_{\text{thr}}$, the only variable in (4.2) is B. Therefore, halving B, the MOS is doubled. Finally, the antenna gain values were taken directly from the manufacturer datasheet. All the values are presented in Tab. 4.4.

4.6 Parallel Implementation of the OMNeT++ INET Framework

In the previous sections, we described a full-stack calibration procedure for Veins-INET framework. Following the above procedure, we can increase the accuracy of our simulator and achieve real-world-like results. However, the better accuracy always comes at the cost of increased simulation time. Even for simplified scenarios, when all the above modules and parameters are fine-tweaked, the simulation time is significantly increased.

The next-generation C-ITSs require more sophisticated and demanding communication frameworks that usually have to be tested in city-scale scenarios. Designing and deploying an actual testbed is very demanding and costly and is almost never an option. Therefore, the issue of running cost-effective city-scale experimentation is often addressed by means of simulation frameworks. Unfortunately, existing simulation frameworks (i.e., ns-2, INET, etc.) are executed in a single-thread fashion. Thus, in city-scale scenarios, one minute of simulation can easily result in hours if not days of computation.

One of the leading solutions to speed up the execution time of a simulation framework is the use of high-performance parallelised computing platforms. However, the existing implementations are quite laborious with regard to the time required to configure new scenarios or reconfigure the existing ones. Also, they do not always assure an improvement in the simulation time [98]. For these reasons, existing simulation frameworks are mainly operated in a single-thread fashion. The necessity for city-scale simulation models motivated us to exploit ways of parallelising Veins-INET framework.

Analysing the way Veins-INET operates, we identified that the most time-demanding opera-

tion is the actual exchange of the packets. Especially for wireless scenarios, such as in the case of vehicular communications, packets are being broadcast to many nodes or are being overheard from the surrounding vehicles causing interference. Furthermore, when a list of obstacles is introduced in a simulation environment, the complexity and the execution time are remarkably increased. For the remaining of this section, we will describe our system analysis and the proposed solution for the above problem.

4.6.1 Parallelising Veins-INET: System Analysis and Proposed Solution

We identify at first the way Veins-INET interprets obstacles and propagated signals. In a vehicular scenario, buildings are the main obstacles that can be found in a city. Veins-INET refers to them as obstacles and parses them within the *PhysicalEnvironment* namespace. All the obstacles are listed in an XML file with an attribute *type*, which defines their shape. Each obstacle is represented by a set of coordinates, that are regarded as the edges of the building in the 3D space. Each building is also associated with a specific *material* that is being used to calculate the attenuation loss caused by the obstacle. The obstacle loss in Veins-INET is calculated by two different models, the *IdealObstacleLoss* and the *DielectricObstacleLoss*. The first determines either the signal to be completely blocked, or not attenuated when intersected with a physical object. The latter computes the power loss based on the material properties, the shape, the position and the orientation of an obstacle.

INET treats the positions of each vehicle as a point on the simulation canvas and updates them by using the SUMO traffic generator [169]. The signal propagation is modelled as a line segment, between point A and point B. The signal attenuation is a function of: 1) the path loss model, 2) the obstacle loss model from the number of intersections calculated as mentioned before. Both models are configured by the user. What is really of interest, is the way Veins-INET calculates the attenuation due to the wall intersections. When a packet is transmitted, INET finds sequentially and iteratively all the intersections with the obstacles. For the given intersections, it calculates the obstacle loss using the function *visit* of the obstacles classes mentioned above. Considering the above and that IEEE 802.11p operates in broadcast mode, it is evident that the above process is computationally expensive. In fact, the computation of the attenuation loss has a computational complexity of $O(mn^2)$, where n is the vehicles and m the number of obstacle intersections. This significantly increases the simulation time in large-scale scenarios.

In order to overcome the aforementioned problem, we developed a multi-thread version of the *PhysicalEnvironment* class by modifying the function *visitObjects*. This function is responsible for parsing all the obstacles and finding the power attenuation. Also, we modified *visit* in *DielectricObstacleLoss*, to ensure flawless operation. In our version of INET, the number of threads can be dynamically changed by the user when initialising the simulation scenario.

In city-scale scenario (namely, maps greater than ≥ 2km), vehicles are not expected to communicate from one side of the city to the other. Despite this, Veins-INET always computes

the intersection map between each pair of vehicles and their signal attenuation, regardless of the distance between them. In order to speed up the execution time even further, we integrate the notion of the transmission radius in the system. As such, we introduced the *distanceBoundary* user parameter, within the *ScalarAnalogueModel* class, under the *RadioMedium* namespace. For all the exchanged packets, we find the distance between the two communicating vehicles, and if it is greater than the given boundary, we regard the packet as non-deliverable. By that, we can avoid unnecessary calculations in the *PhysicalEnvironment* model. Of course, the above improvement can be implemented into other analogue models as well (e.g., *DimensionalAnalogModel*) but we chose the scalar one as a proof of concept.

4.7   Performance Evaluation and Macroscopic View

In Sec. 4.4 we described the experimental testbed that we designed and prototyped, presented its different components and how it operates. We also performed an initial test, in order to ensure that our testbed is fully functional and can be used for real-world experimentation. In Sec. 4.5, we described a full-stack calibration procedure for Veins-INET framework. We also addressed the issues that arise from experimenting in different environments and the way they affect the system performance. Finally, in Sec. 4.6 we explained the necessity for fast execution times when using a simulation framework under vehicular scenarios, as well as the demands for larger scale experimentation. Based on that, we described our system analysis of Veins-INET simulation framework, and our proposed solution to parallelise it and improve its execution time.

In this section, we proceed by presenting the results related to all the contributions of the chapter introduced above. More specifically, we start our performance investigation by an ideal-like scenario within the confined space of our laboratory. By that, we can ensure the flawless operation of our testbed. Our performance investigation continues by conducting a study for the performance of a vehicular communication link under three different environments. We conduct our study using both our calibrated framework and our experimental testbed, and we compare the differences in the results. Later, we evaluate the simulation time improvement using our parallelised Veins-INET framework. We conduct this evaluation under a large-scale city-wide scenario and investigate how the different map sizes and the number of vehicles affects the simulation complexity. Finally, we deploy our ITS-G5 testbed in the City of Bristol and conduct large-scale experimentation evaluating both V2V and V2I links, under various conditions (urban, rural, highway). Our field trials span throughout three days. We use the knowledge acquired from the trials to evaluate the performance of a real-world ITS-G5 deployment, identify the limitations and ways to overcome them. From this research activity, we gained an in-depth knowledge of how a massive city-scale deployment should be approached in the future, and we intend to use this knowledge to extend the features and coverage of our system further.

4.7.1 Performance Investigation of an Ideal Scenario

Before we process with the performance investigation of our system in the real world, we start by evaluating its performance under an ideal scenario. For that, two devices (an RSU and an OBU) were positioned with three-meter distance separation between them. During this scenario, both devices were stationary and placed inside our laboratory under a controlled environment. This distance between them was carefully chosen, based on the equation below. This equation provides the distance separation for two ideal dipole antennas, in order to operate in the far-field and avoid near-field effects. This separation is calculated as [170]:

$$\text{Far Field} \geq \frac{2D^2}{\lambda} \tag{4.3}$$

where D is the antenna length and λ is the wavelength. Given the frequency f, we can calculate the wavelength as $\lambda = c/f$, where c is the speed of light. Using $f = 5.9\,\text{GHz}$ and an antenna length of 20 cm (the longest antenna used during these trials), it was calculated that the radiating near-field distance is ~1.57 m. Therefore, placing the devices at three meters apart we can avoid any effects due to the near field. Of course, no ideal dipole antenna exists in the real world. Therefore, using a relatively longer distance compared to the one calculated, we can ensure that these effects will not disrupt the performance of our testbed. More information about the devices and their configuration can be found in Table 4.4 and Sec. 4.4.2.

Our experiment evaluated the performance of both the HP and the LP transceivers, in order to check whether the different drivers used could potentially affect the performance. The performance evaluation was also conducted for several hours (≥ 5h), to stress the devices even further. From both transceivers and devices, we broadcast a CAM message every 10 ms that is received at the other side. We start with Fig. 4.12. This figure shows the results for the HP transceiver. In that particular case, the heatmap shows the packet delivery rate for all the packets transmitted by the OBU and received at the RSU side. The red dot on the top left corner of the heatmap is the actual position of the RSU. However, as the experiments were conducted within the confined space of our laboratory, we observe that the GPS error was significantly increased. This, however, does not affect our performance investigation as we are particularly interested in just the packet delivery rate in that case. Furthermore, using the same GPS dongle for both the NTP servers, we ensured that the GPS error would be consistent between the devices. Therefore, we will be later able to reconcile the data correctly. As we will see later, when operating these devices in an outdoor environment (Sec. 4.7.4, the position error is minimised.

Fig. 4.12 shows that the packet delivery rate for this close proximity scenario was almost 100%. Similarly, for the LP transceiver, Fig. 4.13 shows that the packet delivery achieves a similar performance. The dark blue areas that have a meagre packet delivery rate represent the packets that were exchanged the first seconds of our experimentation. During that time, GPS error was so significant that the NTP server was not operating correctly, and therefore the packets exchanged were not tagged with the correct position. This effect was avoided when conducting large-scale
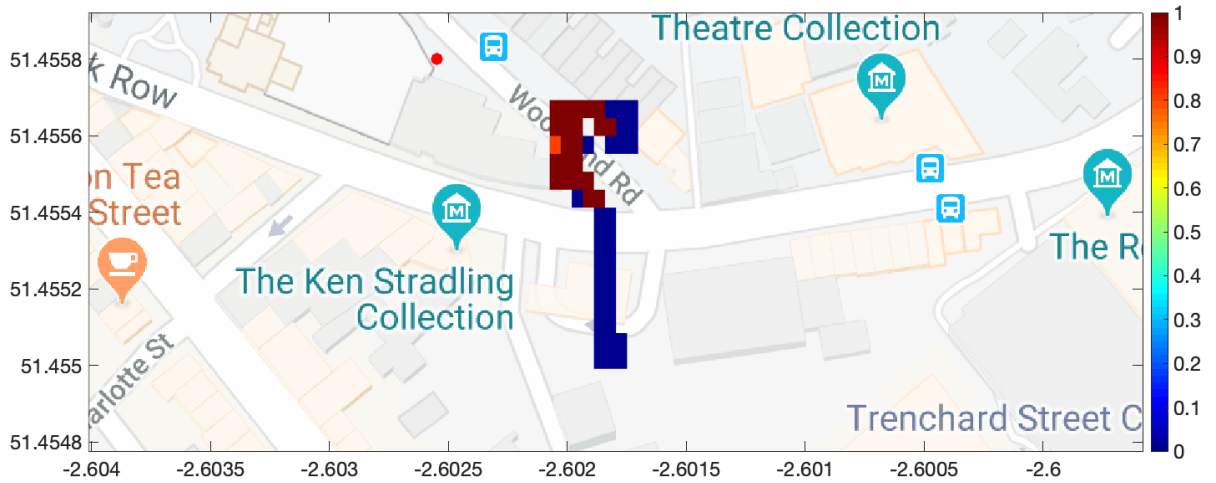
Figure 4.12: Ideal heatmap scenario for the HP transceiver. The distance separation between the devices was three meters.
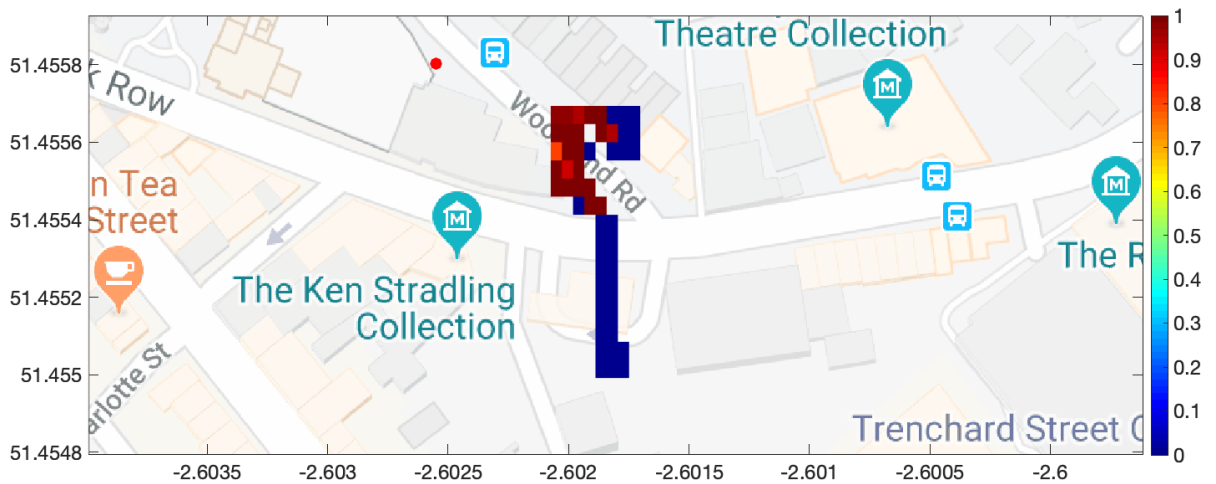
Figure 4.13: Ideal heatmap scenario for the LP transceiver. The distance separation between the devices was three meters.

real-world trials as a strong GPS signal was acquired before starting the experimentation. The above experiments prove that our devices can operate adequately under an ideal like scenario, i.e. being in close proximity and stationary.

4.7.2 Contribution 1: Comparison between Trials and Simulations

In Sec. 4.5, we described a full-stack calibration procedure for Veins-INET framework. We also addressed the issues that arise from experimenting in different environments and the way they affect the system performance. Therefore, we designed and evaluated three different scenarios – urban, suburban, rural. As in our initial calibration scenario (Sec. 4.5.1), we considered two stationary devices, one acting as the RSU and the second as the OBU. For both pairs of
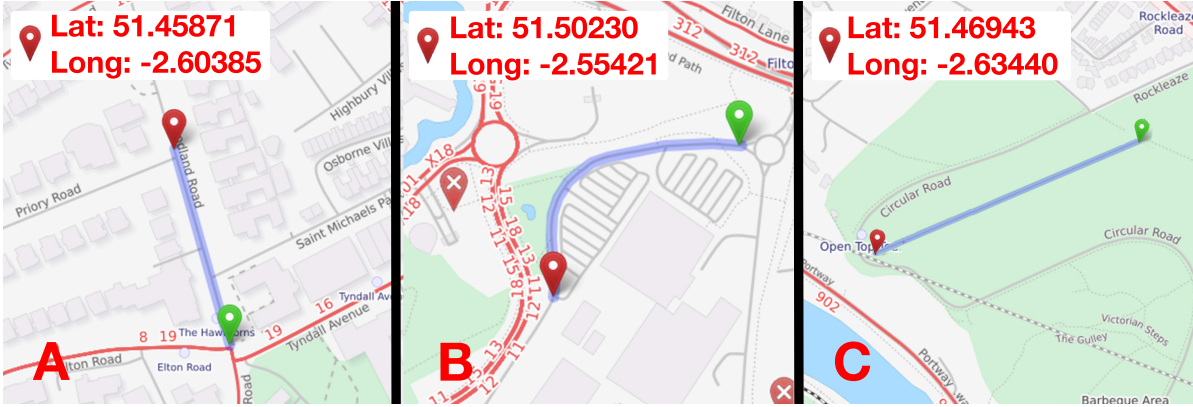
Figure 4.14: The three different scenarios were conducted around the city of Bristol, UK. a) the urban, b) the suburban, c) the rural scenario.

transceivers (HP and LP), we used the same model of antennas to avoid imperfections. The antenna gain for the HP scenario was 9 dBi, while for the LP one was 7 dBi. For both devices, we evaluated their network throughput performance under different distances and MCSs. To do so, we utilised a UDP stream that saturated the channel, and a periodic beaconing, transmitted every 100 ms from the OBU to the RSU. The same scenarios were replicated within Veins-INET, using the parameters from Tabs. 4.3 and 4.4 and the analysis preceded in the previous sections.

Starting with the first scenario, we used an urban road with buildings on both sides (see Fig. 4.14 – A). Both devices were positioned on the edge of the pavement, close to the road and were always in LOS. The buildings surrounding our devices cause multipath distortion. However, being in LOS and conducting the experiment at a relatively short distance, we considered a Rician fading model as the most suitable one to describe the experiment. Veins-INET was configured with a K-factor $k = 3.36$ dB and a path loss exponent $\alpha = 2.3$ [171].

Secondly (see Fig. 4.14 – B), we have a suburban area on a bent and sloppy road with foliage in between the devices and a few buildings on one side of the road. The devices were in LOS for the first ~50 m. Later, the RSU was hidden behind the vegetation and the road slope. Following our system analysis above, we fragmented this scenario in two different subscenarios. At first, we considered a Rician fading model with $k = 2.45$ dB and $\alpha = 2.3$ [171]. When the distance was greater than 50 m, we refer to Rayleigh fading model with $\alpha = 2.5$.

The third scenario refers to a rural environment (see Fig. 4.14 – C). Both the RSU and OBU are always in LOS, and no high buildings or other objects were surrounding the devices apart from some foliage. Therefore, the impact of multipath was minimum. As such, a Rician fading model was considered, with $k = 8$ dB and $\alpha = 2.2$ [172]. For this scenario, only the HP transceiver was used.

During the experiments carried in scenarios A, B and C, the RSU and OBU were fitted on a tripod at ~1.8 m height. A consistent setup has been simulated in Veins. During both the trials

and simulations, each performance metric we measured is the result of an average of multiple experiments. The ARP probe was disabled by manually inserting the addresses of the devices in their respective ARP tables. For all scenarios, we show the most meaningful results and describe the performance for the rest in the text.

With regards to the urban scenario, Fig. 4.15 shows the communication throughput that can be sustained by the OBU as a function of the distance between RSU and OBU, for each MCSs. The distance separation considered were $\{12.5, 25, 37.5, 50, 67.5, 75, 87.5, 110\}$ m. For lower modulations (BPSK, QPSK) and all distances, the same trend and performance were seen as in the calibration process (Fig. 4.10). Again, the difference in the performance between the transceivers, not observed within Veins, is due to the different drivers used. Increasing the MCS and the distance separating the devices, even though the median values remain similar to that shown in Fig. 4.10, the introduced multipath distortion leads to a larger number of outliers. For 16-QAM ½ and ¾, it was seen that as the distance is increased, the SINR drop starts being observed within Veins-INET for the LP configuration (e.g. 110 m) having a different behaviour compared to the trials. Finally, for 64-QAM ⅔ and ¾, the signal received from the LP transceiver within Veins-INET is significantly attenuated. The performance degradation is about 1 Mbps compared to the calibration scenario (50 m) reaching up to 3 Mbps at 110 m.

Fig. 4.16 shows the communication throughput measured in the case of the suburban scenario, for different MCSs and distances $\{30, 60, 90, 120, 200\}$ m. For distances of 30 m and 60 m, we observe that the results follow the same trend as in the urban case (see Fig. 4.15). In particular, despite the RSU being hidden behind the the road slope after 50 m, the overall communication throughput was not severely impacted. For greater distances and the HP scenario, Veins-INET behaves slightly worse compared to the actual device. However, for the LP scenario, the actual device achieved less throughput compared to the simulated result. Especially for 64-QAM ⅔ and ¾ and a distance of 200 m, our LP transceiver achieved zero throughput during the trials whereas the Veins-INET result is around 3 Mbps and 1.2 Mbps respectively. This is due to the BER calculation within Veins-INET that is approximated based on a Gaussian error function not exactly reflecting the reality.

Fig. 4.17 refers to the rural scenario. As before, we considered different distances and MCSs. The distances considered were $\{50, 100, 150, 200, 300, 400, 550, 700\}$ m. Again, for lower modulation schemes (BPSK, QPSK), the same trend was observed as before. For higher MCS, Veins-INET again exhibits a sharp performance degradation when the distance increases. This is clearer at 64-QAM ⅔ and ¾ where Veins-INET outperforms the trial performance at 550 m as expected from the trend observed. However, it is significantly worse at 700 m.

This concludes our performance investigation for the calibrated simulation framework, compared to the real-world field trials. In this section, we summarised our key observations. The downside of a calibrated simulation framework, even though it performs better and can provide the required realism and accuracy, is the increased simulation time. In the next section, we will
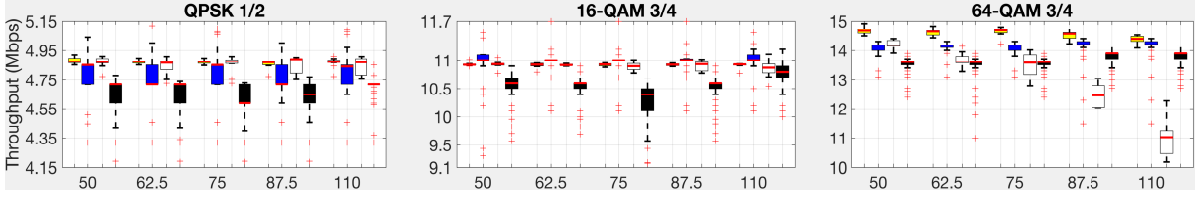
Figure 4.15: Values of network throughput for different MCSs, as a function of the distance between RSU and OBU. Results refer the urban scenario (Fig. 4.14 – A). Each quartet represents the results for a single position with the order (from left to right): 1) Veins-HP, 2) Trials-HP, 3) Veins-LP, 4) Trials-LP.
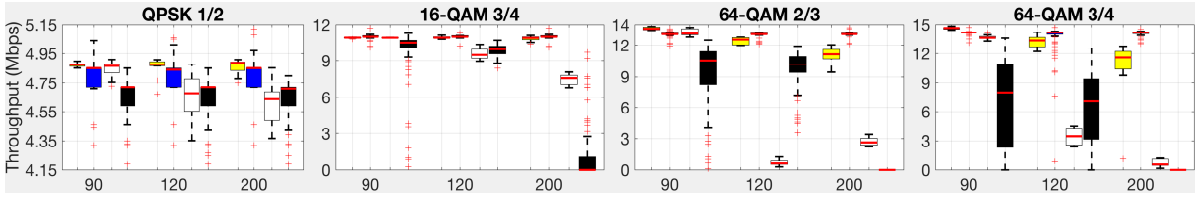
Figure 4.16: Values of network throughput for different MCSs for different MCSs, as a function of the distance between RSU and OBU. Results refer to the suburban scenario (Fig. 4.14 – B). Each quartet follows the same order as in Fig. 4.15.
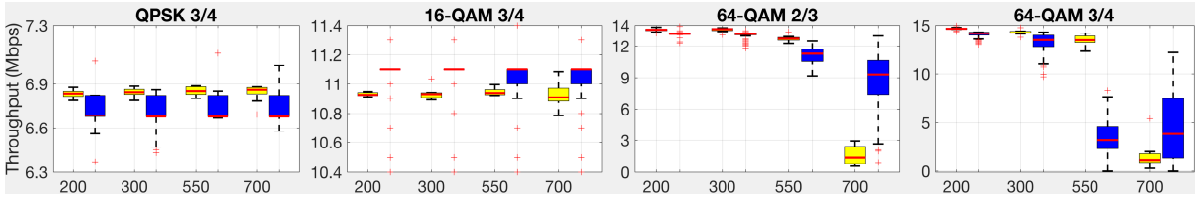
Figure 4.17: Values of network throughput for different MCSs for different MCSs, as a function of the distance between RSU and OBU. Results refer to the rural scenario (Fig. 4.14 – C). Each boxplot pair is: 1) Veins-HP, 2) Trials-HP.

describe our evaluation with regard to the execution time for a city-scale scenario.

4.7.3 Contribution 2: Large-Scale City-Wide Simulations

As discussed above (Sec. 4.6), we implemented a multi-threaded version of the Veins-INET framework. We evaluate the performance of our implementation with two large-scale scenarios in a grid-like fashion. The simulation parameters are as shown in Table 4.5. At first, we evaluate the execution time as a function of the number of vehicles for a map of size 2km^2. Then, we present the execution time as a function of the map size. We consider six map sizes $\{800, 1100, 1400, 1700, 2000, 2300\}\, \text{m}^2$ and 100 vehicles for each scenario. All scenarios have roads equally spread horizontally and vertically every $100\,\text{m}$, without traffic lights at the intersections. Each road is 2-lanes wide. Within each road square, we generated buildings with sides of $950\,\text{m}$

81

Table 4.5: List of simulation parameters for contribution 2.

Parameter	Value	Parameter	Value
Simulation time	100 s	Carrier Frequency	5.9 GHz
TX Power	25 dBm	Channel Bandwidth	10 MHz
TX/RX Antenna Gain	9 dBi	Message Length	140 B
RX Sensitivity	−93 dBm	Pathloss Exponent	2.4
Cable/System Loss	3 dB	Distance Boundary	1000 m
Transmission Interval	0.1 s		

that act as obstacles in our scenarios. Finally, for all scenarios, we generated the vehicle traffic by using SUMO [169]. For our scenarios, we used synthetic map data to avoid inconsistencies in different map areas and provide an easily reproducible scenario.

Fig. 4.18 shows the execution time measured as a function of the number of vehicles. Any multi-threaded execution generates some computational overhead in order to create the threads and handle the content-switch. To that extent, we compared two different number of threads (4 and 10). Overall, by parallelising the functions mentioned above, our version of INET ensures reduced computation times. When four threads are used, we observe an improvement of up to 30%. When ten threads are utilised, the improvement ranges between 10% and 12%. The increased overhead of the number of threads in the second case is the reason for this difference in the performance. We observed that the optimum number of threads is scenario dependent and is related to the average number of intersections between the communicating vehicles (one thread per intersection).

Fig. 4.19 shows the execution time required, as a function of the map size, for 100 vehicles. The synthetic maps we generated have a relatively small number of obstacles compared to a real city. We observe that for small maps (from $800\,\text{m}^2$ to $1.4\,\text{km}^2$) the sequential INET achieves slightly better performance compared to the parallel one. This is caused by the multi-thread overhead. However, for larger maps, we observe that our multi-thread version of INET outperforms the sequential one by reducing the computational time of 43%, for a $2.3\,\text{km}^2$ map. Again, when we increase the number of threads, the increased overhead leads to increased simulation time compared to the 4-thread scenario, but still manages to outperform the sequential execution. Finally what we observe is that for maps $\geq 1.7\,\text{km}^2$, as the size of the map increases, the execution time decreases. This is because of the distance boundary that we introduced. For a fixed number of vehicles that are equally spread on the surface of the map, the distance between them will be greater when the size of the map is increased. Considering the number of buildings that exist in a real-world map, and based on our above observations, we expect our multi-threaded version of the code to enhance the execution time even further.
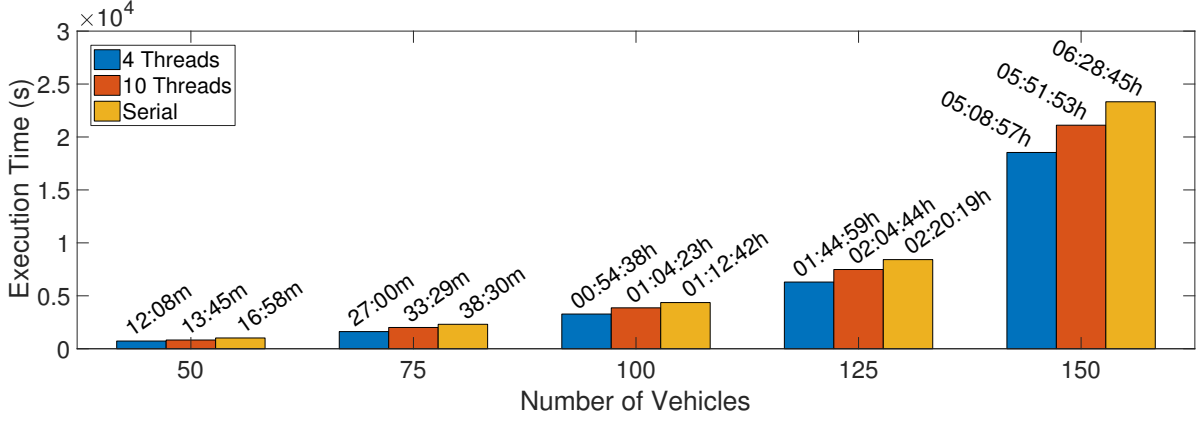
Figure 4.18: The execution time, measured as a function of the number of vehicles for the parallel and the sequential implementation.
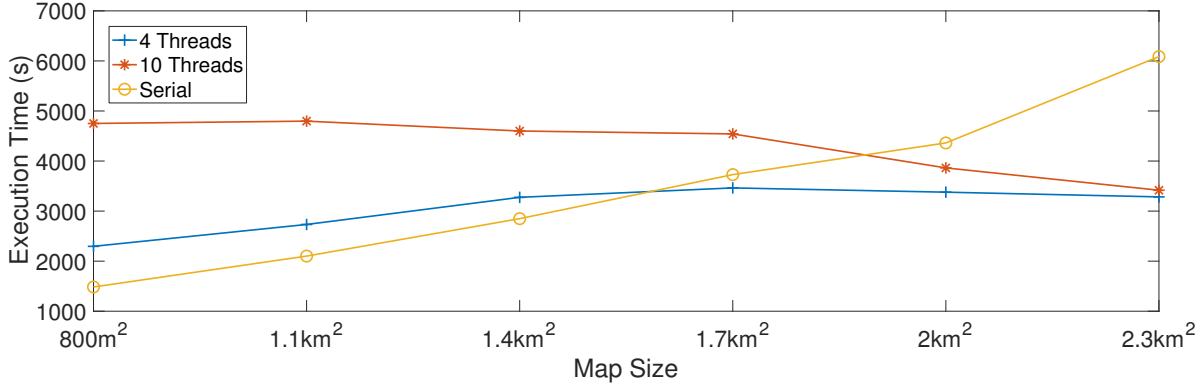
Figure 4.19: The execution time, measured as a function of the map size for the parallel and the sequential implementation.

4.7.4 Contribution 3: Large-Scale City-Wide Experimentation

Our final contribution of this chapter is related with the large-scale city-wide design and deployment of our experimental testbed (Sec. 4.4). We will describe our critical observations of this research activity, investigating the performance of our devices under different environments and links, identifying the limitations of our system, and finally describing ways to overcome them. Firstly, we will investigate the Key Performance Indicators (KPIs) from the perspective of the first car and the RSUs, while the second vehicle acts as an interferer when being within coverage. Secondly, we will present a V2V scenario.

Three RSUs were deployed at three locations around the City of Bristol, UK (as shown in Fig 4.20(a)). *Hydrogen-RSU* was mounted at the height of around ~8 m, on a curvy, narrow road very close to a blind T-junction. The second one (*Helium-RSU*) was installed on the wall of a building next to a straight road with some foliage at the sides at ~5 m. Finally, *Lithium-RSU* was placed on the balcony of a tall building (at ~25 m height), next to a wide road, providing the most

(a) Both Vehicles within RSU coverage - HP transceiver.

(b) Both Vehicles within RSU coverage - LP transceiver.

(c) Only Vehicle 1 within RSU coverage - HP transceiver.

(d) Only Vehicle 1 within RSU coverage - LP transceiver.
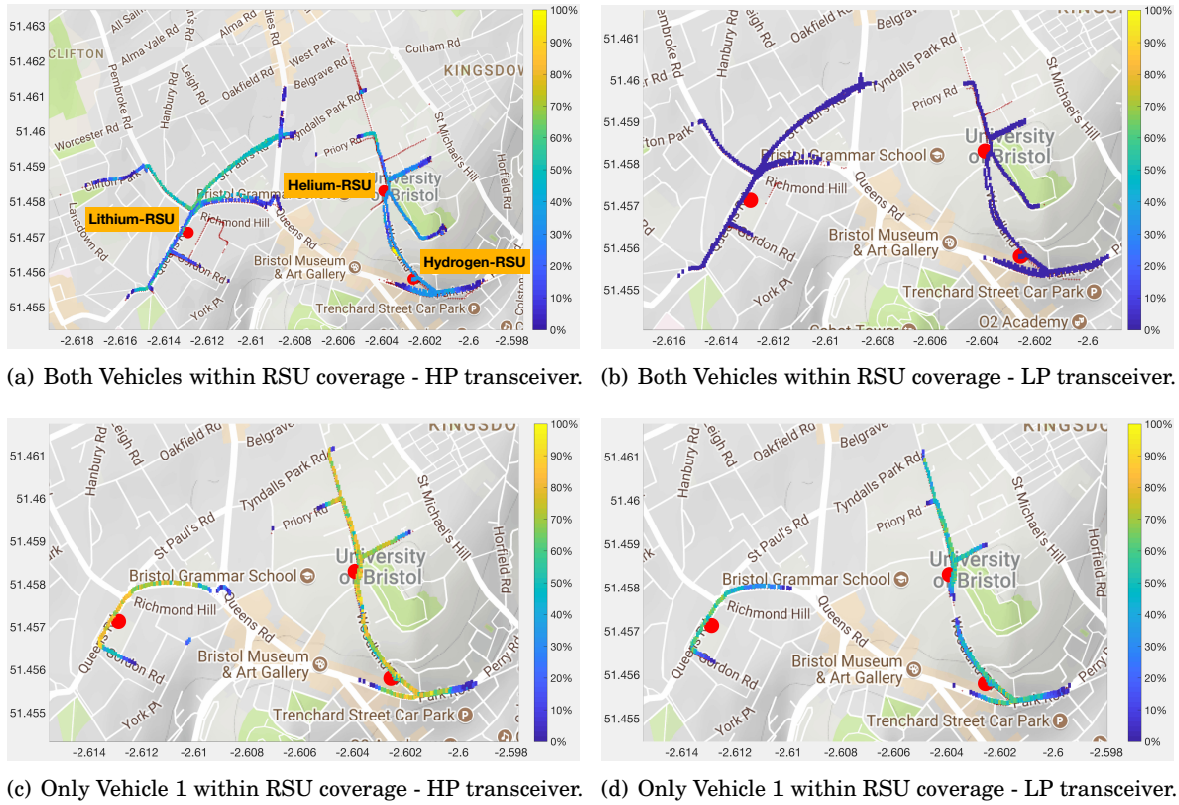
Figure 4.20: Heatmap results for different V2I scenarios (HP and LP NICs).

LOS coverage compared to the other RSUs. The different locations and buildings were chosen to evaluate how the position of an RSU can affect the performance of the network.

Two vehicles (as in Fig. 4.3(c)), equipped with one OBU each, were driving randomly around the city. The second OBU unit shown in Fig. 4.3(b) was there for backup purposes only. All the devices in our system generated and transmitted a CAM per NIC every 10 ms. Each CAM, encapsulating the information described in Sec. 4.4.2, was logged at the transmitter and the receiver side. The log files generated were used later to produce the results that will be described in the next section. Throughout the three days of field trials, we exchanged ~50 million CAMs. Some of our results will use a subset of these exchanged messages. Our entire dataset is available for download in [147]. To the best of our knowledge is one the largest data repositories focused on V2X communications.

Firstly, we start with the V2I scenario. Fig. 4.20 presents the heatmap results for the PDR from all CAMs transmitted from an RSU and received at the vehicle side. The results present the PDR for the vehicle no. 1. Vehicle no. 2 acts as an interferer, as mentioned before, when both vehicles are within the same RSU coverage range. Finally, the red crosses, show the position of a vehicle when a CAM broadcast was successfully received at the RSU side.

Figs. 4.20(a) and 4.20(b) show the PDR results when both vehicles were driving within the
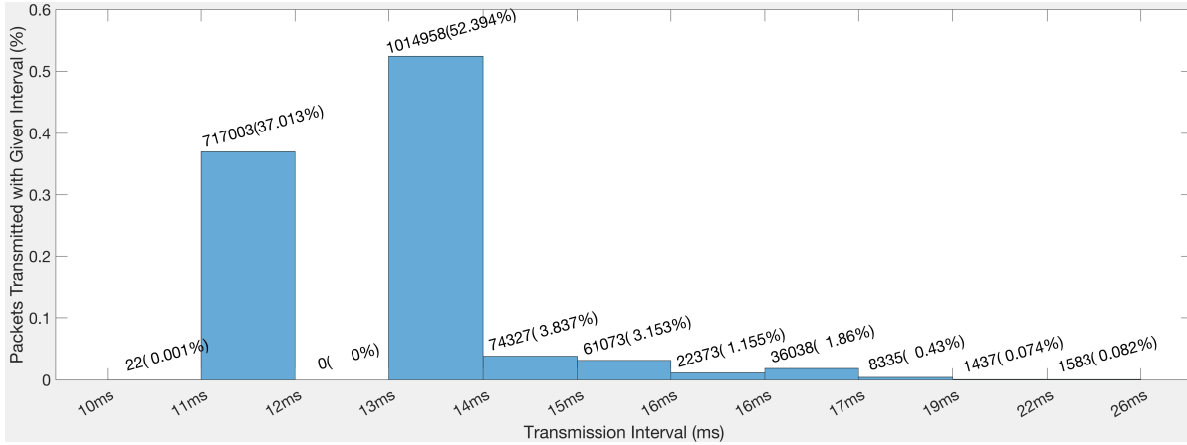
Figure 4.21: Transmission Intervals between two DSRC CAMs.

coverage regions of the RSUs. Figs. 4.20(c) and 4.20(d) show the results when only vehicle no. 1 was within coverage. As described, the DSRC CAMs are being broadcast from all NICs every 10 ms without having any coordination on the channel usage. As shown, there is a significant PDR difference of 30% between the different scenarios, for both the HP and LP transceivers. The reasons behind that could be many and should be taken into consideration. At first, we use two antennas that are not perfectly aligned (as in Sec. 4.7.1). This can degrade the signal performance up to a level that is not easily decoded by a transceiver. Secondly, this is a prototyped testbed with COTS devices that are not meant to be operated on a moving vehicle. From the literature, it is known that vibrations can have a disruptive effect on an antenna [173] and could amplify the interference effects. What is more, when an antenna is placed on a large metal surface like the car rooftop, then the rooftop becomes its groundplane. This could potentially increase the disruptions on the channel as well. The above, including the interference introduced of from the second vehicle, when acting as an interferer, led to an increased number of frame collisions and longer MAC-layer contention intervals at the receiver side. These results show the huge number of factors that needs to be considered when deploying a city-scale vehicular testbed. All in all, these results can be an initial benchmark for a real-world implementation that can be later enhanced in order to become from a prototype to an actual real-world system.

The difference can be observed at the RSU side as well. As shown, the heatmap data overlap with the red crosses in Figs 4.20(c) and 4.20(d), while they do not precisely match the heatmap in the first two figures. This means that when the interfering vehicle was present, vehicle no. 1 was not always able to establish a bidirectional communication link with the RSUs.

In Fig. 4.21, we present the frequencies of the transmission interval between two DSRC CAM. This is an example from Hydrogen-RSU for all CAMs transmitted throughout one day of field trials. The remaining devices and days produced similar results, therefore will not be presented in this chapter. As shown, even though the CAM transmission interval was set at 10 ms, our

85
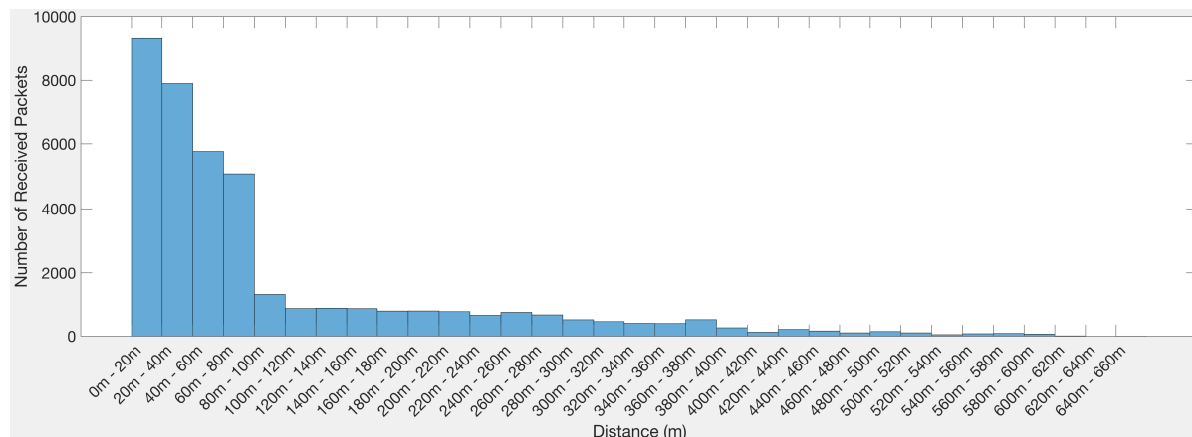
Figure 4.22: Awareness Horizon for the V2V Scenario - HP transceiver.

testbed generates frames at a different rate. Most of the frames are generated and exchanged either every 12 ms or 14 ms. This was expected as our devices are built upon a single-core CPU, which executes tasks with the same priority according to the Linux *Deadline I/O Scheduler*. To that extent, the CPU cannot fetch/push CAMs streams towards the transceivers at a constant I/O rate. These inconsistencies should be taken into account when designing vehicular applications with strict latency requirements. Generating and processing the packets at a stronger Fog node computer, and using the transceivers as the medium to exchange the packets, will significantly improve the consistency of the transmission rate.

Finally, Fig. 4.22 presents the awareness horizon for the V2V scenario, i.e. the Euclidean distance between the vehicles when a CAM is received. For this experiment, two vehicles were driving in opposing directions on a highway section of the road exchanging CAMs every time they were crossing paths. As shown in Fig. 4.21 most packets are being transmitted every 12 ms or every 14 ms. Given that the vehicles drive at a constant speed, we can estimate that a similar number of packets was exchanged at every distance interval. We observe that when the vehicles are in close proximity, a bigger number of packets is being received compared to longer distances. When the vehicles are more than 80 m apart, most of the packets are never delivered. Similar performance can be observed in the rural and urban trials conducted. From the above, we can observe that using the previously described setup, we can achieve adequate V2V communications for up to about 80 m. For sensor features exchange at longer distances, a multi-hop communication using V2V or V2I links is necessary. This section concludes the performance evaluation of all the contributions presented in this chapter.

4.8 Conclusions

In this chapter, we presented our research activities to accurately and cost-effectively integrate IEEE 802.11p/DSRC protocol stack in a next-generation C-ITS. Some of the main challenges

that we addressed were related to the evaluation of large-scale V2X communication scenarios. As discussed, two tools can be used: 1) computer simulations/theoretical models, and 2) real-world field trials. Initially, we discussed the advantages and drawbacks of both approaches and described the necessity for establishing a connection between them. Our activities gave us an excellent insight of how a network simulator operates, we observed the differences compared to real-world devices, and that is very difficult to replicate the inconsistencies of the real-world within a simulated scenario, e.g. the different behaviour of two drivers. Even after our proposed calibration solution for Veins-INET, we can still observe some noticeable difference that should be taken into consideration when evaluating a vehicular network. We also observed that introducing more complex simulation parameters significantly increases the execution time. Our solution strategy was the implementation of a multi-threaded version of Veins-INET that manages to improve the simulation time and make large-scale simulations more feasible for the end user. Finally, as we move towards the deployment of real-world C-ITSs, we focused our research activities towards a real-world city-scale deployment of an IEEE 802.11p/DSCR communication system. We based our system on the ITS-G5 and IEEE 802.11p/DSRC standards. Later, we presented our large-scale field trials, conducted around the City of Bristol. Some of our key observations are the necessity for more sophisticated MAC-coordination schemes as the co-existence of various vehicles under the same coverage area generated interference problems. What is more, our system will be benefited by a Fog-computing service due to the inconsistencies observed during the generation and the exchange of data. This will also help with the centralised control of our network, that will considerably enhance the performance. These are considered as activities for future work.

5

A MmWave MAC-layer Approach for Device-to-Device (D2D) V2X Beam Steering

5.1 Introduction and Contributions

5.1.1 Introduction

Chapter 5 is the first chapter related to mmWaves and presents our work regarding a novel intelligent beamforming algorithm, called Smart Motion Prediction Beam Alignment (SAMBA), able to efficiently align the V2X mmWave beams. Our algorithm operates under a heterogeneous DSRC/mmWave network architecture. Using out-of-band sensor information exchanged via a DSRC link, the RSUs and the CAVs can adapt their beams to compensate for the increased propagation loss of the mmWaves frequencies being. With a better beam alignment, they will be able to establish gigabit-per-second and ultra-low latency link between them and adapt to the highly dynamic vehicular network topologies and CAVs mobility patterns.

In this chapter, we will first present the limitations of the legacy IEEE 802.11ad beamforming strategy (Sec. 5.2). We start by presenting the beamforming technique as described in the standard and analyse the overhead introduced. Later, in Sec. 5.3, we discuss the challenges of the traditional beamforming strategy when used within the highly dynamic vehicular network topology and support our statement with some preliminary results. Sec. 5.4, presents our heterogeneous algorithm, describing the model we designed and how it is affected by the different position errors. Two different optimisation algorithms are introduced in Sec. 5.5 to maximise the data rate of our system. Two different vehicular scenarios, a simple straight road and a more complicated large-scale urban road network, are evaluated in Sec 5.6 and the final remarks of this chapter are presented in Sec. 5.7.

5.1.2 Contributions

For the work presented in Chapter 5 the author:

- Examined the limitations of the legacy beamforming technique used with IEEE 802.11ad standard.

- Derived equations to analyse the collision probability and the overhead generated during the legacy beamforming.

- Discussed the challenges and the limitations of IEEE 802.11ad when particularly used under vehicular communication frameworks using some preliminary results.

- Proposed a new heterogeneous beamforming approach for V2I links that are based on various motion sensor data acquired from a moving vehicle.

- Analysed the individual error components with respect to the estimated position of the vehicle.

- Proposed two beamwidth optimisation algorithms to maximise the system performance, i.e.:

  – Optimisation algorithm based on the position error.
  – Optimisation algorithm based on the distance between an RSU and a CAV.

- Designed a system level simulator in Matlab and evaluated different urban scenarios to identify the feasibility of using the proposed algorithms and improve the performance.

The contents of this chapter were originally presented at two conferences:

- IEEE Vehicular Technology Conference (VTC-Fall) 2017 [37].

- International Conference on Antennas, Propagation & RF Technology for Transport and Autonomous Platforms 2017 [36].

5.2 Traditional IEEE 802.11ad Beamforming Algorithm

IEEE 802.11ad is the dominant standard for mmWave communications [174]. IEEE 802.11ad introduced a novel network called Personal Basic Service Set (PBSS) where nodes communicate in an ad-doc-like manner. One of the participating nodes acts as the PBSS Control Point (PCP), similar to an Access Point (AP), announcing the network and organising the medium access. A PBSS may consist of up to $N \leq 254$ number of Directional Multi-Gigabit (DMG) devices. In IEEE 802.11ad, the interval between two beacon frames is defined as the *Beacon Interval (BI)*. During a BI, management and network information is exchanged at first, followed by the data

transmission. Referring to the standard [174], in this section we are going to describe the different time intervals that can be found within one BI, concerning the beamforming process.

The BI is subdivided into two different access periods named as *Beacon Header Interval (BHI)* and *Data Transmission Interval (DTI)*. These two access periods and a visualisation of the beamforming process can be seen in Fig. 5.1. At first, *BHI* facilitates the exchange of management information and network announcements. The second interval (*DTI*) is responsible for the data transmission. BHI is further subdivided into three shorter access periods that are the following:

- *Beacon Transmission Interval (BTI)*: Used for the network announcement and the beamforming training.

- *Association Beamforming Training (A-BFT)*: The system antennas are trained and paired with the PCP/AP.

- *Announcement Transmission Interval (ATI)*: Management information is exchanged with the associated and beam-trained stations.

The propagation characteristics in the frequency band of 60 GHz result in severe signal attenuation during quasi-omnidirectional communications. Therefore, the MAC layer of IEEE 802.11ad introduces the concept of "virtual" antenna sectors. These sectors divide the azimuth plane into a number of equal portions, depending on the type of the device used (e.g. a PCP/AP will utilise more "virtual" sectors than a handheld device). These "virtual" sectors can be further subdivided with respect to the minimum beamwidth of the antenna. The above create a *two-layer model* representation of the beams and the beamforming process (Fig. 5.1). In such manner, the beamforming in IEEE 802.11ad is performed in two phases. At first the initial best "virtual" sectors are chosen and later, the beam alignment is further refined to achieve the best performance between the two devices. Each phase requires a bidirectional frame exchange to train the "virtual" antenna sectors.

5.2.1 Two-Phase Beamforming Process for IEEE 802.11ad

Consider a typical vehicular network with one PCP/AP acting as an RSU and a vehicle travelling on the road. The first training phase, training the first layer of antenna sectors (as shown in Fig. 5.1), is called *Sector Level Sweep (SLS)*. During this phase, the RSU uses an iterative sweeping process, transmitting directional frames on each sector. The vehicle, listening quasi-omnidirectionally, transmits feedback information for each frame received and the best RSU sector is chosen based on the strongest Signal-to-Noise Ratio (SNR).

Later, during the *Beam Refinement Phase (BRP)* (second-layer), the antenna of the vehicle is trained, and the beams are further refined. With the beamwidth being inversely proportional to the antenna gain, during this phase, a narrower beamwidth is chosen to achieve higher antenna gain. Also, the beams of both devices are more accurately aligned, choosing finally a pair of
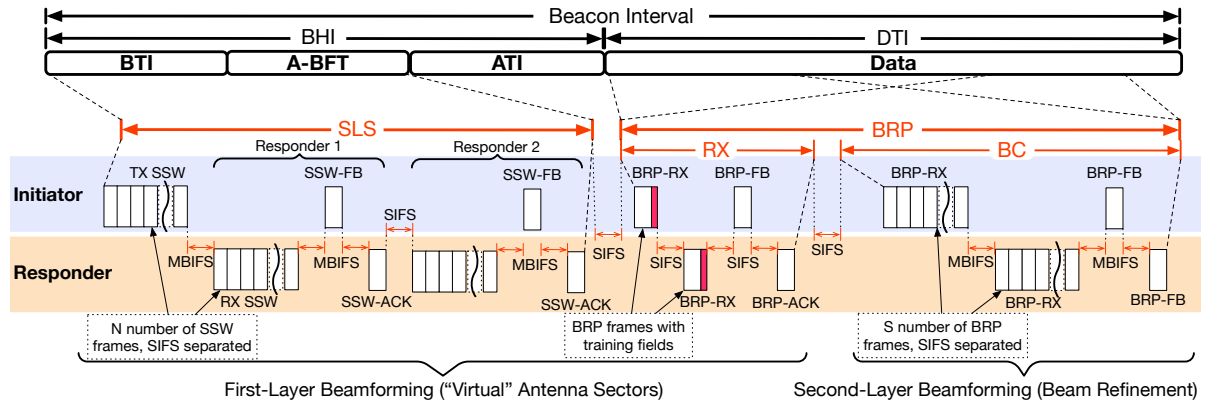
Figure 5.1: The Beacon Interval and the *two-layer* beamforming: An example of two stations
participating in the beamforming process. The frames exchanged, the interframe spacing and the
intervals that each phase requires.

refined beams able to compensate for the channel losses at 60 GHz. In the next section, we are
going to describe in detail the frames exchanged during this two-phase process, and we will try to
derive equations that will calculate the time required for each phase.

5.2.2   Overhead Analysis of the IEEE 802.11ad Beamforming Process

Consider a vehicular network with one RSU and a number of vehicles on the road where all devices
operate with respect to the IEEE 802.11ad standard. According to [103] the beam switching for a
mmWave phased-array antenna is almost instantaneous ($\simeq 50$ ns). Therefore, the IEEE 802.11ad
beamforming training delay is entirely related to the number of frames exchanged. The training
delay will be further analysed later in this section.

Starting with the SLS, it is being executed during BTI and A-BFT in four steps (Fig. 5.1):

1. The *initiator*, in that case, the RSU, transmits one directional training frame per sector
   while the receiving vehicles (*responders*) listen in quasi-omnidirectional mode.

2. The responders reply with a directional frame throughout all their sectors.

3. Feedback information is transmitted from the initiator within the Sector Sweep Feedback
   (SSW-FB) frames.

4. The feedback information is acknowledged by the responders with the Sector Sweep Ac-
   knowledges (SSW-ACKs).

When more than one stations exist in the coverage region of PCP/AP, the slotted A-BFT
introduces a contention-based response period taking values from a uniform random distribution
$U(4,8)$ [174]. Allocating one slot per vehicle, more than one devices can respond to the beacon

92

sweep reducing the collisions of the RX-SSW frames. The total time required for a SLS period can be calculated as:

$$(5.1) \qquad T_{SLS} = K \cdot T_{TX-SSW} + N \cdot (K \cdot T_{RX-SSW} + T_{SSW-FB/ACK}) + T_{IFS}$$

where T_{TX-SSW}, T_{RX-SSW}, T_{SSW-FB} and $T_{SSW-ACK}$ are the required time for the different frames exchanged, K is the number of "virtual" sectors (in this case, the same number was assumed for all the devices), N is the number of stations around the PCP/AP and T_{IFS} is the total interframe spacing time and is equal to:

$$(5.2) \qquad T_{IFS} = (N+1)(K-1) \cdot T_{SBIFS} + 3 \cdot T_{MBIFS}.$$

where T_{SBIFS} is the Short Beamforming IFS time and is equal to $T_{SBIFS} = 1\,\mu s$. Finally, T_{MBIFS} is the Medium Beamforming IFS, equal to $T_{MBIFS} = 3 \cdot T_{SIFS}$ with $T_{SIFS} = 3\,\mu s$.

During the BRP phase, multiple configurations can be tested with one frame transmission, reducing the overhead compared to SLS. BRP can be divided into two subphases. At first, the best RX antenna sector is found by exchanging BRP frames appending *transmit and receive training fields (TRN-TX/RX)* and followed by a feedback frame and an acknowledgement. The time required is given as follows:

$$(5.3) \qquad T_{RX} = 2 \cdot T_{BRP} + T_{BRP-FB} + T_{BRP-ACK} + 3 \cdot T_{SIFS}$$

By the end of the above process, a pair of "virtual" antenna sectors is chosen with respect to the higher SNR. During the second subphase, called *Beam Combining (BC)*, the beams are further refined. A set of pairwise antenna weight vector combinations is tested between the two devices as a directional link between the devices is already established. The time required can be calculated as:

$$(5.4) \qquad T_{BC} = S \cdot T_{BRP} + T_{BRP-FB/ACK} + 3 \cdot T_{MBIFS} + (S-1) \cdot T_{SIFS}$$

where S is the number of antenna weight vector combinations tested. Finally, the total time required for the beamforming process is as follows:

$$(5.5) \qquad T_{all} = T_{SLS} + T_{RX} + T_{BC} + 2 \cdot T_{SIFS}$$

The time intervals introduced in the previous equations can be calculated as the total number of bits exchanged for every interval divided by the bitrate used. For example, all the frames exchanged during the SLS phase, are transmitted with the most resilient MCS - thus at a very low bitrate (27.5 Mbps). On the other hand, the frames during the BRP phase can be exchanged with any given MCS used with IEEE 802.11ad. For our work, the faster bitrate provided (6756.75 Mbps) was chosen, for demonstration purposes only. More information about the length of each frame exchanged can be found in [174].

The analysis above is an approximation of the time required for the beamforming process of
N number of vehicles. However, a perfect channel with zero frame loss was considered, and BRP
can be more complicated (e.g. a BRP setup subphase might be required if BRP does not follow
an SSW-ACK). Therefore, the above equations can give a rough approximation of the minimum
time wasted from the beamforming process of IEEE 802.11ad and may not precisely reflect all
the scenarios.

5.2.3   Frame Collision Analysis during the Beamforming Process

Following the overhead analysis in Sec. 5.2.2, in this section we will analyse the collision proba-
bility during the beamforming process of IEEE 802.11ad. As described before the A-BFT interval
is slotted to accommodate the more devices within the coverage region of a PCP/AP.

However, the slotted A-BFT can lead to collisions when two or more vehicles are randomly
assigned the same slot. What is more, the predefined number of slots (≤ 8 slots   [174]) is
insufficient for urban scenarios, as more than eight vehicles can convene within an RSU coverage
region. The *probability of collision* within an A-BFT slot can be defined as follows:

$$(5.6) \qquad\qquad P_{\text{col}} = 1 - \frac{x!\,(x-1)!}{(x-v)!\,(x+v-1)!}$$

where x is the number of slots, and v is the number of vehicles. The probability of avoiding a
collision is the ratio of the combinations when only one vehicle is allocated per slot $\binom{x}{v}$, over the
number of vehicles allocated to a number of slots $\binom{x+v-1}{v}$. P_{col} is the complement of the above. In
the next section, we are going to present some results with respect to the utilisation of mmWaves
and IEEE 802.11ad with a network of CAVs. We will also discuss the feasibility of using IEEE
802.11ad for vehicular communications and the drawbacks that we identified.

5.3   Challenges of Legacy Beamforming Strategy for Future ITSs

As described in the standard [174], the BI length is limited to 1000 ms [174]. Concerning the
surrounding environment, the length can be optimised to achieve the best performance. Longer
intervals increase throughput and reduce the management frame transmission. However, the
system becomes intolerant to the delay spread. Misalignments between the TX and RX antennas
can lead to more severe delay spread and consequently degradation in the performance. A
typical BI length for indoor environments (zero or low mobility) is around 100 ms. However, for
moving vehicles more frequent beam switching is required (<30 ms) to avoid severe performance
degradation due to beam misalignments from the increased mobility. In this section, we will
try and identify the limitations of the legacy beamforming technique when used for vehicular
communications.
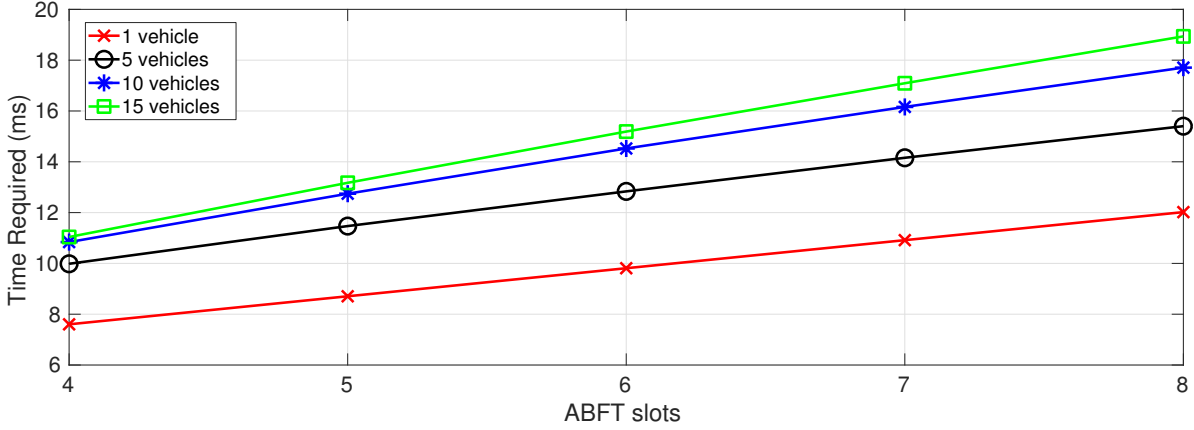
Figure 5.2: Example of the average delay introduced every BI from legacy beamforming training. 16 "virtual" antenna sectors were used for both TX and RX antennas, and a different number of vehicles was considered.

5.3.1 Limitations Concerning the Delay Introduced from Beamforming

As described in Sec 5.2.2, the bidirectional exchange of training frames increases the overhead delay. The time required for the beamforming is predefined and depends on the network and the surrounding environment. Thus, shorter BIs will lead to a bigger portion of time being used for beamforming. An example of the overhead delay can be seen in Fig. 5.2. For this example, we assumed 16 "virtual" antenna sectors for both the TX and RX sides and a variable number of vehicles. The results were calculated with respect to the equations derived in Sec. 5.2.2. Assuming a BI of 30 ms, we see that more than one-third of the BI is misspent for beamforming when the number of vehicles is increased.

Increasing the overhead in our system will reduce the throughput as well. This behaviour was observed in Fig 5.3 where the achievable throughput is presented for the aforementioned scenario and is compared with the theoretical maximum of IEEE 802.11ad (when 64-QAM 13/16 is used). The achievable throughput, in this case, is the data rate that can be achieved averaged over the BI length after the beamforming process (no other management frame exchange or inter-frame spacing was considered). The positions of the vehicles were randomly chosen to be within the coverage range of the RSU, and the averaged throughput was calculated later. As shown, increasing the number of A-BFT slots reduces the throughput. Introducing more vehicles in the system will result in even worse performance with more than half of the theoretical maximum throughput being lost because of the beamforming process.

The strict QoS requirements for the next-generation automotive applications require *tactile-like* end-to-end delays (<10 ms) and increased data rates (≥1 Gbps). As shown in Figs. 5.2 and 5.3, IEEE 802.11ad cannot compensate with these requirements.
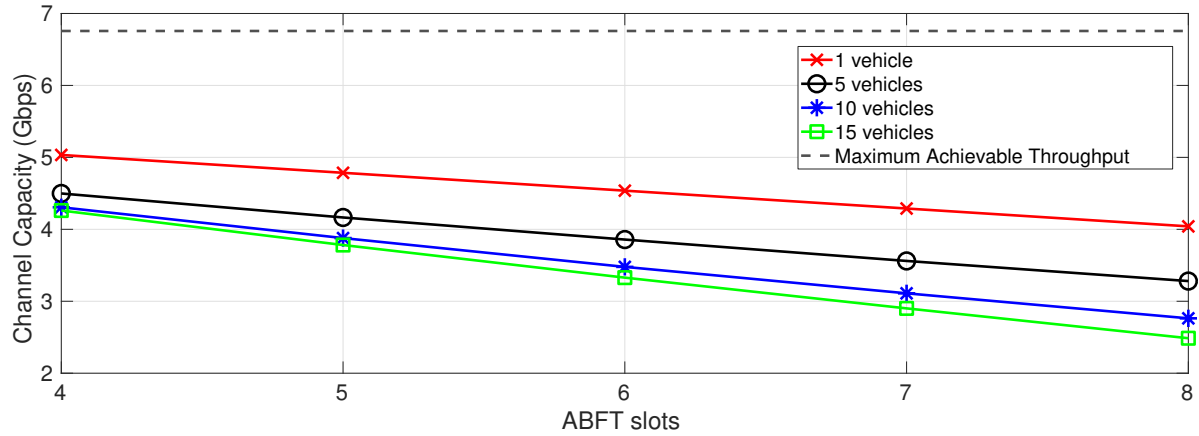
Figure 5.3: Achievable theoretical throughput for a different number of vehicles after the beam-forming process of IEEE 802.11ad.

5.3.2 Limitations due to the Collisions during the Beamforming Process

In the previous section, we presented some preliminary results when only the beamforming delay is considered in our system. In this section, we will present the limitations with respect to the collision probability and why the legacy beamforming technique of IEEE 802.11ad is not suitable for vehicular communications.

In Fig 5.4, we present the probability of at least one collision to happen during an A-BFT slot, for a different number of vehicles. For example, lets consider the scenario that we have four slots and one vehicle occupying each slot. If a fifth vehicle is introduced in the system and as all the slot are occupied, a collision will occur with probability of 1. Obviously, increasing the number of vehicles ever further, the collision probability will always be 1. Fig 5.4 shows that the collision probability dramatically increases during an A-BFT slot as the number of vehicles is increased - thus, significantly reducing the trained antenna beams. This is because the maximum number of A-BFT slots is 8, so increasing the number of vehicles within the coverage region of an RSU will lead to more collisions and consequently less trained antennas. A collision during the beamforming training will lead to a vehicle that will be idle for a BI (in this case for 30 ms) - thus reducing the sensor data flow to and from the vehicle and increasing the probability of accidents.

From those above, it is proven that the strict QoS requirements of Next-Generation ITSs, cannot be fulfilled when using the legacy IEEE 802.11ad beamforming technique. To that extent, we will introduce a new beamforming approach not relying on the in-band information exchanged. Achieving overhead-free beamforming, it will make mmWaves a viable solution for future ITSs. By that, we will be able to fulfil the demanding requirements for increased capacity from the next-generation vehicular applications and consequently improve the road safety.
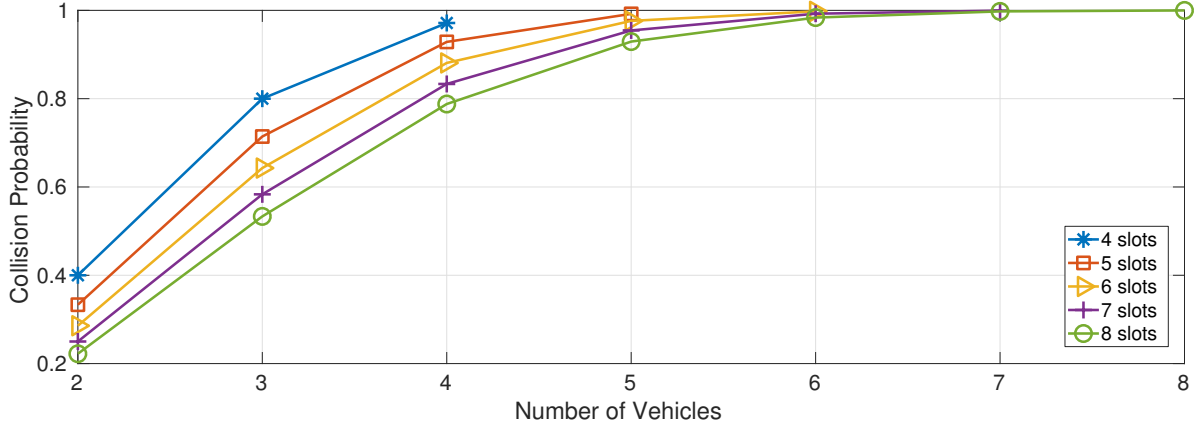
Figure 5.4: Collision probability during one A-BFT for a different number of vehicles.
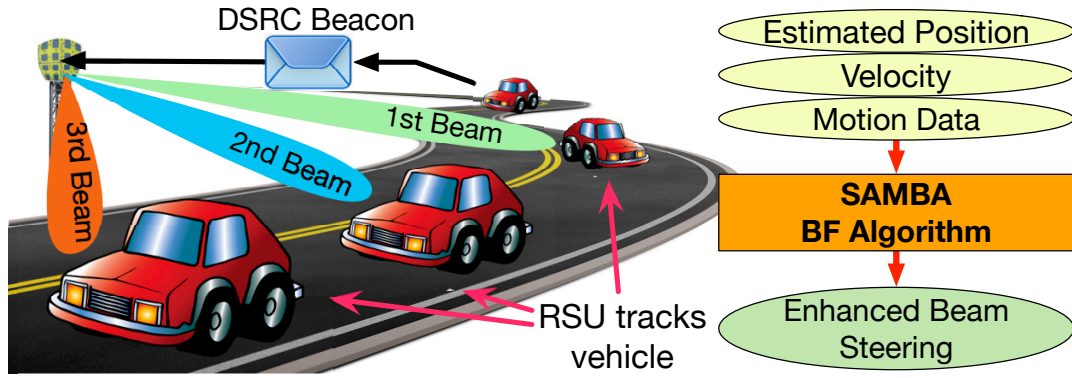
Figure 5.5: SAMBA System design: Position, velocity and motion information encapsulated in DSRC CAM messages are used for smart beamforming. RSUs predict the motion of the vehicle and its position and align their beams accordingly.

5.4 SAMBA: Heterogeneous DSRC/mmWave Beamforming for V2I Links

In Secs. 5.2.2 and 5.2.3 we derived equations that helped us calculate the overhead and the collision probability introduced from the IEEE 802.11ad legacy beamforming technique. Later (Sections 5.3.1 and 5.3.1) we presented some preliminary results with respect to the aforementioned analysis. To solve the previously mentioned problems, we propose the Smart Motion Prediction Beam Alignment (SAMBA) algorithm. Our algorithm operates in a heterogeneous manner combining DSRC and mmWave RATs. Leveraging from the position, the velocity and the motion information broadcast over the DSRC links, SAMBA can provide overhead-free beamforming, reduce the association delays, minimise the beam misalignments and enhance mmWave performance. SAMBA operates as shown in Fig. 5.5.

On the infrastructure side, and as shown in Alg. 2, SAMBA algorithm considers a road

---

Algorithm 2 SAMBA Algorithm: Infrastructure Side

Require: Vehicles encapsulate position, motion and velocity in CAM messages
Ensure: RSU_n has not changed after every update interval.
 1: **while** N number of Vehicles within the network range ($N \geq 1$) **do**
 2:     **if** New CAM Message received **then**
 3:         Find RSU_n for each vehicle $\triangleright RSU_n \rightarrow Closest\ RSU$
 4:         **if** $Received_{pos} \neq P_{pos}$ **then**
 5:             Beamforming: Align RSU_n beam based on $Received_{pos}$
 6:         **else**
 7:             Predict current position of vehicle P_{pos}
 8:             Beamforming: Align RSU_n beam based on P_{pos}
 9:         **end if**
10:     **else**
11:         **repeat** every Update Interval $\triangleright 30\,ms$
12:             Predict current position of vehicle P_{pos}
13:             Beamforming: Align RSU_n beam based on P_{pos}
14:         **until** New CAM message is received
15:     **end if**
16: **end while**

---

network with N number of vehicles, where $N \geq 1$. The information received from a vehicle is used
to decide whether the vehicle has moved, compared to the previously stored position. With respect
to all the positions, the RSUs decide their serving vehicles, i.e., each vehicle is served by its
closest RSU. When an updated position is received, the serving RSU aligns its beam accordingly.
Later, the RSUs can efficiently track the movement of each vehicle predicting its motion and
position. The update interval for SAMBA was predefined at 30 ms. By that, a comparable BI with
IEEE 802.11ad is used, and increased Doppler shift can be avoided.

On the vehicle side (Alg. 3), the vehicles transmit DSRC CAM Messages to all RSUs in range
encapsulating their velocity, their motion data (based on the *vehicle motion dynamics*) and their
estimated position. CAM messages are broadcast every 100 ms (DSRC CAM message interval)
and the acquired information is updated periodically. All CAVs, as smart entities of an ITS, can a
priori know the positions of the RSUs. To that extent, each vehicle aligns its beam towards the
closest RSU.

When more than one vehicles are within the coverage region of an RSU, a dynamic channel
time allocation access mechanism is used, that implements a polling based channel access, similar
to the one of IEEE 802.11ad [174]. Using the same mechanism for both approaches, it can be
ensured that the delays introduced by the resource allocation scheme are negligible for our
results.

Algorithm 3 SAMBA Beamforming Algorithm: Vehicle Side

Require: Vehicle in Range of RSU_i, $i \in \{1, ..., n\}$

  1: **repeat**

  2:     $Transmitted_{pos} \leftarrow$ GPS Coordinates with additive error   ▷ This is the E_{pos}

  3:     Align Beam with RSU_i ▷ Based on RSU positions

  4:     **if** CAM Message Timer=0 **then**

  5:         Transmit CAM Message   ▷ Encapsulate Motion, Velocity, Position data

  6:         Reset Timer to 30 ms

  7:     **end if**

  8: **until** Not in Range

5.4.1 Mobility Model and Position Based Beam Alignment

The synchronised flow traffic model [175] can accurately represent an urban scenario. It represents a continuous traffic flow, with no significant stoppages, where vehicles perform random manoeuvres (braking/accelerating, changing lanes) and tend to synchronise their movement. Velocity varies over time and is averaged around a mean value, following a Normal distribution, i.e., $s \sim \mathcal{N}(s_{\text{avg}}, 2)$. Velocity errors can be easily corrected employing data fusion techniques and were not considered in this model.

The position of the vehicle can be estimated based on acquired position, acquired most frequently from a GPS dongle. GPS positions introduce an error with a mean value of ~3 m and standard deviation of ~1 m [45]. Worse performance can be observed in urban environments due to the urban street canyons effects that are being introduced from the height of the buildings. As discussed though, increased accuracy can currently be achieved by fusing CAVs sensory data (e.g. with Kalman filters), being able to achieve centimetre-accuracy even under urban environments [48].

The estimated position of a vehicle, as said, is affected by this additive error, i.e. $E_{pos} = R_{pos} + e_{pos}$, where R_{pos} is the real position of a vehicle and $\log(e_{pos}) \sim \mathcal{N}(\mu, \sigma_v^2)$ is the log-Normal error. Terms μ and σ are the mean and standard deviation values for non-logarithmized samples with mean m and variance s and are defined as follows:

$$(5.7) \qquad \mu = \log\left(m\left(1 + \frac{s}{m^2}\right)^{-1/2}\right) \qquad \sigma_s = \sqrt{\log\left(1 + \frac{s}{m^2}\right)}$$

Knowing the estimated position, an RSU can steer its beam towards the vehicle. This can be achieved by calculating the angle $k°$ (with respect to the reference plane), using the trigonometric equations for right-angled triangles and the distance of the vehicle from the axis (as in Fig. 5.6(a)).

5.4.2 Vehicle Motion Dynamics and Motion-Prediction

CAV equipped with IMU sensors (e.g. magnetometers, accelerometers, gyroscopes), will be able to measure the motion changes of a vehicle. The acquired sensory data can be combined using data fusion algorithms. Their output is the angular velocity of the vehicle, measured as deg°/s, in

(a) Beam steered based on the position after the additive error.

(b) Position error components and the coverage and outage intervals.

(c) Vehicle moving on the perimeter of a circle (2D representation).
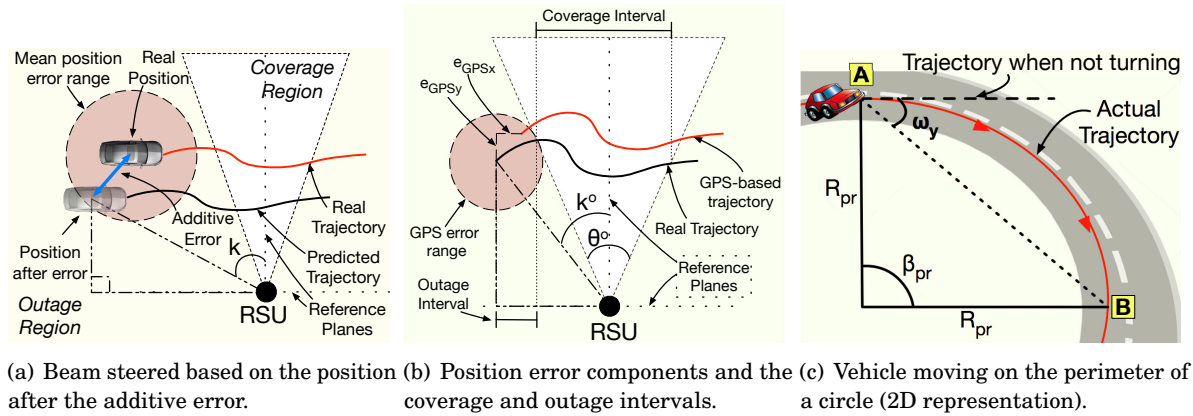
Figure 5.6: a) Beam steering, and how the position error introduces a difference to the trajectories, b) Position error components and the outage and coverage intervals, c) Motion of vehicle within a time interval.
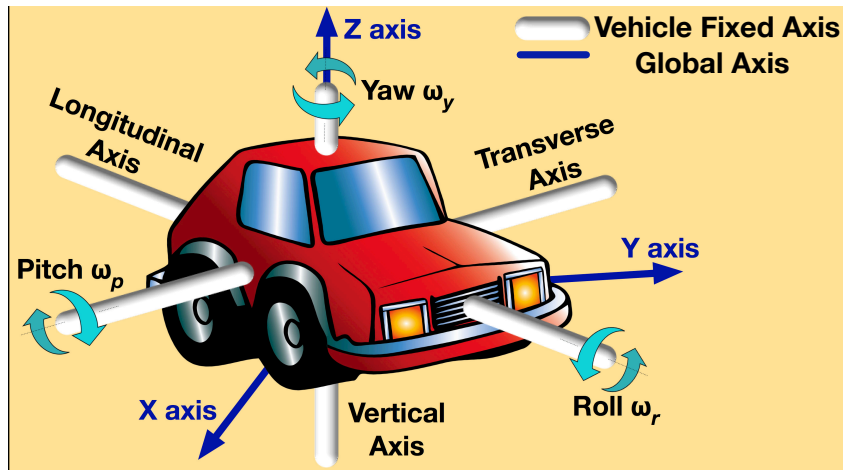
Figure 5.7: Vehicle Dynamics: Yaw ω_y, pitch ω_p and roll ω_r for a moving vehicle compared to the global axis system.

three different axis ($yaw\ \omega_y,\ pitch\ \omega_p,\ roll\ \omega_r$) as shown in Fig. 5.7. Sensory data errors being within the range of $0.2° - 1°$ [48] does not introduce significant errors in the algorithm. Therefore they were considered negligible.

Consider a constant angular speed. A vehicle in motion follows the surface of a sphere (when observed in 3D space). However, a vehicle changes its direction significantly on the vertical axis, as they are changing their direction on the road plane. The changes in the other two axes (longitudinal and transverse) are less drastic for a vehicle following a synchronised flow traffic model. To that extent, in this work, vehicles and RSUs will be considered as 2D objects, positioned on a plane. In this case, a vehicle follows the perimeter of a circle and its motion can be predicted based on ω_y, E_{pos}, and s_{avg}.

With respect to Fig. 5.6(c), a vehicle moving from A to B, will drive a distance of $l\widehat{AB}_{\text{pr}}$. The distance travelled and its angle β_{pr}, can be defined as follows:

$$\text{(5.8)} \qquad \beta_{\text{pr}} = m\widehat{AB} = 2\omega_{\text{y}} t_{\text{pr}} \qquad l\widehat{AB}_{\text{pr}} = s\, t_{\text{pr}}$$

where t_{pr} is the time elapsed from the latest CAM message and s is the velocity of the vehicle. Based on the circle properties and using (5.8), the radius of the circle R_{pr}, and the distance $\overline{AB}_{\text{pr}}$ between points A and B, are given as follows:

$$\text{(5.9)} \qquad R_{\text{pr}} = \frac{l\widehat{AB}_{\text{pr}}}{\pi\,\omega_{\text{y}} t_{\text{pr}}} \qquad \overline{AB}_{\text{pr}} = 2R_{\text{pr}} \sin(\omega_{\text{y}} t_{\text{pr}})$$

Finally, the predicted position $P_{\text{pos}}{}^1$ is calculated as:

$$\text{(5.10)} \qquad P_{\text{pos}}(x,y) = \begin{cases} P_{\text{pos}}(x) = E_{\text{pos}}(x) + \overline{AB}_{\text{pr}} \sin(\beta_{\text{pr}}) \\ P_{\text{pos}}(y) = E_{\text{pos}}(y) + \overline{AB}_{\text{pr}} \cos(\beta_{\text{pr}}) \end{cases}$$

5.4.3 Relation between position error, beamwidth and velocity

Consider a scenario where a number of RSUs are placed at the side of the road, and one vehicle performs a movement as in Fig. 5.8. In an ideal scenario with zero position error, the beamforming algorithm presented can achieve the maximum system performance with no outages (perfect prediction of the motion). However, GPS devices are imperfect and will lead to *outage interval*. The total time that a vehicle travels within the beam (*coverage interval*) is proportional to the beamwidth and the velocity and can be expressed as a function of them $T(s, \theta°)$ (Fig. 5.6(b)).

Distinct errors will affect the system differently. For any given position error, there will be an interval that there is an outage. The length of that interval is related to the position error and is minimised when it is zero. However, zero error cannot be achieved in real-world systems. The position error, as shown in [45], can be decomposed into (e_x, e_y) error components (easting and northing). Each error, with respect to the RSU position, will influence the system differently, as described in Sec. 5.4.2. For example, in Fig. 5.6(b), the e_x will cause a more significant outage than the e_y.

For non-consistent position errors, an analysis of the critical error components is required. Based on this analysis, severe performance degradations can be prevented by either utilising different network topologies or developing error correction algorithms. For example, for a given non-zero error causing misalignments, there is always an optimal non-zero beamwidth with respect to this error that maximises the system performance [176].

5.4.4 Link Budget Analysis, Antenna Gain and Beamwidth Relationship

To maximise the performance with respect to the beamwidth, an antenna model forming a relationship between the gain and beamwidth $\theta°$ should be derived a first. In this work, an ideal

---

[1]The model can be extended to a 3D scenario, by modifying (5.8), (5.9) and (5.10) to fit a spherical object.
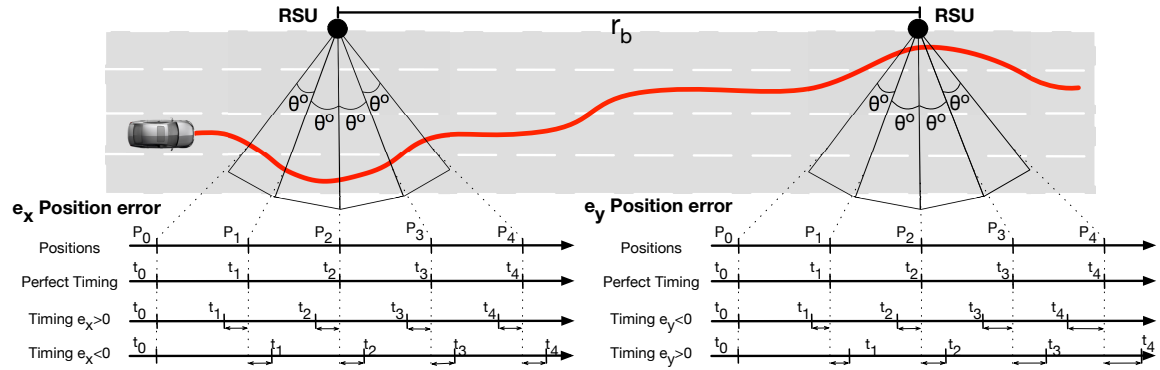
Figure 5.8: Vehicle performs a random movement on the road. Beam misalignments occur due to the position error, and the individual error components affect the misalignment differently. The movement of the vehicle is projected on a straight road.

beam is assumed with uniform gain and no sidelobes. This model can be easily replaced with a better approximation for a specific type of antennas, to achieve more accurate results. The directivity of an antenna, associated with the beam solid angle Ω_A is given as follows [170]:

$$(5.11) \qquad\qquad D = \frac{4\pi}{\Omega_A}$$

For this ideal beam representation, Ω_A can be approximated as $\Omega_A \approx \theta_{1r}\theta_{2r}$ where θ_{1r} and θ_{2r} are the half-power (-3 dB) beamwidths of the elevation and azimuthal polarisation planes respectively.

The antenna gain G can be given as [170]:

$$(5.12) \qquad\qquad G = \eta D$$

where η is the efficiency of the antenna associated with the antenna aperture. For an ideal antenna, the efficiency is equal to 100%, and the gain becomes equal to the directivity. What is more, for an ideal beam $\theta_{1r} = \theta_{2r} = \theta$. So, (5.12) with respect to the beamwidth $\theta°$ (measured in degrees) becomes as:

$$(5.13) \qquad\qquad G(\theta°) \simeq \frac{4\pi}{\theta^{2°}}$$

The received SNR for the above antenna can be expressed as the ratio of the *received power* P_{rx} over the *noise power* P_{noise}. The *received power* is given as follows [177]:

$$(5.14) \qquad\qquad P_{\mathrm{rx}}(\theta°) = P_{\mathrm{tx}} + G_{\mathrm{tx}}(\theta°) + G_{\mathrm{rx}}(\theta°) - PL$$

where P_{tx} is the transmitted power and $G_{\mathrm{rx}}(\theta°)$ and $G_{\mathrm{tx}}(\theta°)$ are the antenna gains for the receiver and the transmitter respectively. Finally, PL is the *path-loss component* and is defined as:

$$(5.15) \qquad\qquad PL = 10\,n\,\log_{10} d + S_f + C_{\mathrm{att}} + A_{\mathrm{att}} + R_{\mathrm{att}}$$

where n is the path-loss exponent and d is the distance separation between the RSU and the vehicle. S_f is the random shadowing effect following a log-Normal distribution $\log(S_f) \sim \mathcal{N}(0, \sigma^2_{S_f})$, with σ equal to 5.8 [178]. A_{att} and R_{att} are the average atmospheric and rain attenuation, respectively. Finally, C_{att} is a constant, representing the channel attenuation for a LOS link in an urban environment, measured at 20 m [177]. For this model, a LOS link is always assumed.

The P_{noise} can be calculated as:

$$(5.16) \qquad\qquad P_{\text{noise}} = N_{\text{fl}} + 10 \log_{10} B + N_{\text{fig}}$$

where N_{fl} is the noise floor value, N_{fig} is the noise figure, and B is the antenna bandwidth. The antenna gain is associated with the beamwidth as described before. The distance d between the RSU and the vehicle changes over time. For a given time t, the estimated position of the vehicle is known, and the distance can be easily calculated. Therefore, the SNR can be expressed as a function of the beamwidth and the time as follows:

$$(5.17) \qquad\qquad SNR(t, \theta°) = \frac{P_{\text{rx}}(t, \theta°)}{P_{\text{noise}}}$$

The rest of the variables are always considered as constant for this model.

5.4.5 Sensitivity Analysis for Individual Error Components

The instantaneous channel capacity for a given beamwidth and time can be calculated from the Shannon-Hartley theorem as follows:

$$(5.18) \qquad\qquad C(t, \theta°) = B \log_2(1 + SNR(t, \theta°))$$

With regard to the prediction model introduced in Sec. 5.4.2, the beam is steered when the vehicles reach the edge of the beam. The interval between two beam realignments is the time between t_i (the system switches to the ith beam) and t_{i+1} (the beam is realigned). For each t_i, a position P_i exists being the real position of the vehicle (Fig. 5.8).

Consider the same scenario as before (Sec. 5.4.3). Even though the vehicle performs a curved movement (Fig. 5.8), in the long-term tends to fend off the starting point, oscillating with respect to the x-axis. Given that only the position error exists (other sensors feedback ideal values), the real movement of the vehicle will be identical with the one predicted but shifted on the two axis. To that extent, the movement within the time interval $[t_i, t_{i+1}]$ is assumed to be a straight line. In contemplation to that, the data rate for a given $\theta°$ and a time interval $[t_i, t_{i+1}]$ can be calculated as:

$$(5.19) \qquad\qquad D_i(t, \theta°) = \int_{t_i}^{t_{i+1}} B \cdot \log_2(1 + SNR(t)) dt$$

However, due to the position error, the beamforming timing will be imperfect leading to outages. The two error components $P_e(e_x, e_y)$ for an estimated position $E_{\text{pos}}(t)$ at a given time t,

103

be calculated as follows:

$$(5.20) \qquad P_e\left(e_x, e_y\right)(t) = \begin{cases} e_x(t) = E_x(t) - R_x(t) \\ e_y(t) = E_y(t) - R_y(t) \end{cases}$$

where $(R_x(t), R_y(t))$ are the two components of the real position $R_{\text{pos}}(t)$ of the vehicle and
$(E_x(t), E_y(t))$ is the acquired estimated position given by the GPS. Both errors can be divided
in two cases: 1) when $e_x \geq 0$ and $e_x < 0$ and 2) when $e_y \geq 0$ and $e_y < 0$. For example, if
$e_x(t) > 0 \Rightarrow E_x(t) > R_x(t)$, meaning that the beam steering will be delayed creating an outage.
Therefore, the moment of the ith beam alignment is given as:

$$(5.21) \qquad \widehat{t_i} = \frac{\sqrt{((P_{ix} - P_{0x}) + e_x)^2 + ((P_{iy} - P_{0y}) + e_y)^2}}{s}$$

where P_0 is the initial position of the vehicle (as shown in Fig. 5.8). Based on the above, the
data rate (5.19) should be calculated for the portion on the interval that the beam is aligned, i.e.
$[\widehat{t_i}, \widehat{t_{i+1}}]$. The above is valid only when there is at least a very short interval where there is no
outage. In the case of total misalignment, the data rate is equal to zero.

Assuming that the error components are not so severe to cause total misalignment, their
impact on the system performance can be analysed. *Differential sensitivity analysis* [179] was
used, meaning that the sensitivity coefficient U for a particular independent variable is calculated
from the partial derivative of the dependent variable with respect to the independent variable.
For a predefined beamwidth $\theta°$ and denoting one of the errors as constant, the relationship of the
individual uncertainty component with the channel capacity for the ith beam can be calculated
as follows:

$$(5.22) \qquad U_i(e^*) = \frac{\partial D_i(t|e^*, \theta°)}{\partial e} = \frac{\partial}{\partial e} \int_{\widehat{t_i}}^{\widehat{t_{i+1}}} B \log_2(1 + SNR(t)) dt$$

The formula of the channel capacity within the integral in (5.19) has an anti-derivative. To
that extend, and denoting it as $c(t)$, can be calculated using the fundamental theorem of calculus:

$$(5.23) \qquad D_i(t|e^*) = \int_{\widehat{t_i}}^{\widehat{t_{i+1}}} c(t) dt = C(\widehat{t_{i+1}}) - C(\widehat{t_i})$$

Now, using the chain rule for the partial derivative of the sensitivity coefficient (5.22) we have:

$$(5.24) \qquad U_i(e^*) = \frac{\partial D_i(t|e^*)}{\partial e} = C'(\widehat{t_{i+1}}) \frac{\partial(\widehat{t_{i+1}})}{\partial e} - C(\widehat{t_i}) \frac{\partial(\widehat{t_i})}{\partial e}$$

The above equations can be numerically evaluated by calculating the sensitivity coefficient
for both the error components e_x and e_y. Analysing the individual errors before developing a new
ITS can significantly enhance the system performance. The impact of systematic errors can be
confined to maximise the performance. This can be done in many ways. For example, changing
the position of the infrastructure RSU devices, a system will easier compensate for these errors.

Another solution is the development of correction algorithms able to tackle specific errors. For this work, a 2D representation was used for our system. GPS northing and easting error will not have significant differences in a real-world system. However, the same analysis can be applied to a 3D system as well, where the errors between the azimuth and the elevation plane have significant differences [45].

5.5 Beamwidth Optimisation Algorithms

As described in Sec. 5.4.5, the two ways of compensating with a systematic position error is either physically changing the RSU positions to minimise it, or by developing correction algorithms. Nevertheless, reducing the influence of the position error will increase the system performance. However, any existing error will always lead to an outage. For an a priory known error, in this section we will describe two different optimisation algorithms that were proposed and evaluated, and discuss their benefits in the overall system performance.

5.5.1 Beamwidth Optimisation based on the Position Error

The position error P_e described before is a random variable. For our system, we are interested in maximising the data rate, i.e. $\arg\max_{\theta°} D_i(t, \theta°|P_e)$ at any given time t and beamwidth $\theta°$ with respect to the position error. For the long-term average value, the maximum data rate can be given as the expectation of the data rate denoting that it is averaging over P_e:

$$(5.25) \qquad \hat{\theta}° = \arg\max_{\theta°} \mathbb{E}_{P_e}[D_i(t, \theta°)]$$

P_e is decomposed in e_x and e_y as described in Sec. 5.4.5 and (5.20), which are both continuous random variables. Therefore, $\mathbb{E}_{P_e}[\cdot]$ is a positive linear function and the equation 5.25 can be rewritten as:

$$(5.26) \qquad \hat{\theta}° = \arg\max_{\theta°} \mathbb{E}_{e_x}[D_i(t, \theta°)] + \arg\max_{\theta°} \mathbb{E}_{e_y}[D_i(t, \theta°)]$$

Denoting $f(e_x)$ and $f(e_y)$ as the probability distribution functions for e_x and e_y respectively, the two expected values can be calculated as:

$$(5.27) \qquad \mathbb{E}_{e_x}[D_i(t, \theta°)] = \int_{-\infty}^{\infty} D_i(t, \theta°) f(e_x) \, de_x$$

$$(5.28) \qquad \mathbb{E}_{e_y}[D_i(t, \theta°)] = \int_{-\infty}^{\infty} D_i(t, \theta°) f(e_y) \, de_y$$

However, the above equations should be limited to consider only the interval that there is no outage. Therefore, to accurately calculate the expected value of the data rate over the P_e, the limits should be updated accordingly.

Total misalignment happens when the magnitude of the P_e is greater than the distance
from one edge of the beam to the other. In the time domain, this can be expressed as $\widehat{t_i} > t_{i+1} \Leftrightarrow$
$\|\widehat{P_i}\|/s > \|P_{i+1}\|/s$ for $e_x < 0$ and $e_y \geq 0$ and $t_i > \widehat{t_{i+1}} \Leftrightarrow \|P_i\|/s > \|\widehat{P_{i+1}}\|/s$ for $e_x \geq 0$ and $e_y < 0$.
From the above it can be calculated that the total misalignment conditions are:

$$(5.29) \qquad e_x = \begin{cases} e_x > P_i + P_{i+1}, & \text{for } e_x \geq 0 \\ e_x < P_{i+1} - P_i, & \text{for } e_x < 0 \end{cases}$$

$$(5.30) \qquad e_y = \begin{cases} e_y > P_i + P_{i+1}, & \text{for } e_y < 0 \\ e_y < P_{i+1} - P_i, & \text{for } e_y \geq 0 \end{cases}$$

So, using the above limits we have:

$$(5.31) \qquad \begin{aligned} \mathbb{E}_{e_x}[D_i(t,\theta°)] = & \int_0^{P_i+P_{i+1}} D_i(t,\theta°|e_x \geq 0) f(e_x) \, de_x \\ & + \int_{P_{i+1}-P_i}^0 D_i(t,\theta°|x_e < 0) f(e_x) \, de_x \end{aligned}$$

$$(5.32) \qquad \begin{aligned} \mathbb{E}_{e_y}[D_i(t,\theta°)] = & \int_0^{P_{i+1}-P_i} D_i(t,\theta°|e_y \geq 0) f(e_y) \, de_y \\ & + \int_{P_i+P_{i+1}}^0 D_i(t,\theta°|e_y < 0) f(e_y) \, de_y \end{aligned}$$

The above equations can be numerically evaluated, and the optimum beamwidth can be found
for a given position error. In Sec. 5.6, we will present a numerical example based on the equations
above to prove the validity of our model.

5.5.2 Beamwidth Optimisation Concerning the Distance from a CAV

Consider a scenario where a vehicle travels on a road, approaching the RSU from a distance,
passing by and fending off until it is outside of the coverage region. It is observed that the
beam covers a much broader area at its edge. A wider beam implies a wider beamwidth and
consequently low antenna gains and SNR. To that extent, narrower beams away from the RSU
and increased beamwidth as the vehicle gets closer, will provide an optimal solution and maximise
the performance.

As SAMBA relies on the position information, the error introduced will lead to misalignments
for very narrow beams. From geometry, it is known that the incentre is the centre of the triangular
area and is equally spaced from the beam edges. To that extent, adapting θ accordingly, the
vehicle can be positioned at the centre of the beam.

Centring the vehicle with respect to the beam edges will maximise the data rate and minimise
the misalignments compensating with the random errors. To do so, the incentre point is assumed
to be I_{pos} and d_{in} is the distance of this point from the RSU. To maximise the data rate, the

Table 5.1: List of simulation parameters for SAMBA V2I scenarios.

Parameter		Value	
Carrier Frequency	f_{c}	60	GHz
Bandwidth	B	2.16	GHz
Path-Loss Exponent	n	2.66	[180]
Atmospheric Attenuation	A_{att}	15	dB km^{-1}
Rain Attenuation	R_{att}	25	dB km^{-1} (in the UK)
Channel Attenuation	C_{att}	70	dB [177]
Transmission power	P_{tx}	10	dBm
Noise Figure	N_{fig}	6	dB
Noise Floor	N_{fl}	−174	dBm
Road Block Length	r_b	40	m
BI IEEE 802.11ad		30	ms
DSRC beacon interval		100	ms
Position update interval		1000	ms

highest MCS should be considered for our model, i.e., the sensitivity threshold S_{max} of the highest MCS (in our model we consider 64-QAM $^{13}/_{16}$ for our maximum sensitivity threshold) is higher than the perceived SNR. The above can be mathematically expressed as follows:

$$(5.33a) \qquad \hat{\theta} = \underset{\theta}{\arg\max} \quad \left\{ D_{\mathrm{i}}(d_{\mathrm{in}}, \theta_{\mathrm{i}}) \right\}$$

$$(5.33b) \qquad \text{subject to} \quad S_{\mathrm{max}} \geq \gamma_{\mathrm{i}}, \quad \forall i \in \{1, ..., N\}$$

$$(5.33c) \qquad \theta_{\mathrm{i}} > 0, \quad \forall i \in \{1, ..., N\}$$

where $\hat{\theta}$ is the adapted beamwidth, D_{i} is the data rate of the chosen MCS at the distance d_{in}, γ_{i} is the given SNR for the position I_{pos} and, finally, N is the number of required beam realignments.

5.6 Performance Evaluation

We proceed to examine two different scenarios which model a vehicle driving in an urban environment. For the first scenario, we will investigate the performance of our system for a vehicle moving in a straight line, and we will present some numerical results for the equations above in Sec 5.5.1. Later, we will investigate a larger scale urban scenario with a more significant number of vehicles driving in a Manhattan-Grid-like urban environment.
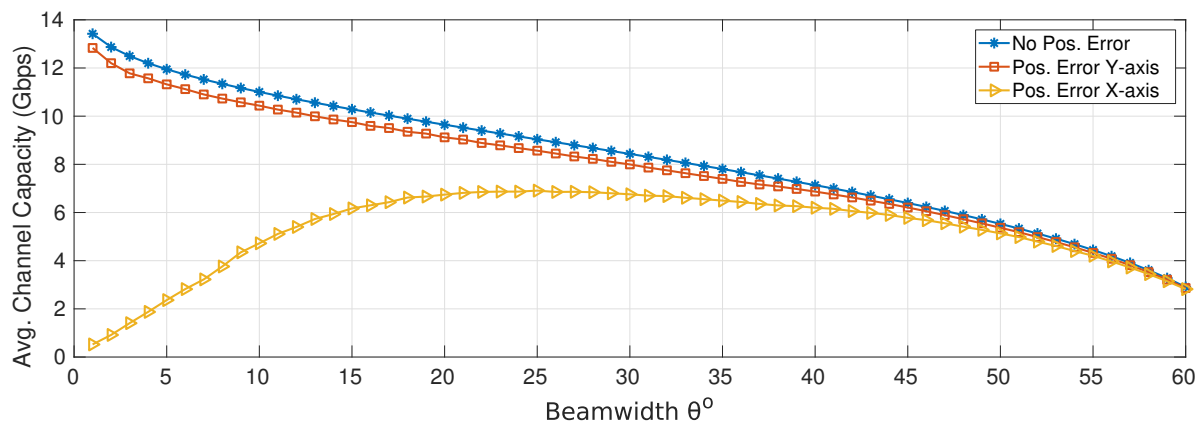
Figure 5.9: Differential sensitivity analysis for the affect of individual position error components
and different beamwidths. The average position error for this example was $3\,\mathrm{m}$.

5.6.1   Scenario 1: Simplified Approach - Straight Road

For the first scenario, we consider a vehicle that travels on a straight road following a linear
motion. An RSU is mounted at the side of the road. The RSU will try and align its beams with the
moving vehicle. At the closest point, the vehicle was ~$18\,\mathrm{m}$ apart from the RSU (the moment that
the vehicle moves perpendicular to the RSU). The distance travelled is one *road block* r_b and the
vehicle travels at a constant speed of $14\,\mathrm{m\,s^{-1}}$. All the simulation parameters are summarised in
Table 5.1.

At first, we present the results of the differential sensitivity analysis for the individual error
components with respect to the different beamwidths. As discussed in Secs. 5.4.5 and 5.5.1, we
described that the individual position error components affect the performance of our system
differently. The results can be seen in Fig. 5.9, were the two different components (error on the
x-axis and error on the y-axis) were considered, and the average achievable channel capacity
was evaluated and compared with the theoretical maximum value. For our scenario, when each
error takes random values, the other one is considered as constant. With respect to Sec. 5.4.5,
each error can take either positive or negative values as shown in Figs. 5.6(b) and 5.8 and it
has a *mean absolute distance* from the real position. Therefore, the averaged constant error
considered equals to zero for each case. For the random error, a mean value of $3\,\mathrm{m}$ was considered
for our demonstration. As it is observed in Fig. 5.9, even though both errors have the same
magnitude, the error on the x-axis influences more the system performance than the one on the
y-axis. Analysing an a priory known position error for a specific road and knowing the effect of
each component is very important for the initial planning of an ITS. With better design, will be
able to compensate with different kind of errors, achieving better performance and enhancing the
road safety.

Finally, Fig. 5.10 presents the average channel capacity and the optimum beamwidth for each
error. The results are compared with the ideal case, where the estimated position matches the
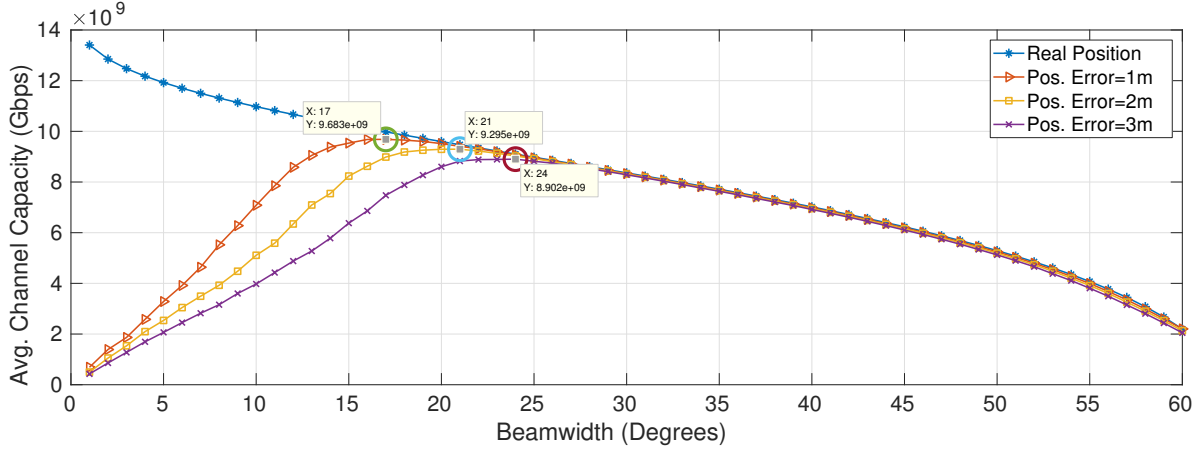
Figure 5.10: Optimum beamwidth values for different position errors compared to the maximum system capacity (given when using the real position).

Table 5.2: MCSs and SNR threshold used for the simulation results.

MCS	Modulation	Coding Rate	Data Rate	L_{margin} Threshold	Sensitivity
MCS1	$\pi/2$-BPSK	1/2	27.5 Mbps	8 dB	-78 dB
MCS2	$\pi/2$-BPSK	1/2	385 Mbps	18 dB	-68 dB
MCS3	SQPSK	5/8	866.25 Mbps	22 dB	-64 dB
MCS4	QPSK	1/2	1386 Mbps	23 dB	-63 dB
MCS5	16-QAM	3/4	4158 Mbps	32 dB	-54 dB
MCS6	64-QAM	5/8	5197.5 Mbps	35 dB	-51 dB
MCS7	64-QAM	13/16	6756.75 Mbps	39 dB	-47 dB

real one, i.e. zero-error exists. A mean error of 3 m was considered for this scenario. As shown, when $\theta°$ tends to zero, even the smallest error can cause a total misalignment - thus, degrading the performance. On the other hand, when $\theta°$ is large, the SNR is decreased leading to lower channel capacity. An optimum beamwidth exists that is different for each position error, which maximises the average channel capacity and is shown as a circle for each case. To that extent, the system can be fine-tuned for a given error to achieve the maximum performance.

5.6.2 Scenario 2: Large Scale Evaluation - Manhattan Grid

For our second scenario, we consider a larger scale urban scenario to test the performance of our algorithm. As shown in Fig. 5.2, IEEE 802.11ad performance degrades significantly as the number of vehicles is increased. SAMBA, as an algorithm with zero in-band overhead, is expected to outperform IEEE 802.11ad for a large number of vehicles. To that extent, SAMBA performance
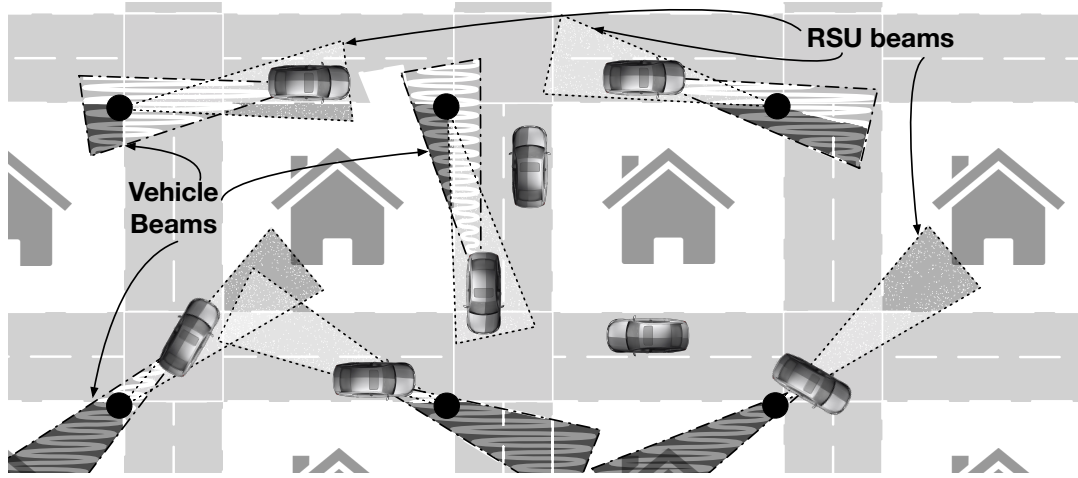
Figure 5.11: System level simulation scenario: Vehicles drive within a Manhattan Grid-like road network, and the system performance is evaluated. The darker beam area represents the coverage region blocked by the buildings.

will be evaluated under various scenarios (different number of vehicles, velocities and position errors) and will be compared with the legacy beamforming technique.

For our evaluation, we utilised a $200\,\text{m}\times200\,\text{m}$ sized Manhattan Grid road network, as in Fig. 5.11, consisting of five horizontal and vertical roads. Each road is $12.8\,\text{m}$ wide (4 lanes of $3.2\,\text{m}$ each) and has two lanes per direction. The RSUs are positioned at the top-right corner of each building block as shown in Fig. 5.11. Our design is a 2D implementation, so we assume that the RSUs are mounted at the same height as the vehicles but are always in LOS. The distance between two RSUs on the same road is ~48 m. The above RSU positions and dimensions were chosen to achieve 100% coverage of the road network (no blind spots) to evaluate our system under a controlled environment, i.e. the vehicles are always within coverage, and we avoid the blockages from the buildings. The motion of the vehicle is random as described in Sec. 5.4.1 and is limited within the road boundaries. A seamless handover was assumed between the RSUs. The vehicle position error is between 1-3 m, following a more conservative approach compared to the centimetre accuracy presented before (Sec. 5.4.1). RSUs positions are a priori known, so no position errors are introduced for them. Finally, when the beamwidth adaptation introduced in Sec. 5.5.2 was not considered, the beamwidth was set to 15°. Finally, the CAM Delivery Ratio (CAM-DR) is set to 1 (no loss). The rest of the simulation parameters can be found in Tab. 5.1.

At first, the average data rate of SAMBA compared to the legacy beamforming technique was evaluated, for a different number of vehicles and position errors. The data rate is calculated for this scenario as the averaged MCS data rate with respect to the achieved SNR (Table. 5.2) from all BIs. For SAMBA, two different scenarios were evaluated, i.e. with and without the beamwidth adaptation introduced in Sec. 5.5.2. As shown in Fig. 5.12, SAMBA can notably improve the system performance as it minimises the beamforming overhead. As expected, increasing the
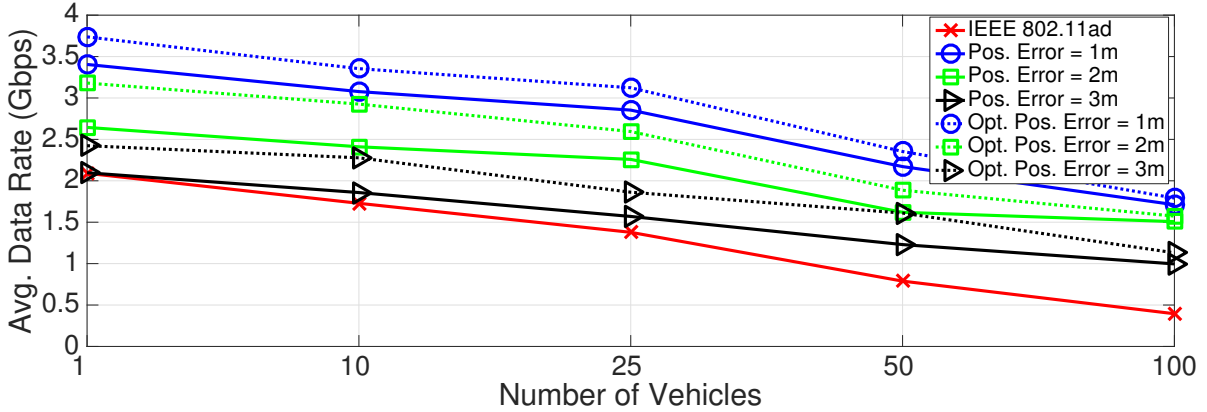
Figure 5.12: Average data rate per vehicle for a different number of vehicles and position errors. The average speed used for this scenario was $14\,\mathrm{m\,s^{-1}}$.

position accuracy improves the performance. For a position error of 3 m, without beamwidth adaptation and a very sparse network (≤ 10 vehicles), both techniques have similar performance. However, when the vehicle density is increased, the number of collisions during A-BFT interval is increased as well (as shown in Fig. 5.4), significantly degrading the performance of the legacy beamforming strategy. On the other hand, SAMBA can recompense with the increased density as the beam alignment is based on out-of-band feedback information.

As shown, the increased number of vehicles reduces the average data rate achieved per vehicle. Another intriguing observation can be seen in Fig. 5.13 where the overall network throughput performance is presented. It is observed that SAMBA significantly outperforms the IEEE 802.11ad legacy procedure. When the number of vehicles is increased (>50) it is shown that a bottleneck is created in the available resources of the network. As the collisions during A-BFT interval limit IEEE 802.11ad performance. For ultra-dense scenarios (e.g., 200 vehicles) IEEE 802.11ad is even slightly degrade. On the other hand, SAMBA can exploit the network resources a lot better achieving almost seven times more data exchange rate. This is very important within a CAV network as the sensory data exchanged are of paramount importance for the navigation and the routing of the vehicles.

In Figs. 5.12 and 5.13 we presented the results for different vehicular scenarios. Observing the results and taking the physiology of the beams and the behaviour of the vehicles into account (approach an RSU from a distance, pass by it and fend off again), we can conclude that a dynamic beam adaptation can enhance the data rate. Therefore, the beamwidth adaptation introduced before when used with SAMBA motion prediction algorithm manages to improve the performance even further. This is because vehicles are centred within the beams; our algorithm can later easier compensate with the increased position error (e.g., 3 m), maximising the SNR and consequently the data rate.

In Fig. 5.14, SAMBA is evaluated with respect to the DSRC CAM-DR. The feedback informa-

111
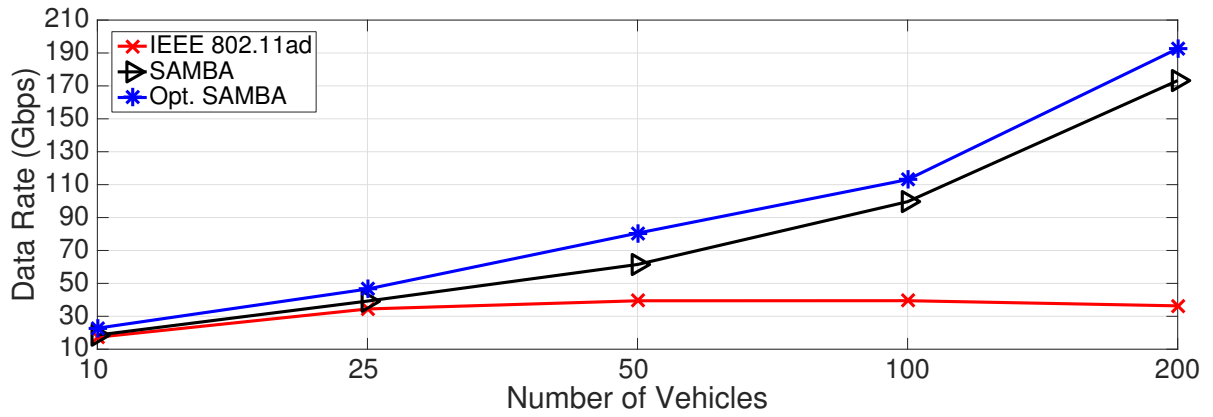
Figure 5.13: Throughput (evaluated under saturation conditions) for the entire network. Velocity is ~14 m s^{-1} and the position error for SAMBA is 3 m.
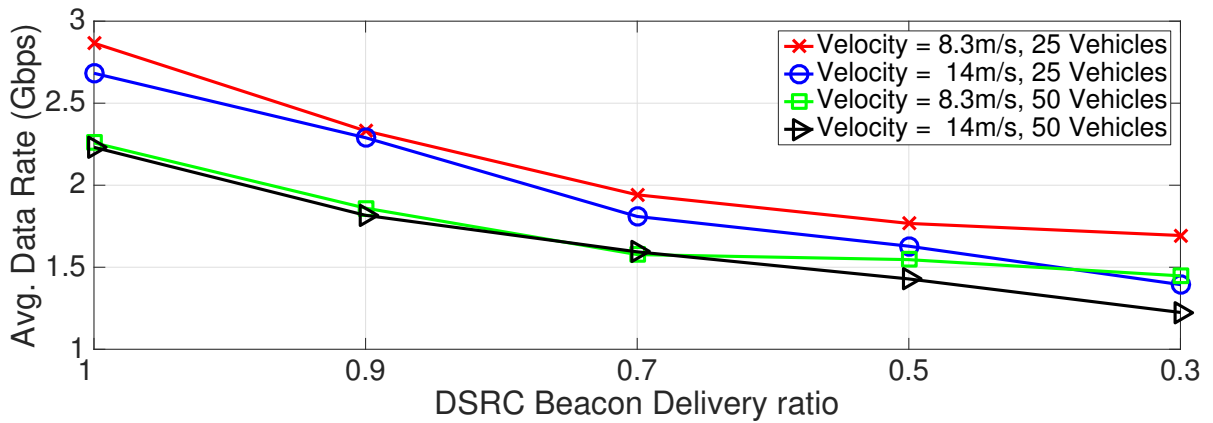
Figure 5.14: Average data rate per vehicle as a function of the beacon delivery ratio and the velocity. The position error was set to 1 m.

tion within the DSRC CAMs is the core of our system making it is obvious that the performance degrades when CAM-DR is decreased. However, even with significant CAM loss ($\geq 50\%$) (Fig. 5.14), SAMBA manages to achieve the required gigabit-per-second performance for future ITS services. This is because vehicles do not change their direction so often and so when observed macroscopically, they tend to move in a straight line making our algorithm tolerant to the microscopic beacon drop interval. Finally, when the vehicle density is increased, the performance tends to become worse. This is because the number of misalignments is increased and therefore more network resources are wasted. The macroscopic analysis that the vehicle moves in a straight line when observed for a longer period can be combined with the sensitivity analysis we introduced in Sec. 5.5.1. Better placement of the RSUs (e.g. instead of the corner of the buildings to be mounted on a traffic light above the road) can enhance more the performance. A combined scenario like the one mentioned before will be evaluated in future work.

Finally, evaluating SAMBA performance with respect to the velocity (Fig. 5.15), it was shown
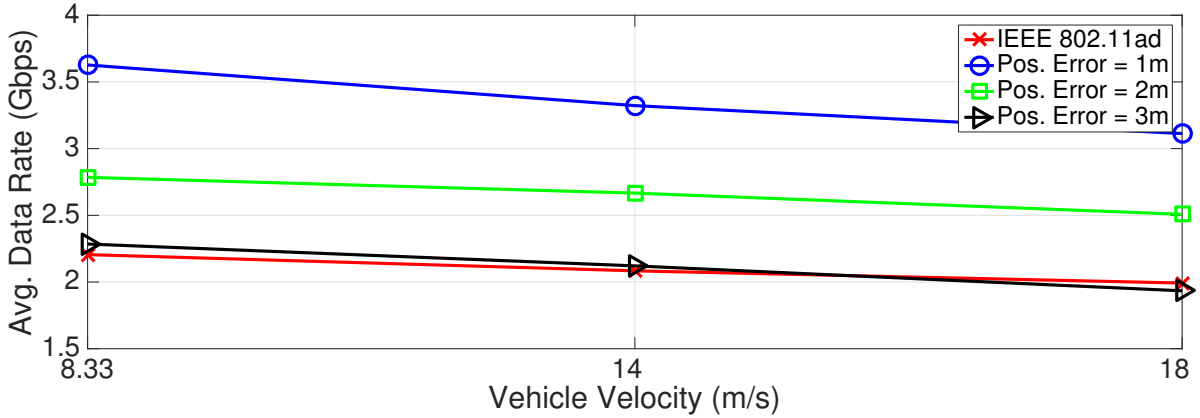
Figure 5.15: Achieved average data rate (one vehicle) for SAMBA algorithm in contrast with the legacy beamforming strategy for different velocities.

that even though the performance slightly decreases as the velocity is increased, SAMBA can compensate well with the increased mobility. Overall, SAMBA achieved improved system performance compared to IEEE 802.11ad. Particularly, reducing the position error will significantly improve the system performance. All in all, SAMBA was proven capable of replacing the legacy BF technique under urban vehicular scenarios.

5.7 Conclusions

In this chapter, we presented the first step in our work towards the design of heterogeneous intelligent beam steering framework that will be used for mmWaves links for future ITSs. The proposed strategy can achieve overhead-free beamforming exploiting the sensor information broadcast over DSRC links. Based on an agile motion prediction algorithm, we can predict the movement of the vehicle and adapt the antenna beams accordingly. The results showed that SAMBA outperforms the legacy beamforming technique with respect to the average data rate and overall network throughput performance while being tolerant to the position errors introduced. Introducing two different beamwidth optimisation algorithms, we managed to enhance the performance even further and provide an all-around model for different scenarios and conditions. As such, SAMBA is a viable solution for the mmWaves beamforming training of the next-generation ITSs. With the experience acquired in this chapter, we will proceed to the next chapter that we will introduce a V2V mmWaves association scheme.

6

EFFICIENT MAC-LAYER BEAM STEERING FOR V2V MMWAVES COMMUNICATIONS

6.1 Introduction and Contributions

6.1.1 Introduction

Chapter 6 presents a multi-link association scheme for mmWave V2V communications, able to utilise the channel better and prioritise the exchange of sensor data between different types of CAVs. In this chapter, we formulate a novel many-to-many link association scheme able to maximise the channel utilisation by taking into account the achievable data rate of mmWave channels, the different types of CAVs and their movement and their position on the road.

As mentioned before, CAVs are expected to generate up to 1 Gbps of sensor data [79]. Considering an urban scenario, where CAVs are in close proximity, IEEE 802.11ad mmWaves protocol can achieve up to 6.75 Gbps of data rate under LOS conditions [84]. Comparing this value with the previously estimated sensor data generation, it is evident that forming point-to-point links will lead to under-utilisation of the channel. What is more, the trajectory of a vehicle not executing emergency manoeuvres (e.g., avoiding a pedestrian), tends to follow a well-defined path [181]. However, a vehicle under emergency conditions will perform more complicated manoeuvres, like the obstacle avoidance and advanced trajectory planning [110]. The increased vehicle mobility and manoeuvrability imply that prioritised and faster access to more sensor data will be mandatory, due to its increased speed and size (e.g. in case of an emergency vehicle).

In this work, we will consider two different types of vehicles, i.e. *emergency CAVs* (e-CAVs) and *regular CAVs* (r-CAVs). The algorithm we present prioritises the sensor data towards the e-CAVs without though limiting the acquired information of the r-CAVs and dominating the

channel. We formulate our association policy as a distributed constraint optimisation problem, where the information required will be exchanged via DSRC links in a heterogeneous manner. To solve our problem, we use the Stable Fixtures (SF) matching theoretic game [118] and formulate a more generalised many-to-many link association scheme that forms bidirectional links taking into account the unique characteristics of each CAV.

This chapter is organised as follows. In Sec. 6.2 we present our system model. We will describe our matching game, introducing the requirements for our algorithm and the concept of the matching capacity with respect to the association policy. Sec. 6.3 describes our problem formulation, the distributed V2V SF matching game, broken down into two phases, and the definitions and criteria for a stable matching. In Sec. 6.4, we describe our simulation setup and present our performance investigation based on the algorithm mentioned above. Finally, this chapter is concluded in Sec. 6.5 where we summarise our findings and the validity of our algorithm.

6.1.2 Contributions

The contribution of the author for Chapter 6 is the following:

- Derived the equations for the matching capacity.

- Identified the requirements for the proposed heterogeneous system and the different variables that constitute the utility function for the system evaluation.

- Derived the equations for the problem formulation, and the maximisation problem introduced.

- Designed a two-phase algorithm based on the framework of Matching Theory that can solve the above-mentioned problem, explaining the different definitions for the matching pairs and their stability.

- Introduced the stability criteria for the Matching game and V2V scenarios.

- Designed and implemented a system level simulator in Matlab, that takes as an input the routes of the vehicles for a given urban scenario and the variables initialised by the user and solves the formulated problem for different scenarios.

- Presented the performance of the algorithm above for a number of different scenarios, using the introduced simulator, and critically discussed the results.

The contents of this chapter were presented at the conference:

- IEEE International Conference on Communications (ICC) 2018 [38].

6.2 System Model

Consider an urban vehicular scenario where mmWaves links are established between moving vehicles in a V2V manner. Let $\mathcal{V} \triangleq \{1,\dots,V\}$ denote the set of CAVs in the system. As described above, CAVs are of two different types: e-CAVs, denoted as $\mathcal{E} \triangleq \{1,\dots,E\}$ and r-CAVs, denoted as $\mathcal{R} \triangleq \{1,\dots,R\}$ with $\mathcal{E},\mathcal{R} \subseteq \mathcal{V}$, $\mathcal{E} \cup \mathcal{R} = \mathcal{V}$, and $\mathcal{E} \cap \mathcal{R} = \emptyset$.

Links are half-duplex, interconnecting two CAVs for a timeslot with duration T_s. Let A_j be the number of CAVs linked with CAV i within a T_s. The formed links are the pairs $\sum_{i \in \mathcal{V}} \left[(i,j) : j \in A_j \right]$. We assume that a scheduling algorithm shares the available time between all the pairs in equal transmission slots $\mathcal{T}$ with a duration of $\{\mathcal{T}_t : T_s \mod \mathcal{T}_t = 0\}$.

6.2.1 MmWave Link Budget analysis

Our link budget equations were previously defined in Sec 5.4.4. However, the analysis that will be presented in Chapter 6 requires a different notation, as we need to introduce the concept of time and the set of vehicles (i,j) that form a link. As discussed before, IEEE 802.11ad standard [174] defines a sensitivity threshold K_{MCS} for each MCS. The different thresholds can be found in Table 5.2. For a link (i,j) between two CAVs at time $t \in \mathcal{T}_t$ and a given SINR, an appropriate MCS can be chosen based on $SINR_{i,j} \geq K_{MCS}$. The chosen MCS indicates the maximum achievable throughput $r_{i,j}^{\max}(t)$ for this communication link.

The $\text{SINR}_{i,j}$ is time dependent and is defined as the ratio between the *received power* $P_{i,j}^{\text{rx}}$ over the *noise power* P_{n} plus the *interference* $\mathcal{I}(t) = \sum_{k \in \mathcal{V}, k \neq j} I_{k,j}(t)$, i.e.:

$$(6.1) \qquad \text{SINR}_{i,j}(t) = \frac{P_{i,j}^{\text{rx}}(t)}{P_{\text{n}} + \mathcal{I}(t)}$$

From the above equation, we define $P_{i,j}^{\text{rx}}$ as [177]:

$$(6.2) \qquad P_{i,j}^{\text{rx}}(t) = P^{\text{tx}} + G_{i,j}^{\text{rx}}(t) + G_{i,j}^{\text{tx}}(t) - PL_{i,j}(t)$$

where P^{tx} is the transmission power, $G_{i,j}^{\text{tx}}$ and $G_{i,j}^{\text{rx}}$ are the TX and RX antenna gains, and PL is the *path-loss component*.

The ideal antenna beams are modelled with a gain defined as $G_{i,j}^{\alpha}$, with $\alpha \in \{tx, rx\}$, and zero sidelobes. $G_{i,j}^{\alpha}$ is a function of the half-power beamwidth θ ($-3\,\text{dB}$), is equal for both polarisation planes and is given as [170]:

$$(6.3) \qquad G_{i,j}^{\alpha}(t) \simeq 4\pi/\theta_{i,j}^{2}(t), \ \text{ for } \left| \vartheta_{i,j}(t) \right| \leq \theta_{i,j}(t)/2$$

where $\vartheta_{i,j}(t)$ is the misplacement error that leads to misalignments. In this work, we assumed perfect beam alignment all the time. The PL is calculated as:

$$(6.4) \qquad PL_{i,j}(t) = 10\,n \log_{10} d_{i,j}(t) + C_{i,j}^{\text{att}}(t) + S_{\text{f}}(t)$$

117
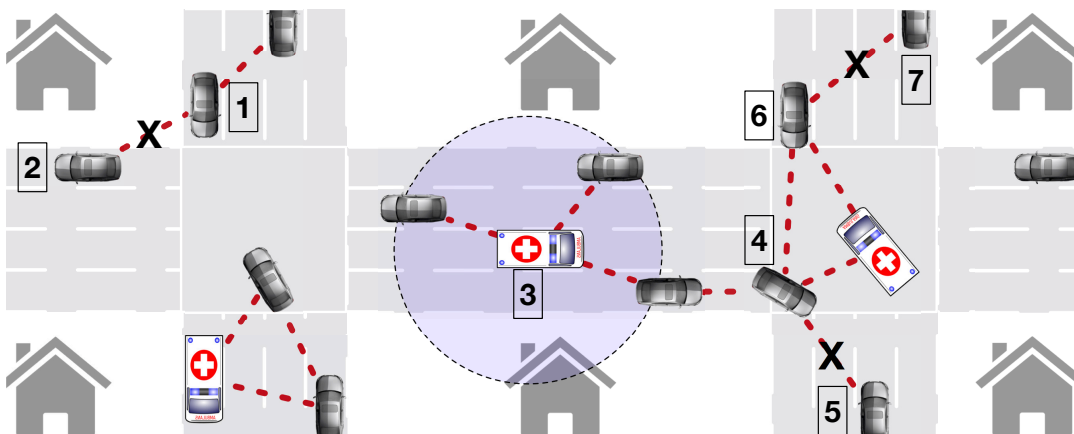
Figure 6.1: A snapshot of our system at time t, generated from the SF matching game using all the utility functions $\mathcal{U}$. Links are formed (dotted lines) or not (lines crossed by an "X") according to our maximisation problem. The capacity for this example was equal to 3.

where n is the path-loss exponent and $d_{i,j}$ is the distance between vehicles i and j. $C_{i,j}^{\text{att}}$ is the channel attenuation with regard to the distance $d_{i,j}$ at time t, given by the rain and atmospheric attenuation and the channel attenuation factor H^{att} for a mmWave LOS link at 60 GHz in urban environments [177], i.e. $C_{i,j}^{\text{att}}(t) = 40 d_{i,j}(t)/1000 + H^{\text{att}}$. Finally, S_{f} represents the shadow fading of the channel following a log-Normal distribution $\log(S_{\text{f}}) \sim \mathcal{N}(0, \sigma_{S_f}^2)$ with $\sigma = 5.8$ [178]. Finally, the noise term P_{n} can be calculated as in (5.16).

With remark to the link budget analysis mentioned above, the average data rate of the vehicle i for all the links $\sum_{i \in \mathcal{V}} \left[(i,j) : j \in A_j \right]$ is given in (6.5). As in Chapter 5, we take into account the sensitivity threshold and find the maximum theoretical data rate from the chosen MCS. That is the theoretical data rate achieved at the physical layer of IEEE 802.11ad. The equation describing the above is the following:

$$(6.5) \qquad C_i(t) = \frac{1}{\mathcal{T}} \sum_{j \in A_j} r_{i,j}^{\max}(t)$$

6.2.2 Matching Capacity for the Association Policy

In this system a CAV i can be potentially linked with all CAVs at its 360° azimuth plain. The connections can be established under some specific conditions that will be described later in Sec. 6.2.3. At time t, CAV i is associated with A_j vehicles, of which A_e are e-CAVs and A_r are r-CAVs with $A_r, A_e \in \mathbb{N}^*, A_r, A_e \subseteq \mathcal{V}$ and $A_j = A_r + A_e$.

A_j can be affected by the number of vehicles in LOS. In fact, the high radio frequency of mmWaves significantly affects the received signal strength, and even small obstacles (e.g. another CAV) can behave as impenetrable blockages. To that extent, our formulated problem considers as potential CAVs for the association all the vehicles that are in LOS with our target i. For example, in Fig. 6.1 CAV no. 1 cannot be paired with CAV no. 2, as their link, is blocked by a building.

The matching capacity $c_i \in \mathbb{N}^*$ is defined as the maximum number of CAVs that i can be associated with. Two factors limit the capacity. Firstly, the maximum number of links that each CAV can establish (predefined at the beginning of each scenario). Secondly, is capped with regard to the exchanged data within a timeslot, i.e. $C_i(t) \leq r_{i,j}^{max}(t)$, $\forall i,j \in \mathcal{V}$. CAV no. 4, for example, cannot be linked with CAV no. 5 (Fig. 6.1) as its matching capacity was reached (connected to three CAVs with higher priority).

6.2.3 Requirements for the System Evaluation

Cooperative manoeuvring on the road can be achieved by sharing the generated sensor data with the surrounding CAVs. The amount of sensor data generated depends on the resolution and the number of equipped sensors, the surrounding environment and the preprocessing that might take place before the transmission and is different for each vehicle. This amount can be quantified as $g \in \{1, \ldots, G\}$, where g is measured in data transfer rate units.

The main idea behind our systems is that a CAV, evaluating the importance and utility of a nearby vehicle can decide whether it wants to connect with that or not. This potentially formed link (i,j), can be quantified with a utility function $u_{i,j}(\cdot) \in [0,1]$ and is based on the opposing vehicles topologies and movement, the generated sensor information, and finally, their types of vehicle.

Our system operates in a heterogeneous manner (DSRC/mmWaves), and the information required for the utility function are encapsulated within the DSRC CAM messages (as in [37]). Observing macroscopically our system, we can say that within a timeslot T_{s_i}, the generated data g will not be very different compared to $T_{s_{i-1}}$. Therefore, the generated data g is estimated from each CAV for the next scheduling period and accompanied with t the information mentioned above, before being broadcast on the network. Using this heterogeneous architecture, we leverage from the increased coverage of the DSRC technology ensuring that vehicles with close proximity will receive the same information. Also, the reduced signal loss minimises the error rate in the CAM reception at close distances. When the information from the surrounding vehicles is received, all CAVs can independently run their SF matching game. The utilities for all CAVs are represented as $\mathcal{U} = \{u_{i,j} | i,j \in \mathcal{V}\}$. The variables constituting the above utility function are the following:

6.2.3.1 Type of Vehicle

The first variable considered is the type of vehicles denoted as τ_i. As discussed in Sec. 6.1.1, it is essential to prioritise links with vehicles with increased mobility performing more complex manoeuvres. Therefore, for τ_i we have $\tau_i \in \{e-CAV_i, r-CAV_i : e-CAV_i = 1, r-CAV_i = 0.5\}$. Therefore, our model prefers the emergency vehicles compared to the regular ones as shown.

6.2.3.2 Regional Sensor Data Generated

The density q_i is defined as the number of CAVs $q \in \{1,\dots,K\}$ with $K \subseteq \mathcal{V}$ within a circular radius $R \in \mathbb{N}^*$. q_i is affected by the urban topology (e.g., buildings cover a large surface area that vehicles cannot use) and the road characteristics (e.g., more road lanes imply that more CAVs can be positioned within R). In Fig. 6.1, the dotted circle centred around CAV no. 3, represents its sensor region R. Targeting a road section with increased density implies that we can achieve a better link utilisation and access more sensor data. Also, using a smaller radius, implies that our target CAV will be associated with vehicles within close proximity. This will enhance the perceived SINR in our system. The regional data generated for CAV i can be given as $Q_i = g_i + \sum_{j=1,\dots,q_i} g_j$. This value is later normalized with respect to the maximum regional data generated per timeslot, i.e. $Q_i^*(t) = \frac{Q_i(t)}{\max(Q(t))} \in [0,1]$.

6.2.3.3 Vehicle Direction and Distance

We assume that a CAV is more interested in sensor data coming from CAVs with coinciding routes. This is important because vehicles can collect more sensor information for an unknown road section. However, the utility of the information degrades with respect to the distance. To that extent, in order to take into account both variables we use the normalized weighted arithmetic mean as below:

$$(6.6) \qquad e_{i,j} = w_1 \frac{R - d_{i,j}}{R} + w_2 \frac{\pi - \phi_n^{i,j}}{\pi} \quad \in [0,1]$$

where $\phi_n^{i,j}$ is the normalized angle difference between the two vehicle directions and w_1 and w_2 are the weight factors. w_1 and w_2 are calculated as: $w_1 = \frac{R - d_{i,j}/R}{R - d_{i,j}/R + \pi - \phi_n^{i,j}/\pi}$ and $w_2 = \frac{\pi - \phi_n^{i,j}/\pi}{R - d_{i,j}/R + \pi - \phi_n^{i,j}/\pi}$. A value ≈ 1 determines a coinciding movement and very close distance between two vehicles.

For all the above quantifying parameters, a utility function can be written as $u_{i,j}(\{\tau_j, Q_j, e_{i,j}\}) \in [0,1]$, evaluating the link (i,j) between two vehicles. More specifically, the more valuable a CAV is, the more important is the connection with that. The utility function is given as:

$$(6.7) \qquad u_{i,j}(t) = \tau_j Q_j(t) e_{i,j}(t) \frac{r_{i,j}(t)}{r_{i,j}^{\max}(t)} \quad \in [0,1]$$

6.3 Matching Game and Problem Formulation

6.3.1 Problem Definition

The communication links that can potentially be established between all vehicles at time $t \in T_s$ are defined with a matrix $\mathcal{M}(t) = (m_{i,j}) \in \mathbb{R}^{x \times y}$, $x,y \leq \mathcal{V}$. The binary value $m_{i,j}$ represents the existence of a link between vehicles i and j, having the following values:

$$(6.8) \qquad m_{i,j}(t) = \begin{cases} 1, & \text{if link } (i,j) \text{ is established for } (t, t + \mathcal{T}_t] \\ 0, & \text{otherwise} \end{cases}$$

Each CAV can be simultaneously matched with less or equal than $c_j \in \mathbb{N}^*$ vehicles, being either e-CAVs or r-CAVs. This value depends on the matching capacity of each vehicle as described in Sec. 6.2.2 and 6.2.3.2. Let $\mathcal{V}_j \subseteq \mathcal{V}$ be the number of CAVs paired with i at time t. The previously introduced utility function should be redefined as:

$$(6.9) \qquad \mathcal{X}_i(\mathcal{M}(t)) \equiv \mathcal{X}_i \left(\{\tau_j, Q_j, e_{i,j}\}_{j \in \mathcal{V}_j(t)} \right) \in [0,1]$$

representing the utility functions of all the vehicles that can be potentially paired with i. The problem can be formulated as the maximisation of (6.9) in order to find the best links between all CAVs at time t:

$$(6.10a) \qquad \underset{\mathcal{M}(t)}{\text{Maximise}} \sum_{i \in \mathcal{V}_i(t)} \mathcal{X}_i(\mathcal{M}(t))$$

$$(6.10b) \qquad \text{subject to: } \sum_{j \in \mathcal{V}_j(t)} m_{i,j}(t) \le c_j \qquad\qquad , \forall j \in \mathcal{V}(t)$$

$$(6.10c) \qquad j \text{ in LOS with } i \qquad\qquad , \forall j \in \mathcal{V}_j(t)$$

$$(6.10d) \qquad j \text{ within } R \qquad\qquad , \forall j \in \mathcal{V}_j(t)$$

$$(6.10e) \qquad m_{i,j}(t) \in (0,1] \qquad\qquad , \forall i,j \in \mathcal{V}_i(t) \times \mathcal{V}_j(t)$$

$$(6.10f) \qquad m_{i,j}(t) = 0 \qquad\qquad , \forall i = j$$

where the above constraints are the number of CAVs that can be associated with the target i and the range of values for the utility function in (6.9).

6.3.2 Distributed Stable Fixtures (SF) Matching Game

The problem formulated in (6.10) is a multigraph problem that is at least NP-complete. Our objective was to develop a mechanism able to run independently on each CAV in a real-time manner and find the best available links with respect to the CAVs utilities. To solve the formulated problem using a reasonable amount of resources we utilised an SF matching game that has a complexity of $O(m)$ [118]. Using this framework, we developed a strategy able to run independently on each CAV and establish the V2V links with the highest utility until the capacity is reached. In the next sections, we will describe the basic definitions of the SF game with respect to our problem formulation in Sec. 6.3.1 and the two-phase operation required for the link establishment.

6.3.3 Definitions for the Distributed SF Matching Game

A matching game is defined by a set of vehicles $\mathcal{V}(t) = \{v_1, v_2, \ldots, v_n\}$ where n is the number of all CAVs. We denote $\sum_{i,j \in \mathcal{V}(t)} (i,j)$ as the number of potential links that will be evaluated within this matching game. Also, we denote m as a matching pair. We assume a strict preference relation for CAV i between vehicles (j, j') with $j \ne j'$ and denote it by $\succsim_i$ that is complete, reflexive and

transitive. For the two potential matches $\mathcal{M}(t)$ and $\mathcal{M}'(t)$ we have $m_i(t) = j$ and $m'_i(t) = j'$. The above implies that:

$$(6.11) \qquad (j, \mathcal{M}(t)) \succsim_i (j', \mathcal{M}'(t)) \iff u_{i,j}(t) > u_{i,j'}(t)$$

meaning that i prefers j over j' due to its higher utility. Intuitively, a stable matching refers to a solution where no players have the intention to change their preferences (switch preference for a higher prioritised CAV). With regard to the links formed, we present two definitions that will result in a pairwise and groupwise stability in our system.

Definition 1: *A link (i,j) is pairwise stable if and only if both i and j decide to associate with each other.*

This implies that there is a mutual approval from both vehicles and there are no *blocking pairs*. For example, for CAVs $i, j, k \in \mathcal{V}(t)$ and $i \neq j \neq k$, we have $\big(u_{i,j}(t) > u_{i,k}(t)\big)$ and $\big(u_{j,i}(t) > u_{j,k}(t)\big), \forall k \in \mathcal{V}(t)$. Comparing the utilities and using (6.11), we have that $\succsim_i$ and $\succsim_j$ are complete and reflexive. Finally, as this also works for $\forall k \in \mathcal{V}(t)$, the preference relations are transitive as well.

Definition 2: *A link (i,j) is group-wise stable if and only if it is not blocked by any coalition of vehicles $S \subseteq \mathcal{V}$.*

For a many-to-many matching game, the stability concept should focus not only on the blocking pairs between vehicles but also on the blocking groups involved in the matching game. If we consider a sequence of pairwise stable links $m_i = \{m_1, m_2, \ldots, m_n\}, n \in \mathcal{V}$ that form a cluster of vehicles with pairwise stability, there should be no other vehicle or cluster of vehicles in the network that will block any of the above links. The mathematical framework behind the above definitions and a more in-depth can be found in [182].

If all the links are pairwise and group-wise stable at time t, we can ensure that SF will agree on a stable match. There is always a chance that a link does not meet these requirements. For example, in Fig. 6.1, CAVs no. 6 and 7 formed a link at first. However, during the execution of the algorithm, it was discovered that CAV no. 6 prefers to be linked with two other vehicles. In that case, the rejected link is removed from the matching game as it will be described in the next section.

6.3.4   Stability Criteria of SF Matching Game for V2V links

The general idea of the SF matching game is that each participating CAV starts with a fixed capacity (defined as the matching capacity c_i before) and builds its preference list P of potential CAVs to be matched with. We denote as $d_i = \min(c_i, |P_i|)$ the *degree* for each CAV i where $|P_i|$ is the length of its preference list. We should note that the *degree* of a player is a property of the problem instance; it does not change its value in the course of the algorithm. Each P_i is classified as *short* ($|P_i| < d_i$) or *long* ($|P_i| > d_i$). SF executes in two phases, reducing the length of P during each phase and concluding with the final matches. This is subject to two stability criteria:

1. $\sum_i d_i$ is not odd. This sum is double the size of a stable matching as it counts every matched pair exactly twice, so if it is odd will imply that there is no pairwise and groupwise stability (as discussed in Sec. 6.3.3).

2. None of the preference lists P_i becomes short during the execution of SF algorithm, i.e. $|P_i| \not< d_i, \forall i \in \mathcal{V}$. If any list is short, then this implies the elimination rotation was unsuccessful (more details about that will be given later) thus the algorithm reports the instance as unsolvable.

In both cases, i.e. $\sum_i d_i$ becomes odd or $\exists i . |P_i| < d_i$, $i \in \mathcal{V}$, the instance is reported as unsolvable.

6.3.5 The two Phases of the SF Matching Game

Right now, we will describe the two phases of the SF matching algorithm. Algorithm 4 presents these two phases with a pseudocode format as well.

Phase 1: During this phase, a sequence of *bids* takes place from one vehicle to the others. These bids are used to construct a set S that is an initial list of the potential matching pairs as well as to identify and delete pairs that cannot belong in a stable matching. For the pair (v_i, v_j), v_j is a *target* for the v_i, and v_i is a *bidder* for the v_j. The target set for v_i is denoted as $A_i = \{v_j : (v_i, v_j) \in S\}$ and the bidder set as $B_i = \{v_j : (v_j, v_i) \in S\}$. We also denote as $a_i = |A_i|$ and $b_i = |B_i|$ the length of each set. Each v_i, bids for its most favourable target and adds the pair (v_j, v_i) in S (line 3). Keep in mind that the pairs in S are ordered so (v_i, v_j) is different than (v_j, v_i). Then v_j checks whether it has exceeded its B_j capacity, i.e. $b_j \geq c_j$ (line 4). If so, it deletes all vehicles within P_j that are worse than the c_jth rank (lines 5-11). The bidding continues as long as $a_i < \min(c_i, |P_i|), \forall v_i \in \mathcal{V}$. The outcome of Phase 1 is a reduced P_i for each v_i and an increased set S of potential matching pairs.

Phase 2: The two stability criteria defined in Sec. 6.3.4 are checked throughout the execution. If any becomes true, then the instance is reported as unsolvable (lines 14 and 20). During this phase, we search for possible *rotations* that can further reduce the length of P_i lists. As a rotation is defined a sequence of ordered pairs $\rho = ((v_{i_0}, v_{j_0}), (v_{i_1}, v_{j_1}), \ldots, (v_{i_{r-1}}, v_{j_{r-1}}))$. To find ρ, we start with a CAV having a long P. For each $k \in [0, r-1]$ we denote $v_{i_k} = v_{l(j_k)}$ and $v_{j_{k+1}} = v_{f(i_k)}$). This means that v_{i_k} is v_{j_k}'s worst bidder, $v_{j_{k+1}}$ is v_{i_k}'s next target and the pair (v_{i_k}, v_{j_k}) is in ρ. The process stops when a CAV has been visited twice. For each (v_i, v_j) within ρ, this pair is removed (line 19) from S and is replaced with the next favourable pair for v_i, found from its P_i list (line 23). The rotation process stops when a preference list becomes short, i.e., there is no stable matching, or when no further possible rotations exist, i.e. a stable matching was found. More details about the SF matching algorithm, such as the mathematical proofs and lemmas, as well as the complexity analysis of the algorithm can be found in [118].

Algorithm 4 SF Matching Game Algorithm

Phase 1:
1: **while** $a_i < \min\left(c_i, |P_i|\right)$ **do**   ▷ $\forall i \in \mathcal{V}$
2:     Take v_j, the first vehicle in P_i, but not in A_i
3:     $S = S \cup \left\{(v_i, v_j)\right\}$ ▷ Starting with S as an empty set
4:     **if** $b_j \geq c_j$ **then**
5:         Find v_k ▷ c_jth ranked bidder for v_j
6:         **repeat** for each preferred vehicle v_l over v_k in P_j
7:             **if** $(v_l, v_j) \in S$ **then**
8:                 $S = S \setminus \left\{(v_l, v_j)\right\}$.
9:                 Remove pair (v_l, v_j) from P
10:             **end if**
11:         **until** All v_l removed from P_j
12:     **end if**
13: **end while**
Phase 2:
14: **if** $\sum_i d_i$ is odd **then**
15:     No stable matching exists
16: **else**
17:     **while** $|P_i| \not< d_i$ **do**   ▷ $\forall i \in \mathcal{V}$
18:         Find a possible rotation ρ pair (v_i, v_j) ▷ Start with a long P_i
19:         For ρ, remove v_i and v_j from preference lists
20:         **if** any P_i becomes short **then**   ▷ i.e. $|P_i| < d_i$
21:             No stable matching exists
22:         **else**
23:             $S = S \setminus \left\{(v_i, v_j)\right\}$
24:         **end if**
25:     **end while**
26:     Return $\mathcal{M} = S$ ▷ Stable Matching $\mathcal{M}$
27: **end if**

6.4 Performance Evaluation

In the previous section, we presented our model for our multi-link association scheme for V2V mmWave communications. In this section, we will evaluate our system performance with respect to the average data exchanged and the link utilisation. For our evaluation framework, we designed and developed a system level simulator in Matlab, that takes into account the vehicle traces and the maps generated from SUMO traffic generator [169]. The effects of the coverage radius as well as the beamwidth were investigated to have a more broad idea of the performance of our system. More details about the simulation framework and the results can be found in the next sections.

6.4.1 Simulation Framework

As a road network, we used a Manhattan Grid-like map ($100\,\text{m} \times 100\,\text{m}$) consisting of three horizontal and perpendicular roads with zero traffic lights. The reason that we chose this road networks is the increased probability of traffic jams generated from the unsupervised four-way intersections. Each road consists of four-lanes (two lanes per direction), and each lane is $3.2\,\text{m}$-wide. Using the SUMO traffic generator [169], we generated realistic vehicle traces for both the e-CAVs and r-CAVs. e-CAVs were generated with a probability of 15%. A vehicle being of type

Figure 6.2: CDF for the link utilisation for all CAVs, and for different matching capacities and beamwidths. The radius is equal to $R = 20\,\text{m}$.

e-CAV implies that will move twice as fast compared to an r-CAV, will be impatient, i.e., will make manoeuvres to overtake other vehicles and will be willing to impede other vehicles with higher priority on the road (e.g., crossing an intersection without using the right-of-way rule). For our channel model [177], we used the same simulation parameters as in Chapter 5. A summary of all the parameters can be found in Table 5.1 firstly introduced in Chapter 5.

Each r-CAV randomly selected the amount of sensor data it generates from $g \in \{0.25, 0.5, 0.75, 1\}$ (in Gbps per timeslot). All e-CAVs constantly generated $1\,\text{Gbps}$ of sensor data. The different radius used are $R \in \{20\,\text{m}, 30\,\text{m}, 40\,\text{m}\}$. Two different beamwidths were used ($\theta = 5°$ and $\theta = 15°$), to investigate the effect of the different antenna beamwidths on our system. Finally, 20 CAVs were used for our scenario. This density was carefully chosen in order to generate a small traffic jam within our network in order to exploit the increased number of links that each CAV can form.

6.4.2 Simulation Results

At first, we evaluated the link utilisation (per timeslot) for our system, defined as the amount of data exchanged, over the maximum data rate $r^{\max}$. The results can be found in Fig. 6.2. The radius used was $R = 20\,\text{m}$ and different results are presented for the two different beamwidths. As shown, increasing the capacity, significantly increases the link utilisation as well. This is because a CAV, being able to form multiple links, tends to better utilise its channel compared to the baseline case where $c_i = 1$, i.e., only point-to-point links are formed. As discussed before this is due to the estimation that vehicles will not always transmit the maximum amount of data required to saturate the channel. Observing the different beamwidth results, we can see that there no significant difference, having the case of $\theta = 15°$ being slightly better compared to the $\theta = 5°$ scenario.

However, using a bigger radius ($R = 40\,\text{m}$), the results significantly change as shown in
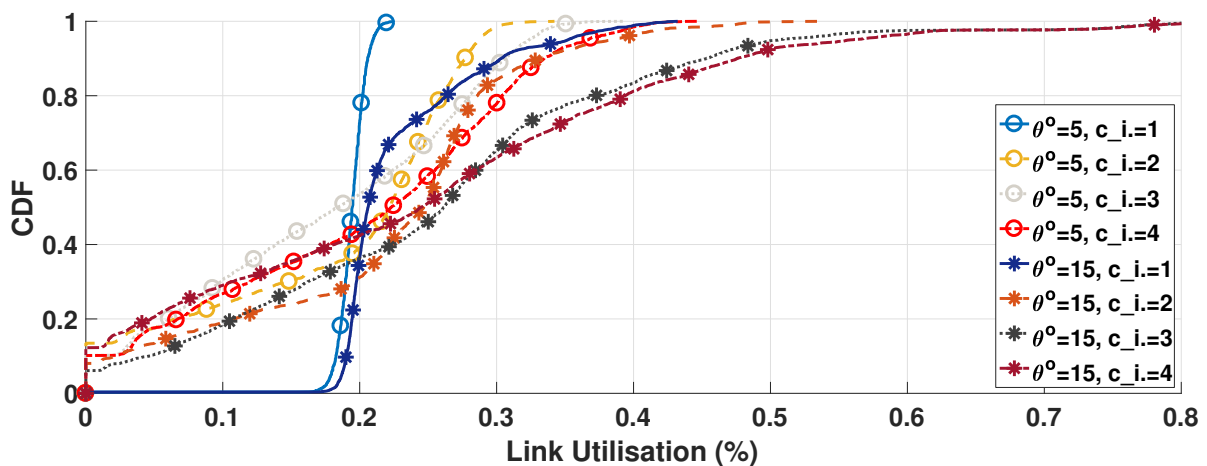
Figure 6.3: CDF for the link utilisation for all CAVs, and for different matching capacities and beamwidths. The radius is equal to $R = 40\,\mathrm{m}$.

Fig. 6.3. Again, we observe that the link utilisation increases as the matching capacity is increased. However, we notice that there is a big difference comparing the results for the two beamwidths. Using a narrow beamwidth, we increase the antenna gain and consequently our data rate. However, a greater data rate implies that this factor will dominate our utility function. So a CAV will prefer surrounding vehicles providing increased channel data rate, i.e., the closest ones, as the SINR is a function of the distance, and not the ones with the more sensor data to offer.

On the other hand, a wider beamwidth increases the link utilisation compared to the narrower case. A wider beamwidth implies that the increased data rate will not dominate our model and vehicles with more sensor information available will compete for the association. This increased offer of data increases the utilisation of our channel and improves our system performance. Taking into account the observations of Chapter 5, we can say that a wider beamwidth will make our system more tolerable to beam misalignment problems. This is very important, especially for e-CAVs where beam misalignments might lead to lost or delayed sensor information thus, delays in their manoeuvres. We showed that the optimum values for the beamwidth are between $\theta \in [17, 21, 24]$ (depending on the different errors) so the beamwidth we used was very close to these values.

Comparing Figs. 6.2 and 6.3, it is worth noting that increasing R, the dissimilarity of the link utilisation increases as well. This is because the increased number of vehicles within R leads to more blocking pairs and less stable matches. For $R = 30\,\mathrm{m}$, the results are similar to the above cases lying between the two cases mentioned earlier. These results can be seen in Fig. 6.4.

Fig. 6.5 presents the average data exchanged (sent and received) for each CAV within a timeslot. As expected, increasing the matching capacity will improve our system performance compared to the baseline point-to-point links. Later, comparing the exchanged data between the
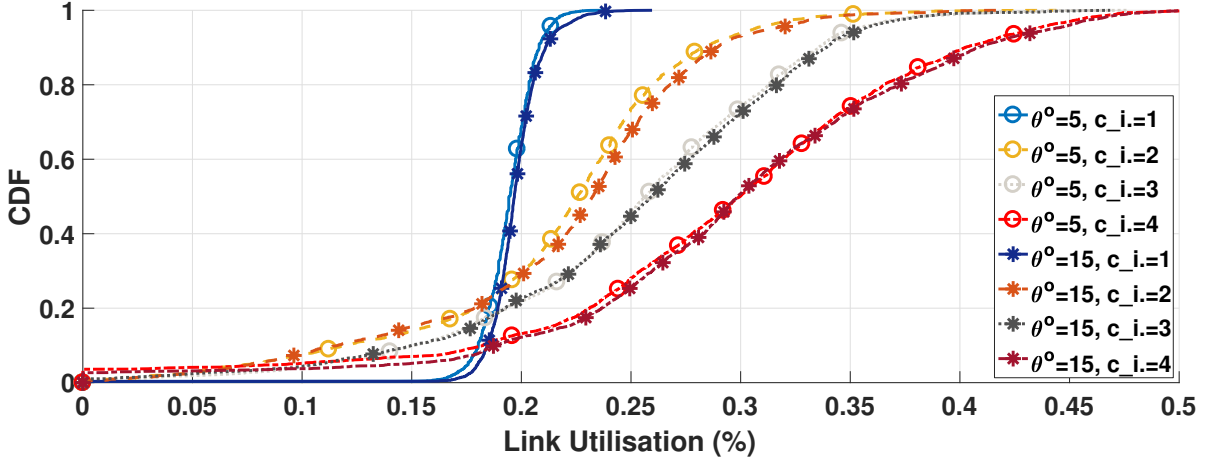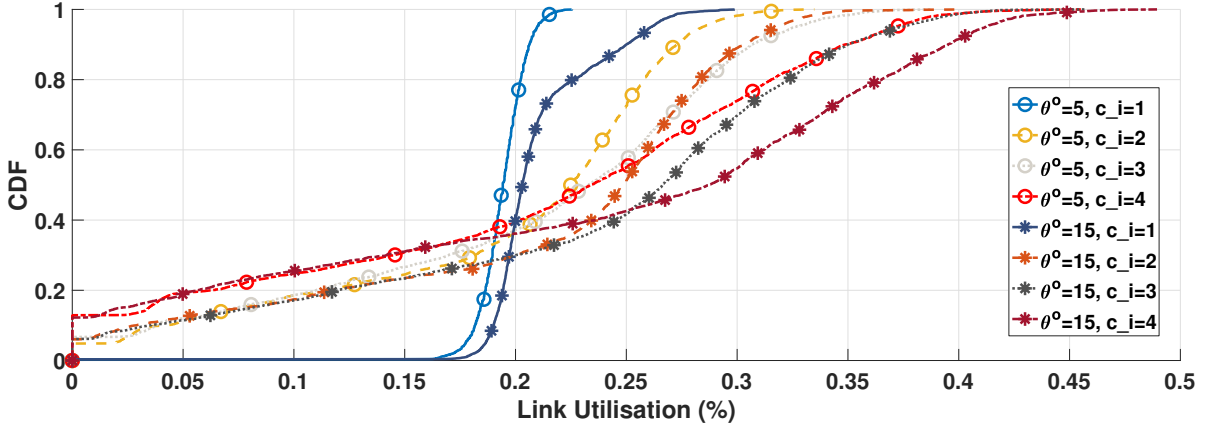
Figure 6.4: CDF for the link utilisation for all CAVs, and for different matching capacities and beamwidths. The radius is equal to $R = 30\,\mathrm{m}$.
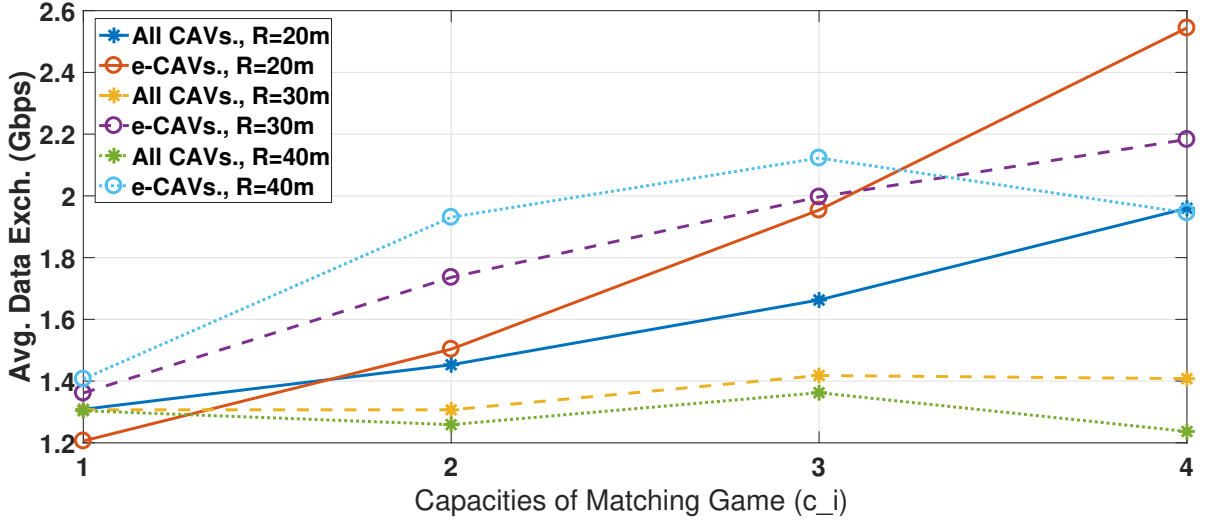
Figure 6.5: Average data exchanged per timeslot for each CAV (either for all CAVs or individually each e-CAV). The beamwidth is $\theta = 15°$.

e-CAVs individually and all the vehicles in the system, we observe that our approach prioritises the traffic towards e-CAVs as intended. Starting with $c_i = 1$, e-CAVs exchange roughly the same amount of data as the remaining vehicles. However, when the capacity is increased, the average exchanged data with e-CAVs becomes greater compared to the remaining vehicles. It is worth noting though that the average performance throughout the network remains almost constant even though the increased capacity leads to a reduced number of stable matches and therefore, less association time. Finally, we observed better performance for $R = 40\,\mathrm{m}$ and for $c_i = 2$ and $c_i = 3$, whereas when $c_i = 4$, the maximum achieved data exchanged was observed when $R = 20\,\mathrm{m}$. This is because the increased matching capacity and radius reduce the number of stable matches, consequently changing the performance of our system.
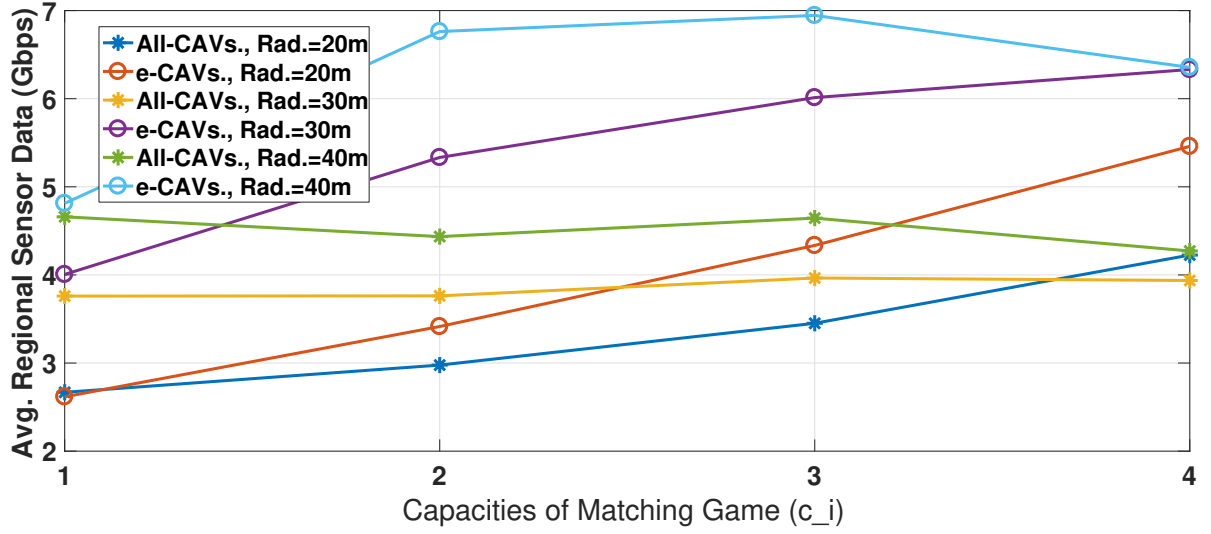
Figure 6.6: Average access to sensor data per timeslot for each CAV (either for all CAVs or individually for each e-CAV). The beamwidth is $\theta = 15°$.

Similarly, in Fig. 6.6, the average access to regional sensor data via the paired CAVs is presented. Again e-CAVs have access to more data compared to r-CAVs (~4.2 Gbps to ~5.5 Gbps for $c_i = 4$ and $R = 20$ m). This value is almost doubled for $R = 40$ m (~4.6 Gbps to ~7.4 Gbps). The above will have a direct impact in the manoeuvrability of e-CAVs as they will have access to more sensor data. Also, and as shown, our system does not limit the sensor data access to the remaining CAVs. The performance when $\theta = 5°$ is similar to the aforementioned ones (Figs. 6.5 and 6.6).

6.4.3 Complexity and Stability Analysis of Stable Fixture Matching Game

In the previous sections, we described our approach for a V2V association scheme using the Stable Fixture matching game. As described in [118] this is an algorithm with complexity of $O(m)$. From the computational complexity theory, as introduced in Computer Science, we know that $O(m)$ implies that the complexity of an algorithm will grow linearly and in direct proportion to the size of the input data set. In that particular case, the input data set is the number of vehicles involved in the game. Also, this will be the worst-case scenario; that means that when a break criterion is introduced the estimated execution time decreases. This is the case of Phase 2 of our algorithm. When the two stability criteria are not met the execution of the algorithm will stop, and the system resources will be freed for other processes.

Given the inability of the vehicles to form NLOS links due to the propagation characteristics of mmWaves, and the relatively short range that the signal can still be decoded, the formulated links will be within a short radius around a given vehicle. This implies that the starting initial input of vehicles participating in the matching game will be small. Of course, the number of

vehicles is related to the road layout, the number of lanes and the speed of vehicles. Generally, we can say that the worst-case scenario for moving vehicles, is the scenario that the target vehicle is in the middle of an intersection being exposed to vehicles coming from all the different directions. For simpler scenarios (e.g., a straight road) the number of potential vehicles to participate in the game is limited to the ones that are in front of behind our target vehicle.

For an SF matching game, that is an algorithm with linear complexity as described before, having a small input data set means that the required execution time will be short as well. To demonstrate the time required for finding a solution, we designed a simple example on our Matlab simulator. The scenario is a cross-junction (the worst-case scenario as discussed before), with two lanes per direction (four lanes per road). A vehicle was positioned at the centre of the intersection, in order to have LOS with all four roads surrounding it. To generate the traffic density, we took the upper limit before a "breakdown" occurs and causes a severe traffic jam. This limit is 67 vehicles per mile per lane [183]. Given that a vehicle will be interested in being connected to a vehicle that is in close proximity ($\leq 40\,\mathrm{m}$) and assuming that the vehicles are equally spread on the road, we calculated the number of CAVs surrounding our target vehicle, i.e., ~27 vehicles. That means that 28 vehicles participate in our matching game.

In order to calculate the execution time, we utilised *timeit* function from Matlab. The *timeit* function calls a specified function multiple times and returns the median of the measurements. The system utilised was a PC with specifications: Intel i7 3.7 GHz, 16 GB RAM, Ubuntu 18.04 LTS, Matlab 2017b. For the above scenario, the returned average was 11 ms. Of course, this is the upper bound for the worst-case scenario and it is system-dependent. The execution time depends on the processing power of the given system. Also, Matlab does not operate very fast when an algorithm is designed with nested loops as in our case. It is expected that if this algorithm is designed in a real-world system using a faster programming language (e.g., C++), will operate a lot faster. All in all, the above observations prove that our algorithm could potentially be utilised for a real-world system design.

In terms of the stability analysis, the two main factors that affect the stability ratio is the size of the input data set and the requested matching capacity per vehicle. For both factors, the bigger the value the more decreased the stability ratio will be. An example can be found in [117], where the authors calculated the stable matching ratio for a constant value of matching capacity and different number of vehicles as an input, under an LTE-V2X scenario. Similarly, we calculated the stable matching ratio for the above-mentioned scenario (the cross-shaped intersection) for a matching capacity of 4 and different number of vehicles. The results can be seen in Table 6.1 and show the stable matching ratio achieved from all vehicles participating in the game. As shown, as the number of vehicles participating in the game is increased, the number of stable matches is decreased, never though going below 87.1%. As described in [118], there are algorithms that lead to the notions of weak stability, strong stability and super-stability relaxing the stable or unstable nature of the SF matching game [184, 185]. These could potentially be incorporated in

Table 6.1: Stable matching ratio as a function of the number of vehicles.

Vehicles Number	5	10	15	20	27
Ratio	98.2%	96.7%	92.5%	89.9%	87.1%

the system in the future, to enhance the further performance.

6.5 Conclusions

Following Chapter 5, in this chapter we presented a novel association scheme for V2V mmWave links. As described above, our system operates in a heterogeneous manner and can enhance the cooperative autonomous driving for the next-generation CAVs. We formulated a utility function based on the information encapsulated within DSRC CAMs. Based on the outcome of our formulation, we prioritise the generated sensor data towards the higher priority CAVs, without though limiting the network access to the rest of the vehicles. We solved our formulated problem using a tool from the Matching Theory field, called Stable Fixtures matching game. When played between the vehicles, the outcome is a list of multipoint-to-multipoint links that combined can increase the network throughput and the channel utilisation for all CAVs. The performance of our system was compared with the traditional point-to-point link association scheme that is usually used for mmWave communications. The performance of our system was evaluated using realistic mobility traces to approach a real-world-like implementation. Utilising the approach mentioned above, e-CAVs at first and the remaining CAVs later, are expected to have enhanced access to valuable sensor data from their surrounding environment enhancing their cooperated manoeuvrability and consequently the road safety.

EFFICIENT MILLIMETRE-WAVE INFRASTRUCTURE PLACEMENT FOR CITY-SCALE ITS

7.1 Introduction and Contributions

7.1.1 Introduction

In this chapter we address the problem of the efficient mmWave-RSU placement within an urban environment. As discussed before, especially in dense urban environments, mmWaves-RSUs will significantly improve the performance of the small-cell access networks [120], as they can enable gigabit-per-second data rates and ultra-low latency for V2I links. RSUs are costly to deploy and maintain. Therefore the desirable number of positions and their location should be carefully chosen. Choosing the suitable locations is vital to improving the network performance, and an efficient and agile method is required, that can operate under different scenarios and environments.

For the remaining of this chapter, we will present our strategy that can automate the mmWave-RSU placement. We take into account the unique road and building layout as well as the LOS requirements and the strict propagation characteristics of the mmWaves technology. To find the best positions, we will incorporate the QoS constraints of vehicular applications within our problem formulation, considering two KPIs that need to be fulfilled. The first one is the LOS network coverage achieved after all RSUs are deployed and the second one is the RSS averaged throughout the considered deployment area. The rest of the chapter is organised as follows. In Sec. 7.2 we describe our system model and how we identify the candidate RSU positions. In Sec. 7.3 we formulate our optimisation problem and outline our proposed strategy. Sec. 7.4 summarises our performance investigation where we compare the aforementioned deployment strategies. Finally, Sec. 7.5 summarises our findings.

7.1.2 Contributions

The contribution of the author for Chapter 7 is the following:

- Designed a scalable system that is able, given a city-map, to identify the potential RSU locations, based on Computational Geometry.

- Developed a model that can find the best mmWave-RSU locations given an initial set of positions.

- Identified the QoS requirements needed for a city-scale implementation and integrated them within the given problem and the proposed algorithm.

- Developed a simulator in Matlab, implementing the above system, that can automate the RSU selection process.

- Simulated in Matlab the different scenarios, comparing the above algorithm with an implementation of a Genetic Algorithm (GC) and a Greedy Construction (GC) algorithm and examined the numerical results.

The contents of this chapter were submitted for publication at the conference:

- IEEE Vehicular Networking Conference (VNC) 2018.

7.2 System Model

We consider an urban city map $\mathcal{M}$ with dimensions $[\mathcal{M}_x, \mathcal{M}_y]$, measured in meters. Let $\mathcal{C} \triangleq \{1, \ldots, C\}$ denote the candidate RSU positions, with all being within the boundaries of $\mathcal{M}$. For all the above positions we denote as $\mathcal{D} \triangleq \{1, \ldots, D\}$ the positions chosen to deploy an RSU, and $\mathcal{R} \triangleq \{1, \ldots, R\}$ the rejected ones. We have that $\mathcal{D} \subseteq \mathcal{C}$, $\mathcal{R} \subseteq \mathcal{C}$, $\mathcal{D} \cup \mathcal{R} = \mathcal{C}$, and $\mathcal{D} \cap \mathcal{R} = \emptyset$ hold.

Our RSU placement algorithm works in two steps:

1. At first, we identify all locations $\mathcal{C}$ to deploy an RSU employing tools from Computational Geometry.

2. Then, we choose a subset $\mathcal{D} \subseteq \mathcal{C}$ in order to maximise the outdoor coverage and the RSS for the entire map.

In our system model, we assume that all the RSUs are mounted at the height of a lamp post or a traffic light. By that, we avoid most of the low-level obstacles (e.g., kiosks, vehicles, trees, etc.). For simplicity, we consider a 2D network planning. Finally, we refer to OpenStreetMap [186] to obtain our city maps.

The problem of identifying $\mathcal{C}$ can be approached by using tools from Computational Geometry. Similarly to [130], we introduce the notions of *Simple Polygon (SP)* and *Polygon With Holes (PWH)*
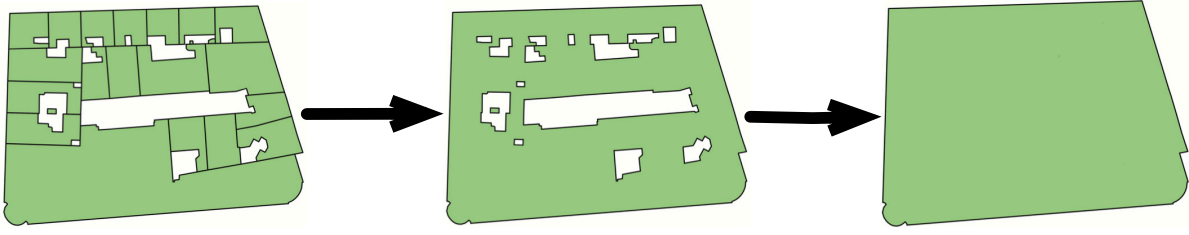
Figure 7.1: Example of how adjacent buildings (left), are concatenated to a single polygon (centre) at first, and the inside holes are removed (right), using the union polygon operation [4]. By that, we can decrease the execution time of our simulation.

in our system, describing the buildings and the roads, respectively. For the buildings, we process the map data acquired to generate an SP for each city block. Similarly, the road data are parsed to generate a PWH representing the roads in our system. All the SPs are used to determine if a link is in LOS or not.

7.2.1 Identifying Potential RSU Locations

An SP is considered a flat shape consisting of straight, non-intersecting line segments, that, when joined pair-wise, they form a closed path. In our system, all buildings $\mathcal{E} \triangleq \{1,\ldots,E\}$ are considered as SPs, with the sides of the buildings being the sides of the SPs. Given a city map, many SPs have adjacent sides, being part of the same city block B_i. These buildings can be concatenated using the polygon union operation [4] to decrease the execution time. Concatenating all the adjacent SPs, we usually end up having a hole in the middle (e.g. a courtyard). Removing the inner holes of all city blocks and representing the entire city block as an SP, we are given a new set of SP $\mathcal{B} \triangleq \{1,\ldots,B\}$ that can use to find the LOS coverage rate. An example of the above operation can be seen in Fig. 7.1.

On the other hand, PWH is a polygon with an irregular shape that contains one or more holes or cutouts in it. As mentioned before, we assume that the RSUs will be placed either on lamp posts or traffic lights. OpenStreetMap data represent the road as a solid line. However, having access to the metadata of each road with, we can later accurately calculate the different polygons representing each road. Each polygon is later concatenated with the others, having finally a concave PWH that will be used for the RSU placement.

Knowing the generated SPs and PWHs, we can identify $\mathcal{C}$ for a given map. Our algorithm searches along the sides of PWHs for sharp edges and long straight sections. The edges, being the corner of two roads, are usually the best positions for an RSU, as they can maximise the LOS coverage (as shown in [127]). We also consider the length of a road. A road qualifies as long when the distance between two intersections is greater than a given threshold RSU_t. For any long road, we consider more possible positions, equally spaced between the two intersections. The number of these positions is given as the ceiling function from the division of the length l_i of the road i,

over the given threshold, i.e. $\lceil l_i/\text{RSU}_t \rceil$. Combining both lists, $\mathcal{C}$ is found.

7.3   Problem Formulation and Solution

Given $\mathcal{C}$, the objective is to find $\mathcal{D} \subseteq \mathcal{C}$ to maximise the network coverage and achieve a minimum RSS throughout the network. The LOS coverage rate is modelled by equally spacing $\mathcal{Z} \triangleq \{1, \dots, Z\}$ grid points on the map with equal weights. Each point represents a squared tile having the same RSS throughout its surface. Using a tile-like approach with relatively small tiles, we can decrease the processing power required without having a significant loss of accuracy.

We determine $\mathcal{D}$ by taking into account two different constraints. For a given $\mathcal{Z}$, we consider a set of $\mathcal{N}$ reference points, with $\mathcal{N} \subseteq \mathcal{Z}$, being the grid points on top of the road polygons. We define the binary variable γ_n, to denote the state of a reference point n at location (x, y) as follows:

$$(7.1) \qquad \gamma_n(x, y) = \begin{cases} 1, & \text{if } n \text{ is covered by at least one RSU} \\ 0, & \text{otherwise} \end{cases}$$

Our first constraint imposes that all at least $|\mathcal{N}|$ tiles are in LOS with at least a RSU, i.e.:

$$(7.2) \qquad \sum_{n=1}^{|\mathcal{N}|} \gamma_n(x, y) \geq \tau \, |\mathcal{N}|$$

where $\tau \in [0, 1]$, is a tolerance factor that can relax the constraint.

We say that P_{tx} and G_{tx} are the transmission power and antenna gain of each RSU. Also, $L_{\text{LOS}}(d)$ signifies our propagation loss at a distance d. Given the above, we can calculate the RSS for each tile as $\text{RSS}(d) = P_{\text{tx}} + G_{\text{tx}} - L_{\text{LOS}}(d)$, with $L_{\text{LOS}} = 10\,n\,\log_{10}(d) + C_{\text{att}}$. For all $\mathcal{D}$, there is a number of tiles K_i that surrounds it. Each tile can be served by more than one RSU. We define as $\max_{i \in \{1, \dots, \tau \, |\mathcal{N}|\}} \{K_i^{\text{RSS}}\}$ the highest received RSS from all RSUs that serve this tile. The interference between the deployed RSUs is not taken into account. For the entire covered area, we sort all tiles with respect to their received RSS value, we take the first $\tau \, |\mathcal{N}|$ and denote them as $\text{RSS}_{\text{i,k}}^{\max}$. Our second constraint ensures that a target number of tiles has an average RSS that is greater than or equal to a threshold RSS_{th}:

$$(7.3) \qquad \sum_{i=1}^{|\mathcal{D}|} \sum_{k=1}^{\tau \, |\mathcal{N}|} \text{RSS}_{\text{i,k}}^{\max} \geq \text{RSS}_{th}$$

7.3.1   Problem Formulation

Let $e_{\mathcal{C}}$ be the vector that defines the state of each RSU i in $\mathcal{C}$. We have:

$$(7.4) \qquad e_i = \begin{cases} 1, & \text{if } i \text{ is deployed} \\ 0, & \text{if } i \text{ is not deployed} \end{cases}$$

134
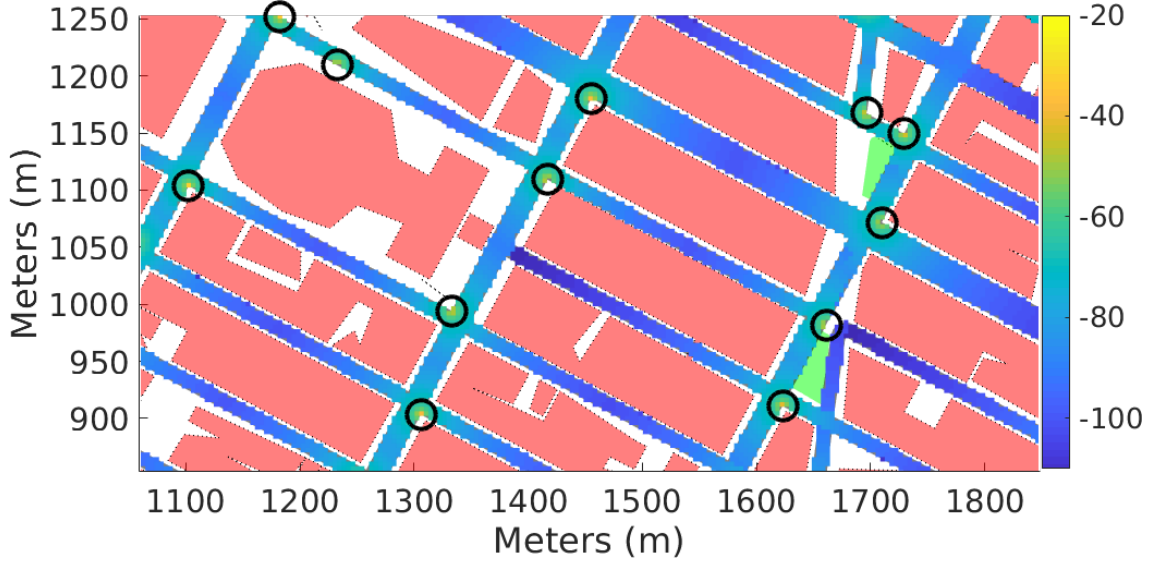
Figure 7.2: Example of 12 RSUs chosen for Manhattan by our strategy with their corresponding RSS. The street geometry affects the perceived RSS.

In order to find the best RSU locations, our problem is formulated as the minimisation of (7.4). More specifically, we have:

$$\text{(7.5a)} \qquad \underset{e_1,\dots,e_\mathcal{C}}{\text{Minimise}} \sum_{i=1}^{|\mathcal{C}|} e_i$$

$$\text{(7.5b)} \qquad \text{subject to:} \sum_{n=1}^{|\mathcal{N}|} \gamma_n(x,y) \geq \tau\,|\mathcal{N}|$$

$$\text{(7.5c)} \qquad \sum_{i=1}^{|\mathcal{D}|} \sum_{k=1}^{\tau|\mathcal{N}|} \text{RSS}_{i,k}^{\max} \geq \text{RSS}_{th} \qquad\qquad , \forall i \in \mathcal{D}$$

$$\text{(7.5d)} \qquad e_i \in [0,1] \qquad\qquad , \forall i \in \mathcal{C}$$

$$\text{(7.5e)} \qquad \text{RSS}_{i,k}^{\max} \text{ for } k \text{ in LOS} \qquad\qquad , \forall i \in \mathcal{D}$$

where the above are the constraints described in (7.2) and (7.3). Also, the RSS is calculated for all tiles being in LOS with a RSU, i.e. a SP does not block the link. An example solution for a small urban area can be seen in Fig. 7.2. In this particular example, we can observe the effect of the street geometry on the RSS.

7.3.2 Proposed Algorithm

To solve (7.5a)-(7.5e), we propose a novel algorithm to calculate the list $\mathcal{D}$. A city-scale RSU placement problem can be computationally expensive. So, we designed our algorithm to operate in three phases, to minimise the execution time. Our algorithm (Alg. 5) works are as follows:

Phase 1: We start by calculating the number of tiles k_i' required to achieve a mean RSS value greater than or equal to RSS_{th} for all $\mathcal{C}$. Later, we iteratively add to $\mathcal{D}$ the RSU with the most

---

Algorithm 5 Agile RSU Placement

Output: Return lists with RSUs: $\mathcal{D}$ and $\mathcal{R}$

Phase 1:

1: Calculate the tiles k_i' required to achieve the RSS_{th} for each $\mathcal{C}$
2: **while** Constraint (7.5b) not met **do**
3:     Find served tiles in the system and remove from lists k_i'.
4:     Find RSU i with the longest k_i' list and add it in $\mathcal{D}$.
5: **end while**

Phase 2: – Skip if (7.5b) and (7.5c) are met.

6: **while** Constraint (7.5c) is not met **do**
7:     **for all** RSUs in $\mathcal{C} \notin \mathcal{D}$ **do**
8:         Calculate number of non-covered tiles that they can serve.
9:     **end for**
10:     **if** Non-covered tiles on map **then**   ▷ i.e., (7.5b) is not maximised
11:         Find RSU i that covers the most non-covered tiles.
12:         Add i in list of chosen RSUs $\mathcal{D}$.
13:     **else**
14:         **for all** RSUs in $\mathcal{C} \notin \mathcal{D}$ **do**
15:             Calculate the potential mean RSS if RSU is chosen.
16:         **end for**
17:         Find RSU i that maximises the mean RSS for the system.
18:         Add i in list of chosen RSUs $\mathcal{D}$.
19:     **end if**
20: **end while**

Phase 3:

21: **repeat**
22:     **for all** RSUs i in $\mathcal{D}$ **do**
23:         Find k in $\mathcal{C} \notin \mathcal{D}$ that improves constraints (7.5b) and (7.5c).
24:         Replace i with k.
25:     **end for**
26: **until** $\mathcal{D}$ cannot be improved more   ▷ i.e., no more swaps can be done

---

non-served tiles within RSS_{th} found before, until constraint (7.5b) is met. Thus, we ensure a sufficient amount of coverage in our system in a fast greedy-addition-like fashion. If (7.5c) is also fulfilled, we proceed to Phase 3. If not, we continue with Phase 2.

Phase 2: We add more RSUs in the system until both constraints are met. We identify the non-covered areas of the map and prioritise our RSU placement towards them, as this can increase our system performance. Then, we find the areas that are not adequately served by the existing RSUs, and we add RSUs that can fulfil (7.5c). When both constraints are fulfilled, we have two admissible lists $\mathcal{D}$ and $\mathcal{R}$ and we proceed to the next phase.

Phase 3: From the above two phases, we may not always achieve an ideal solution for $\mathcal{D}$. This happens especially when the requirements for a specific scenario are more relaxed (e.g. a low coverage rate is required). To improve the performance at this point, and we search in $\mathcal{R}$ if it exists an RSU that can improve (7.5b) or (7.5c). If so, we replace these two. We iterate, until no other RSUs can be swapped, meaning that we have our final $\mathcal{D}$.

Table 7.1: List of map areas used and simulation parameters.

Urban Area	Manhattan, NY, USA	Paris, FR
Centre	$-73.9841°W, 40.75545°N$	$2.33235°W, 48.875°N$
Number of Maps	2×2	2×2
Map Size $[\mathcal{M}_x, \mathcal{M}_y]$	$1\,\text{km} \times 1\,\text{km}$	$1\,\text{km} \times 1\,\text{km}$
Area of Interest	$900\,\text{m} \times 900\,\text{m}$	$900\,\text{m} \times 900\,\text{m}$
Grid-Tile Size	$4\,\text{m} \times 4\,\text{m}$	$4\,\text{m} \times 4\,\text{m}$

Parameter	Symbol	Value
Transmission power	P_{tx}	10 dBm
TX Antenna Gain	G_{tx}	15 dBi
Path-Loss Exponent	n	2.66
Channel Attenuation	C_{att}	70 dB
Distance Threshold	RSU_t	100 m

7.4 Simulation Results

Our strategy is compared against a GC and a GA approach. GC is scalable, but it cannot fulfil the required KPIs, as it does not take into account the RSS threshold. The second one is notoriously computationally expensive but generates high-quality solutions. We test all the algorithms in two urban areas from Manhattan (New York, USA) and Paris (FR). An average road lane is 2.9 m to 4.6 m wide. Therefore, we considered a grid with a side equal to 4 m, so each tile covers roughly the width of a road lane. The considered areas and the simulation parameters are summarised in Table 7.1.

Each area is $4\,\text{km}^2$ and divided into four equal sections. Each section is considered as an independent map in our simulation with dimensions $[\mathcal{M}_x, \mathcal{M}_y]$ and a surface of $1\,\text{km}^2$. The centre coordinates given in Table 7.1 present the point that the edges of all four sections meet. For each section, we consider an area of interest of $810\,\text{m}^2$, to avoid border effects. Four different tolerance parameters are considered, namely, $\tau \in \{0.85, 0.90, 0.95, 0.99\})$, with 90% being an average coverage rate, while 99% being the extreme case. Also, four RSS thresholds are considered, namely, $\text{RSS}_{th} \in \{\text{Inf}, -90, -84, -79\}$. The first value signifies the case where no RSS threshold is considered. The last one was chosen based on the sensitivity threshold of IEEE 802.11ad, i.e., the minimum RSS (without considering the RX antenna gain) to achieve one-gigabit-per-second data rate.

In Fig. 7.3, we present the RSS results per-tile for Manhattan and $\tau = 0.99$. We observe that GC always produces similar results, comparable with our scheme (when the RSS constraint is disregarded). That is because GC cannot take into account the RSS threshold. Obviously, as the RSS_{th} is taken into consideration, we observe that both our algorithm and GA perform better than
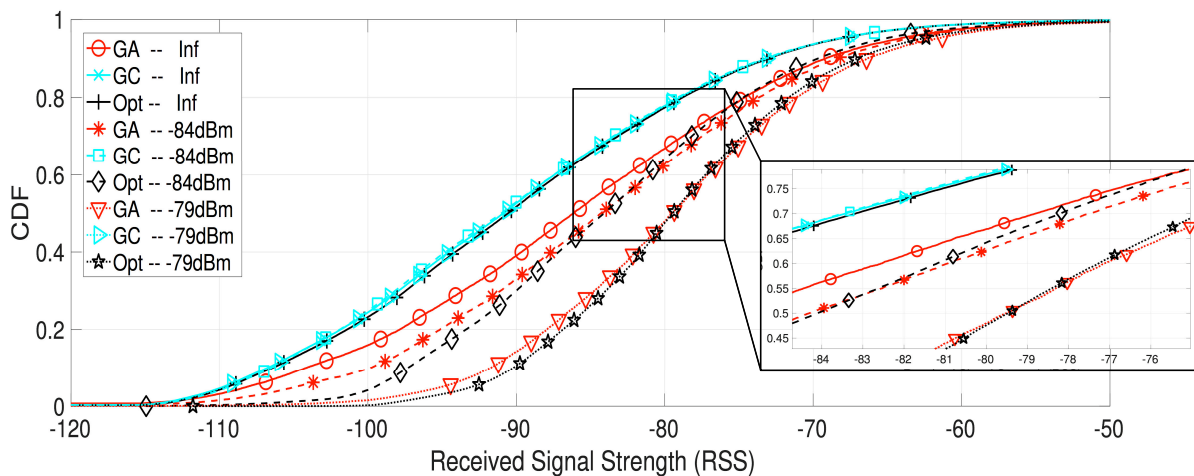
Figure 7.3: The empirical CDF of the RSS for the map of Manhattan. A tolerance $\tau = 0.99$ and three different RSS_{th} were used for this scenario.
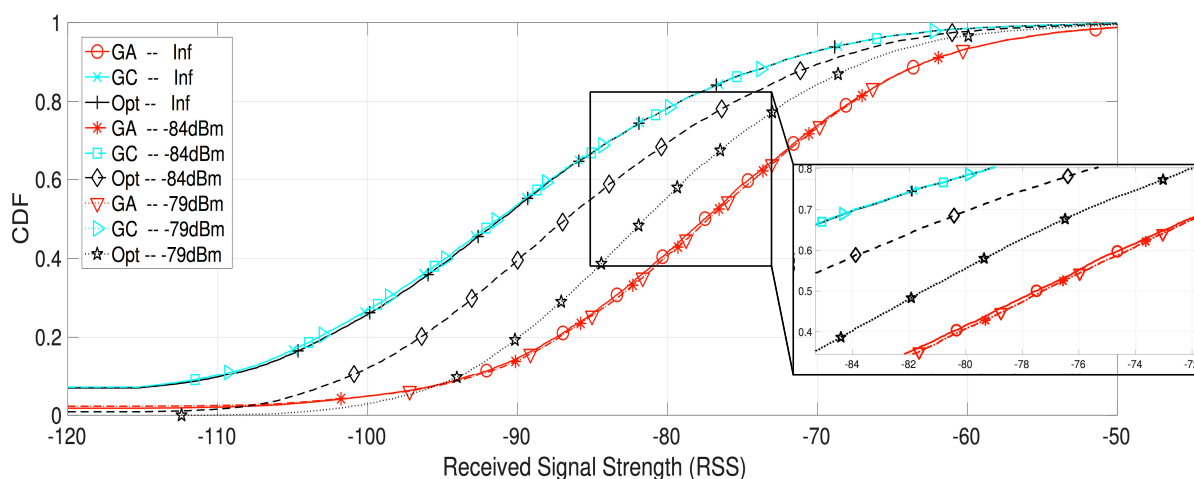
Figure 7.4: The empirical CDF of the RSS for the map of Paris. A tolerance $\tau = 0.90$ and three different RSS_{th} were used for this scenario.

GC. In particular, our strategy achieved a city-wide mean RSS of -83.6 dBm and -78.9 dBm, while GA achieved -84.3 dBm and -79.2 dBm for $RSS_{th} = -84$ dBm and $RSS_{th} = -79$ dBm, respectively.

Similarly, Fig. 7.4 presents the RSS results for Paris and $\tau = 0.90$. Again, GC algorithm behaves as before, having similar performance for all thresholds and being comparable with our algorithm when RSS_{th} is not taken into account. However, the main difference compared to the Manhattan results is the performance difference between our algorithm and GA. We observed that, even with relaxed parameters GA always finds an extreme solution (mean RSS of -78.8 dBm and -78.5 dBm for $RSS_{th} = -84$ dBm and $RSS_{th} = -79$ dBm respectively), while our algorithm behaves as before (mean RSS of -86.2 dBm and -80.8 dBm). This is because of the highly irregular building shapes of Paris, compared to the grid-like shape of Manhattan, making
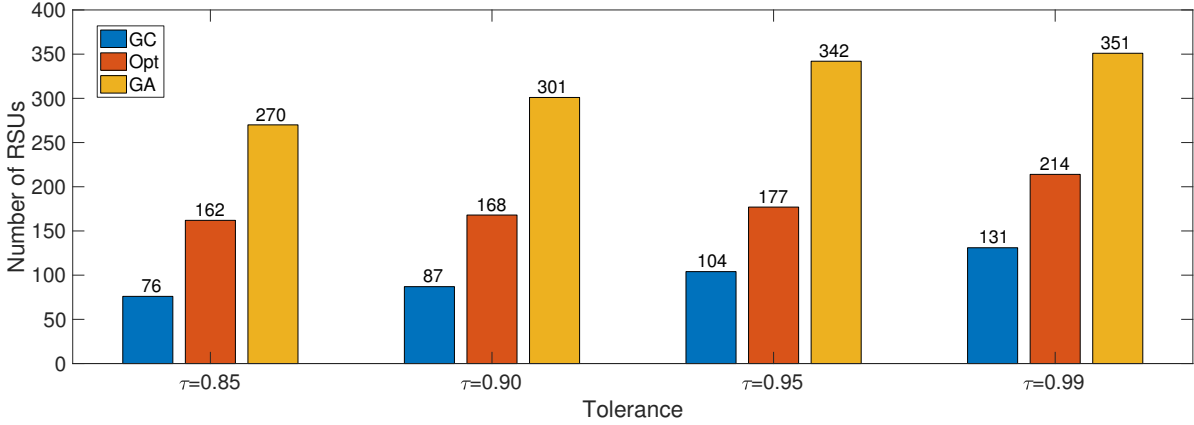
Figure 7.5: The number of RSUs (Manhattan), given by all algorithms, for all different tolerance parameters. The RSS_{th} is equal to $-84\,\text{dBm}$.
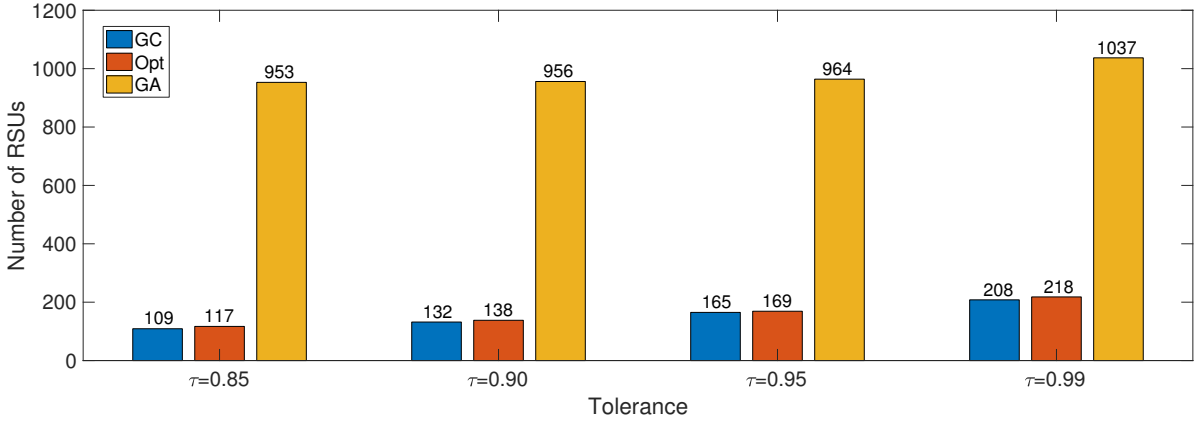
Figure 7.6: The number of RSUs (city of Paris), given by all algorithms, for all different tolerance parameters. The RSS_{th} is equal to $-90\,\text{dBm}$.

it very difficult for GA to find the best RSU positions in the city and always getting stuck to a local maximum. For both cities, all the algorithms behave similarly for other tolerance parameters and will not be presented here due to the limited space.

In Figs. 7.5 and 7.6, we present the number of RSUs required to fulfil the QoS constraints required, for all τ. Fig. 7.5 refers to the Manhattan scenario, for $\text{RSS}_{th} = -84\,\text{dBm}$. Once more, GC utilises fewer RSUs, but it does not take into account the RSS_{th}. Comparing the number of RSUs obtained with the proposed approach and with GA, we observe that our scheme ensures a reduction of up to 50% in the number of RSUs compared to GA. From Figs. 7.3 and 7.5, we observe that our algorithm achieves comparable results, fulfilling the QoS requirements with a smaller number of RSUs. Moving on to Fig. 7.6, the difference between the number of RSUs and the two algorithms is greater, having GA solving our optimisation problem with almost eight times as many RSUs. Again, as before, this is because of the irregularity of the building and road

shapes.

7.5   Conclusions

In this chapter, we propose an agile and efficient strategy for city-wide mmWave-RSU placement. We present a scalable algorithm, able to compute high-quality RSU deployment on a map. In doing so, the proposed strategy takes into account two KPIs for vehicular communications – the coverage rate and the RSS threshold. Our approach is compared to the GC and GA strategies. GC is fast and scalable, but it cannot satisfy the above mentioned KPIs, while GA is computationally expensive. We showed that our strategy meets target coverage constraints with a smaller number of RSUs, making it a suitable solution for large-scale experimentation.

8

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

The work in this thesis was motivated by the need to build an agile communication framework, able to accommodate the exchange of sensor data and manoeuvring intentions within a next-generation C-ITS and the forthcoming autonomous vehicles. CAVs, becoming cooperative entities, require the exchange of sensor data with stringent QoS constraints, such as multi-gigabit-per-second data rate and ultra-low end-to-end latency. Therefore, an efficient, highly reliable and robust communication plane is of paramount importance. While existing technologies have been preliminarily employed around the globe to accommodate vehicular applications data in their infancy, we identified the need for more sophisticated solutions to meet the strict QoS requirements of the future. Our work focused on the design of a novel heterogeneous C-ITS framework able to handle scalable city-wide ITS application data streams. Having in mind the necessity for large-scale deployment we experimented using both real-world field trials and simulation models for sizeable scenarios. We tackled different drawbacks of our envisioned system, evaluating at first the performance of the existing solutions and proposing novel algorithms to overcome the current problems. More specifically, the thesis main contributions per chapter are as follows:

- Chapter 3: We presented our envisioned C-ITS multi-RAT architecture that will enhance the CAV-related MaaS paradigm. Our system components, i.e. the Decision-Making Agent, the ITS Agent and their sub-components can provide the necessary flexibility, manageability and configurability required for a C-ITS framework. Incorporating an SDN-like design with Fog and Cloud computing capabilities, we can not only enable the central system coordination of a city-scale implementation, but also intelligently allocate the existing resources, accommodate different data streams, and fulfil the demanding QoS constraints.

  We envision our paradigm to play a pivotal role in the next-generation C-ITSs. However, we

understand the complexity of the above-mentioned system and the time and effort required for a city-scale deployment. This difficulty arises from the need to precisely design, test and integrate the different system components before they harmonically cooperate as a whole. With that in mind, for the remaining contribution chapters, we focused on various subsystems of our framework, delving specifically into IEEE 802.11p/DSRC and mmWave wireless technologies. For the above, we identify existing problems and propose solutions to overcome them.

- Chapter 4: IEEE 802.11p/DSRC is with no doubt a technology that has been evaluated and enhanced with numerous ways from the research community in the past years. Though, we identified that most research activities are either based on inaccurate simulation models or test the performance of small testbeds, usually within the control environment of a laboratory. An algorithm, to move from the experimental phase to becoming a standard and be used in the real world, needs a more precise evaluation approach. In this chapter, we described at first how OMNeT++ Veins-INET could be microscopically calibrated to represent the reality better. We based our calibrations on a number of control outdoor experiments within different environments and shown that especially for longer distances, the existing simulation models are inaccurate. We also described the steps to incorporate different device profiles into an existing simulation framework and the benefits from that.

  Later, we discussed the necessity for large-scale performance evaluation of the vehicular communication framework. However, a city-scale evaluation significantly increases the computational complexity, making the evaluation almost prohibitive with the existing software and hardware tools. Especially, for a full-stack simulation framework like OM-NeT++ Veins-INET, the complexity is of $O(mn^2)$. To overcome this problem, we proposed our multi-threaded version of OMNeT++ Veins-INET that is able almost to halve the required simulation time. By that, it becomes more feasible to evaluate the performance of larger city-wide scenarios. To take that one step further, we designed and deployed a city-scale ITS-G5 network around the City of Bristol, based on the current standards. We also conducted some preliminary real-world large-scale field trials using two vehicles and three RSUs. Some key findings from our experimental campaign are the need for more sophisticated MAC-layer schemes as well as the benefits of a Fog-computing infrastructure.

  The above research activities gave us an excellent insight into how a network simulator operates and how it can be enhanced. We also gained valuable experience in the area of large-scale deployments and how to design an experimental testbed. Of course, our design is not perfect. We identified the various drawbacks and described them in this chapter, and we intend to fix them in the future. We also came up with promising ideas, that will not only enhance our system but will be able to get us one step closer to making CAVs part of our everyday life.

- Chapter 5: This chapter presented the first step towards the integration of the different technologies we discussed in Chapter 3 and accommodate the needs for a C-ITS. Our work is based on a heterogeneous intelligent beam steering algorithm for mmWaves. Our strategy uses out-of-band information and a motion prediction algorithm to steer the beams accordingly. We also present an overhead analysis of the IEEE 802.11ad beamforming process in order to strengthen the benefits of our novel design. More specifically, our performance investigation shows that our overhead-free algorithm outperforms the legacy beamforming technique both concerning the average data rate and the overall network throughput performance. Introducing two beamforming algorithms, we provided an all-around model for different scenarios and conditions.

  Our proposed solution could still be optimised in a variety of ways. For example, we conducted a preliminary analysis of position error, its relation with the RSU position on the road and how it affects the network throughput performance. This can be further investigated in the future and come up with more optimisation algorithms based on different scenarios. Furthermore, access to real mmWave devices will benefit our research activities as we will be able to evaluate our algorithms in the real world.

- Chapter 6: This chapter presented a novel association scheme for V2V mmWave multipoint-to-multipoint links that prioritises traffic towards emergency vehicles, without though dictating the channel and degrading the service for the regular vehicles. We build on top of our heterogeneous system utilising both IEEE 802.11p/DSRC and mmWaves technologies, described in the previous chapter. Our formulated utility function is based on the information encapsulated within DSRC CAMs. The physical constraints always limit the link capacity. Also, mmWaves are usually point-to-point links. However, using out-of-band information, we can easier form multipoint-to-multipoint links and later, allocate the resources better and enhance the exchange of valuable sensor data between the vehicles. This will enhance the cooperated manoeuvrability of CAVs and consequently the road safety.

  However, our approach assumes perfect CAM message delivery rate. The integration of a real-world CAM delivery behaviour will make our system a better representation of the reality. Furthermore, we always considered ideal beam alignment. Introducing the beamforming concept in the future will help us investigate how our system behaves under various environments and conditions. Finally, the effect of the physical size of the vehicles and their different profiles was not examined, as well as and their blockage effect, and this is considered a useful extension for the future.

- Chapter 7: The final chapter of this thesis proposed an agile and efficient strategy for city-wide mmWave-RSU placement. We designed a scalable algorithm, that takes into account the coverage rate and the RSS threshold for vehicular communications, and computes a high-quality RSU deployment on a city map. We compared our algorithm with Greedy

Construction and a Genetic Algorithm. Our performance evaluation showed that even though the first one is scalable and fast, it cannot satisfy the above KPIs, while the later, is not easily scalable and not suitable for all environments. Our strategy, outperforming both GC and GA, and meets the given constraints with a smaller number of RSUs. This makes it a suitable solution for large-scale experimentation.

However, our algorithm only considered the link quality when one vehicle should be served. Using realistic mobility traces, we intend to evaluate the performance when more than one vehicles should be served from one RSU and extend our algorithm to meet the increased QoS requirements. Furthermore, a Cell Switch-Off (CSO) approach on top of our placement algorithm can benefit our system. By that, we can adapt the number of enabled devices based on the system requirements and the number of CAVs that should be served on each road. The policies from a system like that can be applied in an SDN-like fashion.

All the above contributions are viable solutions for a next-generation C-ITS as they meet the strict QoS constraints of vehicular applications. As said before, C-ITSs and CAVs are still in their infancy, and we discussed future research directions that will enhance their performance. All things considered, the author believes that the above contributions can bring us one step closer to the cooperated manoeuvring between CAVs, inching towards using them in our everyday life and having as an ultimate goal to provide safer means of transport for everybody.

8.1   Future Research Directions

The previous section presented a critical review of the main contribution of this thesis. Different aspects of the connectivity within the context of a C-ITS were investigated, such as the integration of the existing wireless technologies in a vehicular communication framework and how their performance can be enhanced, multi-RAT architecture designs accommodating scalable sensor data, calibration of the existing experimental tools in order to achieve a more meaningful performance evaluation, real-world trials to perceive the performance of a city-scale testbed under various conditions, etc. However, even though the above topics were investigated, the connectivity of self-driving vehicles is still in its infancy, it is an ongoing area of research, and it is by no means complete. In this chapter, we will discuss the directions that the author believes could lead to fruitful research topics in the future.

8.1.1   Enhanced V2X mmWave Beam Steering

Chapter 5 presented a novel beamforming algorithm for mmWave-V2X communications, named as SAMBA. Our algorithm can align the mmWave beams using out-of-band sensor information exchanged via a DSRC link, reducing the beam training overhead and time required, while increasing the data-rate. SAMBA was evaluated under a dense urban environment for V2I links.

In the future, the performance of our algorithm under a V2V scenario could be evaluated. As discussed in Chapter 5, the performance is affected by the relative difference in the direction of the two vehicles. Given the differences between the azimuth (both northing and easting) and the elevation plane, it is expected to see significant differences under various scenarios. Solving the above problem performance degradation under a V2V scenario will probably require more sophisticated algorithms for the motion prediction of the vehicles.

Furthermore, in our algorithm, we did not consider multiple vehicles being connected with the same RSU within the same BI. Our algorithm assumes that an RSU is always connected to its closest vehicle. Introducing a real-world-like system, where multiple vehicles will be served from one RSU within one BI (using a resource allocation algorithm, as in Chapter 6), will introduce the problem of blockage from the nearby vehicles. This is an exciting problem for investigation and a step closer to a real-world implementation. Also, our system will be benefited by a 3D representation model. As it stands right now, a 2D system was considered. However, in the real-world-like system, the elevation angle should be taken into account as well.

8.1.2 Improving V2X Multipoint-to-Multipoint Association Policy

Chapter 6 described our novel algorithm for a multipoint-to-multipoint link establishment for V2V communications. Our work can be further extended and approach a real-world-like implementation by taking into account all the transmission and alignment delays introduced from the mmWaves links. More specifically, a theoretical analysis can be performed, to accurately calculate the alignment time penalties under different scenarios and number of vehicles. This can be later combined with our introduced motion prediction algorithm in Chapter 5, to have a more accurate representation of the alignment delays and the position of the vehicle every BI.

What is more, our resource allocation strategy did not take into account a queuing model and the buffering of the various packets. Upon arrival to a certain queue, a packet can be either transmitted or dropped if expired. The time a packet spent in the queue contributes to the overall delay discussed before. If a link between two vehicles is active, but the channel conditions are not good enough to deliver all the pending packets, and a new traffic arrival event is initiated, all the old packets will be dropped from the queue. Using this assumption, newer traffic could be prioritised and ensure minimum-delay communication for the demanding next-generation C-ITS applications. Similar activities can be found in [114, 187].

8.1.3 Enhanced Models for better Representation of the Reality

Throughout this thesis, and for all the chapters related to mmWaves, we utilised a generalised shadow fading model to represent our variation of the signal attenuation. Furthermore, we did not consider the impact of the interference in our performance evaluation. Finally, all the above-presented works did not consider NLOS communication scenarios. As it was proven in [188], the wall reflections at the frequency band of 60 GHz can affect the perceived signal, when being in

close proximity to a wall. As it was shown, the first reflection can constructively or disruptively affect the signal. Especially under a dense urban scenario, it is expected that the reflections from a building or the surrounding vehicles, will probably reach the target device.

The author of this thesis believes that utilising more accurate propagation models based on real-world measurements and a model to calculate the impact of the interference, will provide a better representation of the reality for all our proposed solutions. Of course, an approach like that will increase the time required for the performance evaluation as it will increase the computational complexity, as discussed in Chapter 4. To that extent, a careful selection of the appropriate models is required in order to compensate between the required accuracy and the computational complexity.

8.1.4 Improvements in IEEE 802.11p/DSRC Testbed - Centrally Managing a Fog Area

In Chapter 4 we described our efforts in designing an agile calibration method for IEEE 802.11p and Veins-INET simulation framework, we presented our research activities on parallelising the framework mentioned above and discussed our preliminary results using our experimental testbed for a real-world experimental campaign. One of our key observations was that the legacy IEEE 802.11p/DSRC standard lacks a sophisticated MAC-coordination scheme and the co-existence of various vehicles can cause significant interference problems. This is an ongoing area of research for several years (e.g. [189, 190]), however to the best of our knowledge all these proposed solutions have not been tested on a real-world environment. Using an experimental testbed as the one we designed, it will be a great activity to investigate the performance of existing or newly designed algorithms further and perceive their real-world behaviour.

Furthermore, we identified the necessity for a Fog-computing infrastructure and the requirement for clustering a big area into smaller Fog Areas and centrally managing each one of them using an FO. A system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum is of paramount importance and can significantly enhance the performance of our system. Open-source Fog implementations already exist (e.g. Fog05, OpenFog, OpenVolcano) that can be integrated with an existing platform and provide fog-computing capabilities. What is more, the problems identified, related to the synchronisation of the devices and the inconsistencies observed during the generation and the exchange of data, could be tackled in the future. It will also be interesting to investigate the Received Signal Strength Indication (RSSI) of the received packets. This will give a better insight into how such devices operate under the different scenarios and distances. Finally, a cybersecurity framework will benefit the above system. Digital certificates can be used, i.e. a Certificate Revocation List (CRL) and a public key infrastructure, as it described in X.509 Standard [191], in order to encrypt our wireless links (as in [144]). A very interesting topic of research is the investigation of the overhead generated and how it affects the performance of a

vehicular network.

8.1.5 Performance Bench-marking for LTE-A/LTE-A Pro Networks

The research activities in this thesis mainly focused on IEEE 802.11p and mmWaves wireless technologies. In Chapters 2 and 3, we briefly described the operation of LTE-A and LTE-A Pro and how it can be integrated with vehicular communications. We also introduced some use-cases that an LTE link will be required for a next-generation C-ITS. However, we did not investigate the performance of the links mentioned above under vehicular environments. This can be an exciting area of research for the future. Similar research activities can be found in the literature. For example, and similarly to the activity introduced in Sec.4.4.3, the authors in [192] compared the handover performance of a V2I LTE-A link under real-world field trials and a virtual-driving emulated scenario. Their work described how an emulation framework could be used to evaluate the performance of a real-world testbed and similarly, various other aspects of the network performance of an LTE-A or LTE-A Pro link can be investigated.

As we discussed before, the LTE-A and LTE-A Pro support high data-rates and low-to-medium latency, due to their flat network architecture. However, the strict *tactile-like* latency requirements, smaller than 10 ms, cannot be fulfilled in the presence of higher cellular traffic load. What is more, the additional latency of 110 ms, introduced when establishing a link for the first time can lead to severe performance degradation for safety-critical applications. The above problems are topics of ongoing research. For example, authors in [193] described a new D2D discovery scheme, offloading part of the D2D discovery load from the cellular network, which reduces the consumption of valuable resources in the LTE-A network. They also conducted a performance investigation related to the end-to-end delay benefits of their algorithm. Similarly, [194] described a novel adaptive scheduling scheme that can reduce the end-to-end delay while increasing the user satisfaction. Finally, authors in [195] described an adaptive resource allocation scheme that can improve the throughput performance under dense vehicular environments. Overall, the open questions related to the integration of LTE-A within a C-ITS are endless, and it is considered a fascinating area of research for the future.

8.1.6 Implementing SAMBA and our Association Policy algorithm on an Experimental Testbed

Various resource allocation and beam alignment algorithms found in the literature can impact the performance of a C-ITS; however most of them are not implemented in a widely adopted fashion. Experimental mmWave devices already exist at the frequency band of 60 GHz. Some example start-up companies providing this equipment are the Blu Wireless Technologies, Peraso, Nitero, etc. Also, similar solutions can be found in large companies and organisations like Qualcomm, Huawei, and Samsung.

Designing and implementing the algorithms described in Chapters 5 and 6 in a real-world system will help us further evaluate their performance and identify potential drawbacks. This will also be a significant step forward for large-scale adoption of 5G WiGig solutions in our everyday life and the next step for an actual C-ITS.

8.1.7   Active Jammers, Link Security and Various Malicious Attacks

As we discussed in the previous chapters, the flawless operation of an autonomous vehicle relies on its highly reliable communication framework, the small number of lost packets, the high data rates and the low-latency. One crucial question that was left unexplored is the effect of an active attacker and how it can disrupt the various services of a C-ITS. Similar activities can be found in [196]. Even though various simulation models exist that can evaluate the performance of a device under an attack, we showed in Chapter 4, that these simulation models are not always accurate and cannot reflect the reality.

Having an operational IEEE 802.11p/DSRC system, a future activity can be the implementation of a jammer (as in [196]) and test various malicious attacks. For example, earlier in this chapter, we discussed the potential of designing a cybersecurity framework on top of an experimental system and evaluate its performance. A complementary research activity can be an investigation of how a Man-in-the-Middle (MitM) attack can generate revoked certificates in the network by eavesdropping the packets exchanged. Also, the Denial-of-Service (DoS) or the Distributed DoS (DDoS) attacks can affect the system by flooding the channel and disrupting various services related to a C-ITS. The above attacks can be investigated at different layers of the OSI model and evaluate how a real-world system behaves under them at first, and potential ways to overcome them.

MmWave-V2X MATLAB Simulation Framework

T his Appendix describes the simulation frameworks designed and developed in MATLAB in order to evaluate the algorithms presented in Chapters 5, 6 and 7. Two different simulation frameworks were designed. The first one is a discrete time system level simulator that is capable of parsing the mobility traces of the vehicles, the maps generated, and calculates the different metrics presented in Chapters 5 and 6. The second one is a system level simulation framework having as an input a map and generating the optimal RSU positions, as described in Chapter 7. Both frameworks take as an input various simulation parameters, predefined by the user before the simulation and outputs the different figures presented in the previous chapters. In this Appendix, we will briefly describe the main functionality of these frameworks, analyse the statistical accuracy of the results generated, and briefly mention the external functions utilised to achieve the necessary functionality.

A.1 MmWave-V2X Simulation – SAMBA

Starting with the first framework, in Fig. A.1 the reader can see all the implemented functions. Each subfolder is a separate class, and their unique features will be briefly described in the next sections. At the end of this section, the reader can find the MATLAB code of the primary function of the simulator as a reference for the steps followed.

Before we proceed with the core functionality of the simulation, the external functions and libraries used were:

- Function calculating whether a point is inside a polygon or not[1].

---

[1]https://uk.mathworks.com/matlabcentral/fileexchange/10391-inpoly-a-fast-points-in-polygon-test

Figure A.1: The different functions implemented for the MATLAB MmWave-V2X simulation framework.

- Function for finding the line segment intersections, when given an array of lines[2].

- Function to control the log output verbosity level – this was just for debugging purposes only[3].

- Function that converts an XML file into a MATLAB structure for easy access to the data.[4]

The execution of the simulation framework starts by calling *simSettingsSAMBA* that loads all the simulation parameters needed. Example parameters are the file paths for the map and mobility trace files, the maximum theoretical data rates of the different MCSs, constants used for calculating the propagation loss (e.g. speed of light, pathloss exponent, etc.), the matching capacity requested by the user and many more.

Later, an existing map is loaded and converted into a MATLAB structure (i.e. *loadNetwork* function) or a new one is created using the given simulation parameters (i.e. *mapCreate* function). The supported map formats are either a downloaded map from Open Street Map [186], given in *.osm* format, or a network file from SUMO traffic generator [169], given as a *.net* file. This

---

[2]https://uk.mathworks.com/matlabcentral/fileexchange/27205-fast-line-segment-intersection
[3]https://uk.mathworks.com/matlabcentral/fileexchange/42909-log-output-verbosity-control
[4]https://uk.mathworks.com/matlabcentral/fileexchange/28518-xml2struct

is followed by importing the mobility traces of the vehicles calling *loadVehicles* function. These traces should be generated using SUMO beforehand and exported in a geolocation format. They should also be saved in the appropriate folders (i.e. the folder called *mobilityFiles*).

For the V2I scenarios, *rsuCreate* function finds the upper-right corners of a road-block and places an RSU on each corner. More details about that can be found in Chapter 5. This function works well when we have grid-like maps, and we require a fully covered road layout without the increased complexity of solving an optimisation problem. Later, we will further describe the simulation framework designed to find the optimal RSU positions. Furthermore, if the number of vehicles provided by the trace file is larger than the requested number from the user, the imported traces are post-processed within our simulator, and the exceeding vehicles are discarded. This helps us to quickly test scenarios with a different number of vehicles without the need of generating new mobility traces and maps every time. Finally, the beamwidth adaptation introduced in Chapter 5 is enabled or disabled by the user. When it is enabled, the different beams are pre-calculated given the link budget equations (having an idealised triangular shape), in order to minimise the execution time.

When the initial configuration of the simulator is completed, we start by simulating the different scenarios. We can configure the simulator to run for different position errors and for a given number of times in a Monte Carlo fashion. The entire simulation time (real-world time) is split into equal timeslots. The length of each timeslot is predefined by the user based on the required granularity of the results. On each step, our simulator computes all the data required for the entire simulation time before going to the next step. This was very important, as, by designing our simulator in such fashion, we managed to exploit MATLAB capabilities for matrix multiplications and vectorised data arrays. By that, we significantly decreased the execution time minimising the nested loops and being able to run in parallel the different processes. To speed up the execution time even further, our simulation framework is capable of running in parallel different system configurations in order to exploit the multi-core CPUs found today. Finally, we kept in mind to design a framework that easily readable and modifiable by the user.

Each scenario is initiated by calculating the perceived vehicle positions (*calcGPSPos* function). This depends on the given position error and the position update interval, specified by the user. The position error is chosen from a random distribution and added on the real-world traces loaded from SUMO. This can create more realistic vehicle traces taking into account the GPS errors of the real world. Later, all the DSRC beacons are generated and exchanged for the V2I links (*beaconV2I* function). Their packet delivery ratio, as well as the transmission interval, are predefined by the user. The packets are dropped based on a uniform random distribution. As discussed in Chapter 6, based on the different types of vehicles, they generate a different amount of sensor data per timeslot. This feature is implemented using the *genTraffic* function that is part of the matching game implementation. Again, as before a uniform random distribution is utilised to choose the amount of data generated per vehicle and per timeslot.

The function called as *vehicleDensity* is responsible for calculating the number of vehicles within a region – this notion was introduced in Chapter 6. This is based on the radius provided by the user, and this region is always circle-shaped. As before, function *beaconV2V* calculates the generated and exchanged DSRC beacons for the V2V links, based on the simulation parameters specified by the user. Knowing the number of received beacons, we can later run our algorithm for the motion prediction of a vehicle (functions *predictMovement* and *predictMovementV2V*). This motion prediction is calculated for all beacons exchanged (both V2I and V2V links). For our system design described in Chapter 5 only the V2I implementation was utilised.

The following functions are all related to the Stable Fixture matching game described above. We start with *findLOSLinks* that finds all the LOS links between vehicles for all the timeslots and the given road layout. We later calculate, based on the estimated positions, the closest vehicles that have a LOS link with a target vehicle. Given all the vehicle motion dynamics, we then calculate all the variables constituting our utility function. We start with the *timeliness*, related to the vehicle direction and distance. Later, we estimate the regional data exchanged between vehicles, using the *regionalData* function, based on the above-given data, and the matching capacity for each vehicle (*setMatchingCapacity*). Having all the above, we can later run our fitness function (*fitnessFunction*) for all the given data and start the matching game using *matchGame*. This function returns the matching set of vehicles that is our final solution. Using this solution and the function *linkUtilisation* we can calculate the results shown in Chapter 6.

The above matching game for V2V links is followed by the beamwidth adaptation algorithms introduced in Chapter 5. Given the predicted vehicle's position and motion, we can calculate the shape of the new beams using *updateBeamsSAMBA*. This function aligns the beams for all the V2I links. In our simulation framework, the beams have a triangular shape as discussed before. The same feature was implemented for the V2V links as well, using the function *alignBeamsV2V* but was not used for generated the results of this thesis. For all the above links we later have the performance investigation of the different proposed algorithms. The functions responsible for that are *meanDataRateV2I*, *perfEvaluation*, and *perfEvaluationV2V*. Finally, an animation environment was designed for demonstration and debug purposes (functions *printAnimate* and *printAnimateDebug*) and is called at the end of each configuration, if the higher level of verbosity was chosen by the user. All the simulation results are later saved in a .mat file to be easily accessible in the future. By that, the execution of our simulation framework is concluded.

A.1.1   Statistical Analysis of Simulation Frameworks

All the presented performance metrics in Chapters 5 and 6 are values estimated employing multiple Monte Carlo iterations. To this end, this section will briefly describe the validity of the simulation models.

In the next figures, each error bar identifies a binomial proportion confidence interval,
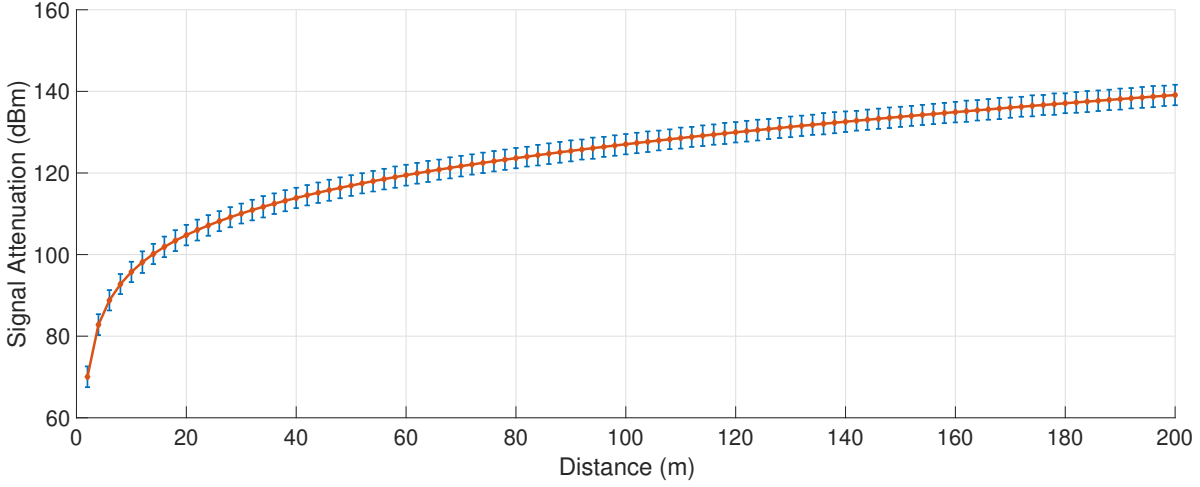
Figure A.2: The signal attenuation as a function of the distance.

approximated by means of a normal approximation and is defined as in [197]:

$$(A.1) \qquad \left[\hat{p} - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \ \ \hat{p} + z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right]$$

where $\hat{p}$ is the estimated probability value, n is the number of Monte Carlo iterations and z is the $(1 - 0.5 \cdot e)$-th quantile, for $0 \leq e \leq 1$. In the updated version of the manuscript, we refer to the $(1 - 0.5 \cdot 0.05)$-th quantile, i.e., we refer to the confidence intervals of 95%.

At first, in Fig. A.2 we present the signal attenuation as a function of the distance. The link budget analysis, the way the antenna gain was calculated, and the relationship with the beamwidth can be found in Secs. 5.4.4 and 6.2.1. The signal attenuation presented is derived by $n = 1000$ Monte Carlo simulations and using the simulation parameters found in Table 5.1. The beamwidth θ^o was set to $\theta = 15°$. The figure shows the mean value calculated per distance as well as the confidence intervals. The statistical error shown originates from the random nature of the shadow fading S_f, introduced in the mmWave channels. As described in Chapter 5, S_f follows a log-Normal distribution $\log(S_f) \sim \mathcal{N}(0, \sigma_{S_f}^2)$, with σ equal to 5.8 [178].

The same error is introduced in the packet delivery rate and consequently in the averaged data rate as well. Using the same large scale evaluation scenario described in Sec. 5.6.2, we generated the confidence intervals of 95% for different velocities and position errors. The results can be found in Fig. A.3. As in Fig. A.2, 1000 Monte Carlo simulations were used to generate this result. This figure presents the average data rate of one vehicle in contrast with the legacy beamforming strategy. Again, as shown the shadow fading introduced in our algorithm introduces a variation in the perceived performance. From both figures, we see that the random nature of signal attenuation slightly changes the results per iteration. However, using a large number of Monte Carlo iterations, like the one used in our performance investigation in Chapters 5 and 6, and averaging the results later, we can have a thorough and valid system investigation.

153
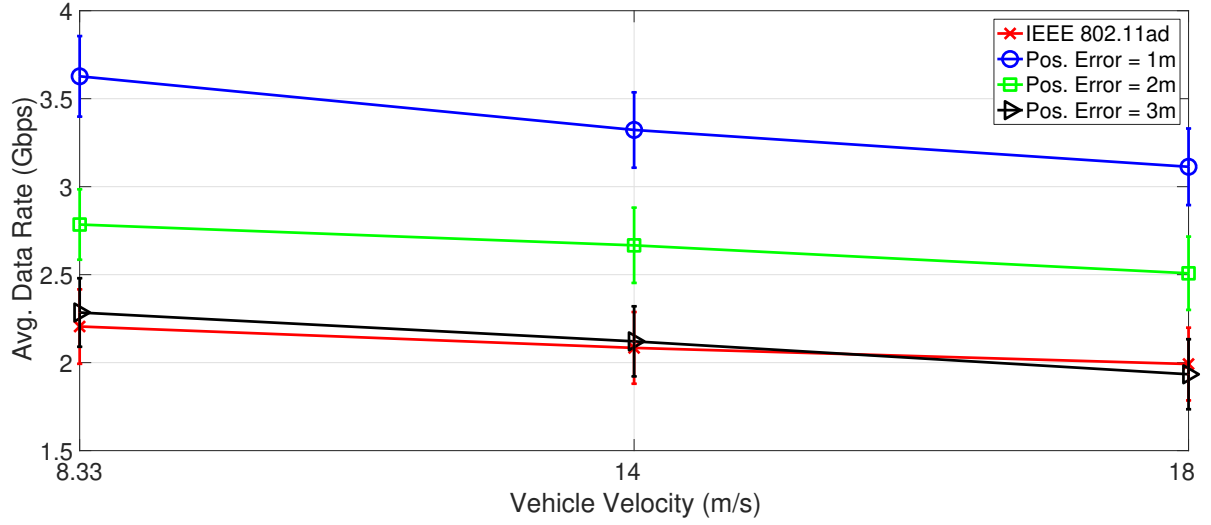
Figure A.3: Achieved average data rate (one vehicle) for SAMBA algorithm in contrast with the legacy beamforming strategy for different velocities. The confidence intervals of 95% are presented for each data rate.

MATLAB Code - runSimSAMBA.m

```matlab
1 % RUNSIMSAMBA Runs the simulation.
2 %   Loads the settings from simSettingsSAMBA.m and runs the simulator.
3 %
4 % Usage: runSimSAMBA
5 %
6 % Copyright (c) 2017-2018, Ioannis Mavromatis
7 % email: ioan.mavromatis@bristol.ac.uk
8
9 clf; clc; clear; close all;
10
11 simSettingsSAMBA; % Load all the simulation settings from the file.
12
13 %% Parse SUMO map from network file and create boundary box
14 SUMOMap = loadFiles.loadNetwork(inputFileMap); % Load the map from the SUMO network file
15 map = maps.mapCreate(map); % keep it for future reference
16
17 %% Parse SUMO mobility model and construct the RSUs on the road
18 [vehiclesArray, vehicleStruct, numTimesteps, numVehiclesPerTimestep, numVehicles] = loadFiles.loadVehicles(inputFile,
        SUMOMap.bBoxForVehicles, numVehicles, startTime);
19 rsuStruct = parsing.rsuCreate(SUMOMap, linkBudget, vehicleStruct);
20
21 for nodeIndex = 1:vehicleStruct.numNodes
22     vehicleStruct.vehNode(nodeIndex).V_TIME = vehicleStruct.vehNode(nodeIndex).V_TIME - numOfVeh;
23     vehicleStruct.vehNode(nodeIndex).V_TIME = round(vehicleStruct.vehNode(nodeIndex).V_TIME,3);
24 end
25
26 vehiclesArray.TIMESTEP = vehiclesArray.TIMESTEP - numOfVeh;
27 vehiclesArray.TIMESTEP = round(vehiclesArray.TIMESTEP,3);
28
29 %% Run the utility function for beamwidth Adaptation or initialise an empty array if beamwidth adaptation is not
        enabled.
30 if beamWidthAdaptation
31     beamAdapt = parsing.beamWidthAdapt(linkBudget);
32     fprintf('The Beamwidth adaptation is enabled.\n');
33 else
```

```
34      beamAdapt = [];
35        fprintf('The Beamwidth adaptation is disabled.\n');
36 end
37
38 %% Run Simulation - Different errors, no. of Iterations
39 for posError = 1:1
40      % For parallel processing to optimise the running time,
41      % change the loop below to "parfor"
42      for iteration = 1:1
43
44          vehicleStructTmp = vehicleStruct;
45          rsuStructTmp = rsuStruct;
46          fprintf('SAMBA: Pos. Error.: %d, PDRratio: %d, Iteration: %d\n',posError, PDRratio, iteration);
47
48          %% Estimate the GPS position based on the given error
49          [ vehicleStructTmp ] = gps.calcGPSPos(vehicleStructTmp,posError,gpsUpdateInterval);
50
51          %% Create and transmit the DSRC beacons
52          [ vehicleStructTmp ] = beacons.beaconV2I(vehicleStructTmp,vehiclesArray,beaconInterval,PDRratio);
53
54          %% Generate the network traffic for each vehicle (per timeslot)
55          [ vehicleStructTmp ] = matchingGame.genTraffic(vehicleStructTmp,genData);
56
57          %% Vehicle Density
58          [ closeVehicles, distanceCloseVeh, timeIndicesVeh, indicesVehTimeslot ] = parsing.vehicleDensity(
                 vehicleStructTmp,densityRadius);
59
60          %% Beacons between vehicles (V2V) are generated and exchanged
61          [ vehicleStructTmp ] = beacons.beaconV2V(vehicleStructTmp,beaconInterval,PDRratio);
62
63          %% Based on beacons and GPS position, predict the movement of the vehicle
64          [ vehicleStructTmp, rsuStructTmp ] = movementPrediction.predictMovement(vehicleStructTmp,rsuStructTmp);
65          [ vehicleStructTmp ] = movementPrediction.predictMovementV2V(vehicleStructTmp,beaconInterval,
                 gpsUpdateInterval);
66
67          %% LOS links between vehicles based on the beacon information
68          [ predictedMovement, linksLOS, vehCombs ] = losLinks.findLOSLinks(vehicleStructTmp,map,SUMOMap,
                 closeVehicles,distanceCloseVeh,1);
69          [ closestVehiclesLOSLinks, linksLOStimestamps, distanceVehicles ] = losLinks.closestVehicle(linksLOS,
                 vehicleStructTmp, closeVehicles, distanceCloseVeh, vehCombs );
70
71          %% Timeliness / Direction Comparison
72          [ vehicleStructTmp ] = matchingGame.timeliness(vehicleStructTmp);
73
74          %% Regional Data Generated for each vehicle
75          [ vehicleStructTmp ] = matchingGame.regionalData(vehicleStructTmp,closeVehicles);
76
77          %% Matching Game for all Vehicle
78          [ matchingCapacity ] = matchingGame.setMatchingCapacity(matchingCapacity,vehicleStructTmp);
79          [ utility,calcDataRate ] = matchingGame.fitnessFunction(vehicleStructTmp,linksLOS,closeVehicles,
                 distanceCloseVeh,linkBudget, MCS, utilFuncMode,densityRadius);
80          [ matchingSet, matchingSetProcessed ] = matchingGame.matchGame( matchingCapacity,utility);
81
82          % Results of the Matching Game
83          [ linkUtil, dataRate, regDataAccess, emerVehLUtil, emerVehDataRate, regDataAccessEmer ] = matchingGame.
                 linkUtilisation(vehicleStructTmp, calcDataRate, matchingSetProcessed);
84          linkUtilAll(iteration,:) = linkUtil;
85          emerVehLUtilAll(iteration,:) = emerVehLUtil;
86          linkUtilAllData(iteration,:) = dataRate;
87          emerVehLUtilAllData(iteration,:) = emerVehDataRate;
88          regDataAccessAll(iteration,:) = regDataAccess;
89          regDataAccessEmerAll(iteration,:) = regDataAccessEmer;
90          %% Realign the beams
91          [ vehicleStructTmp, rsuStructTmp ] = beamAlign.updateBeamsSAMBA(vehicleStructTmp,rsuStructTmp,linkBudget,
                 beamAdapt);
```

155

```
92        [ beamsV2V, predictPosMatrix ,closestVehiclesLOSLinks ] = beamAlign.alignBeamsV2V(predictedMovement,
                closestVehiclesLOSLinks ,linksLOStimestamps ,linkBudget ,vehicleStructTmp);
93
94        %% Evaluate the system performance
95        [ vehicleStructTmp ] = performance.perfEvaluation(vehicleStructTmp , rsuStructTmp , linkBudget , MCS, 0);
96        [ networkThroughputV2V{iteration} ] = performance.perfEvaluationV2V(beamsV2V, distanceVehicles ,
                vehicleStructTmp , closestVehiclesLOSLinks , linksLOStimestamps , linkBudget , MCS, 0);
97
98        netThroughputPerIterV2I(iteration) = performance.meanDataRateV2I(vehicleStructTmp);
99        netThroughputPerIterV2V(iteration) = mean(networkThroughputV2V{iteration});
100
101       animate.printAnimate(vehicleStructTmp ,rsuStructTmp ,SUMOMap,beamsV2V ,predictPosMatrix);
102       animate.printAnimateDebug(vehicleStructTmp ,rsuStructTmp ,SUMOMap,beamsV2V ,predictPosMatrix ,2100);
103    end
104    meanNetThroughputV2I(posError) = mean(netThroughputPerIterV2I)/1000000000;
105    meanNetThroughputV2V(posError) = mean(netThroughputPerIterV2V)/1000000000;
106
107 end
108
109 linkUtilAll(linkUtilAll >1)=1;
110 linkUtilAll(isnan(linkUtilAll))=0;
111 meanLinkUtil = mean(linkUtilAll ,1);
112
113 % Print the Results
114 for posEr=1:posError
115     fprintf('Mean Network Throughput for pos. error %d: 1) V2I – %d, 2) V2V – %d\n', posEr , meanNetThroughputV2I(
                posEr), meanNetThroughputV2V(posEr) )
116 end
117
118 % Save the workspace in a MAT file.
119 [~, nameVehicles, ~] = fileparts(inputFile);
120 filenameMAT = ['results/samba_mCap_' num2str(matchingCapacity(1)) '_dRad_' num2str(densityRadius) '_uFunc_'
                num2str(utilFuncMode) '_' nameVehicles '.mat'];
121 save(filenameMAT)
```

A.2   RSU MmWave-Placement Simulation Framework

This section will briefly describe the functionality of the second simulation framework designed and implemented. Again, as before this framework was designed in MATLAB. In Fig. A.4 the reader can see all the implemented functions. This framework was used to evaluate the performance of our proposed algorithm, described in Chapter 7, and compare it against two more optimisation algorithms. Again, at the end of this section, the reader can find the MATLAB code of the main function of the simulator as a reference for the steps followed. Before we proceed with the core functionality, we will present the external functions and libraries used. Some of them were presented in Sec A.1. The rest are:

- Function that converts latitude and longitude coordinates into UTM (WGS84)[5].

- Fecursive function implementing Douglas-Peucker simplification algorithm – used to simplify the map layout[6].

- Function that calculates the Hausdorff Distance between two points[7].

---

[5]https://uk.mathworks.com/matlabcentral/fileexchange/10915-deg2utm
[6]https://uk.mathworks.com/matlabcentral/fileexchange/21132-line-simplification
[7]https://uk.mathworks.com/matlabcentral/fileexchange/27905-hausdorff-distance

- The MATLAB geometrical toolbox library for 2D/3D geometric computing[8].

- Library containing all the functions for working with OpenStreetMap XML Data files[9].

This simulator was designed in order to be able to provide the necessary granularity to the user. Using a small map and a large tile size, the user can easily and quickly test a new feature implemented. Increasing the map size and decreasing the tile size, the precision can be enhanced and provide a more real-world like solution. As before the user executes the main file (*runSimulator.m*) that initiates the simulation. Again, different parameters are predefined in the *simSettings* file (e.g., tile size, map size, distance threshold, RSS threshold, etc.). Later, the resources for different modules of the simulation framework are preallocated in the system using *setupSimulator* function. When the setup is over, the main part of the simulator is executed, responsible for finding the optimum positions for the RSUs using three different algorithms.

That starts by loading the map file. As before, the map could be either an OpenStreetMap file or a SUMO network file. Later, based on the configuration parameters, the different map regions, as well as the tiles of the map, are calculated (*breakMapSmallerTiles* function). Furthermore, the potential RSU positions (function *potentialRSUPositions*) are found based on the imported map. As described in Chapter 7 the potential positions are the corners or the roads at first and are equally spread on long roads. The given map is later divided into tiles and map regions using the function *mapTiles*. Based on all the above, we can later start our performance evaluation.

At first we find all the tiles that are within the requested area of interest (functions *withinMap* and *tilesWithinEdge*). As described, this is done to avoid the boundary effects. For all the potential RSU positions, we find the tiles that have LOS with each position and calculate their RSS (function *losMapTiles*). Having all the above information we can later solve our optimisation problem. Three different functions were designed for each optimisation algorithm introduced, i.e. *gaSolver*, *autoOptimisation*, and *greedyAddition*. These functions find the best positions to deploy a mmWave RSU basestation as described in Chapter 7. The simulation concludes with the generation of all the result figures and the heatmaps as well as saving the simulation variables for future reference.

MATLAB Code - runSimulator.m

```matlab
1 % RUNSIMULATOR Runs the simulation.
2 %   Loads the settings from simSettings.m and runs the simulator.
3 %
4 % Usage: runSimulation
5 %
6 % Copyright (c) 2017-2018, Ioannis Mavromatis
7 % email: ioan.mavromatis@bristol.ac.uk
8
9 clc; clf; clear; clear global; close all;
10
```

---

[8]https://github.com/mattools/matGeom
[9]https://uk.mathworks.com/matlabcentral/fileexchange/35819-openstreetmap-functions
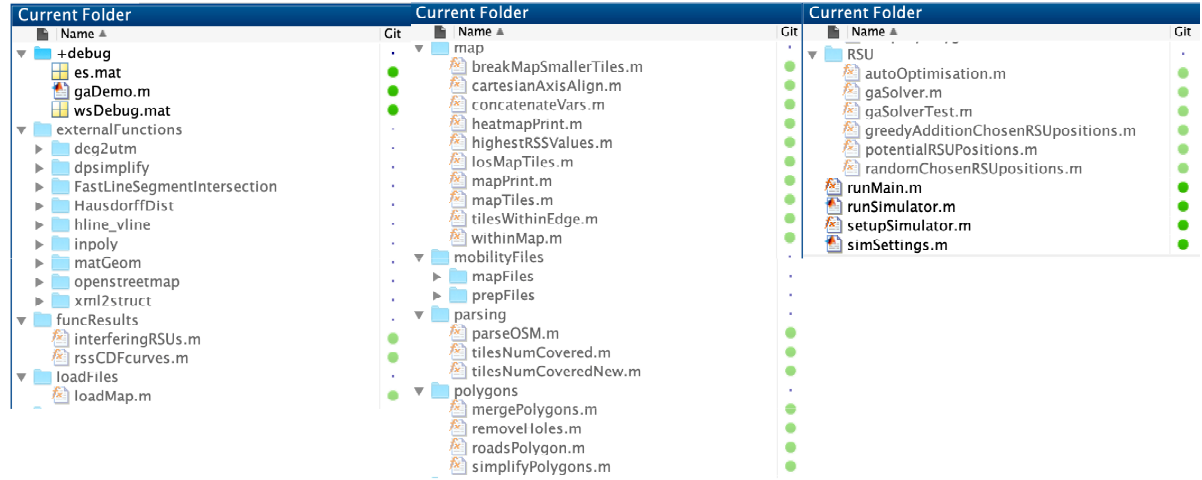
Figure A.4: The different functions implemented for the MATLAB RSU MmWave-Placement simulation framework.

```matlab
11 fprintf('V2X MmWave Simulator for City-Scale Simulations\n');
12 fprintf('Automated Road-Side Unit Placement for City-Scale Maps\n');
13 fprintf('Copyright (c) 2017-2018, Ioannis Mavromatis\n');
14 fprintf('email: ioan.mavromatis@bristol.ac.uk\n\n');
15
16 % Load the simulation settings (for further explanation see simSettings.m)
17 simSettings;
18
19 % Add the different modules of the simulator to MATLAB path
20 setupSimulator(simulator);
21
22 % Main function (for further explanation see runMain.m)
23 runMain(map, RSU, linkBudget, simulator);
```

MATLAB Code - runMain.m

```matlab
1 function runMain(map, RSU, linkBudget, simulator)
2 %RUNMAIN This is the main function of RSU mmWave placement simulation framework
3
4 % Copyright (c) 2017-2018, Ioannis Mavromatis
5 % email: ioan.mavromatis@bristol.ac.uk
6
7     % Parse the map the file
8     outputMap = loadMap(map,simulator);
9
10    if isempty(outputMap)
11        fprintf('No network file was loaded!')
12        return;
13    end
14
15    [ ~, nameMap, ~ ] = fileparts(map.mapFile);
16    outputMap = breakMapSmallerTiles(outputMap,map);
17    potRSUPos = potentialRSUPositions(outputMap,RSU,simulator);
18
19    [ mapTileIncentres, outputMap ] = mapTiles(outputMap,map,simulator);
20
21    if simulator.verboseLevel == 2
22        mapPrint(outputMap)
23        alpha(0.5)
24        hold on
```

```
25      end
26      [toRun,~] = size(outputMap.mapToRunX);
27
28      pathLos = [ './mobilityFiles/prepFiles/losRSSTiles/' nameMap '_' num2str(map.tileSize) ];
29      if exist([ pathLos '.mat' ],'file')
30          load([ pathLos '.mat' ], '-mat')
31      end
32      if exist('losTileIDs','var')
33          dontRun = 0;
34      else
35          dontRun = 1;
36      end
37
38      for i = 1:toRun
39          [ mapTileIncentresTmp{i},potRSUPosTmp{i},tileIDs{i} ] = withinMap(outputMap,mapTileIncentres,potRSUPos,i);
40          [ mapTileWithinEdge{i}, tileWithinEdgeIDs{i} ] = tilesWithinEdge(mapTileIncentresTmp{i},outputMap,map,
                  simulator,i);
41          [ losTileIDs{i}, rssTile{i}, losTileIDsWithinNoiseFloor{i}, perTileBSIDs{i}, perTileRSS{i} ] = losMapTiles
                  (outputMap,mapTileIncentresTmp{i},potRSUPosTmp{i},RSU,linkBudget,simulator);
42
43          [ choseRSUposGA{i},tilesCoveredGA{i},tilesCoveredGAIDs{i},highestRSSGA{i} ] =...
44              gaSolver(RSU,potRSUPosTmp{i},losTileIDs{i},rssTile{i},tileWithinEdgeIDs{i});
45          [ choseRSUposOpt{i},tilesCoveredOpt{i},tilesCoveredOptIDs{i},highestRSSOpt{i} ] =...
46              autoOptimisation(RSU,potRSUPosTmp{i},losTileIDs{i},rssTile{i},tileWithinEdgeIDs{i},simulator);
47          [ chosenRSUposGC{i},tilesCoveredGC{i},tilesCoveredGCIDs{i},highestRSSGC{i} ] =...
48              greedyAddition(potRSUPosTmp{i},losTileIDs{i},RSU,rssTile{i},tileWithinEdgeIDs{i},simulator);
49
50          if simulator.verboseLevel == 2
51              plot(potRSUPosTmp{i}(choseRSUposOpt{i},2),potRSUPosTmp{i}(choseRSUposOpt{i},1),'yo','MarkerSize',10);
52              hold on
53          end
54      end
55
56      [potRSUPosAll,chosenRSUposOptAll,chosenRSUposRndAll,chosenRSUposGCAll,...
57              tilesCoveredOptIDsAll,tilesCoveredGCIDsAll,tilesCoveredRndIDsAll,...
58              losTilesIDsAll, rssTileAll,tileWithinEdgeIDsAll,mapTileIncentresAll,...
59              mapTilesIDsToCheck,highestRSSOptAll,highestRSSGCAll,highestRSSRndAll,...
60              highestRSSGAAll,tilesCoveredGAIDsAll,chosenRSUposGAAll]...
61              = concatenateVars(toRun,mapTileIncentresTmp,...
62              potRSUPosTmp,choseRSUposGA,choseRSUposOpt,chosenRSUposGC,chosenRSUposRnd,...
63              tilesCoveredGAIDs,tilesCoveredOptIDs,tilesCoveredGCIDs,tilesCoveredRndIDs,...
64              losTileIDs,rssTile,tileWithinEdgeIDs,tileIDs,mapTileIncentres,...
65              highestRSSOpt,highestRSSGC,highestRSSRnd,highestRSSGA);
66
67      %% Plot all the results
68       [ servingRSUsGC,servingRSUsRand,servingRSUsOpt, statsGC,...
69          statsNormGC, statsRand, statsNormRand, statsOpt, statsNormOpt ] =...
70          interferingRSUs(perTileBSIDs,tileWithinEdgeIDs,chosenRSUposGC,chosenRSUposRnd,choseRSUposOpt,simulator);
71
72      rssCDFcurves(highestRSSGCAll,highestRSSRndAll,highestRSSOptAll,simulator)
73
74      %% Plot all the maps with the deployed basestations
75      if (simulator.verboseLevel >= 1)
76          h = findobj('type','figure'); % Get the number of figures open for results
77
78          %% Greedy Addition Algorithm Plot
79          h(length(h)+1) = figure(length(h)+1);
80          heatmapPrint(outputMap,mapTileIncentres,mapTilesIDsToCheck,map,tilesCoveredGCIDsAll,highestRSSGCAll,
                  tileWithinEdgeIDsAll,chosenRSUposGCAll,potRSUPosAll);
81          title("Greedy Addition");
82
83          %% Random Algorithm Plot
84          h(length(h)+1) = figure(length(h)+1);
85          heatmapPrint(outputMap,mapTileIncentres,mapTilesIDsToCheck,map,tilesCoveredRndIDsAll,highestRSSRndAll,
                  tileWithinEdgeIDsAll,chosenRSUposRndAll,potRSUPosAll);
```

159

```matlab
86            title("Random Placement");
87
88        %% Optimised Algorithm Plot
89        h(length(h)+1) = figure(length(h)+1);
90        heatmapPrint(outputMap,mapTileIncentres,mapTilesIDsToCheck,map,tilesCoveredOptIDsAll,highestRSSOptAll,
                 tileWithinEdgeIDsAll,chosenRSUposOptAll,potRSUPosAll);
91            title("Optimised Placement");
92    end
93
94    %% Save Results
95    path = [ './results/' nameMap ];
96    save(path, '-regexp', '^(?!(h)$).');
97    save([ pathLos '.mat' ], '-mat','losTileIDs','rssTile')
98    savefig(h, [ './results/' nameMap '_figures.fig' ])
99
100 end
```

BIBLIOGRAPHY

[1]   P. M. d'Orey and M. Ferreira, "ITS for Sustainable Mobility: A Survey on Applications and Impact Assessment Tools," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 477–493, Apr. 2014. doi: 10.1109/TITS.2013.2287257

[2]   "Autonomous Road Vehicles - PostNote," Tech. Rep. 443, Sep. 2013. [Online]. Available: http://researchbriefings.parliament.uk/ResearchBriefing/Summary/POST-PN-443

[3]   "Xiph.org Video Test Media [derf's collection]," accessed: 2018-09-30. [Online]. Available: https://media.xiph.org/video/derf/

[4]   F. Martínez, A. J. Rueda, and F. R. Feito, "A New Algorithm for Computing Boolean Operations on Polygons," *Computers & Geosciences*, vol. 35, no. 6, pp. 1177 – 1185, 2009. doi: https://doi.org/10.1016/j.cageo.2008.08.009

[5]   Thomson Reuters, "Connected Car Data: Let the Data Flow," [Online; accessed on 30 June 2018]. Available: https://blogs.thomsonreuters.com/answerson/connected-car-data-open-data/.

[6]   M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1617–1655, Sep. 2016. doi: 10.1109/COMST.2016.2532458

[7]   UK association for Intelligent Transportation Systems, "Intelligent Transport Systems (ITSs) and their Benefits," *Whitepaper*, Feb. 2017, [Online]. Available: http://its-uk.org.uk/wp-content/uploads/2017/02/ITS-UK-Benefits-of-ITS.pdf.

[8]   M. Weiser, "The Computer for the 21st Century," *SIGMOBILE Mobile Computing and Communications Review*, vol. 3, no. 3, pp. 3–11, Jul. 1999. doi: 10.1145/329124.329126

[9]   "ERTICO - ITS Europe: Shaping Intelligent Mobility for Europe Together," [Online; accessed on 30 June 2018]. Available: http://ertico.com/.

[10]   "Autopilot Project - Automated Driving Progressed by Internet of Things," [Online; accessed on 30 June 2018]. Available: https://autopilot-project.eu/.

[11] "ADASIS Project - Advancing map-enhanced driver assistance systems," [Online; accessed on 30 June 2018]. Available: https://adasis.org/.

[12] "Newcastle Railway Project," [Online; accessed on 30 June 2018]. Available: https://www.ncl.ac.uk/newrail/.

[13] "European Truck Platooning Project," [Online; accessed on 30 June 2018]. Available: https://eutruckplatooning.com/default.aspx.

[14] "FABRIC - Feasibility analysis and development of on-road charging solutions for future electric vehicles," [Online; accessed on 30 June 2018]. Available: https://www.fabric-project.eu/.

[15] "Intelligent Transportation Society of America," [Online; accessed on 30 June 2018]. Available: https://www.itsa.org/.

[16] "Intelligent Transportation Society of Asia and Japan," [Online; accessed on 30 June 2018]. Available: http://www.its-jp.org/english/.

[17] "VENTURER Project," [Online; accessed on 30 June 2018]. Available: http://www.venturer-cars.com/.

[18] "Flourish Project," [Online; accessed on 30 June 2018]. Available: http://www.flourishmobility.com/.

[19] H. Hartenstein and L. P. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, Jun. 2008. doi: 10.1109/M-COM.2008.4539481

[20] 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," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 584–616, Apr. 2011. doi: 10.1109/SURV.2011.061411.00019

[21] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of Network and Computer Applications*, vol. 37, pp. 380 – 392, Feb. 2014. doi: https://doi.org/10.1016/j.jnca.2013.02.036. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S108480451300074X

[22] "Global Market Insight Report for Smart Autonomous Markets," [Online; accessed on 30 June 2018]. Available: https://globenewswire.com/news-release/2018/04/18/1480664/0/en/Smart-Transportation-Market-to-surpass-130bn-by-2024-Global-Market-Insights-Inc.html.

[23] E. Hollnagel, *Barriers and Accident Prevention*. Routledge, 2004, vol. 895.

[24] J. Cheng, L. Li, H. Li, and F. Wang, "SAR Target Recognition based on Improved Joint Sparse Representation," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 87, Jun 2014. doi: 10.1186/1687-6180-2014-87

[25] M. Bijelic, T. Gruber, and W. Ritter, "A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down?" in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 760–767. doi: 10.1109/IVS.2018.8500543

[26] B. K. Kim and Y. Sumi, "Performance evaluation of safety sensors in the indoor fog chamber," in *2017 IEEE Underwater Technology (UT)*, Feb. 2017, pp. 1–3. doi: 10.1109/UT.2017.7890317

[27] J. Glancy, Dorothy, "Autonomous and automated and connected cars—oh my! first generation autonomous cars in the legal ecosystem," *Journal of Law, Science and Technology*, vol. 16, no. 2, pp. 619–691, 2015.

[28] "World's Population Increasingly Urban with more than Half Living in Urban Areas," The United Nations, Tech. Rep., Jul. 2014. [Online]. Available: http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html

[29] A. Palazzi, G. Borghi, D. Abati, S. Calderara, and R. Cucchiara, "Learning to Map Vehicles into Bird's Eye View," in *Proceedings of 2017 International Conference on Image Analysis and Processing (ICIAP)*, S. Battiato, G. Gallo, R. Schettini, and F. Stanco, Eds., 2017, pp. 233–243.

[30] "IEEE 802.11p Standard - IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std. 802.11p-2010*, Jul. 2010. doi: 10.1109/IEEESTD.2010.5514475

[31] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, Apr. 1980. doi: 10.1109/TCOM.1980.1094702

[32] "INET Framework," https://inet.omnetpp.org/, 2018, [Online; accessed 24-August-2018].

[33] I. Mavromatis, A. Tassi, G. Rigazzi, R. J. Piechocki, and A. Nix, "Multi-Radio 5G Architecture for Connected and Autonomous Vehicles: Application and Design Insights," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems 2018*, vol. 5, no. 15, Jul. 2018.

[34] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Agile calibration process of full-stack simulation frameworks for V2X communications," in *Proceedings of 2017 IEEE Vehicular Networking Conference (VNC)*, Nov. 2017, pp. 89–96. doi: 10.1109/VNC.2017.8275604

[35] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "A City-Scale ITS-G5 Network for Next-Generation Intelligent Transportation Systems: Design Insights and Challenges," in *Proceedings of 2018 International Conference on Ad Hoc Networks and Wireless – AdHoc-Now 2018*, Sep. 2018, pp. 53–63. doi: 10.1007/978-3-030-00247-3_5

[36] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Beam Alignment for Millimetre Wave Links with Motion Prediction of Autonomous Vehicles," in *Proceedings of 2017 International Conference in Antennas, Propagation & RF Technology for Transport and Autonomous Platforms*, Feb. 2017, pp. 1–8. doi: 10.1049/ic.2017.0023

[37] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "MmWave System for Future ITS: A MAC-Layer Approach for V2X Beam Steering," in *Proceeding of 2017 IEEE Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–6. doi: 10.1109/VTC-Fall.2017.8288267

[38] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Efficient V2V Communication Scheme for 5G mmWave Hyper-Connected CAVs," in *Proceedings of 2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6. doi: 10.1109/ICCW.2018.8403780

[39] J. Harri, F. Filali, and C. Bonnet, "Mobility Models for Vehicular Ad-Hoc Networks: A Survey and Taxonomy," *IEEE Communications Surveys Tutorials*, vol. 11, no. 4, pp. 19–41, Dec. 2009. doi: 10.1109/SURV.2009.090403

[40] B. E. Bilgin and V. C. Gungor, "Performance Comparison of IEEE 802.11p and IEEE 802.11b for Vehicle-to-Vehicle Communications in Highway, Rural, and Urban Areas," *International Journal of Vehicular Technology*, vol. 2013, pp. 1–10, Sep. 2013.

[41] F. J. Ros, J. A. Martinez, and P. M. Ruiz, "A survey on modelling and simulation of vehicular networks: Communications, mobility, and tools," *Computer Communications*, vol. 43, pp. 1–15, May 2014. doi: https://doi.org/10.1016/j.comcom.2014.01.010

[42] "Electric Vehicles Batteries Lifetime," [Online; accessed on 30 June 2018]. Available: https://cleantechnica.com/2016/05/31/battery-lifetime-long-can-electric-vehicle-batteries-last/.

[43] "Nvidia's New Supercomputer Will Enable the Highest Level of Automated Driving," [Online; accessed on 30 June 2018]. Available: https://www.theverge.com/2017/10/10/16449416/nvidia-pegasus-self-driving-car-ai-robotaxi.

[44]   C. Huang, R. Lu, and K. K. R. Choo, "Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, Nov. 2017. doi: 10.1109/MCOM.2017.1700322

[45]   Hughes, William J., "Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report," FAA GPS Performance Analysis Report, Tech. Rep. 94, July 2016.

[46]   K. A. Ahmad, M. Sahmoudi, and C. Macabiau, "Characterization of GNSS Receiver Position Errors for User Integrity Monitoring in Urban Environments," in *ENC-GNSS 2014*, 04 2014.

[47]   G. MacGougan, G. Lachapelle, R. Klukas, K. Siu, L. Garin, J. Shewfelt, and G. Cox, "Performance analysis of a stand-alone high-sensitivity receiver," *GPS Solutions*, vol. 6, no. 3, pp. 179–195, Dec. 2002. doi: 10.1007/s10291-002-0029-z

[48]   S. Zhao, Y. Chen, and J. A. Farrell, "High-Precision Vehicle Navigation in Urban Environments Using an MEM's IMU and Single-Frequency GPS Receiver," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2854–2867, Oct. 2016. doi: 10.1109/TITS.2016.2529000

[49]   D. C. Salmon and D. M. Bevly, "An Exploration of Low-Cost Sensor and Vehicle Model Solutions for Ground Vehicle navigation," in *Proceedings of 2014 IEEE/ION Position, Location and Navigation Symposium - PLANS*, May 2014, pp. 462–471. doi: 10.1109/PLANS.2014.6851404

[50]   D. Gingras, "Data Fusion Techniques and Positioning Estimation for Land Vehicle Navigation Systems: An Overview," in *Automotive Informatics and Communicative Systems: Principles in Vehicular Networks and Data Exchange*.   IGI Global, Jun. 2009, ch. 12, pp. 219–246.

[51]   N. Rola and S. Razvan, "Quality of Service Provisioning in Wireless Vehicular Networks: Challenges and Mechanisms," in *Wireless Vehicular Networks for Car Collision Avoidance*.   Springer New York, May 2013, pp. 37–69.

[52]   N. Bouchemal, "Quality of Service Provisioning and Performance Analysis in Vehicular Network," Ph.D. dissertation, Universite de Versailles-Saint Quentin en Yvelines, 2015.

[53]   K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2377–2396, Jun. 2015. doi: 10.1109/COMST.2015.2440103

[54] P. Fernandes and U. Nunes, "Platooning of Autonomous Vehicles with Intervehicle Communications in SUMO Traffic Simulator," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 1313–1318. doi: 10.1109/ITSC.2010.5625277

[55] C. Campolo, A. Molinaro, G. Araniti, and A. O. Berthet, "Better Platooning Control Toward Autonomous Driving : An LTE Device-to-Device Communications Strategy That Meets Ultralow Latency Requirements," *IEEE Vehicular Technology Magazine*, vol. 12, no. 1, pp. 30–38, Mar. 2017. doi: 10.1109/MVT.2016.2632418

[56] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions," *CoRR*, vol. abs/1708.02562, May 2017. [Online]. Available: http://arxiv.org/abs/1708.02562

[57] M. Cartwright and L. Knoop, "C-its roadmap for european cities," Tech. Rep., Nov. 2016. [Online]. Available: http://cimec-project.eu/wp-content/uploads/2016/12/CIMEC_D3.2-Draft-Roadmap.pdf

[58] "IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture," *IEEE Std 1609.0-2013*, pp. 1–78, Mar. 2014. doi: 10.1109/IEEESTD.2014.6755433

[59] "Car 2 Car - Communication Consortium: Car 2 Car - Communication Consortium: Mission & Objectives," [Online; accessed on 30 June 2018]. Available: https://www.car-2-car.org/index.php?id=5.

[60] M. Saggi and K. Sandhu, "A Survey of Vehicular Ad Hoc network on Attacks & Security Threats in VANETs," in *Proceedings of International Conference on Research and Innovations in Engineering and Technology (ICRIET 2014)*, Dec. 2014, pp. 1–8.

[61] X. Chen and D. Yao, "An Empirically Comparative Analysis of 802.11n and 802.11p Performances in CVIS," in *Proceedings of 2012 International Conference on ITS Telecommunications*, Nov. 2012, pp. 848–851. doi: 10.1109/ITST.2012.6425303

[62] R. Bhakthavathsalam and S. Nayak, "Operational Inferences on VANETs in 802.16e and 802.11p with Improved Performance by Congestion Alert," in *Proceedings of 2011 IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2011, pp. 467–471. doi: 10.1109/CCNC.2011.5766514

[63] I. C. Msadaa, P. Cataldi, and F. Filali, "A Comparative Study between 802.11p and Mobile WiMAX-based V2I Communication Networks," in *Proceedings of 2010 International Conference on Next Generation Mobile Applications, Services and Technologies*, Jul. 2010, pp. 186–191. doi: 10.1109/NGMAST.2010.45

[64] H. Sawant, J. Tan, Q. Yang, and Q. Wang, "Using Bluetooth and Sensor Networks for Intelligent Transportation Systems," in *Proceedings of 2004 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2004, pp. 767–772. doi: 10.1109/ITSC.2004.1398999

[65] T. Ohyama, S. Nakabayashi, Y. Shiraki, and K. Tokuda, "A Study of Real-Time and Autonomous Decentralized DSRC System for Inter-Vehicle Communications," in *Proceedings of 2000 IEEE Intelligent Transportation Systems (ITSC)*, Oct. 2000, pp. 190–195.

[66] V. Va, X. Zhang, and R. W. Heath, "Beam Switching for Millimeter Wave Communication to Support High Speed Trains," in *Proceedings of 2015 IEEE Vehicular Technology Conference (VTC-Fall)*, Sep. 2015, pp. 1–5. doi: 10.1109/VTCFall.2015.7390855

[67] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Beam Design for Beam Switching Based Millimeter Wave Vehicle-to-Infrastructure Communications," in *Proceedings of 2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6. doi: 10.1109/ICC.2016.7511414

[68] N. Dreyer, A. Moller, Z. H. Mir, F. Filali, and T. Kurner, "A Data Traffic Steering Algorithm for IEEE 802.11p/LTE Hybrid Vehicular Networks," in *Proceedings of 2016 IEEE Vehicular Technology Conference (VTC-Fall)*, Sep. 2016, pp. 1–6. doi: 10.1109/VTC-Fall.2016.7880850

[69] E. G. Strom, "On Medium Access and Physical Layer Standards for Cooperative Intelligent Transport Systems in Europe," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1183–1188, Jul. 2011. doi: 10.1109/JPROC.2011.2136310

[70] "LTE-V2X and IEEE802.11p/ITS-G5 Spectrum Sharing at 5.9 GHz," Tech. Rep., Nov. 2017. [Online]. Available: goo.gl/aZg5gT

[71] R. Molina-Masegosa and J. Gozalvez, "LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range Vehicle-to-Everything Communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, Dec. 2017. doi: 10.1109/MVT.2017.2752798

[72] K. Lee, J. Kim, Y. Park, H. Wang, and D. Hong, "Latency of Cellular-Based V2X: Perspectives on TTI-Proportional Latency and TTI-Independent Latency," *IEEE Access*, vol. 5, pp. 15 800–15 809, Jul. 2017.

[73] A. Ghosh, T. A. Thomas, M. C. Cudak, R. Ratasuk, P. Moorut, F. W. Vook, T. S. Rappaport, G. R. MacCartney, S. Sun, and S. Nie, "Millimeter-Wave Enhanced Local Area Systems: A High-Data-Rate Approach for Future Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1152–1163, Jun. 2014. doi: 10.1109/JSAC.2014.2328111

[74]   E. Uhlemann, "Introducing Connected Vehicles," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 23–31, Mar. 2015. doi: 10.1109/MVT.2015.2390920

[75]   M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen, "From LTE to 5G for Connected Mobility," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 156–162, Mar. 2017. doi: 10.1109/MCOM.2017.1600778CM

[76]   T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: Directional 60 GHz Communication For Multi-Gigabit-per-Second Wi-Fi," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, Dec. 2014. doi: 10.1109/M-COM.2014.6979964

[77]   Y. Ghasempour, C. R. C. M. da Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11ay: Next-Generation 60 GHz Communication for 100 Gb/s Wi-Fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, Dec. 2017. doi: 10.1109/MCOM.2017.1700393

[78]   3GPP, "3GPP Release 15," 3rd Generation Partnership Project (3GPP), Tech. Rep., Jun. 2017, version 15.

[79]   J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-Wave Vehicular Communication to Support Massive Automotive Sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, Dec. 2016. doi: 10.1109/M-COM.2016.1600071CM

[80]   A. Tassi, M. Egan, R. J. Piechocki, and A. Nix, "Wireless vehicular networks in emergencies: A single frequency network approach," in *Proceedings of 2017 International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*, Jan. 2017, pp. 27–32. doi: 10.1109/SIGTELCOM.2017.7849790

[81]   V. Vukadinovic, K. Bakowski, P. Marsch, I. Dexter Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. , and I. Thibault, "3GPP C-V2X and IEEE 802.11p for Vehicle-to-Vehicle Communications in Highway Platooning Scenarios," *Ad Hoc Networks*, vol. 74, 03 2018. doi: 10.1016/j.adhoc.2018.03.004

[82]   G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for Vehicular Networking: A Survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013. doi: 10.1109/MCOM.2013.6515060

[83]   "C-ITS Platform," European Commission, Tech. Rep., Jan. 2016. [Online]. Available: http://ec.europa.eu/transport/themes/its/doc/c-its-platform-final-report-january-2016.pdf

[84]   A. Tassi, M. Egan, R. J. Piechocki, and A. Nix, "Modeling and Design of Millimeter-Wave Networks for Highway Vehicular Communication," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, Aug. 2017. doi: 10.1109/TVT.2017.2734684

[85] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *Proceedings of 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 6–9. doi: 10.1109/PERCOMW.2017.7917508

[86] J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 122–128, Dec. 2015. doi: 10.1109/MWC.2015.7368833

[87] S. Bitam and A. Mellouk, "ITS-cloud: Cloud computing for Intelligent transportation system," in *Proceedings of 2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 2054–2059. doi: 10.1109/GLOCOM.2012.6503418

[88] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, and N. Kumar, "Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing," *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 55–64, Sep. 2017. doi: 10.1109/MVT.2017.2667499

[89] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. Uusitalo, B. Timus, and M. Fallgren, "Scenarios for 5G mobile and wireless communications: The Vision of the METIS Project," *IEEE Communication Magazine*, vol. 52, no. 5, pp. 26–35, May 2014. doi: 10.1109/MCOM.2014.6815890

[90] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: architecture and challenges," *IEEE Network*, vol. 30, no. 4, pp. 72–80, Jul. 2016. doi: 10.1109/MNET.2016.7513867

[91] J. Li, X. Zhang, S. Wang, and W. Wang, "Context-oriented multi-RAT user association and resource allocation with triple decision in 5G heterogeneous networks," *China Communications*, vol. 15, no. 4, pp. 72–85, Apr. 2018. doi: 10.1109/CC.2018.8357702

[92] "ETSI Intelligent Transport Systems (ITS) Communications Architecture," *ETSI - EN 302 665*, pp. 1–44, 2010.

[93] "Intelligent Transport Systems – Communications Access for Land Mobiles (CALM) – Architecture," *ISO 21217:2014*, pp. 1–50, Mar. 2014.

[94] P. Demestichas, A. Georgakopoulos, K. Tsagkaris, and S. Kotrotsos, "Intelligent 5G Networks: Managing 5G Wireless/Mobile Broadband," *IEEE Vehicular Technology Magazine*, vol. 10, no. 3, pp. 41–50, Sep. 2015. doi: 10.1109/MVT.2015.2446419

[95] C. Sommer, O. K. Tonguz, and F. Dressler, "Adaptive Beaconing for Delay-Sensitive and Congestion-aware Traffic Information Systems," in *Proceedings of 2010 IEEE Vehicular Networking Conference*, Dec. 2010, pp. 1–8. doi: 10.1109/VNC.2010.5698242

[96] C. Han, M. Dianati, and M. Nekovee, "Effective Decentralised Segmentation-based Scheme for Broadcast in Large-scale Dense VANETs," in *Proceedings of 2016 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2016, pp. 1–6.

[97] F. A. Teixeira, V. F. e Silva, J. L. Leoni, D. F. Macedo, and J. M. Nogueira, "Vehicular Networks Using the IEEE 802.11p Standard: An Experimental Analysis," *Vehicular Communications*, vol. 1, no. 2, pp. 91 – 96, 2014.

[98] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *Proceedings of 2008 IEEE Conference on Software Testing (ICST)*, Mar. 2008.

[99] M. Stoffers, R. Bettermann, J. Gross, and K. Wehrle, "Enabling Distributed Simulation of OMNeT++ INET Models," in *Proceedings of 2014 OMNeT++ Community Summit*, Sep. 2014, pp. 1–5.

[100] A. Maltsev, I. Bolotin, A. Lomayev, A. Pudeyev, and M. Danchenko, "User Mobility Impact on Millimeter-Wave System Performance," in *IEEE 2016 European Conference on Antennas and Propagation (EuCAP)*, Apr. 2016, pp. 1–5. doi: 10.1109/EuCAP.2016.7481505

[101] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE 2014*, vol. 102, no. 3, pp. 366–385, Mar. 2014. doi: 10.1109/JPROC.2014.2299397

[102] P. Kumari, J. Choi, N. González-Prelcic, and R. W. Heath, "IEEE 802.11ad-Based Radar: An Approach to Joint Vehicular Communication-Radar System," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3012–3027, Apr. 2018. doi: 10.1109/TVT.2017.2774762

[103] A. Natarajan, S. K. Reynolds, M. D. Tsai, S. T. Nicolson, J. H. C. Zhan, D. G. Kam, D. Liu, Y. L. O. Huang, A. Valdes-Garcia, and B. A. Floyd, "A Fully-Integrated 16-Element Phased-Array Receiver in SiGe BiCMOS for 60-GHz Communications," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 5, pp. 1059–1075, May 2011. doi: 10.1109/JSSC.2011.2118110

[104] Z. Weixia, G. Chao, D. Guanglong, W. Zhenyu, and G. Ying, "A new Codebook Design Scheme for Fast Beam Searching in Millimeter-Wave Communications," *China Communications*, vol. 11, no. 6, pp. 12–22, Jun. 2014. doi: 10.1109/CC.2014.6878999

[105] B. Li, Z. Zhou, H. Zhang, and A. Nallanathan, "Efficient Beamforming Training for 60-GHz Millimeter-Wave Communications: A Novel Numerical Optimization Framework," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 703–717, Feb. 2014. doi: 10.1109/TVT.2013.2279694

[106] Z. Wei-Xia, D. Guang-Long, L. Bin, C. Zhi-Fang, H. Yu-Cong, and Z. Fang, "A Novel Beam Search Algorithm for 60 GHz Millimeter Wave Communication," *Journal Of Electronics & Information Technology*, vol. 34, no. 3, pp. 683–688, May 2012. doi: 10.3724/SP.J.1146.2011.00436

[107] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with Eyes Closed: mm-Wave Beam Steering without In-Band Measurement," in *Proceedings of 2015 IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2015, pp. 2416–2424.

[108] W. Ding, Y. Niu, H. Wu, Y. Li, and Z. Zhong, "QoS-Aware Full-Duplex Concurrent Scheduling for Millimeter Wave Wireless Backhaul Networks," *IEEE Access*, vol. 6, pp. 25 313–25 322, Apr. 2018. doi: 10.1109/ACCESS.2018.2828852

[109] J. Vestin and A. Kassler, "Low frequency assist for mmWave backhaul - the case for SDN resiliency mechanisms," in *Proceedings of 2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 205–210. doi: 10.1109/ICCW.2017.7962658

[110] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Combined Speed and Steering Control in High-Speed Autonomous Ground Vehicles for Obstacle Avoidance Using Model Predictive Control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8746–8763, Oct. 2017. doi: 10.1109/TVT.2017.2707076

[111] B. Coll-Perales, M. Gruteser, and J. Gozalvez, "Evaluation of IEEE 802.11ad for mmWave V2V communications," in *Proceedings of 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Apr. 2018, pp. 290–295. doi: 10.1109/WCNCW.2018.8369031

[112] M. Ozpolat, E. Kampert, P. A. Jennings, and M. D. Higgins, "A Grid-Based Coverage Analysis of Urban mmWave Vehicular Ad Hoc Networks," *IEEE Communications Letters*, pp. 1–1, Jun. 2018. doi: 10.1109/LCOMM.2018.2846562

[113] C. Perfecto, J. D. Ser, M. Bennis, and M. N. Bilbao, "Beyond WYSIWYG: Sharing Contextual Sensing Data through MmWave V2V Communications," in *Proceedings of 2017 European Conference on Networks and Communications (EuCNC)*, Jun. 2017. doi: 10.1109/EuCNC.2017.7980726

[114] C. Perfecto, J. D. Ser, and M. Bennis, "Millimeter-Wave V2V Communications: Distributed Association and Beam Alignment," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 2148–2162, Sep. 2017. doi: 10.1109/JSAC.2017.2719998

[115] C. Perfecto, J. D. Ser, and M. Bennis, "On the interplay between scheduling interval and beamwidth selection for low-latency and reliable V2V mmWave communications," in *Proceedings of 2017 Conference on Innovations in Clouds, Internet and Networks (ICIN)*, Mar. 2017, pp. 1–8.

[116] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching Theory for Future Wireless Networks: Fundamentals and Applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, May 2015. doi: 10.1109/MCOM.2015.7105641

[117] Y. Gu, L. X. Cai, M. Pan, L. Song, and Z. Han, "Exploiting the Stable Fixture Matching Game for Content Sharing in D2D-Based LTE-V2X Communications," in *Proceedings of 2016 Global Communications Conference (GLOBECOM)*, Dec. 2016. doi: 10.1109/GLOCOM.2016.7841832

[118] R. W. Irving and S. Scott, "The Stable Fixtures Problem - A many-to-many Extension of Stable Roommates," *Discrete Applied Mathematics*, vol. 155, no. 16, pp. 2118 – 2129, 2007.

[119] P. Belanovic, D. Valerio, A. Paier, T. Zemen, F. Ricciato, and C. F. Mecklenbrauker, "On Wireless Links for Vehicle-to-Infrastructure Communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 269–282, Jan. 2010. doi: 10.1109/TVT.2009.2029119

[120] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" *IEEE Access*, vol. 1, pp. 335–349, May 2013. doi: 10.1109/ACCESS.2013.2260813

[121] M. Abouelseoud and G. Charlton, "System Level Performance of Millimeter-Wave Access Link for Outdoor Coverage," in *Proceedings of 2013 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013, pp. 4146–4151.

[122] S. S. Sehra, J. Singh, and H. S. Rai, "A Systematic Study of OpenStreetMap Data Quality Assessment," in *Proceeding of 2014 International Conference on Information Technology (ICIT)*, Apr. 2014, pp. 377–381. doi: 10.1109/ITNG.2014.115

[123] M. Fogue, J. A. Sanguesa, F. J. Martinez, and J. M. Marquez-Barja, "Improving Roadside Unit Deployment in Vehicular Networks by Exploiting Genetic Algorithms," *Applied Sciences*, vol. 8, Jan. 2018. doi: 10.3390/app8010086

[124] W. Yan, S. Xin-xin, and S. Yan-ming, "Study on the application of genetic algorithms in the optimization of wireless network," *International Workshop on Automobile, Power and Energy Engineering*, vol. 16, pp. 348 – 355, 2011. doi: https://doi.org/10.1016/j.proeng.2011.08.1094

[125] C. Soto, D. H. Covarrubias, and S. Villarreal, "Base Station Placement Optimization Algorithm for Heterogeneous Distributions of Mobile Users with Multi-Service Requirements," *IEEE Latin America Transactions*, vol. 10, no. 5, pp. 2032–2039, Sep. 2012. doi: 10.1109/TLA.2012.6362345

[126] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[127] N. Palizban, S. Szyszkowicz, and H. Yanikomeroglu, "Automation of Millimeter Wave Network Planning for Outdoor Coverage in Dense Urban Areas Using Wall-Mounted Base Stations," *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 206–209, Apr. 2017. doi: 10.1109/LWC.2017.2659732

[128] T. J. Wu, W. Liao, and C. J. Chang, "A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks," *IEEE Transactions on Communications*, vol. 60, no. 8, pp. 2295–2303, Aug. 2012. doi: 10.1109/TCOMM.2012.062512.100550

[129] R. Zhang, F. Yan, W. Xia, S. Xing, Y. Wu, and L. Shen, "An Optimal Roadside Unit Placement Method for VANET Localization," in *Proceedings of 2017 Global Communications Conference (GLOBECOM)*, Dec. 2017, pp. 1–6. doi: 10.1109/GLOCOM.2017.8253971

[130] S. S. Szyszkowicz, A. Lou, and H. Yanikomeroglu, "Automated Placement of Individual Millimeter-Wave Wall-Mounted Base Stations for Line-of-Sight Coverage of Outdoor Urban Areas," *IEEE Wireless Communications Letters*, vol. 5, no. 3, pp. 316–319, Jun. 2016. doi: 10.1109/LWC.2016.2552168

[131] Y. Chen, H. Peng, and J. W. Grizzle, "Fast Trajectory Planning and Robust Trajectory Tracking for Pedestrian Avoidance," *IEEE Access*, vol. 5, pp. 9304–9317, Jun. 2017. doi: 10.1109/ACCESS.2017.2707322

[132] E. Uhlemann, "Connected-Vehicles Applications Are Emerging," *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 25–96, Mar. 2016. doi: 10.1109/MVT.2015.2508322

[133] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016. doi: 10.1109/TITS.2015.2498841

[134] X. Chen, Y. Miao, M. Jin, and Q. Zhang, "Driving Decision-making Analysis of Lane-Changing for Autonomous Vehicle under Complex Urban Environment," in *Proceedings*

of 2017 Chinese Control and Decision Conference (CCDC), May 2017, pp. 6878–6883. doi: 10.1109/CCDC.2017.7978420

[135] M. Obst, L. Hobert, and P. Reisdorf, "Multi-sensor Data Fusion for Checking Plausibility of V2V Communications by Vision-based Multiple-object Tracking," in *Proceedings of 2014 IEEE Vehicular Networking Conference (VNC)*, Dec. 2014, pp. 143–150. doi: 10.1109/VNC.2014.7013333

[136] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "Carspeak: A Content-Centric Network for Autonomous Driving," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 259–270, Aug. 2012. doi: 10.1145/2377677.2377724

[137] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 27–51, Jun. 2015. doi: 10.1109/COMST.2014.2330903

[138] C. McManus, B. Upcroft, and P. Newman, "Learning Place-Dependant Features for Long-Term Vision-Based Localisation," *Autonomous Robots*, vol. 39, no. 3, pp. 363–387, Oct. 2015. doi: 10.1007/s10514-015-9463-y

[139] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous Driving in Urban Environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, Jun. 2008.

[140] M. McNaughton, C. Urmson, J. M. Dolan, and J. W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4889–4895. doi: 10.1109/ICRA.2011.5980223

[141] M. Gerla, E. K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proceedings of 2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 241–246. doi: 10.1109/WF-IoT.2014.6803166

[142] A. Asvadi, P. Girão, P. Peixoto, and U. Nunes, "3D object tracking using RGB and LIDAR data," in *Proceedings of 2016 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 1255–1260. doi: 10.1109/ITSC.2016.7795718

[143] G. Zhao, X. Xiao, and J. Yuan, "Fusion of Velodyne and Camera Data for Scene Parsing," in *Proceedings of 2012 International Conference on Information Fusion*, Jul. 2012, pp. 1172–1179.

[144] G. Rigazzi, A. Tassi, R. J. Piechocki, T. Tryfonas, and A. Nix, "Optimized Certificate Revocation List Distribution for Secure V2X Communications," in *Proceedings of 2017 IEEE*

Vehicular Technology Conference (VTC-Fall), Sep. 2017, pp. 1–7. doi: 10.1109/VTC-Fall.2017.8288287

[145] A. S. Khan, A. Tassi, and I. Chatzigeorgiou, "Rethinking the Intercept Probability of Random Linear Network Coding," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1762–1765, Oct. 2015. doi: 10.1109/LCOMM.2015.2470662

[146] I. Chatzigeorgiou and A. Tassi, "Decoding Delay Performance of Random Linear Network Coding for Broadcast," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7050–7060, Aug. 2017. doi: 10.1109/TVT.2017.2670178

[147] "FLOURISH Car Trials - Dataset of three days of field trials," https://seis.bristol.ac.uk/~eerjp/v2xtrials/, accessed: 31-08-2018.

[148] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," European Telecommunications Standards Institute (ETSI), Standard, Nov. 2014.

[149] S. A. M. Ahmed, S. H. S. Ariffin, and N. Fisal, "Overview of Wireless Access in Vehicular Environment (WAVE) Protocols and Standards," *Indian Journal of Science and Technology*, vol. 6, no. 7, 2013.

[150] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," European Telecommunications Standards Institute (ETSI), Standard, Sep. 2014.

[151] E. Belyaev, P. Molchanov, A. Vinel, and Y. Koucheryavy, "The Use of Automotive Radars in Video-Based Overtaking Assistance Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1035–1042, Sep. 2013.

[152] R. G. Sargent, "An Introductory Tutorial on Verification and Validation of Simulation Models," in *Proceedings of 2015 Winter Simulation Conference (WSC)*, Dec. 2015, pp. 1729–1740.

[153] F. Klingler, F. Dressler, and C. Sommer, "IEEE 802.11p Unicast Considered Harmful," in *Proceedings of 2015 IEEE Vehicular Networking Conference (VNC)*, Dec. 2015, pp. 76–83. doi: 10.1109/VNC.2015.7385550

[154] C. Cooper, M. Ros, F. Safaei, D. Franklin, and M. Abolhasan, "Simulation of Contrasting Clustering Paradigms under an Experimentally-Derived Channel Model," in *Proceedings of 2014 IEEE Vehicular Technology Conference (VTC2014-Fall)*, Sep. 2014, pp. 1–6. doi: 10.1109/VTCFall.2014.6966179

175

[155] C. Sommer, S. Joerer, and F. Dressler, "On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation," in *Proceedings of 2012 IEEE Vehicular Networking Conference (VNC)*, Nov. 2012, pp. 64–69. doi: 10.1109/VNC.2012.6407446

[156] C. Sommer, D. Eckhoff, and F. Dressler, "IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1733–1745, Aug. 2014. doi: 10.1109/TMC.2013.80

[157] P. J. Roache, *Verification and Validation in Computational Science and Engineering*. Hermosa Albuquerque, NM, 1998, vol. 895.

[158] R. Fantacci, D. Tarchi, and A. Tassi, "A Novel Routing Algorithm for Mobile Pervasive Computing," in *Proceedings of IEEE 2010*, Dec. 2010. doi: 10.1109/GLOCOM.2010.5683765

[159] "Mikrotik RB433 Data Sheet." [Online]. Available: https://mikrotik.com/product/RB433

[160] "Mikrotik R52H Data Sheet." [Online]. Available: https://mikrotik.com/product/R52H

[161] "Mikrotik R5SHPn Data Sheet." [Online]. Available: https://routerboard.com/R5SHPn

[162] E. Estopace, "Netflix adaptive video streaming," accessed: 2018-09-30. [Online]. Available: http://www.philstar.com/technology/2016/06/06/1590535/netflixs-answer-slow-internet-adaptive-streaming

[163] I. Politis, L. Dounis, and T. Dagiuklas, "H.264/svc vs. h.264/avc video quality comparison under qoe-driven seamless handoff," *Signal Processing: Image Communication*, vol. 27, no. 8, pp. 814 – 826, 2012. doi: http://dx.doi.org/10.1016/j.image.2012.01.006 Special issue on: pervasive mobile multimedia.

[164] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012. doi: 10.1109/TCSVT.2012.2221192

[165] A. Abdelgader and W. Lenan, "The Physical Layer of the IEEE 802.11p WAVE Communication Standard: The Specifications and Challenges," in *Proceedings of 2014 World Congress on Engineering and Computer Science (WCECS)*, vol. 2, Oct. 2014.

[166] R. Reinders, M. van Eenennaam, G. Karagiannis, and G. Heijenk, "Contention Window Analysis for Beaconing in VANETs," in *Proceedings of 2011 International Wireless Communications and Mobile Computing Conference*, Jul. 2011, pp. 1481–1487.

[167] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," United States, Tech. Rep., 2003.

[168] A. Paier, D. Faetani, and C. F. Mecklenbräuker, "Performance Evaluation of IEEE 802.11p Physical Layer Infrastructure-to-Vehicle Real-World Measurements," in *Proceedings of 2010 International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*, Nov. 2010, pp. 1–5. doi: 10.1109/ISABEL.2010.5702844

[169] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, Dec. 2012.

[170] C. A. Balanis, *Antenna Theory: Analysis and Design, 4th Edition*. John Wiley & Sons, Mar. 2016.

[171] O. Renaudin, V. M. Kolmonen, P. Vainikainen, and C. Oestges, "Wideband MIMO Car-to-Car Radio Channel Measurements at 5.3 GHz," in *Proceedings of 2008 IEEE Vehicular Technology Conference (VTC)*, Sep. 2008, pp. 1–5.

[172] S. Zhu, T. S. Ghazaany, S. M. R. Jones, R. A. Abd-Alhameed, J. M. Noras, T. V. Buren, J. Wilson, T. Suggett, and S. Marker, "Probability Distribution of Rician K-Factor in Urban, Suburban and Rural Areas Using Real-World Captured Data," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 7, pp. 3835–3839, Jul. 2014.

[173] I. M. Kilgore, S. A. Kabiri, A. W. Kane, and M. B. Steer, "The Effect of Chaotic Vibrations on Antenna Characteristics," *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1242–1244, 2016. doi: 10.1109/LAWP.2015.2503264

[174] "IEEE Std. 802.11ad - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band," *IEEE Std. 802.11ad-2014*, pp. 1–634, Mar. 2014. doi: 10.1109/IEEESTD.2014.6774849

[175] B. Kerner, "Synchronized Flow as a New Traffic Phase and Related Problems for Traffic Flow Modelling," *Journal of Mathematical and Computer Modelling*, vol. 35, no. 5, pp. 481 – 508, 2002.

[176] V. Va and R. W. Heath, "Basic Relationship between Channel Coherence Time and Beamwidth in Vehicular Channels," in *Proceedings of 2015 IEEE Vehicular Technology Conference (VTC-Fall)*, Sep. 2015, pp. 1–5. doi: 10.1109/VTCFall.2015.7390852

[177] A. Yamamoto, K. Ogawa, T. Horimatsu, A. Kato, and M. Fujise, "Path-Loss Prediction Models for Intervehicle Communication at 60 GHz," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 65–78, Jan. 2008. doi: 10.1109/TVT.2007.901890

[178] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, Jun. 2014. doi: 10.1109/JSAC.2014.2328154

[179] D. Hamby, "A Comparison of Sensitivity Analysis Techniques," *Health physics*, vol. 68, no. 2, pp. 195–204, 1995.

[180] E. Ben-Dor, T. S. Rappaport, Y. Qiao, and S. J. Lauffenburger, "Millimeter-Wave 60 GHz Outdoor and Vehicle AOA Propagation Measurements Using a Broadband Channel Sounder," in *Proceedings of 2011 IEEE Global Telecommunications Conference (GLOBE-COM)*, Dec. 2011, pp. 1–6. doi: 10.1109/GLOCOM.2011.6133581

[181] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-Time Trajectory Planning for Autonomous Urban Driving: Framework, Algorithms, and Verifications," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 740–753, Apr. 2016. doi: 10.1109/TMECH.2015.2493980

[182] P. Biró, W. Kern, D. Paulusma, and P. Wojuteczky, "The Stable Fixtures Problem with Payments," *Games and Economic Behavior*, 2017. doi: 10.1016/j.geb.2017.02.002

[183] John van Rijn, "Road Capacities," INDEVELOPMENT, Tech. Rep., Jul. 2014.

[184] R. W. Irving, "The Stable Roommates Problem with Ties," *Journal of Algorithms*, vol. 43, p. 85–105, 2002.

[185] R. W. Irving, "Stable Marriage and Indifference," *Discrete Applied Mathematics*, vol. 48, no. 16, p. 261–272, 1994.

[186] "OpenStreetMap Project," https://www.openstreetmap.org, 2017.

[187] O. Semiari, W. Saad, Z. Daw, and M. Bennis, "Matching Theory for Backhaul Management in Small Cell Networks with MmWave Capabilities," in *Proceedings of 2015 IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 3460–3465. doi: 10.1109/ICC.2015.7248860

[188] A. A. Goulianos, A. L. Freire, T. Barratt, E. Mellios, P. Cain, M. Rumney, A. Nix, and M. Beach, "Measurements and Characterisation of Surface Scattering at 60 GHz," in *Proceedings of 2017 IEEE Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–5. doi: 10.1109/VTCFall.2017.8288373

[189] Q. Wang, S. Leng, H. Fu, and Y. Zhang, "An IEEE 802.11p-Based Multichannel MAC Scheme With Channel Coordination for Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 449–458, Jun. 2012. doi: 10.1109/TITS.2011.2171951

[190] D. N. M. Dang, H. N. Dang, V. Nguyen, Z. Htike, and C. S. Hong, "HER-MAC: A Hybrid Efficient and Reliable MAC for Vehicular Ad Hoc Networks," in *Proceedings of 2014 IEEE International Conference on Advanced Information Networking and Applications*, May 2014, pp. 186–193. doi: 10.1109/AINA.2014.27

[191] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," United States, Tech. Rep., 2003.

[192] J. Cao, D. Kong, M. Charitos, D. Berkovskyy, A. A. Goulianos, T. Mizutani, F. Tila, G. Hilton, A. Doufexi, and A. Nix, "Design and Verification of a Virtual Drive Test Methodology for Vehicular LTE-A Applications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3791–3799, May 2018. doi: 10.1109/TVT.2018.2794263

[193] H. Chour, Y. Nasser, H. Artail, A. Kachouh, and A. Al-Dubai, "VANET Aided D2D Discovery: Delay Analysis and Performance," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8059–8071, Sep. 2017. doi: 10.1109/TVT.2017.2690238

[194] A. Ragaleux, S. Baey, and M. Karaca, "Standard-Compliant LTE-A Uplink Scheduling Scheme With Quality of Service," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7207–7222, Aug. 2017. doi: 10.1109/TVT.2017.2654299

[195] E. Iradier, J. Montalban, G. Araniti, M. Fadda, and M. Murroni, "Adaptive resource allocation in LTE vehicular services using LDM," in *Proceedings of 2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2016, pp. 1–6. doi: 10.1109/BMSB.2016.7521985

[196] O. Puñal, C. Pereira, A. Aguiar, and J. Gross, "Experimental Characterization and Modeling of RF Jamming Attacks on VANETs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 524–540, Feb. 2015. doi: 10.1109/TVT.2014.2325831

[197] T. P. Ryan, *Modern Engineering Statistics*. John Wiley & Sons, Nov. 2006.